Xpert.press

Gisela Engeln-Müllges
Klaus Niederdrenk
Reinhard Wodicka

Numerik-Algorithmen

2 CD-ROMs

Springer

Xpert.press

Die Reihe **Xpert.press** vermittelt Professionals in den Bereichen Softwareentwicklung, Internettechnologie und IT-Management aktuell und kompetent relevantes Fachwissen über Technologien und Produkte zur Entwicklung und Anwendung moderner Informationstechnologien.

Gisela Engeln-Müllges · Klaus Niederdrenk
Reinhard Wodicka

Numerik-Algorithmen

Verfahren, Beispiele, Anwendungen

Neunte, vollständig überarbeitete und erweiterte Auflage
mit zahlreichen Abbildungen und Beispielen
sowie 2 CD-ROMs

Springer

Prof. Dr. rer. nat. Gisela Engeln-Müllges
Fachbereich Maschinenbau und Mechatronik
Fachhochschule Aachen
Goethestraße 1
52064 Aachen
engeln-muellges@fh-aachen.de

Prof. Dr. rer. nat. Klaus Niederdrenk
Fachbereich Chemieingenieurwesen
Fachhochschule Münster
Stegerwaldstraße 39
48565 Steinfurt
niederdrenk@fh-muenster.de

Stud. Prof. Dr. rer. nat. Reinhard Wodicka
Am Kupferofen 34
52066 Aachen

Bibliografische Information der Deutschen Bibliothek
Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.ddb.de abrufbar.

Die Neuauflage fasst die 5. Auflage von „Numerische Mathematik für Ingenieure“ von Gisela Engeln-Müllges und Fritz Reutter(ursprünglich Bibliographisches Institut/Brockhaus AG, Mannheim) und die 8. Auflage von „Numerik-Algorithmen“ von Gisela Engeln-Müllges und Fritz Reutter (ursprünglich VDI-Verlag, Düsseldorf) zusammen.

ISSN 1439-5428
ISBN 3-540-62669-7 Springer Berlin Heidelberg New York

Springer ist ein Unternehmen von Springer Science+Business Media
springer.de

Satz: Druckfertige Daten der Autoren
Herstellung: LE-TEX Jelonek, Schmidt & Vöckler GbR, Leipzig
Umschlaggestaltung: KünkelLopka Werbeagentur, Heidelberg
Gedruckt auf säurefreiem Papier 33/3142/YL - 5 4 3 2 1 0

Im Gedenken an unseren akademischen Lehrer
Prof. Dr. rer. techn. Fritz Reutter (1911–1990)
Rheinisch-Westfälische Technische Hochschule Aachen

Vorwort zur 9. Auflage

Das vorliegende Werk steht in einer langen Tradition. Die erste Auflage erschien 1974 mit den Autoren Gisela Engeln-Müllges und Fritz Reutter unter dem Titel „Formelsammlung zur Numerischen Mathematik mit FORTRAN-Programmen“. Der Inhalt des Buches orientierte sich an Vorlesungen über Numerische Mathematik für Studierende der Ingenieurwissenschaften.
Bei den nachfolgenden Auflagen, an denen Fritz Reutter bis zu seiner schweren Erkrankung 1983 mitarbeitete, nahm der Umfang des Buches stark zu, und die Programmanhänge wurden um die Sprachen Turbo Pascal, C, Modula 2 und Quick-Basic erweitert. Dies hing zusammen mit der raschen Entwicklung der Computertechnologie und ihren Impulsen auf die Numerische Mathematik, die zu einer enormen qualitativen wie auch quantitativen Zunahme numerischer Verfahren führte.

Grundlage des vorliegenden Buches ist die achte Auflage des Buches von Gisela Engeln-Müllges und Fritz Reutter mit dem Titel „Numerik-Algorithmen“ (VDI-Verlag, 1996), die erstmals eine CD-ROM mit Fortran 77/90-, Turbo Pascal- und ANSI C-Programmen enthielt.

Wie bisher wird auch in dieser neunten Auflage von den unten genannten Verfassern besonderer Wert gelegt auf die Erläuterung der Prinzipien der behandelten Verfahren der Numerischen Mathematik, auf die exakte Beschreibung leistungsfähiger Algorithmen und die Entwicklung und Dokumentation zugehöriger Programme. Die ausführliche Darstellung der mathematischen Grundlagen, ergänzt durch viele Abbildungen und durchgerechnete Beispiele, soll den Zugang zur Numerischen Mathematik erleichtern und das Verständnis des algorithmischen Vorgehens fördern. Zahlreiche Beispiele aus ingenieurwissenschaftlichen Anwendungen belegen die Notwendigkeit der behandelten numerischen Verfahren.

Das Buch wendet sich in erster Linie an Ingenieure sowie an Naturwissenschaftler und Informatiker in Studium und Beruf, ferner an Lehrkräfte für mathematisch-naturwissenschaftlichen Unterricht. Deshalb ist besonders auf die Verwendbarkeit der behandelten Themen in der Praxis geachtet worden.

Beigefügt ist dem Buch eine CD-ROM, die umfassend getestete C-Programme zu nahezu allen angegebenen Algorithmen enthält, sowie weitere Software, zu der Informationen auf den folgenden Seiten zu finden sind.

Die inhaltliche Erweiterung des Buches erfordert es, die bisher in ihm enthaltenen numerischen Verfahren für Anfangs- und Randwertprobleme bei gewöhnlichen Differentialgleichungen in einen weiteren Band zu verlagern. Dieser wird, einem häufig geäußerten Wunsch folgend, zusätzlich Numerik partieller Differentialgleichungen sowie wichtige stochastische Methoden, insbesondere statistische Schätz- und Prüfverfahren enthalten und ebenfalls bei Springer erscheinen. Verfasser dieses Buches sind Gisela Engeln-Müllges, Klaus Niederdrenk und Wieland Richter.

Ganz besonders herzlich danken wir den Autoren der Programme und ebenso Doris und Uli Eggermann, die mit Hilfe des Satzprogramms LaTeX das reproduktionsreife Manuskript sowie eine Vielzahl der Abbildungen mit äußerstem Engagement und sehr großem Geschick fertig gestellt haben. Darüber hinaus hat uns Uli Eggermann tatkräftig beim Korrekturlesen und beim Rechnen der Beispiele zur Kubatur unterstützt.
Ein herzlicher Dank gilt nicht zuletzt Herrn Hermann Engesser vom Springer-Verlag für die hervorragende und effektive Zusammenarbeit.

Aachen, Münster, Juli 2004

Gisela Engeln-Müllges
Klaus Niederdrenk
Reinhard Wodicka

Informationen zur beigefügten Software (CD-1, CD-2)

Dem Buch sind zwei CDs beigefügt, welche unterschiedliche Demonstrationsprogramme enthalten sowie ANSI-C-Quellen zur Verwendung in selbstgeschriebenen Programmen.

Informationen zur CD-ROM mit C-Programmen u.a. (CD-1)

Auf dieser CD befinden sich Ansi-C-Quellen von Unterprogrammen zu den meisten der im Buch angegebenen Algorithmen. Diese Quellen können compilerunabhängig in eigenen C-Programmen verwendet werden. Mittels beigefügter Makefile-Datei ist man in der Lage,

- gezielt einzelne Module zu übersetzen
- eine eigene Bibliothek zusammenzustellen, die zu selbstgeschriebenen Programmen hinzugelinkt werden kann
- spezielle Testprogramme zu den Unterprogrammen zu erstellen und mit geeigneten Testdatensätzen ablaufen zu lassen.

Weitergehende Informationen zur Verwendung der Ansi-C-Quellen und Antworten zu Compiler- und Makefile-Fragen sind auf der CD in der Datei *ReadMe.htm* und in der DMAKE-Datei *Makefile.mk* angegeben.

Systemvoraussetzungen

Die Unterprogramme wurden unter verschiedenen Betriebssystemen getestet, u.a. MS-DOS, Windows (95, 98, 2000, NT), OS/2, TOS 4.04, UNIX, Linux. Dabei wurden diverse C-Compiler verwendet.

C++-Programme und Campuslizenzen

Die C-Programme der CD-ROM sind bei der FH-Aachen auch in C++ erhältlich. Informationen über Lizenzen zu den C++-Programmen sowie über Campuslizenzen zu den C- und C++-Programmen können per e-Mail angefordert werden:

engeln-muellges@fh-aachen.de

Weitere Software auf der CD-1

Auf der CD-1 sind noch folgende Programme zu finden:

- CurveTrac
- CurveView
- Interaktive Lehrunterstützung

Zu allen drei Programmen finden Sie Informationen in den folgenden Abschnitten. In der HTML-Datei *ReadMe* auf der CD sind sie ebenfalls kurz beschrieben und können (unter Windows) direkt aus dem Browser (z.B. IE, Netscape, Mozilla, Opera) heraus gestartet werden.

Systemvoraussetzungen

Betriebssystem:	Windows (98, 2000 oder XP)
Arbeitsspeicher:	mindestens 256 MB RAM
Browser:	Netscape ab 6.2, Internet Explorer ab 5.5, Mozilla ab 1.4
Java:	aktivieren (ab Version 1.2); wenn von CD gestartet (aus ReadMe.htm), wird mitgeliefertes Java verwendet.

Informationen zum Expertensystem „CurveTrac“ (CD-1)

CurveTrac ist ein Baukastensystem von numerischen Anwendungen, die auf den C-Programmen aus diesem Buch basieren. Die Benutzer-Oberfläche ist in Java programmiert. Dadurch ist gewährleistet, dass die Anwendung auf den gängigsten Betriebsystemen genutzt werden kann. Getestet wurde CurveTrac unter Windows und Linux. In der hier vorliegenden Ausbaustufe ist das Modul „Höhenlinien“ eingebunden. Informationen über weitere Module können Sie bei engeln-muellges@fh-aachen.de anfordern. Dies sind im Einzelnen:

- Expertensystem zur Numerischen Mathematik
- Berechnung der Schnittkurve zweier Flächen im Raum
- Splinemodul (Flächensplines und Kurvensplines)
- u. a. m. ...

Beschreibung des Moduls „Höhenlinien“

Mit dem Modul „Höhenlinien“ können beliebig angeordnete Wertetripel $\big(x_i, y_i, z_i = f(x_i, y_i)\big)$, z. B. Messdaten, eingegeben werden. Die Funktion $f(x, y)$ wird näherungsweise mittels eines zweidimensionalen Oberflächenspline berechnet und als dreidimensionale Grafik dargestellt. Neben dieser Funktionalität ist es zusätzlich möglich, beliebige Schnitte parallel zur x, y-Ebene zu berechnen und das daraus erzeugte Höhenliniendiagramm in einer zweidimensionalen Grafik darzustellen. Bei Bedarf kann das Höhenliniendiagramm mit frei definierbaren Farben ausgefüllt werden.
CurveTrac mit dem Modul Höhenlinien ist auf der CD-ROM mit den C-Programmen zu finden.
Programmiert wurde CurveTrac von Dominikus Bartusch, Frank Hähling und Thomas Layh.

Informationen zu dem Modul „CurveView“ (CD-1)

CurveView ist wie CurveTrac ein in Java programmiertes Baukastensystem ohne Benutzeroberfläche. Mittels in XML-Dateien abgelegten numerischen Berechnungsvorschriften (basierend auf den C-Programmen aus diesem Buch) können die Ergebnisse einer Berechnung als zwei- bzw. dreidimensionale Grafik bereitgestellt werden. In der hier vorliegenden Ausbaustufe ist das Berechnungsmodul „Kurvensplines“ mit vielen Beispielen eingebunden. Wenn Interesse an weiteren Modulen besteht, z. B.

- Flächensplines
- Nullstellenverfahren
- Differentialgleichungen
- Matrizen
- Statistik
- u. a. m. ...

senden Sie eine Nachricht an engeln-muellges@fh-aachen.de.
Programmiert wurde CurveView von Stefan Kleemann.

Informationen zum Demo-Framework „Interaktive Lehrunterstützung“ auf der CD-ROM (CD-1)

Mit dem Framework können die Lernenden Themengebiete aus Lehrveranstaltungen selbstständig mit interaktiven Elementen vertiefen. Es gibt eine Auswahl von über 100 Interaktionen und Animationen, die gezielt zu speziellen Themengebieten zusammengestellt und dem Lernenden zur Verfügung gestellt werden können. In dieser Demo-Version sind zu jedem der folgenden Themengebiete beispielhaft Interaktionen und Animationen vorhanden:

- Quadratur
- Nullstellen
- Splines
- Differentialgleichungen

Wenn Interesse an weiteren Interaktionen und Animationen besteht, können Informationen unter der Mail-Adresse engeln-muellges@fh-aachen.de angefordert werden.
Programmiert wurde das Framework von Michael Neßlinger.

Informationen zur Demo-CD-ROM „NUMAS“ (CD-2)

Die beiliegende Demo-CD-ROM enthält das Lernfeld „Kurvensplines“ aus dem multimedialen Lehr- und Lernsystem NUMAS zur Numerischen Mathematik und Statistik. Der

Inhalt der Demo-CD-ROM entspricht den Kapiteln über Kurvensplines in diesem Buch und informiert auch über den Gesamtinhalt des Lernsystems.
NUMAS bietet didaktisch aufbereitetes Wissen zur Numerischen Mathematik und Statistik. Es werden Inhalte angeboten, die für die Hochschulausbildung vieler Fachrichtungen grundlegend sind und die sich gleichzeitig hervorragend dazu eignen, in Formen des angeleiteten Selbststudiums aufgenommen zu werden. Lernerinnen und Lerner werden zeitgemäß in ihren Selbstlern- und Selbstorganisationskompetenzen gefördert.
Für die Vollversion NUMAS wurden in großen Teilen die Inhalte dieses Buches und des im Vorwort angekündigten weiteren Bandes mit Differentialgleichungen und Statistik verwendet.
NUMAS ist aus einem vom BMBF und vom Land NRW geförderten Projekt innerhalb der Ausschreibungen „Neue Medien in der Bildung“ und „Neue Medien in der Hochschullehre“ hervorgegangen. An dem Projekt waren die Fachhochschule Aachen (Prof. Dr. Engeln-Müllges), die Freie Universität Berlin (Prof. Dr. Martus), die Fachhochschule Münster (Prof. Dr. Niederdrenk) und die Fachhochschule Südwestfalen (Prof. Dr. Richter) beteiligt.
NUMAS ist auch im Internet unter http://www.numas.de/ zu erreichen.

Der Umgang mit der CD-ROM selbst ist denkbar einfach: Nach dem Einlegen der CD-ROM in das Laufwerk erscheint selbstständig ein Auswahlbildschirm mit den im Gesamtsystem zur Verfügung stehenden Lernfeldern. Von hier aus kann man sich den Inhalt ansehen und bearbeiten. Bei der Demo-CD-ROM ist nur das Lernfeld Kurvensplines freigeschaltet.
Die CD-ROM stellt nur einen Teil der Funktionalität des Lernsystems zur Verfügung. Zum Beispiel sind die Kommunikationswerkzeuge, die an das Internet gebunden sind, nur im Online-System verfügbar.

Systemvoraussetzungen

Ihr Computersystem sollte folgende Bedingungen erfüllen, um einen reibungslosen Betrieb von NUMAS erreichen zu können:

Betriebssystem:	Windows (98, 2000 oder XP)
Browser:	Netscape ab 6.2, Internet Explorer ab 5.5, Mozilla ab 1.4
Javascript:	aktivieren (ab Version 1.2)
Cookies:	aktivieren

Um auch die interaktiven und multimedialen Elemente von NUMAS nutzen zu können, müssen zusätzlich folgende Plug-Ins installiert sein:
Macromedia Flash Player und SUN Java (JRE) ab Version 1.4 (mit Ausnahme der Version 1.4.1_03)
Weitere Informationen zu NUMAS können Sie unter der Mail-Adresse info@numas.de anfordern.

Bezeichnungen

$\Rightarrow$	wenn – dann bzw. hat zur Folge
$\Leftrightarrow$	dann und nur dann
$a := b$	a wird definiert durch b
$\stackrel{!}{=}$	geforderte Gleichheit
$<\quad \leq$	kleiner, kleiner oder gleich
$>\quad \geq$	größer, größer oder gleich
$a << b$	a ist wesentlich kleiner als b
$\approx$	ungefähr gleich
$\equiv$	identisch
$\sim$	proportional bzw. gleichmäßig zu
$\{a_1, a_2, \ldots\}$	Menge aus den Elementen a_1, a_2, ...
$\{x \mid \ldots\}$	Menge aller x für die gilt ...
$\in$	Element von
$\notin$	nicht Element von
$\subseteq$	enthalten in oder Teilmenge von
$\subset$	echt enthalten in oder echte Teilmenge von
$\not\subset$	nicht Teilmenge von
$\mathbb{N}$	Menge der natürlichen Zahlen
$\mathbb{N}_0$	Menge der natürlichen Zahlen mit Null
$\mathbb{Z}$	Menge der ganzen Zahlen
$\mathbb{Q}$	Menge der rationalen Zahlen
$\mathbb{R}$	Menge der reellen Zahlen
$\mathbb{C}$	Menge der komplexen Zahlen
$\mathbb{R}^+$, $\mathbb{R}^-$	Menge der positiven bzw. negativen reellen Zahlen
(a, b)	offenes Intervall von a bis b, $a < b$
$[a, b]$	abgeschlossenes Intervall von a bis b, $a < b$
$[a, b)$	halboffenes Intervall von a bis b (rechts offen), $a < b$
$(a, b]$	halboffenes Intervall von a bis b (links offen), $a < b$
$n!$	n Fakultät mit $n! = 1 \cdot 2 \cdot 3 \cdots n$, $n \in \mathbb{N}$, $0! := 1$
$\binom{n}{k}$	n über k mit $\binom{n}{k} := \frac{n!}{k!(n-k)!}$, $k \in \mathbb{N}_0$, $k \leq n$, $n \in \mathbb{N}$
$\prod_{i=1}^{n} a_i$	$a_1 \cdot a_2 \cdot a_3 \ldots a_n$
$\sum_{i=1}^{n} a_i$	$a_1 + a_2 + \ldots + a_n$
$\int_a^b$	Integral in den Grenzen a und b

i	imaginäre Einheit $\mathrm{i} := \sqrt{-1}$
e	Eulersche Zahl $= 2.718\,281\,828\,459\ldots$
$\lvert a\rvert$	Betrag von a mit $\lvert a\rvert := \begin{cases} a & \text{für} \quad a \geq 0 \\ -a & \text{für} \quad a < 0 \end{cases}$
$\lVert\cdot\rVert$	Norm von $\cdot$
$\{a_k\}$	Folge von a_k
$\lim\limits_{k\to\infty} a_k$	Limes von a_k für $k \to \infty$
$\max\{f(x) \lvert x \in [a,b]\}$	Maximum aller Funktionswerte $f(x)$ für $x \in [a,b]$
$\min\{\lvert M_i\rvert \,\big\vert\, i = 1,2,\ldots,m\}$	Minimum aller $\lvert M_i\rvert$ für $i = 1,2,\ldots,m$
(x,y)	geordnetes Paar
$(x_1,x_2,\ldots,x_n)$	geordnetes n-Tupel
$f: I \to \mathbb{R}$	Abbildung f von I nach $\mathbb{R}$
$x \mapsto f(x), x \in D$	x wird $f(x)$ zugeordnet für $x \in D$
$f', f'', f''', f^{(4)}, \ldots, f^{(n)}$	erste, zweite, dritte, vierte, …, n-te Ableitung von f
$C[a,b]$	die Menge der auf $[a,b]$ stetigen Funktionen
$C^n[a,b]$	die Menge der auf $[a,b]$ n-mal stetig differenzierbaren Funktionen
$\mathbb{R}^n$	n-dimensionaler euklidischer Raum
$A = O(h^q)$	$\lvert A/h^q\rvert \leq C$ für $h \to 0$, $C =$ const.
$i = m(1)n$	$m,n \in \mathbb{Z}, \quad m \leq n, \quad i = m, m+1, \ldots, n$
sign (a), sgn (a)	$\begin{cases} 1 & \text{für} \quad a > 0 \\ 0 & \text{für} \quad a = 0 \\ -1 & \text{für} \quad a < 0 \end{cases}$
$\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \ldots$	Vektoren
$\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \ldots$	Matrizen
$\boldsymbol{0}$	Nullmatrix bzw. Nullvektor
$\boldsymbol{x} \times \boldsymbol{y}$	Vektorprodukt bzw. Kreuzprodukt
$\boldsymbol{E}$	Einheitsmatrix
$\boldsymbol{A}^{\mathsf{T}}$	transponierte Matrix von $\boldsymbol{A}$
$\boldsymbol{x}^{\mathsf{T}} = (x_1,x_2,\ldots,x_n)$	transponierter Vektor zu $\boldsymbol{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$
$\boldsymbol{A}^{-1}$	inverse Matrix von $\boldsymbol{A}$
$\lvert\boldsymbol{A}\rvert$, $\det(\boldsymbol{A})$	Determinante von $\boldsymbol{A}$
[Verweis]	s. im Literaturverzeichnis unter [Verweis]
o. B. d. A.	ohne Beschränkung der Allgemeinheit
$\square$	Ende eines Beispiels oder eines Beweises

Inhaltsverzeichnis

Kapitel 1

Darstellung von Zahlen und Fehleranalyse, Kondition und Stabilität

1.1 Definition von Fehlergrößen

Ein numerisches Verfahren liefert im Allgemeinen anstelle einer gesuchten Zahl a nur einen Näherungswert A für diese Zahl a. Zur Beschreibung dieser Abweichung werden Fehlergrößen eingeführt.

Definition 1.1. (*Wahrer* und *absoluter Fehler*)
Ist A ein Näherungswert für die Zahl a, so heißt die Differenz

$$\Delta_a = a - A$$

der *wahre Fehler* von A und deren Betrag

$$|\Delta_a| = |a - A|$$

der *absolute Fehler* von A.

Sehr oft wird in der mathematischen Literatur Δ_a bereits als absoluter Fehler und $|\Delta_a|$ als Absolutbetrag des Fehlers bezeichnet. In ingenieurwissenschaftlichen Anwendungen ist allerdings die Schreibweise in Definition 1.1 häufiger anzutreffen.

In den meisten Fällen ist die Zahl a nicht bekannt, so dass weder der wahre noch der absolute Fehler eines Näherungswertes A angegeben werden können. Daher versucht man, für den absoluten Fehler $|\Delta_a|$ von A eine möglichst kleine obere Schranke $\varepsilon_a > 0$ anzugeben, so dass $|\Delta_a| \leq \varepsilon_a$ gilt.

Definition 1.2. (*Fehlerschranke für den absoluten Fehler, absoluter Höchstfehler*)
Ist $|\Delta_a|$ der absolute Fehler eines Näherungswertes A und ist $\varepsilon_a > 0$ eine obere Schranke für $|\Delta_a|$, so dass

$$|\Delta_a| \leq \varepsilon_a$$

gilt, dann heißt ε_a eine *Fehlerschranke für den absoluten Fehler* von A oder *absoluter Höchstfehler* von A.

Bei bekanntem ε_a ist wegen $|\Delta_a| = |a - A| \leq \varepsilon_a$

$$A - \varepsilon_a \leq a \leq A + \varepsilon_a\,, \qquad \text{also} \qquad a \in [A - \varepsilon_a, A + \varepsilon_a]\,. \tag{1.1}$$

Um einen Näherungswert A unabhängig von der Größenordnung von a beurteilen zu können, wird der relative Fehler eingeführt.

Definition 1.3. (*Relativer Fehler*)
Ist $|\Delta_a|$ der absolute Fehler eines Näherungswertes A für die Zahl a, so heißt der Quotient

$$|\delta_a| = \frac{|\Delta_a|}{|a\,|} \quad \text{für} \quad a \neq 0$$

der *relative Fehler* von A.

Streng genommen müsste man $\delta_a = \Delta_a/a$ als relativen Fehler von A bezeichnen. Das Vorzeichen von δ_a gibt dann eine zusätzliche Information über die Richtung des Fehlers, d. h. für eine positive Zahl a hat $\delta_a = (a - A)/a < 0$ demnach $a < A$, also einen zu großen Näherungswert A für a zur Folge.

Da in der Regel a unbekannt ist, wird häufig auch

$$|\delta_a| = \frac{|\Delta_a|}{|A|} \qquad \text{für } A \neq 0$$

relativer Fehler von A genannt. Da dann auch Δ_a nicht exakt angebbar ist, wird man sich wieder mit der Angabe einer möglichst guten oberen Schranke für den relativen Fehler behelfen müssen.

Definition 1.4. (*Fehlerschranke für den relativen Fehler, relativer Höchstfehler*)
Ist $|\delta_a|$ der relative Fehler eines Näherungswertes A und gilt mit einem $\varrho_a > 0$

$$|\delta_a| \leq \varrho_a\,,$$

dann heißt ϱ_a eine *Fehlerschranke für den relativen Fehler* $|\delta_a|$ oder *relativer Höchstfehler* von $|\delta_a|$.

Ist ε_a ein absoluter Höchstfehler von A, so ist

$$\varrho_a = \varepsilon_a/|a| \quad \text{bzw.} \quad \varrho_a = \varepsilon_a/|A|$$

ein relativer Höchstfehler von A.

Definition 1.5. (*Prozentualer Fehler, Fehlerschranke für den prozentualen Fehler*)
Ist $|\delta_a|$ der relative Fehler des Näherungswertes A, so heißt

$$|\delta_a| \cdot 100$$

der *prozentuale Fehler* von A (relativer Fehler in Prozent), und σ_a mit

$$|\delta_a| \cdot 100 \;\leq\; 100 \cdot \varrho_a \;=\; \sigma_a$$

heißt eine *Fehlerschranke für den prozentualen Fehler*.

Beispiel 1.6.
Für die Zahl $x = \pi$ sind $X = 3.14$ ein Näherungswert und $\varepsilon_x = 0.0016$ eine Schranke für den absoluten Fehler $|\Delta_x|$. Also gilt nach (1.1)

$$3.1384 = 3.14 - 0.0016 \leq \pi \leq 3.14 + 0.0016 = 3.1416\,,$$

und der relative Höchstfehler ergibt sich zu

$$|\delta_x| \leq \varrho_x = \frac{0.0016}{3.14} \approx 0.00051\,.$$

Für den prozentualen Fehler folgt nach Definition 1.5

$$100 \cdot |\delta_x| = 0.051\%\,.$$

□

1.2 Zahlensysteme

1.2.1 Darstellung ganzer Zahlen

Für jede ganze Zahl a gibt es genau eine Potenzentwicklung zur Basis 10 (*Zehnerpotenzen*) der Gestalt

$$a \;= v \cdot (a_n 10^n + a_{n-1}10^{n-1} + \ldots + a_1 10^1 + a_0 10^0) = v \cdot \sum_{k=0}^{n} a_k 10^k$$

mit dem Vorzeichen $v \in \{-1, 1\}$, den Koeffizienten $a_k \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ und einer nicht negativen ganzen Zahl n ($n \in \mathbb{N}_0$). Diese *Dezimaldarstellung* von a erhält man, indem man die Ziffern, die zur Bezeichnung der Zahlen a_k dienen, in absteigender Reihenfolge aufschreibt:

$$a \;= v \cdot a_n a_{n-1} \ldots a_1 a_0 \tag{1.2}$$

Die Ziffern a_k in der Dezimaldarstellung (1.2) werden auch *Stellen* von a genannt; die Stellung einer Ziffer in der Zahl gibt ihren Wert (Einer – Zehner – Hunderter usw.) wieder.

Es gibt keinen triftigen Grund, nur das uns gewohnte und vertraute *Dezimalsystem* zu benutzen. Wegen der einfachen technischen Realisierbarkeit benutzen digitale Rechenanlagen durchweg das *Dualsystem*, das auf den zwei verschiedenen Ziffern 0 und 1 beruht. Jede ganze Zahl a kann damit in eindeutiger Weise als Potenzentwicklung zur Basis 2 (Zweierpotenzen)

$$a = v \cdot \sum_{k=0}^{n} a_k \cdot 2^k \qquad \text{mit } a_k \in \{0, 1\}$$

und einem Vorzeichen $v \in \{-1, 1\}$ geschrieben werden. Und wiederum gibt nur die Stellung einer Ziffer a_k Auskunft über ihren Stellenwert, so dass die Angabe der dualen Ziffern in der richtigen Reihenfolge zur Charakterisierung ausreicht:

$$a = v \cdot (a_n a_{n-1} \ldots a_1 a_0)_2$$

Der Index 2 soll hierbei auf die Darstellung als Dualzahl verweisen; weggelassen wird nur der Index 10 für die üblichen Dezimalzahlen. So hat man beispielsweise

$$\begin{aligned} 2004 &= (+1) \cdot (2004)_{10} \\ &= (+1) \cdot (1 \cdot 2^{10} + 1 \cdot 2^9 + 1 \cdot 2^8 + 1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^4 + 1 \cdot 2^2) \\ &= (+1) \cdot (11111010100)_2 \,. \end{aligned}$$

Definition 1.7. (*Stellenwertsystem zur Basis β*)
Sei $\beta \in \mathbb{N}$, $\beta \geq 2$. Das System der β verschiedenen Ziffern 0, 1, …, $\beta-1$ bildet ein *Stellenwertsystem zur Basis β*, und jede ganze Zahl a lässt sich darin in der Form

$$a = v \cdot \sum_{k=0}^{n} a_k \cdot \beta^k = v \cdot (a_n a_{n-1} \ldots a_1 a_0)_\beta$$

mit eindeutig bestimmten Ziffern $a_k \in \{0, 1, \ldots, \beta-1\}$ und einem Vorzeichen $v \in \{-1, 1\}$ darstellen.

Die Wahl $\beta = 10$ führt auf das geläufige Dezimalsystem und $\beta = 2$ auf das Dualsystem. Anwendungen finden immer wieder auch die Basiswahlen $\beta = 8$ (*Oktalsystem*) und $\beta = 16$ (*Hexadezimalsystem* mit den Ziffern 0, 1, …, 9, A, B, C, D, E, F)

Die gegenseitige Konvertierung von Zahlen in unterschiedlichen Stellenwertsystemen lässt sich einfach bewerkstelligen, wobei es reicht, als ein Bezugssystem das Dezimalsystem anzunehmen. Die Umwandlung einer β-Zahl in die zugehörige Dezimalzahl erfolgt ökonomisch mit dem *Horner-Schema* (s. Abschnitt 3.2) für Polynome:

$$\begin{aligned} a &= v \cdot (a_n \beta^n + a_{n-1}\beta^{n-1} + a_{n-2}\beta^{n-2} + \ldots + a_1\beta + a_0) \\ &= v \cdot \Big(\Big\{ \ldots \big[\underbrace{(a_n\beta + a_{n-1})}_{s_{n-1}} \beta + a_{n-2} \big] \beta + \ldots + a_1 \Big\} \beta + a_0 \Big) \\ &\qquad \underbrace{\qquad\qquad\qquad\qquad}_{s_{n-2}} \\ &\qquad\quad \ddots \\ &\qquad \underbrace{\qquad\qquad\qquad\qquad\qquad\qquad\qquad}_{s_1} \\ &\quad \underbrace{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}_{s_0} \end{aligned}$$

Algorithmus 1.8. (*Umwandlung einer β-Zahl in eine Dezimalzahl*)
Berechnet man für eine im Stellenwertsystem zur Basis β gegebene ganze Zahl $a = v \cdot (a_n a_{n-1} \dots a_1 a_0)_\beta$, $v \in \{+1, -1\}$, ausgehend von $s_n = a_n$, nacheinander die Größen

$$s_k = \beta \cdot s_{k+1} + a_k\,, \qquad k = n{-}1,\, n{-}2,\, \dots,\, 1,\, 0\,,$$

dann ist $a = v \cdot s_0$ im Dezimalsystem.

Umgekehrt folgt aus dem letzten Zwischenergebnis

$$a \;= v \cdot s_0 = v \cdot (s_1 \cdot \beta + a_0)\,,$$

dass a_0 als der Rest der Division der ganzen Zahl s_0 durch β angesehen werden kann:

$$\frac{s_0}{\beta} = s_1 + \frac{a_0}{\beta} \qquad \text{oder} \qquad \frac{s_0}{\beta} = s_1 \quad \text{Rest } a_0$$

Geht man schrittweise weiter zurück, so folgen die nächsten β-Ziffern a_1, a_2 und so weiter.

Algorithmus 1.9. (*Umwandlung einer Dezimalzahl in eine β-Zahl*)
Berechnet man für eine im Dezimalsystem gegebene ganze Zahl a, ausgehend von $s_0 = |a|$, aus der Gleichung

$$\frac{s_k}{\beta} = s_{k+1} \quad \text{Rest } a_k$$

nacheinander für $k = 0,\, 1,\, 2,\, \dots$ die Größen s_1 und a_0, s_2 und a_1 usw., bis für ein $k = n$ der Wert $s_{n+1} = 0$ ist, dann ist mit den auf diese Weise gewonnenen Ziffern $a_0, a_1, \dots, a_n \in \{0, 1, \dots, \beta{-}1\}$ die β-Darstellung von a gegeben durch

$$a \;= v \cdot (a_n a_{n-1} \dots a_1 a_0)_\beta\,,$$

wobei $v \in \{-1, 1\}$ das Vorzeichen von a bezeichnet.

Die Zahl 2004 besitzt demnach wegen

$$\begin{array}{rcrll}
\frac{2004}{2} & = & 1002 & \text{Rest } 0 & (= a_0) \\
\frac{1002}{2} & = & 501 & \text{Rest } 0 & (= a_1) \\
\frac{501}{2} & = & 250 & \text{Rest } 1 & (= a_2) \\
\frac{250}{2} & = & 125 & \text{Rest } 0 & (= a_3) \\
\frac{125}{2} & = & 62 & \text{Rest } 1 & (= a_4) \\
\frac{62}{2} & = & 31 & \text{Rest } 0 & (= a_5) \\
\frac{31}{2} & = & 15 & \text{Rest } 1 & (= a_6) \\
\frac{15}{2} & = & 7 & \text{Rest } 1 & (= a_7) \\
\frac{7}{2} & = & 3 & \text{Rest } 1 & (= a_8) \\
\frac{3}{2} & = & 1 & \text{Rest } 1 & (= a_9) \\
\frac{1}{2} & = & 0 & \text{Rest } 1 & (= a_{10})
\end{array} \quad \uparrow$$

die Dualdarstellung

$$2004 = (+1) \cdot (11111010100)_2\,.$$

Rechnerintern wird eine ganze Zahl $a = v \cdot |a|$ als Dualzahl durch eine Reihe von Bits ("binary digits") abgespeichert, gewöhnlich in zwei oder vier Bytes (1 Byte = 8 Bits). Bei einer 2-Byte-Hinterlegung bedeutet das:

1. Byte								2. Byte							
$\mu \vdots \alpha_{14}$	α_{13}	α_{12}	α_{11}	α_{10}	α_9	α_8		α_7	α_6	α_5	α_4	α_3	α_2	α_1	α_0

Das Vorzeichen wird dann über

$$\mu = \begin{cases} 0\,, & \text{falls } v = +1 \\ 1\,, & \text{falls } v = -1 \end{cases}$$

gewählt, und im Fall einer positiven Zahl entsprechen α_{14} bis α_0 den Dualziffern von a:

$$a \ = (+1) \cdot (\alpha_{14}\alpha_{13} \ldots \alpha_1\alpha_0)_2$$

Als größte darstellbare positive Zahl ergibt sich dann

$$a \ = (+1) \cdot (111111111111111)_2 = \sum_{k=0}^{14} 1 \cdot 2^k = \frac{2^{15}-1}{2-1} = 32\,767\,.$$

Damit eine Subtraktion auf eine Addition über $a - b = a + (-b)$ zurückgeführt werden kann, werden negative Zahlen rechnerintern durch ein Komplement dargestellt, d. h. im Fall $\mu = 1$ bzw. $v = -1$ stimmen die Hinterlegungen α_k nicht mehr mit den Dualziffern a_k der Zahl $a = (-1) \cdot (a_{14}a_{13} \ldots a_1a_0)_2$ überein.

Vorsicht ist geboten, wenn bei einer Addition oder Subtraktion das Ergebnis nicht mehr im darstellbaren Bereich liegt: Addiert man beispielsweise bei einer 1-Byte-Zahl [größte darstellbare positive Zahl: $2^7-1 = 127$] $97 = (+1)\cdot(1100001)_2$ und $43 = (+1)\cdot(0101011)_2$, so folgt

$$\begin{array}{rr} & 0 \vdots 1\ 1\ 0\ 0\ 0\ 0\ 1 \\ + & 0 \vdots 0\ 1\ 0\ 1\ 0\ 1\ 1 \\ \hline & 1 \vdots 0\ 0\ 0\ 1\ 1\ 0\ 0 \end{array}$$

d. h. ein Übertrag beeinflusst letztendlich das Vorzeichenbit und führt auf ein völlig inplausibles negatives Resultat! Solche Effekte treten leider häufiger auf, ohne dass vom Rechnersystem eine Fehlermeldung oder zumindest eine Warnung ausgesprochen wird. Deshalb ist **bei Rechnerergebnissen stets eine Prüfung auf Plausibilität** erforderlich!

1.2.2 Darstellung reeller Zahlen

Jede nicht ganze, reelle Zahl a besitzt eine Entwicklung der Form

$$a \ = v \cdot \left(\sum_{k=0}^{n} a_k 10^k + \sum_{k=1}^{\infty} b_k 10^{-k} \right)$$

mit einem Vorzeichen $v \in \{-1, 1\}$ und Ziffern a_k, $b_k \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, wobei mindestens ein Term

$$b_k 10^{-k} \neq 0$$

auftritt. Die Dezimaldarstellung einer nicht ganzen Zahl heißt *Dezimalbruch*. Im Dezimalbruch einer Zahl a wird zwischen a_0 und b_1 ein Punkt, der sog. *Dezimalpunkt*, gesetzt:

$$a = v \cdot a_n a_{n-1} a_{n-2} \ldots a_1 a_0 \,.\, b_1 b_2 \ldots b_t \ldots \tag{1.3}$$

Die rechts vom Punkt notierten Stellen heißen *Dezimalstellen* oder *Dezimalen*. Gibt es in einem Dezimalbruch eine Dezimale $b_j \neq 0$, so dass alle folgenden Dezimalen $b_{j+1} = b_{j+2} = ... = 0$ sind, dann heißt der Dezimalbruch *endlich*, andernfalls *unendlich*. Eine Darstellung (1.3) mit endlich vielen Dezimalstellen heißt *Festpunktdarstellung*. Demnach vergegenwärtigt $2\frac{1}{8} = 2.125$ einen endlichen Dezimalbruch und $\frac{1}{3} = 0.3333\ldots$ nicht. Dezimalbruchdarstellungen sind, abgesehen vom Sonderfall der Neunerperiode (z. B. ist $-1.29999\ldots = -1.2\overline{9} = -1.3$), eindeutig.

In einem Stellenwertsystem zur Basis β gilt analog die allgemeine Darstellung

$$\begin{aligned} a &= v \cdot \left(\sum_{k=0}^{n} a_k \, \beta^k + \sum_{k=1}^{\infty} b_k \, \beta^{-k} \right) \\ &= v \cdot (a_n a_{n-1} \ldots a_1 a_0 \,.\, b_1 b_2 \ldots)_\beta \end{aligned}$$

mit $v \in \{+1, -1\}$ und a_k, $b_k \in \{0, 1, \ldots, \beta-1\}$; der Index β verweist wieder auf das zugrunde liegende Stellenwertsystem und entfällt nur im vertrauten Fall $\beta = 10$. Der zwischen den Ziffern a_0 und b_1 gesetzte Punkt heißt nun *β-Punkt*, und man nennt $(a_n a_{n-1} \ldots a_1 a_0)_\beta$ – manchmal auch unter Berücksichtigung des Vorzeichens v – den *ganzzahligen* Anteil und $(\,.\, b_1 b_2 b_3 \ldots)_\beta$ den *gebrochenen* oder *fraktionierten* Anteil von a. Der Sonderfall $\beta = 2$ führt nun zur *Dualbruchdarstellung* einer reellen Zahl. Gibt es nur endliche viele „Nachkomma“-Stellen, so spricht man von einem *endlichen β-Bruch*, andernfalls von einem *unendlichen β-Bruch*.

Es gilt der

Hilfssatz 1.10.
Jede rationale Zahl p/q mit teilerfremden $p \in \mathbb{Z}$ und $q \in \mathbb{N}$ wird durch einen endlichen oder durch einen unendlichen periodischen Dezimal- oder Dualbruch dargestellt, jede irrationale Zahl durch einen unendlichen nicht periodischen Dezimal- oder Dualbruch.

Zur Umwandlung reeller Zahlen in ein β-System reicht es, den ganzzahligen Anteil – siehe Abschnitt 1.2.1 – und den gebrochenen Anteil unabhängig voneinander zu konvertieren. Beispielsweise folgt mit $\beta = 2$ aus

$$\begin{aligned} 0.625 &= (+1) \cdot (\,.\, b_1 b_2 b_3 \ldots)_2 \qquad\qquad \text{mit } b_k \in \{0, 1\} \\ &= b_1 \, 2^{-1} + b_2 \, 2^{-2} + b_3 \, 2^{-3} + \ldots \end{aligned}$$

durch Multiplikation mit $\beta = 2$

$$\begin{aligned} 1.25 &= b_1 + b_2 \, 2^{-1} + b_3 \, 2^{-2} + \ldots \\ &= (b_1 \,.\, b_2 b_3 \ldots)_2 \,. \end{aligned}$$

Somit entspricht b_1 dem ganzzahligen Anteil 1 von 1.25 und weiter

$$0.25 \;=\; (.\, b_2 b_3 \ldots)_2 \;=\; b_2\, 2^{-1} + b_3\, 2^{-2} + \ldots$$

Fährt man nun mit einer erneuten Multiplikation mit $\beta = 2$ fort

$$0.5 \;=\; (b_2\,.\, b_3 \ldots)_2 \;=\; b_2 + b_3\, 2^{-1} + \ldots\ ,$$

so führt der Vergleich der ganzzahligen und der gebrochenen Anteile in dieser Gleichung auf $b_2 = 0$ und $0.5 = (.\, b_3 b_4 \ldots)_2$. Der nächste gleichartige Schritt ergibt über

$$1.0 \;=\; (b_3\,.\, b_4 \ldots)_2 \;=\; b_3 + b_4\, 2^{-1} + \ldots$$

schließlich $b_3 = 1$ und $b_k = 0$ für $k \geq 4$. Mithin gilt

$$0.625 = (+1) \cdot (.\, 101)_2\ ,$$

was auch plausibel ist, denn 0.625 ist die Summe von $\frac{1}{2} = 2^{-1}$ und $\frac{1}{8} = 2^{-3}$.

Allgemein formuliert heißt das:

Algorithmus 1.11. (*Umwandlung einer echt gebrochenen Zahl in einen β-Bruch*)
Gegeben sei eine echt gebrochene Zahl a, $-1 < a\ < 1$.
Berechnet man, ausgehend von $c_0 = |a|$, für $k = 1, 2, 3, \ldots$ nacheinander die Größen

$$\begin{array}{ll} b_k = \text{int}\ (\beta \cdot c_{k-1}) & \text{„ganzzahliger Anteil von } \beta \cdot c_{k-1}\text{“}, \\ c_k = \beta \cdot c_{k-1} - b_k & \text{„echt gebrochener Anteil von } \beta \cdot c_{k-1}\text{“}, \end{array}$$

dann ist

$$a \;= v \cdot (.\, b_1 b_2 b_3 \ldots)_\beta\ ,$$

wobei $v \in \{-1, 1\}$ das Vorzeichen von a wiedergibt.

Die Berechnung wird abgebrochen, wenn hinreichend viele „Nachkomma“-Stellen ermittelt wurden.

Beispiel 1.12.
Algorithmus 1.11 führt mit $a = 0.1$ auf die folgenden Größen für das Dualsystem:

$$\begin{array}{ll} b_1 = \text{int}\ (0.2) = 0\,, & c_1 = 0.2 \\ b_2 = \text{int}\ (0.4) = 0\,, & c_2 = 0.4 \\ b_3 = \text{int}\ (0.8) = 0\,, & c_3 = 0.8 \\ b_4 = \text{int}\ (1.6) = 1\,, & c_4 = 0.6 \\ b_5 = \text{int}\ (1.2) = 1\,, & c_5 = 0.2 \\ b_6 = \text{int}\ (0.4) = 0\,, & c_6 = 0.4 \\ \ldots & \ldots \end{array}$$

Man erkennt, dass sich die Dualziffernfolge periodisch wiederholt:

$$\begin{array}{rcl} 0.1 & = & (+1) \cdot (.\, 00011001100110011 \ldots)_2 \\ & = & (+1) \cdot (.\, 0\overline{0011})_2\ . \end{array}$$

Mithin entspricht 0.1 einem unendlichen periodischen Dualbruch. □

Definition 1.13. (*Tragende Ziffern*)
Alle Ziffern in der β-Darstellung einer Zahl $a = v \cdot (a_n a_{n-1} \dots a_0 \,.\, b_1 b_2 \dots)_\beta$ mit $v \in \{+1, -1\}$ und $a_k,\ b_k \in \{0, 1, \dots, \beta-1\}$, beginnend mit der ersten von Null verschiedenen Ziffer ($a_n \neq 0$), heißen *tragende Ziffern*.

Beispiel 1.14.

$a = 0.0024060$	besitzt 7 Dezimalen	und	5 tragende Ziffern,
$a = 1573800$	besitzt keine Dezimalen	und	7 tragende Ziffern.
$a = 47.110$	besitzt 3 Dezimalen	und	5 tragende Ziffern.
$a = 0.1$	besitzt 1 Dezimale	und	1 tragende Ziffer.

Für die letzte Zahl ergibt sich im Dualsystem $a = (0.0\overline{0011})_2$, eine Zahl mit unendlich vielen Dualstellen und somit unendlich vielen tragenden Ziffern. □

Definition 1.15. (*Normalisierte Gleitpunktdarstellung*)
Jede reelle Zahl $a \neq 0$ kann als β-Zahl in der Form

$$a = v \cdot (\,.\, d_1 d_2 d_3 \dots d_s d_{s+1} \dots)_\beta \cdot \beta^k \tag{1.4}$$

für ein $k \in \mathbb{Z}$ dargestellt werden, wobei $v \in \{+1, -1\}$ das Vorzeichen von a angibt und $d_1 \neq 0$ gilt. (1.4) heißt *normalisierte β-Gleitpunktdarstellung* von a,
$m = (\,.\, d_1 d_2 d_3 \dots)_\beta$ ihre *β-Mantisse*, und k ihr *β-Exponent*. Besitzt die Mantisse s tragende Ziffern, $s \in \mathbb{N}$, so heißt sie *s-stellig*.

So besitzen $a = 346.5201$ und $b = -0.005386$ im Dezimalsystem die normalisierten Gleitpunktdarstellungen

$$\begin{aligned} a &= 0.3465201 \cdot 10^3 \quad \text{(7-stellige Mantisse)}, \\ b &= -0.5386 \cdot 10^{-2} \quad \text{(4-stellige Mantisse)}. \end{aligned}$$

Einem Computer stehen für Berechnungen nur endlich viele in ihm darstellbare Zahlen, die *Maschinenzahlen*, zur Verfügung. Die Mantissen m dieser Maschinenzahlen haben gewöhnlich eine feste Anzahl von Ziffern. Ferner ist der Exponent $k \in \mathbb{Z}$ durch $-k_1 \leq k \leq k_2$ mit $k_1,\ k_2 \in \mathbb{N}$ begrenzt. Eine weltweit gültige Norm nach ANSI (American National Standards Institute) und IEEE (Institute of Electrical and Electronics Engineers) schreibt für reelle Zahlen, als Dualbruch mit 4 Bytes (= 32 Bits) im Rechner hinterlegt, das Format

$\mu \,\vdots\, e_1$	$\dots$	$e_8 \,\vdots\, d_2$	d_3	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	d_{24}

vor, wobei

- $\rightarrow \mu$ das Vorzeichen der Zahl wiedergibt,
- $\rightarrow\ e_1 \dots e_8$ zur Darstellung des Exponenten k in (1.4) gehört; als nicht negative ganze Zahl, auch *Charakteristik* genannt, wird hiervon stets die Konstante 127 abgezogen, d. h. $k = (e_1 \dots e_8)_2 - 127$,
- $\rightarrow\ d_2 \dots d_{24}$ die Mantisse (bei negativen Zahlen wieder komplementär genommen) der Zahl darstellt, wobei $d_1 = 1$ zu ergänzen ist (als normalisierte Gleitpunktdarstellung bleibt für $d_1 \neq 0$ im Dualsystem nur $d_1 = 1$ übrig!).

Hinzu kommen zusätzliche Kennungen für Fehlerfälle wie "Overflow" (etwa bei Division durch 0) und anderes mehr.

Für die Darstellung von $0.1 = (+1) \cdot (.\,0\overline{0011})_2 = (+1) \cdot (.\,\overline{1100})_2 \cdot 2^{-3}$ bedeutet das

$$\boxed{0 \,\vdots\, \underbrace{01111100}_{=124} \,\vdots\, \underbrace{100\,1100\,1100\,1100\,1100\,1100}_{d_2 \dots d_{23}}} \,1100\,11\ldots$$

Als 2-Exponent ergibt sich $124 - 127 = -3$, und $d_1 = 1$ ist unterdrückt worden. Mithin ist klar, dass 0.1 im Rechner nicht exakt wiedergegeben werden kann! Rundet die Rechnerarithmetik auf, so wird d_{24} zu 1 gewählt, und die interne Maschinenzahl ist größer als 0.1 (genauer: 0.1000000015...); wird abgerundet, also $d_{24} = 0$ gesetzt, dann hat die interne Maschinenzahl einen Wert unterhalb von 0.1 (genauer: 0.09999999404...). Deshalb wird ein Rechner – übrigens unabhängig vom Speicherformat – für das Produkt $10 \cdot 0.1$ nie genau den Wert 1 ermitteln.

Dass mit Gleitpunktzahlen auf Rechenanlagen überhaupt gearbeitet wird, liegt daran, dass man damit bei gleicher Speichergröße einen enorm viel größeren Zahlenbereich überstreicht.

Beispiel 1.16.
Nimmt man – abgesehen vom Vorzeichen – an, dass 8 dezimale Speichereinheiten zur Verfügung stehen, so ließen sich damit in den einzelnen Systemen die folgenden positiven Zahlen darstellen:

1) Ganzzahlsystem
   - kleinste darstellbare positive Zahl 0000 0001
   - größte darstellbare positive Zahl 9999 9999
2) Festpunktsystem mit 4 Dezimalen
   - kleinste darstellbare positive Zahl 0000.0001
   - größte darstellbare positive Zahl 9999.9999
3) Normalisiertes Gleitpunktsystem mit 6 Mantissen- und 2 Exponentenziffern (Exponent = Charakteristik−50)
   - kleinste darstellbare positive Zahl $.100\,000 \cdot 10^{-50}$
   - größte darstellbare positive Zahl $.999\,999 \cdot 10^{49}$

Da alle Versionen von demselben Speicheraufwand ausgehen, lassen sich jeweils gleich viele verschiedene Maschinenzahlen darstellen. Die zugehörigen Zahlenwerte sind allerdings auf dem reellen Zahlenstrahl ungleich verteilt:

- Bei dem Ganzzahlsystem lassen sich im positiven Bereich alle ganzen Zahlen zwischen 1 und 99 999 999 (=100 Mio. −1) hinterlegen.
- Bei dem angegebenen Festpunktsystem lassen sich positive Zahlen zwischen 0.0001 (= 1 Zehntausendstel) und 9999.9999 (= 1 Zehntausendstel unter 10 000) wiederum gleichabständig, diesmal im Abstand von 1 Zehntausendstel, als Maschinenzahlen abspeichern.
- In dem betrachteten normaliserten Gleitpunktsystem liegen zwischen jeder zulässigen Zehnerpotenz – also zwischen 10^{-50} und 10^{-49} genauso wie zwischen 10^{-49} und 10^{-48} usw. – gleich viele Maschinenzahlen (genauer: jeweils 899 999). Das heißt das Intervall zwischen 0.1 und 1 enthält genauso viele darstellbare Zahlen wie das Intervall zwischen 1 und 10 oder das Intervall zwischen 10 und 100. Mithin sind diese Zahlen höchst uneinheitlich dicht und unregelmäßig verteilt: Grob gesprochen liegen zwischen 0 und 1 genauso viele Gleitpunktzahlen wie oberhalb von 1. □

1.3 Rechnung mit endlicher Stellenzahl

Den Vorteil der Gleitpunktzahlen, riesige Zahlenbereiche zu erfassen, steht allerdings ein erhöhter Aufwand für die Anwendung der Grundrechenarten gegenüber. Eine Rechnung mit s-stelliger Gleitpunktarithmetik bedeutet, dass alle Zwischenergebnisse und das Endresultat ebenfalls eine s-stellige Mantisse besitzen. Dadurch geht Genauigkeit verloren.

Um derartige Einflüsse einschätzen zu können, sollen im Folgenden vorwiegend anhand des vertrauten Dezimalsystems besondere Betrachtungen angestellt werden. Die erste Schwierigkeit tritt bereits bei der Hinterlegung einer reellen Zahl im Rechner auf, wenn der Speicherplatz nicht genügend Ziffern aufnehmen kann; dies entspricht der folgenden Situation:

Definition 1.17. (*Korrekte Rundung, gültige oder sichere Dezimalen*)
Einer Zahl a in der Darstellung (1.3) mit mehr als t Dezimalstellen wird die Näherungszahl A mit t Dezimalstellen durch korrekte Rundung zugeordnet, wenn

$$|a - A| \leq \frac{1}{2}10^{-t} = \varepsilon_a$$

gilt (vgl. Definition 1.2). A besitzt dann t *gültige (sichere) Dezimalen.*

Daraus folgt:
Der absolute Fehler $|a-A|$ gibt Auskunft über die Anzahl gültiger Dezimalen.

Beispiel 1.18.
Sei $a = 180.1234567$. Gerundet auf 4 Dezimalen erhält man $A = 180.1235$, so dass gilt

$$|a - A| = 0.0000433 \leq 0.5 \cdot 10^{-4};$$

damit besitzt A also 4 gültige Dezimalen. □

Beispiel 1.19.
Jede der folgenden Zahlen ist auf vier tragende Ziffern (d. h. auf eine dezimale Gleitpunktzahl mit 4-stelliger Mantisse) nach Definition 1.17 zu runden und als normalisierte dezimale Gleitpunktzahl (Definition 1.15) darzustellen.

$$\begin{array}{lcll} 0.012358 & \Rightarrow & 0.01236 = & 0.1236 \cdot 10^{-1} \\ 4.2354 & \Rightarrow & 4.235 \ = & 0.4235 \cdot 10^{1} \\ 4.235499 & \Rightarrow & 4.235 \ = & 0.4235 \cdot 10^{1} \\ 4.2698 & \Rightarrow & 4.270 \ = & 0.4270 \cdot 10^{1} \\ 4.2998 & \Rightarrow & 4.300 \ = & 0.4300 \cdot 10^{1} \\ 3.2355 & \Rightarrow & 3.236 \ = & 0.3236 \cdot 10^{1} \\ 3.2345 & \Rightarrow & 3.235 \ = & 0.3235 \cdot 10^{1} \\ 42354 & \Rightarrow & 0.4235 \cdot 10^{5} & \\ 42698 & \Rightarrow & 0.4270 \cdot 10^{5} & \end{array}$$

□

Definition 1.20. *(Statistisch korrekte Rundung)*
Gilt exakt

$$| \, a \ - A \ | = \frac{1}{2} 10^{-t},$$

so wird abgerundet, falls in (1.3) b_t gerade ist, und aufgerundet, falls b_t ungerade ist.

Beispiel 1.21.
Jede der folgenden Zahlen ist nach Definition 1.20 auf vier tragende Ziffern (4-stellige Mantisse) zu runden:

$$\begin{array}{lcl} 3.2355 & \Rightarrow & 3.236\,, \\ 3.2345 & \Rightarrow & 3.234\,, \\ 3.234500 & \Rightarrow & 3.234\,, \\ 3.234501 & \Rightarrow & 3.235\,. \end{array}$$

□

Definition 1.22. (*Gültige oder sichere Ziffern*)
Ist A aus a durch korrekte Rundung auf t Dezimalen entstanden, so heißen die Ziffern in A, die in der Position 10^{-t} und davor stehen, *gültige bzw. sichere Ziffern*; führende Nullen werden ignoriert.

Die letzte Ziffer einer Näherungszahl sollte immer eine gültige (sichere) Ziffer sein.

Beispiel 1.23.

1. Sei X eine Näherungszahl von x. Wird keine Angabe über den absoluten Fehler $|\Delta_x|$ von X gemacht, so sollte die letzte tragende Ziffer der Zahl eine sichere Ziffer sein. Die Schreibweise $X = 3.14$ sollte bedeuten, dass für den absoluten Fehler von X gilt $|\Delta_x| \leq 0.5 \cdot 10^{-2}$, d. h. dass X zwei sichere Dezimalen bzw. drei sichere Ziffern besitzt.

2. Die Angabe $X = 0.004534 \pm 0.000004$ bedeutet, dass X wegen $|x - X| \leq 0.5 \cdot 10^{-5}$ 5 sichere Dezimalen und 3 sichere Ziffern besitzt. Dagegen heißt $X = 0.004534 \pm 0.000006$, dass X wegen $|\Delta_x| > 0.5 \cdot 10^{-5}$, aber $|\Delta_x| \leq 0.5 \cdot 10^{-4}$ nur vier sichere Dezimalen und zwei sichere Ziffern hat.

Die Anzahl sicherer Dezimalen liefert somit eine Abschätzung des absoluten Fehlers. Die Anzahl der sicheren Ziffern hingegen liefert eine grobe Schätzung des relativen Fehlers. □

Es gilt der

Satz 1.24.
a habe die Darstellung (1.4) im Dezimalsystem ($\beta = 10$). A sei eine s-stellige dezimale Gleitpunktzahl, die aus a durch Rundung auf eine s-stellige Mantisse entstanden ist. Dann gelten folgende Abschätzungen:

1. Für den absoluten Fehler von A gilt
$$|\Delta_a| = |a - A| \leq \frac{1}{2} 10^{k-s},$$

2. Für den relativen Fehler von A, $a \neq 0$, gilt
$$|\delta_a| = \left| \frac{\Delta_a}{a} \right| \leq 5 \cdot 10^{-s}.$$

Beweis.
Sei a in der normalisierten Darstellung (1.4) gegeben
$$a \ = v \cdot (.\, d_1 d_2 \ldots d_s d_{s+1} \ldots) \cdot 10^k \,, \quad v \in \{+1, -1\}$$
und werde auf s Mantissenstellen gerundet
$$A \ = v \cdot (.\, d_1 d_2 \ldots D_s) \cdot 10^k \,.$$
Dann gilt
$$|A - a| \leq \frac{1}{2} \cdot 10^{-s} \cdot 10^k = \frac{1}{2} 10^{k-s}$$
und weiter
$$\left| \frac{A - a}{a} \right| \ \leq \ \frac{0.5 \cdot 10^{k-s}}{(0.d_1 d_2 \ldots)\, 10^k} \ = \ \frac{0.5 \cdot 10^{-s}}{.d_1 d_2 \ldots} \ \leq \ \frac{0.5 \cdot 10^{-s}}{0.1} \ = \ 5 \cdot 10^{-s} \,.$$

□

Daraus folgt:
Der relative Fehler gibt Auskunft über die Anzahl gültiger Ziffern einer Zahl.

Beispiel 1.25.
Gegeben ist die Zahl $a = 180.1234567$, die in normalisierter dezimaler Gleitpunktdarstellung lautet

$$a = .1801234567 \cdot 10^3 \quad (\text{also } k = 3)\,.$$

Bei Rundung auf 6-stellige Mantisse ($s = 6$) erhält man

$$A = .180123 \cdot 10^3\,.$$

Für den absoluten Fehler gilt somit

$$|A - a| \;=\; .000\,000\,4567 \cdot 10^3 \;=\; .4567 \cdot 10^{-6} \cdot 10^3 \;\leq\; 0.5 \cdot 10^{-3}\,;$$

A besitzt deshalb 3 gültige Dezimalen. Für den relativen Fehler gilt

$$\left|\frac{A-a}{a}\right| \leq \frac{0.5 \cdot 10^{-3}}{0.1801234567 \cdot 10^3} \leq 5 \cdot 10^{-6}\,;$$

zu A gehören demnach 6 gültige Ziffern. □

Addition von Gleitpunktzahlen

Um zwei Gleitpunktzahlen addieren zu können, müssen zunächst die Exponenten verglichen und womöglich angepasst werden, wobei dann die Mantisse der Zahl mit dem kleineren Exponenten so weit nach rechts verschoben wird, bis die Exponenten übereinstimmen. Dann können die Mantissen addiert werden. In einem letzten Schritt ist unter Umständen die Darstellung zu normalisieren.

Beispiel 1.26.
Es sollen die Zahlen $a = 0.054\,320\,69$ und $b = 999.964\,88$ in einer 5-stelligen dezimalen Gleitpunktarithmetik addiert werden. Die Zahlen a und b werden zunächst im gültigen Format dargestellt:

$$\begin{array}{rcccl} A & = & \mathrm{gl}\,(a) & = & 0.54321 \cdot 10^{-1} \\ B & = & \mathrm{gl}\,(b) & = & 0.99996 \cdot 10^{3} \end{array}$$

$\mathrm{gl}\,(\cdot)$ bezeichne hierbei die zulässige Gleitpunkthinterlegung und -rechnung. Damit ergibt sich

$$\begin{array}{rcl} & A & 0.00005 \,\vdots\, 4321 \cdot 10^3 \\ + & B & 0.99996 \,\vdots\, \cdot 10^3 \\ \hline & & 1.00001 \,\vdots\, \cdot 10^3 \\ & & = 0.10000 \,\vdots\, 1 \cdot 10^4 \quad \text{(normalisiert)} \end{array}$$

also $\mathrm{gl}\,(A + B) = 0.10000 \cdot 10^4 = 1000.0$.

Exakt wären $a + b = 1\,000.019\,200\,69$ und $A + B = 1\,000.014\,321$. Nach Satz 1.24 liegen die absoluten und relativen Fehler zu Beginn bei

$$|\Delta_a| \;=\; |a - A| \;\leq\; 0.5 \cdot 10^{-6}, \qquad |\delta_a| \;=\; \left|\tfrac{\Delta_a}{a}\right| \;\leq\; 5 \cdot 10^{-5},$$

$$|\Delta_b| \;=\; |b - B| \;\leq\; 0.5 \cdot 10^{-2}, \qquad |\delta_b| \;=\; \left|\tfrac{\Delta_b}{b}\right| \;\leq\; 5 \cdot 10^{-5}.$$

Bezeichnet $x = a + b$ die exakte Summe und X deren Näherung in der benutzten Arithmetik, so erhält man entsprechend:

$$|\Delta_x| = |x - X| = 1.92\ldots \cdot 10^{-2}, \quad |\delta_x| = \left|\frac{\Delta_x}{x}\right| = 1.92 \cdot 10^{-5}.$$

Obwohl das Resultat X dem auf die benutzte Gleitpunktarithmetik gerundeten exakten Ergebnis $\mathrm{gl}\,(x)$ entspricht, hat sich der absolute Fehler in Relation zu den Eingangsfehlern doch merklich vergrößert. □

Multiplikation von Gleitpunktzahlen

Um zwei Gleitpunktzahlen zu multiplizieren, genügt es, die jeweiligen Mantissen zu multiplizieren und die zugehörigen Exponenten zu addieren. Zum Schluss muss unter Umständen das Ergebnis der normalisierten Darstellung wieder angepasst werden.

Beispiel 1.27.
Soll das Produkt der Zahlen $a = 0.030\,121\,48$ und $b = 109.9761$ in einer 5-stelligen dezimalen Gleitpunktarithmetik berechnet werden, so ergibt sich mit den gerundeten Werten $A = \mathrm{gl}\,(a)$ und $B = \mathrm{gl}\,(b)$ über

$$\begin{array}{rrl} & A & 0.30121 \vdots \quad \cdot 10^{-1} \\ \cdot & B & 0.10998 \vdots \quad \cdot 10^{3} \\ \hline & & 0.03312 \vdots 707\ldots \cdot 10^{2} \\ & & = 0.33127 \vdots 07\ldots \quad \cdot 10^{1} \qquad \text{(normalisiert)} \end{array}$$

schließlich $\mathrm{gl}\,(A \cdot B) = 0.33127 \cdot 10^1 = 3.3127$.

Exakt hätten sich $a \cdot b = 3.312\,707\,58\ldots$ und $A \cdot B = 3.312\,707\,58\ldots$ ergeben; wiederum entspricht das Resultat $X = \mathrm{gl}\,(A \cdot B)$ dem exakten Ergebnis $x = a \cdot b$, gerundet auf die verwendete Arithmetik: $X = \mathrm{gl}\,(x)$.

Die absoluten und relativen Eingangsfehler nach Satz 1.24

$$\begin{array}{rclcrcl} |\Delta_a| & = & |a - A| & \leq & 0.5 \cdot 10^{-6}, & |\delta_a| & = \left|\frac{\Delta_a}{a}\right| \leq 5 \cdot 10^{-5}, \\ |\Delta_b| & = & |b - B| & \leq & 0.5 \cdot 10^{-2}, & |\delta_b| & = \left|\frac{\Delta_b}{b}\right| \leq 5 \cdot 10^{-5} \end{array}$$

führen in der benutzten Arithmetik zu den entsprechenden Ergebnisfehlern

$$|\Delta_x| = |x - X| = 7.58\ldots \cdot 10^{-6}, \quad |\delta_x| = \left|\frac{\Delta_x}{x}\right| = 0.2288\ldots \cdot 10^{-5};$$

die Gleitpunkt-Multiplikation scheint keinesfalls Eingangsfehler zu vergrößern. □

Man muss beachten, dass mathematische Gesetze wie das Assoziativ- und das Distributivgesetz bei einer Rechnung in einer Gleitpunktarithmetik ihre Allgemeingültigkeit verlieren. Beispielsweise ist mit 5-stelliger Mantissenrechnung

$$\mathrm{gl}\left((0.12345 \cdot 10^{-2} + 0.22232 \cdot 10^{2}) - 0.22223 \cdot 10^{2}\right) = 0.10000 \cdot 10^{-1},$$

hingegen

$$\mathrm{gl}\left(0.12345 \cdot 10^{-2} + (0.22232 \cdot 10^{2} - 0.22223 \cdot 10^{2})\right) = 0.10235 \cdot 10^{-1}.$$

Definition 1.28. (*Maschinengenauigkeit* oder *elementarer Rundungsfehler*)
Die *Maschinengenauigkeit* bzw. der *elementare Rundungsfehler* ϱ ist die kleinste positive Maschinenzahl A, die auf 1 addiert eine Maschinenzahl ungleich 1 liefert:

$$\varrho = \min\{\, A \ > 0 \,|\, \mathrm{gl}\,(1 + A) \neq 1 \,\}\,.$$

$\mathrm{gl}\,(a)$ ist wieder die normalisierte Gleitpunktdarstellung (1.4) der Zahl a.

Beispiel 1.29.
Es sei $a = 1$, die Darstellung erfolge mit 6-stelliger dezimaler Mantisse. Dann gilt

$$\begin{aligned} \mathrm{gl}\,(1) &= .100\,000 \cdot 10^{1}\,, \\ \varrho &= 5 \cdot 10^{-6}\,. \end{aligned}$$

Denn: Bei der Addition $1 + \varrho$ in 6-stelliger Gleitpunktarithmetik ergibt sich

$$\begin{array}{rcl} 1 & = & .100\,000 \cdot 10^{1} \\ +5 \cdot 10^{-6} & = & .000\,0005 \cdot 10^{1} \\ \hline 1 + 5 \cdot 10^{-6} & = & .100\,001 \cdot 10^{1} \ \neq \ 1\,. \end{array}$$

Jede positive Zahl, die kleiner als $5 \cdot 10^{-6}$ ist, wäre bei der Addition unberücksichtigt geblieben. □

Für einen Rechner, der reelle Zahlen a auf Gleitpunktzahlen $A = \mathrm{gl}\,(a)$ mit s-stelliger Mantisse zur Basis β rundet, gilt

$$\left|\frac{A - a}{a}\right| \leq \frac{\beta}{2} \cdot \beta^{-s} = \varrho\,,$$

wobei ϱ die Maschinengenauigkeit nach Definition 1.28 bedeutet. Im gewohnten Dezimalsystem führt das auf $\varrho = 5 \cdot 10^{-s}$. Für Rechner mit Dualzahlarithmetik ergibt sich demnach $\varrho = 2^{-s}$, also für den zuvor angegebenen ANSI- bzw. IEEE-Standard mit $s = 24$ dann $\varrho = 2^{-24} \approx 5.96 \cdot 10^{-8}$, was nach Satz 1.24 über den relativen Fehler mit einer etwa 7-stelligen dezimalen Genauigkeit gleichzusetzen ist.

Kombinierter Test

AbsErr und RelErr seien Schranken für den absoluten bzw. relativen Fehler. Für den Einsatz in Programmen ist der folgende *kombinierte Test* zweckmäßig, der wahlweise eine Abfrage auf den absoluten oder den relativen Fehler erlaubt:

$$|a - A\,| \leq |A|\ \mathrm{RelErr} + \mathrm{AbsErr} \qquad \text{mit} \quad \left\{ \begin{array}{ll} \text{a)} & \mathrm{RelErr} = 0 \text{ und } \mathrm{AbsErr} > 0 \\ \text{b)} & \mathrm{RelErr} > 0 \text{ und } \mathrm{AbsErr} = 0 \end{array} \right. \tag{1.5}$$

(1.5) ist mit a) eine Abfrage auf den absoluten Fehler und mit b) eine Abfrage auf den relativen Fehler.

Es macht unter Umständen Sinn, zugleich beide Fehlerschranken verschieden von Null zu wählen, etwa dann, wenn die Größenordnung des Ergebnisses nicht bekannt ist und in Abhängigkeit davon eine gewisse Genauigkeit erreicht werden soll (bei betragsmäßig großen Zahlen überwiegt so die relative Fehlerschranke, bei betragsmäßig kleinen die absolute Fehlerschranke).

1.4 Fehlerquellen

Bei der numerischen Behandlung eines Problems treten verschiedene Fehlerquellen auf. Der gesamte Fehler einer Berechnung von der Eingabe bis zur Ausgabe setzt sich im Allgemeinen zusammen aus:

- Eingabefehlern (Eingangsfehlern)
- Verfahrensfehlern (Abbruchfehlern, Diskretisierungsfehlern)
- Fortpflanzungsfehlern
- Rechnungsfehlern

1.4.1 Eingabefehler

Eingabefehler sind die Fehler, mit denen die Eingabedaten behaftet sind, z. B. wegen fehlerhafter Messungen oder Rundung. Diese unvermeidbaren Fehler wirken sich auf die Ausgabedaten eines Algorithmus aus. Daher müssen numerische Algorithmen so konzipiert werden, dass der Einfluss von Eingabefehlern möglichst begrenzt wird (siehe auch Abschnitt 1.4.4).

Beispiel 1.30.
Für das Integral

$$I_n = \int_0^1 (1-x)^n \cdot \sin x \, \mathrm{d}x$$

lässt sich mit Hilfe zweimaliger partieller Integration die Rekursionsformel

$$I_n = 1 - n\,(n-1)\,I_{n-2}$$

gewinnen, die für gerades $n \geq 2$, ausgehend von $I_0 = \int_0^1 \sin x \, \mathrm{d}x = 1 - \cos(1)$, nacheinander auf die Werte I_2, I_4, I_6 usw. führt.

In keinem Rechner wird der Wert $\cos(1) = 0.540\,302\,3\ldots$ exakt darstellbar sein. Verschiedene Genauigkeiten bei diesem Eingabedatum haben schon nach wenigen Schritten erstaunliche Konsequenzen:

Eingabewert für $\cos(1)$	Eingabe-ungenauigkeit	resultierender Integralwert I_8	Fehler
exakt	0	$0.011\,027\ldots$	0
10-stellige dezimale Genauigkeit	$9.3 \cdot 10^{-10}$	$0.011\,0261\ldots$	$1.3 \cdot 10^{-6}$
0.54030	$2.3 \cdot 10^{-6}$	0.104	$9.3 \cdot 10^{-2}$
0.540	$3 \cdot 10^{-4}$	12.2	12.2

Man erkennt deutlich, welch verheerende Auswirkungen durch Eingabefehler entstehen können. □

1.4.2 Verfahrensfehler

Verfahrensfehler sind solche, die durch die verwendete numerische Methode verursacht werden. Das Ersatzproblem für eine zu lösende Aufgabe muss so formuliert werden, dass

- es numerisch gelöst werden kann und
- seine Lösung nicht wesentlich von derjenigen des gegebenen Problems abweicht.

Bei einem geeignet formulierten Ersatzproblem wird dessen Lösung eine Näherungslösung für das gegebene Problem sein. Die Differenz zwischen diesen beiden Lösungen heißt der Verfahrensfehler. Dieser Verfahrensfehler hängt in hohem Maße vom gegebenen Problem und von dem ausgewählten Ersatzproblem ab. Der Verfahrensfehler berücksichtigt weder Eingabe- noch Rechnungsfehler im Verfahren selbst.

Beispiele:

1. Die Berechnung eines bestimmten Integrals wird ersetzt durch die Berechnung einer endlichen Summe. Dann ist der Verfahrensfehler die Differenz
$$\left| \int_a^b f(x)\,\mathrm{d}x - \sum_{k=0}^{n} A_k\, f(x_k) \right| .$$
2. Die Berechnung der 1. Ableitung einer Funktion f wird ersetzt durch die Berechnung des vorderen Differenzenquotienten. Dann ist der Verfahrensfehler
$$\left| f'(x) - \frac{f(x+h) - f(x)}{h} \right| .$$

1.4.3 Fehlerfortpflanzung und die Kondition eines Problems

Fehler der Ausgabedaten eines Problems, die durch Fehler der Eingabedaten erzeugt werden, heißen Fortpflanzungsfehler. So lassen sich die Ergebnisse in Beispiel 1.30 dadurch erklären, dass sich ein Eingabefehler ε_0, wenn statt I_0 ein Wert $\widetilde{I}_0$ mit $|\, I_0 - \widetilde{I}_0 \,| \le \varepsilon_0$ genommen wird, fortlaufend vervielfacht: Statt $I_2 = 1 - 2 \cdot 1 \cdot I_0$ berechnet man dann $\widetilde{I}_2 = I_2 - 2\,\varepsilon_0$, anstelle von $I_4 = 1 - 4 \cdot 3 \cdot I_2$ dann $\widetilde{I}_4 = 1 - 4 \cdot 3 \cdot \widetilde{I}_2 = I_4 + 24\,\varepsilon_0$, und entsprechend folgt $\widetilde{I}_6 = I_6 - 720\,\varepsilon_0$ sowie schließlich $\widetilde{I}_8 = I_8 + 40\,320\,\varepsilon_0$ für die berechneten Werte.

Um die Auswirkungen von Fortpflanzungsfehlern allgemein zu untersuchen, nimmt man an, dass das Resultat y eine reellwertige Funktion f sei, die sich aus den Argumenten x_1,x_2,...,x_n berechnen lässt:

$$y = f(x_1, x_2, \ldots, x_n) =: f(\boldsymbol{x}) \quad \text{mit} \quad \boldsymbol{x} = (x_1, x_2, \ldots, x_n)^{\mathsf{T}}.$$

Sind nun statt der Eingabedaten x_i nur Näherungswerte X_i bekannt, so erhält man anstelle des gesuchten Funktionswertes y einen Näherungswert

$$Y = f(X_1, X_2, \ldots, X_n) =: f(\boldsymbol{X}) \;\text{ mit }\; \boldsymbol{X} = (X_1, X_2, \ldots, X_n)^{\mathsf{T}}.$$

Im Folgenden wird eine obere Schranke für den Fortpflanzungsfehler $\Delta_y = y - Y$ bei gegebenen Eingabefehlern $\Delta_{x_i} = x_i - X_i$ der Eingabedaten x_i angegeben.

Satz 1.31.
Es sei $f : G \subset \mathbb{R}^n \to \mathbb{R}$ eine auf einem Gebiet G stetig differenzierbare Funktion, ferner seien $\boldsymbol{x} = (x_1, \ldots, x_n)^{\mathsf{T}} \in G$ und $\boldsymbol{X} = (X_1, \ldots, X_n)^{\mathsf{T}} \in G$.
Dann gibt es ein $\overline{\boldsymbol{x}} = (\overline{x}_1, \ldots, \overline{x}_n)^{\mathsf{T}} \in G$ mit $\overline{x}_i$ zwischen x_i und X_i für $i = 1(1)n$, so dass für den Fortpflanzungsfehler gilt:

$$\Delta_y = y - Y = f(\boldsymbol{x}) - f(\boldsymbol{X}) = \sum_{i=1}^{n} \frac{\partial f(\overline{\boldsymbol{x}})}{\partial x_i} \Delta_{x_i} \;,$$

wobei $\Delta_{x_i} = x_i - X_i$ der Eingabefehler von x_i ist.

Praktisch werden die Eingabefehler nicht exakt bekannt sein, sondern nur obere Schranken ε_{x_i} mit $|\, x_i - X_i \,| \le \varepsilon_{x_i}$ dafür. Ebenso lässt sich, abgesehen von Ausnahmefällen, die „Zwischenstelle“ $\overline{\boldsymbol{x}}$ nicht genau angeben. Deshalb wendet man die Formel aus Satz 1.31 gewöhnlich wie folgt an:

$$|\, \Delta_y \,| \le \sum_{i=1}^{n} \left| \frac{\partial f(\boldsymbol{X})}{\partial x_i} \right| \cdot \varepsilon_{x_i} \tag{1.6}$$

Für kleine Eingabefehler ε_{x_i} liegt auch $\overline{\boldsymbol{x}}$ nahe bei $\boldsymbol{X}$, und wegen der Stetigkeit der partiellen Ableitungen von f bleibt dann die in (1.6) auf der rechten Seite erfolgte Inkorrektheit in überschaubaren Grenzen.

Das Ergebnis aus Satz 1.31 lässt sich leicht auf den relativen Fehler übertragen:

$$\delta_y = \frac{\Delta_y}{y} = \sum_{i=1}^{n} \frac{x_i}{f(\boldsymbol{x})} \cdot \frac{\partial f(\overline{\boldsymbol{x}})}{\partial x_i} \cdot \delta_{x_i} \,, \tag{1.7}$$

wobei $\delta_{x_i} = \dfrac{\Delta_{x_i}}{x_i} = \dfrac{x_i - X_i}{x_i}$ der relative Eingabefehler von x_i ist.
Für die praktische Anwendung heißt das wiederum

$$\left| \frac{\Delta_y}{y} \right| \leq \sum_{i=1}^{n} \left| \frac{X_i}{f(\boldsymbol{X})} \right| \cdot \left| \frac{\partial f(\boldsymbol{X})}{\partial x_i} \right| \cdot \varrho_{x_i}$$

mit den oberen Schranken $\left| \dfrac{x_i - X_i}{x_i} \right| \leq \varrho_{x_i}$ für die relativen Eingabefehler.

Beispiel 1.32.
Man gebe eine Abschätzung des Eingangsfehlers an, der bei der Bestimmung des spezifischen Gewichtes γ eines Zylinders vom Gewicht G, dem Radius r und der Höhe h entsteht, wenn ΔG, Δr die wahren Messfehler von G, r sind und $\Delta\pi$ der wahre Rundungsfehler für π ist; h sei genau gemessen.

Lösung: Es gilt

$$\gamma = \frac{G}{V} = \frac{G}{\pi r^2 h} = \gamma(G, r, \pi, h)\,.$$

Da h genau angenommen wird, setzt man

$$\gamma(G, r, \pi, h) = \gamma^*(G, r, \pi)\,.$$

Mit

$$f(x_1, x_2, x_3) = \gamma^*\,,$$

$$x_1 = G,\ \ x_2 = r,\ \ x_3 = \pi,$$

$$|\Delta G| \leq \varepsilon_G,\ \ |\Delta r| \leq \varepsilon_r, |\Delta\pi| \leq \varepsilon_\pi\,,$$

$$\frac{\partial\gamma^*}{\partial G} = \frac{1}{\pi r^2 h},\ \ \frac{\partial\gamma^*}{\partial r} = -\frac{2G}{\pi r^3 h},\ \ \frac{\partial\gamma^*}{\partial\pi} = -\frac{G}{\pi^2 r^2 h}$$

erhält man nach (1.6)

$$\begin{aligned} |\Delta\gamma^*| \ \leq\ & \frac{1}{(\pi - \varepsilon_\pi)(r - \varepsilon_r)^2 h}\,\varepsilon_G + \frac{2(G + \varepsilon_G)}{(\pi - \varepsilon_\pi)(r - \varepsilon_r)^3 h}\,\varepsilon_r \\ & + \frac{G + \varepsilon_G}{(\pi - \varepsilon_\pi)^2 (r - \varepsilon_r)^2 h}\,\varepsilon_\pi\,. \end{aligned}$$

Bei kleinen Radien wirken sich also alle Messfehler stärker aus als bei großen Radien. □

Die Bestimmung des durch Eingabefehler verursachten Fortpflanzungsfehlers ist Aufgabe der Fehleranalysis. Für eine Eingabe $\boldsymbol{x}$ hängt der relative Fortpflanzungsfehler stark von $\boldsymbol{x}$ und $f(\boldsymbol{x})$ ab, d. h. von dem Problem selbst. Dabei sind die Faktoren

$$K_i = \frac{x_i}{f(\boldsymbol{x})} \frac{\partial f(\overline{\boldsymbol{x}})}{\partial x_i}$$

als Verstärkungsfaktoren der relativen Eingabefehler δ_{x_i} anzusehen. Man erkennt, dass sich Funktionen f mit betragsmäßig kleinen partiellen Ableitungen f_{x_i} bezüglich der

Fehlerfortpflanzung günstig verhalten. Man nennt das Problem, $f(\boldsymbol{x})$ aus $\boldsymbol{x}$ zu berechnen, ein gut konditioniertes Problem für die Funktion f, wenn alle Verstärkungsfaktoren nicht wirklich die Eingabefehler vergrößern, also betragsmäßig in der Größenordnung von 1 liegen. Die Zahlen K_i heißen die *Konditionszahlen* des Problems. Im Idealfall, wenn $|K_i| \leq 1$ ist für alle i, wird keine Verschlechterung der Genauigkeit eintreten. Tatsächlich wäre man schon mit $|K_i| \leq 10^2$ sehr zufrieden, da in diesem Fall der „Output" $f(\boldsymbol{x})$ nur etwa 2 Ziffern an Genauigkeit gegenüber der vorliegenden Eingabegenauigkeit verlieren würde. Der Bezug der Konditionszahl auf die relativen Fehler macht die Betrachtung übrigens unabhängig von der Skalierung der Ein- und Ausgabegrößen.

Für große Probleme ist eine Abschätzung von Fortpflanzungsfehlern und Konditionszahlen sehr kompliziert und selten möglich. In solchen Fällen ist der Einsatz statistischer Fehlerabschätzungen sinnvoll, siehe [HENR1972], Bd.2, S.381.

Im Folgenden werden die relativen Fortpflanzungsfehler für die elementaren arithmetischen Operationen zusammengestellt:

1. *Fortpflanzungsfehler einer Summe*

   Es sei
   $$y = f(x_1, x_2) = x_1 + x_2 \ .$$

   Dann sind $f_{x_1} = f_{x_2} = 1$ und nach (1.7)
   $$\delta_y = \frac{x_1}{x_1 + x_2}\delta_{x_1} + \frac{x_2}{x_1 + x_2}\delta_{x_2} = \sum_{i=1}^{2} K_i \delta_{x_i} \ .$$

   Wenn die Eingabedaten x_1 und x_2 dasselbe Vorzeichen haben, ist die Addition eine gut konditionierte Operation (wegen $|K_i| = |x_i/(x_1 + x_2)| < 1$). Wenn jedoch x_1 und x_2 verschiedene Vorzeichen haben und dem Betrage nach nahezu gleich sind, ist $|x_1 + x_2|$ sehr klein und somit $|K_i| = |x_i|/|x_1 + x_2|$ sehr groß; die Addition ist dann schlecht konditioniert. In diesem Fall gehen bei der Addition tragende Ziffern verloren, ein Effekt, den man *Auslöschung* nennt.

Beispiel 1.33.
Gegeben seien die drei Werte
$$x_1 = 123.454 \text{ Mrd. €},$$
$$x_2 = 123.446 \text{ Mrd. €} \text{ und}$$
$$x_3 = 123.435 \text{ Mrd. €}.$$

Legt man eine 5-stellige dezimale Gleitpunktarithmetik zugrunde, so wird mit
$$X_1 = \mathrm{gl}\,(x_1) = 0.12345 \cdot 10^{12},$$
$$X_2 = \mathrm{gl}\,(x_2) = 0.12345 \cdot 10^{12} \text{ und}$$
$$X_3 = \mathrm{gl}\,(x_3) = 0.12344 \cdot 10^{12}$$

weiter gerechnet. Man erhält so anstelle von
$x_1 - x_2 = x_1 + (-x_2) = \ 8$ Mio. € den Wert 0 und anstelle von
$x_1 - x_3 = x_1 + (-x_3) = 19$ Mio. € den Wert $0.10000 \cdot 10^8$, also 10 Mio.

Die jeweils in der fünften tragenden Stelle liegenden Rundefehler nehmen schon nach einer Rechenoperation Einfluss auf die Größenordnung des Ergebnisses!

Bei der ersten Rechnung $x_1 - x_2$ lauten die Konditionszahlen

$$K_1 = \frac{123.454 \text{ Mrd.}}{8 \text{ Mio.}} = 15431.75 \qquad \text{und} \qquad K_2 = \frac{123.446 \text{ Mrd.}}{8 \text{ Mio.}} = 15430.75\,,$$

und diese größeren Werte sollten dazu anhalten, die mit der Rechnerarithmetik erhaltenen Ergebnisse äußerst kritisch zu überprüfen. □

Der Ausdruck $x-(x-y)$ entspricht korrekt dem Wert y. Mit $x = 10^{30}$ und $y = 10^6$ würde allerdings ein Rechner stattdessen das Ergebnis 0 produzieren, da in der vorgegebenen Auswertungsreihenfolge die Gleitpunktarithmetik den Klammerausdruck zu x werden lässt. In bestimmten Fällen lassen sich durch geschickte Umformungen derartige Gegebenheiten vermeiden.

Beispiel 1.34.

Es soll der Ausdruck $\frac{x}{\sin(1+x)-\sin(1)}$ für x nahe 0 auswertet werden. Setzt man in (Taschen-)Rechnern für x Werte wie 10^{-5}, 10^{-10}, 10^{-15}, 10^{-20} usw. ein, so wird man schon bald eine Fehlermeldung erhalten („Division durch Null"). Dies liegt natürlich daran, dass die beiden Werte im Nenner rechnerintern nicht mehr unterschieden werden können und deshalb die Differenz auf eine 0 führt.

Nutzt man das Additionstheorem

$$\sin(a) - \sin(b) = 2\,\cos\left(\frac{a+b}{2}\right)\,\sin\left(\frac{a-b}{2}\right)$$

mit $a = 1 + x$ und $b = 1$, so folgt daraus

$$\sin(1+x) - \sin(1) = 2\,\cos\left(1+\frac{x}{2}\right)\,\sin\left(\frac{x}{2}\right)\,.$$

Damit gilt die Identität

$$\frac{x}{\sin(1+x)-\sin(1)} = \frac{x}{2\,\cos\left(1+\frac{x}{2}\right)\,\sin\left(\frac{x}{2}\right)}\,.$$

Die Auswertung des umgeformten Ausdrucks erweist sich als numerisch stabil: Für auf 0 zugehende Werte von x liefert auch ein (Taschen-)Rechner Werte in der korrekten Größenordnung von

$$\lim_{x\to 0}\ \frac{x}{\sin(1+x)-\sin(1)} = \frac{1}{\cos(1)} = 1.850\,815\,7\ldots$$

□

Beispiel 1.35.
Die Funktion

$$f(x,y) = 9\,x^4 - y^4 + 2\,y^2$$

besitzt an der Stelle $(x,y) = (10864, 18817)$ den Wert 1.

Eine rechnerische Auswertung in der Reihenfolge $(9\,x^4 - y^4) + 2\,y^2$ schließt Auslöschungseffekte bei der Differenzbildung ein und führt so auf ein Ergebnis in der Größenordnung von $2\,y^2$. Auch die Auswertungsreihenfolge $(9\,x^4 + 2\,y^2) - y^4$ vermeidet keine Auslöschung und liefert wiederum ein fehlerhaftes Resultat, u. U. eine „zufällige“ 0. Erst die Umformung

$$f(x,y) = (3\,x^2 + y^2)\,(3\,x^2 - y^2) + 2\,y^2$$

lässt unangenehme rechnerinterne Verfälschungen außen vor und reproduziert den exakten Wert 1. □

Die letzten Beispiele lassen evident erscheinen, dass man sich keinesfalls auf jedes Rechnerergebnis verlassen darf; man sollte **stets eine Plausibilitätsprüfung** vornehmen! Selbst wenn das Resultat vertrauenswürdig ist, kann man nie allen angegebenen Stellen Glauben schenken.

Einige Rechnersysteme bieten die Möglichkeit, mit unterschiedlicher Mantissenstellenzahl s zu rechnen. Auslöschungseffekte lassen sich umso weiter nach hinten drängen, je größer die Anzahl der mitgeführten Stellen ist, so dass man eine diesbezügliche Einflussnahme zumindest in zweifelhaften Fällen vorteilhaft nutzen kann.

2. Fortpflanzungsfehler eines Produktes

Es sei

$$y = f(x_1, x_2) = x_1 x_2\ .$$

Hier sind $f_{x_1} = x_2$ und $f_{x_2} = x_1$, und somit gilt (wegen $K_i \approx 1$)

$$\delta_y = \frac{x_1}{x_1\ x_2} \cdot \overline{x}_2 \cdot \delta_{x_1} + \frac{x_2}{x_1\ x_2} \cdot \overline{x}_1 \cdot \delta_{x_2} \approx \delta_{x_1} + \delta_{x_2}\ .$$

Also ist die Multiplikation gut konditioniert.

3. Fortpflanzungsfehler eines Quotienten

Es sei

$$y = f(x_1, x_2) = \frac{x_1}{x_2}\ .$$

Hier sind $f_{x_1} = 1/x_2$ und $f_{x_2} = -x_1/x_2^2$, und somit gilt (wegen $K_1 \approx 1$ und $K_2 \approx -1$)

$$\delta_y = \frac{x_1}{x_1/x_2} \cdot \frac{1}{\overline{x}_2} \cdot \delta_{x_1} + \frac{x_2}{x_1/x_2} \cdot \left(\frac{-\overline{x}_1}{\overline{x}_2^2}\right) \cdot \delta_{x_2} \approx \delta_{x_1} - \delta_{x_2}\ .$$

Also ist auch die Division gut konditioniert.

4. Fortpflanzungsfehler von Potenzen

Es sei

$$y = f(x_1) = x_1^p \quad \text{für} \quad p > 0\ ,\ x_1 > 0\ .$$

Hier ist $f_{x_1} = px_1^{p-1}$ und (wegen $K_1 \approx p$) $\delta_y \approx p\delta_{x_1}$; damit sind Wurzeln im Allgemeinen gut konditioniert und Potenzen mäßig schlecht konditioniert für große p. Dies ist einer der Gründe dafür, dass man keine vernünftigen Resultate erwarten kann, wenn versucht wird, ein Polynom

$$P_n(x) = a_0 + a_1x + a_2x^2 + \ldots + a_nx^n$$

auszuwerten, indem Terme der Form $a_k\, x^k$ summiert werden; man sollte stattdessen immer das besser konditionierte Horner-Schema benutzen, in dem abwechselnd multipliziert und addiert, aber nicht potenziert wird (vgl. Abschnitt 3.2).

1.4.4 Rechnungsfehler und numerische Stabilität

Zur Durchführung eines numerischen Verfahrens muss ein Algorithmus formuliert werden.

Definition 1.36. (*Algorithmus*)
Ein Algorithmus ist eine endliche Menge genau beschriebener Anweisungen (arithmetische und logische Operationen und Ausführungshinweise), die in einer bestimmten Reihenfolge auszuführen sind, um mit Hilfe der eingegebenen Daten die gesuchten Ausgabedaten zu ermitteln.

In diesem Sinne wurden bereits zuvor die „Algorithmen“ 1.8, 1.9 und 1.11 formuliert.

Während der Ausführung der Rechenoperationen ergibt sich durch Anhäufung lokaler Rechnungsfehler ein akkumulierter Rechnungsfehler. Die lokalen Rechnungsfehler entstehen z. B. dadurch, dass irrationale Zahlen wie π, e, $\sqrt{2}$ durch endliche Dezimalbrüche (Maschinenzahlen) ersetzt werden; dadurch werden Abbruch- oder Rundungsfehler erzeugt. Hinreichend kleine Größen, die durch Unterlauf entstehen, werden vernachlässigt. Führende genaue Stellen können bei der Subtraktion fast gleich großer Zahlen ausgelöscht werden. Mit der Anzahl der Operationen in einem Algorithmus wächst somit die Gefahr, dass völlig falsche Ergebnisse entstehen.

Algorithmen, die eine Verstärkung und Anhäufung von Rundungsfehlern vermeiden, werden *numerisch stabil* genannt. Es gibt unterschiedliche Definitionen für den Begriff der numerischen Stabilität:

1. Sind sämtliche Rundungsfehler eines Algorithmus von derselben Größenordnung wie der Fehler $f_x \cdot \Delta_x$ für den Eingabefehler Δ_x, so nennt Bauer [BAUE1965] den Algorithmus numerisch stabil.

2. Stewart [STEW1973], S.76, unterscheidet zwischen dem theoretischen Algorithmus f und seiner numerischen Realisation f^*. Er nennt den Algorithmus f^* numerisch stabil, wenn es für jeden Eingabewert x ein benachbartes x^* gibt, so dass $f(x^*)$ dicht bei $f^*(x)$ liegt. Mit dieser Definition nähert für gut konditionierte Probleme das berechnete Ergebnis eines stabilen Algorithmus die exakte Lösung auch gut an.
   Wenn f^* numerisch stabil ist im Sinne von Stewart, dann gibt es für ein schlecht konditioniertes Problem ein x^* nahe bei x, für welches sich $f(x^*)$ und $f^*(x^*)$ in gleichem Maße von $f(x)$ unterscheiden.

3. Eine sehr häufig benutzte Definition für numerische Stabilität fordert die Existenz eines x^* nahe bei x, so dass $f^*(x^*) = f(x)$ ist, vgl. [WILK1969].

Beispiel 1.37.
Hat man eine Gleichung der Form

$$y_{n+1} = a\, y_n + b\, y_{n-1} \qquad (n \in \mathbb{N}),$$

auch Differenzengleichung genannt, mit reellen Konstanten a, b und $n \in \mathbb{N}$, so lässt sich jeder Wert y_n hierüber nach Angabe von zwei konkreten „Startwerten“ y_0 und y_1 berechnen. Eine allgemeine Lösung der Differenzengleichung kann mit Hilfe des Ansatzes

$$y_n = \xi^n \tag{1.8}$$

mit einem festen Wert ξ gewonnen werden. Einsetzen ergibt

$$\xi^{n+1} = a\, \xi^n + b\, \xi^{n-1}$$

bzw. nach Division durch $\xi^{n-1} \neq 0$

$$\xi^2 - a\, \xi - b = 0\,.$$

Die Nullstellen dieser quadratischen Gleichung sind gegeben durch

$$\xi_{1,2} = \frac{a}{2} \pm \sqrt{\frac{a^2}{4} + b}\,,$$

und Fehler in der Angabe von ξ werden durch das Potenzieren in (1.8) nur dann nicht verstärkt, wenn

$$\left| \frac{a}{2} \pm \sqrt{\frac{a^2}{4} + b} \right| \leq 1$$

erfüllt ist. Dann ist ein über obige Differenzengleichung gegebener Algorithmus numerisch stabil. □

Die Verwendung instabiler Algorithmen ist praktisch sinnlos. Aber selbst dann, wenn ein stabiler Algorithmus verwendet wird, hat es natürlich keinen Sinn, zwar mit einer exakten Prozedur zu arbeiten, aber die Berechnung mit nur geringer Genauigkeit auszuführen; es werden dann natürlich große Rechnungsfehler auftreten.

Gewöhnlich sind numerisch stabile Algorithmen und gut konditionierte Probleme notwendig, um überhaupt zufriedenstellende Resultate erreichen zu können; diese Bedingung ist jedoch keineswegs hinreichend, da das Instrument „Computer“, mit dem das Ergebnis erzeugt wird, mit begrenztem Speicherplatz und begrenzter Zeit arbeitet. Computer-Operationen setzen sich zusammen aus dem Schreiben in den Speicher, dem Lesen aus dem Speicher, Overhead (z. B. zusätzlicher Verwaltungsaufwand) und den arithmetischen Operationen.

Alle arithmetischen Operationen (Potenzen, Wurzeln, Auswertung trigonometrischer Funktionen u.a.m.) arbeiten mit internen Prozeduren, die wiederum nur Additionen und Multiplikationen benutzen. Deshalb ist das Zählen der elementaren Operationen (wie Anzahl benötigter Multiplikationen) eine Möglichkeit für den Vergleich unterschiedlicher Algorithmen zur Lösung einer und derselben Problemstellung. Additionen bleiben dabei meist unberücksichtigt, weil sie in der Regel nur einen Bruchteil der Rechenzeit im Vergleich zu den Multiplikationen und Divisionen (also den *Punktoperationen*) benötigen. Spezialrechner sind heutzutage allerdings schon in der Lage, eine Multiplikation in etwa in der Rechenzeit einer Addition zu erledigen. Bei sehr vielen Algorithmen wird insbesondere auf die Anzahl der erforderlichen Punktoperationen hingewiesen.

Ergänzende Literatur zu Kapitel 1

[BART2004]; [BRON1991], 2.1.1, 2.1.2; [CONT1987], 1; [DEUF2002] Bd.1, 2; [ENGE1996], Kap.1; [HAMM1994], 1.; [HILD1987], 1.; [OVER2001]; [PREU2001], 1; [RICE1993], Kap.3; [STOE1999] Bd.1, Kap.4; [STOE2002], 1; [TORN1990] Bd.1, 1.7, 1.8; [UBER1995], 2; [WERN1993], I §6.

Kapitel 2

Numerische Verfahren zur Lösung nichtlinearer Gleichungen

2.1 Aufgabenstellung und Motivation

Ist f eine in einem abgeschlossenen Intervall $I = [a, b]$ stetige und reellwertige Funktion, so heißt eine Zahl $\xi \in I$ eine *Nullstelle der Funktion* f oder eine *Lösung der Gleichung*

$$f(x) = 0\,, \tag{2.1}$$

falls $f(\xi) = 0$ ist.

Wenn f ein *algebraisches Polynom* der Form

$$f(x) \equiv P_n(x) = \sum_{j=0}^{n} a_j x^j, \quad a_j \in \mathbb{R}\,, \quad a_n \neq 0\,, \quad n \in \mathbb{N}$$

ist, heißt die Gleichung (2.1) *algebraisch*, und die natürliche Zahl n heißt der *Grad* des Polynoms bzw. der algebraischen Gleichung. Jede Gleichung (2.1), die nicht algebraisch ist, heißt *transzendent* (z. B. $\ln x - 1/x = 0$; $x - \sin x = 0$).

In diesem Kapitel werden Verfahren zur Bestimmung einfacher und mehrfacher Nullstellen $\xi \in I$ von f vorgestellt. Dabei wird zwischen den klassischen Iterationsverfahren (allgemeines Iterationsverfahren, Newton-Verfahren, Sekantenverfahren) und den sogenannten Einschlussverfahren (Bisektion, Pegasus-Verfahren, Verfahren von Anderson-Björck, Verfahren von King) unterschieden. Die angegebenen Einschlussverfahren benötigen zwei Startwerte, in denen die Funktion f unterschiedliche Vorzeichen hat. Die Startwerte schließen dann (mindestens) eine Nullstelle ungerader Ordnung von f ein. Dieser Einschluss bleibt im Laufe der Rechnung erhalten. Diese Verfahren sind jedoch grundsätzlich unter Anwendung von Satz 2.3 auch im Falle von Nullstellen gerader Ordnung anwendbar. Einschlussverfahren höherer Konvergenzordnung sind im Allgemeinen den klassischen Iterationsverfahren vorzuziehen (vgl. Abschnitt 2.9). Verfahren zur Bestimmung sämtlicher Nullstellen algebraischer Polynome ohne Kenntnis von Startwerten werden im Kapitel 3 behandelt.

Wo in den Anwendungen treten nichtlineare Gleichungen auf? Die meisten derartigen Probleme führen nicht direkt auf eine transzendente Gleichung oder eine Polynomgleichung, sondern diese treten an irgendeiner Stelle des Lösungsprozesses auf. So stößt man etwa auf Polynomgleichungen bei der Lösung von linearen Differentialgleichungen n-ter Ordnung mit konstanten Koeffizienten, bei der Berechnung der Eigenwerte von Matrizen und bei der Bestimmung der Eigenfrequenzen linearer Schwingungssysteme mit n Freiheitsgraden. Transzendente Gleichungen treten beispielsweise bei der Berechnung der kritischen Drehzahl einer Welle in der Gestalt

$$f(x) = \cos x \cosh x \pm 1 = 0 \quad \text{oder} \quad f(x) = \tan x - \tanh x = 0$$

auf. Bei der Berechnung des Druckverlustes in einer Rohrströmung muss man den Rohrreibungsbeiwert λ für hydraulisch glatte Rohre bei turbulenter Strömung nach dem universellen Prandtlschen Widerstandsgesetz bei vorgegebener Reynoldszahl Re aus der transzendenten Gleichung

$$f(\lambda) = \frac{1}{\sqrt{\lambda}} - 2\lg\left(\mathrm{Re}\sqrt{\lambda}\right) + 0.8 = 0$$

ermitteln. Ebenfalls auf eine transzendente Gleichung stößt man bei der Betrachtung der Ausstrahlung eines vollkommenen „schwarzen Körpers“. Wird ein schwarzer Körper erhitzt, so sendet er elektromagnetische Wellen verschiedener Wellenlänge λ (Wärmestrahlen, Licht usw.) aus.

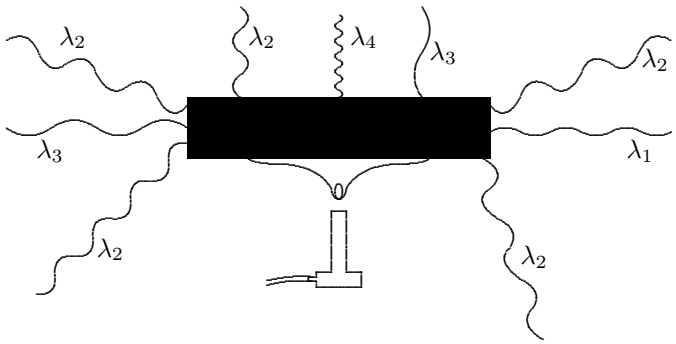


Abb. 2.1. Ausstrahlung eines vollkommenen schwarzen Körpers

Bei einer bestimmten Temperatur T liegt das Maximum der emittierten Strahlung $E(\lambda, T)$ bei $\lambda_{\max}$. Mit steigender Temperatur verschiebt sich die Stelle maximaler Emission nach kürzeren Wellenlängen; es gilt (*Wiensches Verschiebungsgesetz*)

$$\lambda_{\max}(T) = \frac{\alpha}{z \cdot T}$$

mit $\alpha = 14.3881 \cdot 10^{-3}\,\mathrm{m\,K}$. Die Konstante z ist die Lösung der transzendenten Gleichung

$$\mathrm{e}^{-z} = 1 - \frac{1}{5}z \qquad \text{bzw.} \qquad f(z) = \mathrm{e}^{-z} - 1 + \frac{1}{5}z = 0\,, \quad z \in \mathbb{R}\,, \quad z \neq 0\,.$$

Mit dem so ermittelten $z \approx 4.9651$ lässt sich die Formel für $\lambda_{\max}(T)$ angeben. Aus dieser Formel kann dann z. B. ein Näherungswert für die absolute Temperatur der Sonnenober-

fläche bestimmt werden, wobei für das Sonnenspektrum nach Langley das Maximum der Strahlung bei $\lambda_{\max} = 5 \cdot 10^{-7}\,\mathrm{m}$ liegt.

Weitere Anwendungsbeispiele sind in Abschnitt 2.8 zu finden.

2.2 Definitionen und Sätze über Nullstellen

Im Anschluss werden einige Definitionen und Sätze über Nullstellen angegeben, die in den folgenden Abschnitten benötigt werden. Die Beweise findet man in jedem Werk zur Analysis.

Definition 2.1.
Eine Nullstelle ξ einer Funktion $f \in \mathrm{C}[a,b]$ heißt *j-fache Nullstelle* oder Nullstelle der Ordnung j ($j \in \mathbb{N}$), falls f sich auf $[a,b]$ in der Form

$$f(x) = (x-\xi)^j h(x)$$

darstellen lässt mit einer stetigen Funktion h, für die $h(\xi) \neq 0$ ist.
Im Fall $j = 1$ heißt ξ *einfache Nullstelle*, für $j \geq 2$ *mehrfache Nullstelle*.

Ist ξ eine Nullstelle ungerader Ordnung, so hat f in $x = \xi$ einen Vorzeichenwechsel; ist ξ von gerader Ordnung, so berührt der Graph von f die x-Achse, und es gibt keinen Vorzeichenwechsel.

Satz 2.2.
Die Funktion f sei im Intervall I j-mal stetig differenzierbar ($j \in \mathbb{N}$). Dann ist $\xi \in I$ genau dann eine j-fache Nullstelle von f , wenn gilt

$$f(\xi) = f'(\xi) = \ldots = f^{(j-1)}(\xi) = 0\,, \quad f^{(j)}(\xi) \neq 0\,.$$

Mit der Aussage des folgenden Satzes 2.3 lassen sich Einschlussverfahren auch zur Berechnung von mehrfachen Nullstellen einsetzen, insbesondere auch von Nullstellen gerader Ordnung (also ohne Vorzeichenwechsel).

Satz 2.3.
Ist $\xi \in I$ eine j-fache Nullstelle von f, $j \geq 2$, und ist f $(j+1)$-mal stetig differenzierbar, so ist ξ eine einfache Nullstelle von g mit

$$g(x) = \frac{f(x)}{f'(x)}\,.$$

Beweis. Nach Satz 2.2 gelten für $j \geq 2$ $f(\xi) = f'(\xi) = \ldots = f^{(j-1)}(\xi) = 0$ und $f^{(j)}(\xi) \neq 0$. Zu zeigen ist, dass für

$$g(x) = \frac{f(x)}{f'(x)} \quad \text{und} \quad g'(x) = 1 - \frac{f(x)\, f''(x)}{{f'}^2(x)}$$

nach Satz 2.2 gelten

$$g(\xi) = 0 \quad \text{und} \quad g'(\xi) \neq 0\,.$$

Für den Nachweis werden die Taylorentwicklungen von f, f' und f'' an der Stelle ξ benötigt (sie finden auch in Abschnitt 2.5.3 Verwendung).

$$\begin{aligned} f(x) &= \frac{(x-\xi)^j}{j!}\, f^{(j)}(\xi) + O\big((x-\xi)^{j+1}\big) = (x-\xi)^j\, h_0(x) && (2.2)\\ &\quad \text{mit} \quad h_0(x) = \frac{1}{j!}\, f^{(j)}(\xi) + O(x-\xi) \quad \text{und} \\ h_0(\xi) &= \frac{1}{j!}\, f^{(j)}(\xi) \neq 0\,; && (2.3) \end{aligned}$$

$$\begin{aligned} f'(x) &= \frac{(x-\xi)^{j-1}}{(j-1)!}\, f^{(j)}(\xi) + O\big((x-\xi)^j\big) = (x-\xi)^{j-1}\, h_1(x) && (2.4)\\ &\quad \text{mit} \quad h_1(x) = \frac{1}{(j-1)!}\, f^{(j)}(\xi) + O(x-\xi) \quad \text{und} \\ h_1(\xi) &= \frac{1}{(j-1)!}\, f^{(j)}(\xi) \neq 0\,; && (2.5) \end{aligned}$$

$$\begin{aligned} f''(x) &= \frac{(x-\xi)^{j-2}}{(j-2)!}\, f^{(j)}(\xi) + O\big((x-\xi)^{j-1}\big) = (x-\xi)^{j-2}\, h_2(x) && (2.6)\\ &\quad \text{mit} \quad h_2(x) = \frac{1}{(j-2)!}\, f^{(j)}(\xi) + O(x-\xi) \quad \text{und} \\ h_2(\xi) &= \frac{1}{(j-2)!}\, f^{(j)}(\xi) \neq 0\,. && (2.7) \end{aligned}$$

Mit (2.2) und (2.4) ergibt sich

$$g(x) = \frac{f(x)}{f'(x)} = (x-\xi)\, \frac{h_0(x)}{h_1(x)}\,.$$

Mit (2.3) und (2.5) ist

$$\frac{h_0(\xi)}{h_1(\xi)} = \frac{f^{(j)}(\xi)\,(j-1)!}{j!\, f^{(j)}(\xi)} = \frac{1}{j} \neq 0\,; \qquad (2.8)$$

also ist $g(\xi) = 0$.

Weiter ist mit (2.2), (2.4) und (2.6)

$$g'(x) = 1 - \frac{f(x)\, f''(x)}{{f'}^2(x)} = 1 - \frac{h_0(x)\, h_2(x)}{h_1^2(x)}\,.$$

Für $g'(\xi)$ ergibt sich mit (2.3), (2.5) und (2.7)

$$g'(\xi) = 1 - \frac{h_0(\xi)\, h_2(\xi)}{h_1^2(\xi)} = 1 - \frac{(j-1)!\,(j-1)!}{j!\,(j-2)!} = 1 - \frac{j-1}{j} = \frac{1}{j} \neq 0\,.$$

Die j-fache Nullstelle ξ von f ist also einfache Nullstelle von g. □

Satz 2.4. (*Satz von Bolzano, Zwischenwertsatz*)
Sei f in $I = [a, b]$ stetig mit $f(a) \cdot f(b) < 0$. Dann besitzt f in (a, b) mindestens eine Nullstelle ξ ungerader Ordnung.

2.3 Allgemeines Iterationsverfahren

2.3.1 Konstruktionsmethode und Definition

Anstelle der Gleichung $f(x) = 0$ wird eine Gleichung der Form

$$x = \varphi(x) \tag{2.9}$$

betrachtet. Dabei sei

$$\varphi : I \to \mathbb{R} \qquad \text{mit} \qquad x \mapsto \varphi(x)$$

eine in einem abgeschlossenen Intervall I stetige Funktion, und $\xi \in I$ heißt eine Lösung von (2.9) bzw. ein Fixpunkt der Abbildung φ, wenn $\varphi(\xi) = \xi$ ist; darum heißt (2.9) auch *Fixpunktgleichung*.

Die Untersuchung von Gleichungen der Form $x = \varphi(x)$ bedeutet keine Beschränkung der Allgemeinheit, denn es gilt der

Hilfssatz 2.5.
Sind f und g stetige Funktionen in einem abgeschlossenen Intervall I und ist $g(x) \neq 0$ für alle $x \in I$, dann besitzen die Gleichungen $f(x) = 0$ und $x = \varphi(x)$ mit

$$\varphi(x) := x - f(x)g(x) \tag{2.10}$$

im Intervall I dieselben Lösungen, d. h. die beiden Gleichungen sind äquivalent.

Beweis. Ist $\xi \in I$ Lösung von $f(x) = 0$, so folgt wegen $f(\xi) = 0$ aus (2.10) $\varphi(\xi) = \xi$. Ist umgekehrt $\xi \in I$ Lösung von $x = \varphi(x)$, so folgt wegen $\xi = \varphi(\xi)$ aus (2.10) $f(\xi)g(\xi) = 0$; wegen $g(\xi) \neq 0$ ist also $f(\xi) = 0$. □

Jede geeignete Wahl von g liefert eine zu $f(x) = 0$ äquivalente Gleichung $x = \varphi(x)$. Häufig kann eine Gleichung $f(x) = 0$ auf die Form $x = \varphi(x)$ gebracht werden, indem irgendeine Auflösung nach x vorgenommen wird.

Beispiel 2.6.

Gegeben: Die algebraische Gleichung

$$f(x) = x^2 + x - 2 = 0$$

mit den Lösungen $\xi_1 = 1$ und $\xi_2 = -2$.

Gesucht: Zur gegebenen Gleichung äquivalente Gleichungen der Form $x = \varphi(x)$.

Lösung: Durch verschiedenartige Umformung bzw. Auflösung nach x erhält man

$$\begin{aligned} &\text{(I)} && x = 2 - x^2 = \varphi(x)\,, \\ &\text{(II)} && x = \sqrt{2-x} = \varphi(x)\,, && x \le 2\,, \\ &\text{(III)} && x = \tfrac{2}{x} - 1 = \varphi(x)\,, && x \neq 0\,. \end{aligned}$$

Bei der Angabe eines Intervalls I, in dem die Gleichungen äquivalent zur gegebenen sind, müssen die Einschränkungen für x berücksichtigt werden. □

Nun sei eine Gleichung der Form $x = \varphi(x)$ mit dem zugehörigen Intervall I gegeben. Dann konstruiert man mit Hilfe eines *Startwertes* $x^{(0)} \in I$ eine Zahlenfolge $\{x^{(\nu)}\}$ nach der Vorschrift

$$x^{(\nu+1)} := \varphi(x^{(\nu)})\,, \quad \nu = 0, 1, 2, \ldots \tag{2.11}$$

Diese Folge lässt sich nur dann sinnvoll konstruieren, wenn für $\nu = 0, 1, 2, \ldots$

$$x^{(\nu+1)} = \varphi(x^{(\nu)}) \in I$$

ist, da φ nur für $x \in I$ erklärt ist. Durch φ muss also eine Abbildung des Intervalls I in sich gegeben sein, d. h. der Graph von $y = \varphi(x)$ muss im Quadrat

$$Q = \{(x, y) | x \in I, y \in I\}$$

liegen; in der Abbildung 2.2 ist $I = [a, b]$.

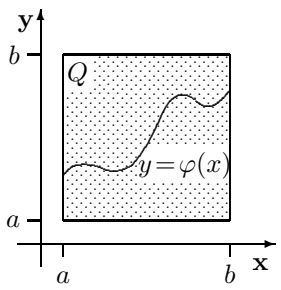


Abb. 2.2.

Wenn die Folge $\{x^{(\nu)}\}$ konvergiert, d. h. wenn die Zahlen $x^{(1)}, x^{(2)}, x^{(3)}, \ldots$ gegen ξ streben, somit

$$\lim_{\nu\to\infty} x^{(\nu)} = \xi$$

ist, dann ist ξ eine Lösung der Gleichung (2.9). Es gilt wegen der Stetigkeit von φ

$$\xi = \lim_{\nu\to\infty} x^{(\nu)} = \lim_{\nu\to\infty} x^{(\nu+1)} = \lim_{\nu\to\infty} \varphi(x^{(\nu)}) = \varphi(\lim_{\nu\to\infty} x^{(\nu)}) = \varphi(\xi)\,.$$

Ein solches *Verfahren der schrittweisen Annäherung* wird *Iterationsverfahren* genannt. Die Vorschrift (2.11) heißt *Iterationsvorschrift*; sie stellt für jedes feste ν einen *Iterationsschritt* dar. Die Funktion φ wird *Schrittfunktion* genannt. Die Folge $\{x^{(\nu)}\}$ heißt *Iterationsfolge*.

Die Iterationsschritte (2.11) für $\nu = 0(1)N$ bilden zusammen mit dem Startwert $x^{(0)}$ das algorithmische Schema des Iterationsverfahrens:

$$\left\{\begin{array}{ll} x^{(0)} & = \text{Startwert}\,, \\ x^{(1)} & = \varphi(x^{(0)})\,, \\ x^{(2)} & = \varphi(x^{(1)})\,, \\ \cdot & \\ \cdot & \\ \cdot & \\ x^{(N+1)} & = \varphi(x^{(N)})\,. \end{array}\right. \tag{2.12}$$

Dabei muss $\varphi(x^{(\nu)}) \in I$ gelten für $\nu = 0(1)N$.

Beispiel 2.7. (Fortsetzung von Beispiel 2.6)
Für die Umformung (I): $x = 2 - x^2$ der Gleichung $x^2 + x - 2 = 0$ lautet die Iterationsvorschrift (2.11)

$$x^{(\nu+1)} = \varphi(x^{(\nu)}) = 2 - (x^{(\nu)})^2, \qquad \nu = 0, 1, 2, \ldots,$$

und es sei $I = [-50, 0]$. Das algorithmische Schema (2.12) des Iterationsverfahrens lautet mit dem Startwert $x^{(0)} = -3$:

$$\begin{array}{lcl} x^{(0)} & = & -3 \in I\,, \\ x^{(1)} & = & \varphi(x^{(0)}) = 2 - 3^2 = -7 \in I\,, \\ x^{(2)} & = & \varphi(x^{(1)}) = 2 - 7^2 = -47 \in I\,, \\ x^{(3)} & = & \varphi(x^{(2)}) = 2 - 47^2 = -2207 \notin I\,. \end{array}$$

Der Verlauf der Rechnung zeigt, dass die so konstruierte Folge $\{x^{(\nu)}\}$ nicht gegen die Lösung $\xi_2 = -2 \in I$ konvergiert.

Mit der Umformung (III): $x = \frac{2}{x} - 1$ der Gleichung $x^2 + x - 2 = 0$ und $I = [-3, -1]$ erhält man dagegen die Iterationsvorschrift

$$x^{(\nu+1)} = \varphi(x^{(\nu)}) = \frac{2}{x^{(\nu)}} - 1\,, \qquad \nu = 0, 1, 2, \ldots,$$

und mit dem Startwert $x^{(0)} = -3$ das algorithmische Schema:

$$
\begin{aligned}
x^{(0)} &= -3 \in I\,,\\
x^{(1)} &= \tfrac{2}{-3} - 1 = -\tfrac{5}{3} = -1.6666667 \in I\,,\\
x^{(2)} &= -2.2000000 \in I\,,\\
x^{(3)} &= -1.9090909 \in I\,,\\
x^{(4)} &= -2.0476190 \in I\,.
\end{aligned}
$$

Diese vier Iterationsschritte zeigen bereits, dass sich die so konstruierte Folge der Lösung $\xi_2 = -2$ immer mehr nähert, d. h. gegen die gesuchte Lösung konvergiert. □

Es stellt sich also die Frage nach Bedingungen für die Konvergenz einer Iterationsfolge. Die folgenden Aussagen über die Existenz einer Lösung der Gleichung (2.9) und deren Eindeutigkeit dienen der Beantwortung dieser Frage. Es wird sich zeigen, dass die Bedingungen für die Existenz und Eindeutigkeit auch hinreichend für die Konvergenz der mit (2.11) konstruierten Iterationsfolge sind.

2.3.2 Existenz einer Lösung und Eindeutigkeit der Lösung

Zum Nachweis der Existenz einer Lösung $\xi \in I = [a, b]$ der Gleichung $x = \varphi(x)$ folgen nun lediglich geometrische Überlegungen; zum analytischen Nachweis s. [HENR1972] Bd.1, S. 85/87. Man erhält ξ als Abszisse des Punktes, in dem die Graphen der Gerade $y = x$ und der Funktion $y = \varphi(x)$ sich in dem Quadrat Q schneiden. Es wird also davon ausgegangen, dass für $x \in I$ auch $\varphi(x) \in I$ ist. Wenn $a \neq \varphi(a)$ und $b \neq \varphi(b)$ sind, müssen $\varphi(a) > a$ und $\varphi(b) < b$ sein. Die Punkte $\big(a, \varphi(a)\big)$ und $\big(b, \varphi(b)\big)$ liegen also auf verschiedenen Seiten der Diagonale $y = x$ im Quadrat Q. Ist φ stetig, so garantiert der stetige Verlauf des Graphen von $y = \varphi(x)$ in Q die Existenz von mindestens einem Schnittpunkt mit der Gerade $y = x$ (Abb. 2.3(a)). Ist φ dagegen nicht stetig, so existiert nicht notwendig ein solcher Schnittpunkt (Abb. 2.3(b)).

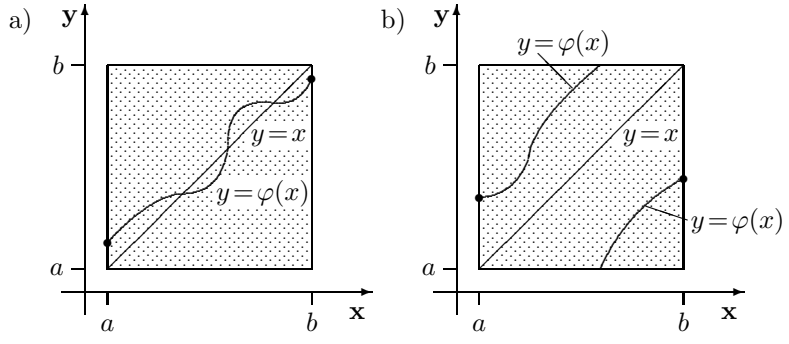


Abb. 2.3. (**a**) φ stetig; (**b**) φ nicht stetig

Satz 2.8. (*Existenzsatz*)
Die in dem endlichen, abgeschlossenen Intervall I definierte Funktion $\varphi : I \to \mathbb{R}$ erfülle die folgenden Bedingungen:

(i) $\varphi(x) \in I$ für alle $x \in I$,
(ii) φ ist stetig in I.

Dann besitzt die Gleichung $x = \varphi(x)$ in I mindestens eine Lösung ξ.

Anhand eines Beispiels wird gezeigt, dass die Stetigkeit (ii) für die Existenz einer Lösung $\xi \in I$ allein nicht hinreicht.

Beispiel 2.9.

Gegeben: Die Gleichung $e^x = 0$, $I = [-a, 0]$, $a > 0$, von der bekannt ist, dass sie keine endliche Lösung besitzt.

Gesucht: Eine dazu äquivalente Gleichung der Form $x = \varphi(x)$.

Lösung: Mit $g(x) = 1$ ergibt sich gemäß (2.10) die Gleichung:

$$x = \varphi(x) = x - e^x.$$

Die Funktion φ ist stetig in $[-a, 0]$. Wegen $\varphi(-a) = -a - 1/e^a < -a$ ist $\varphi(-a) \notin I$ und somit die Bedingung (i) des Satzes 2.8 nicht erfüllt. □

Die Frage nach der Eindeutigkeit einer Lösung der Gleichung $x = \varphi(x)$ kann man beantworten, wenn φ in I einer sogenannten *Lipschitzbedingung* genügt. Eine Funktion φ heißt *lipschitzbeschränkt*, wenn es eine Konstante L mit $0 \le L < 1$ gibt, so dass

$$|\varphi(x) - \varphi(x')| \le L|x - x'| \qquad \text{für alle } x, x' \in I \tag{2.13}$$

gilt. L wird *Lipschitzkonstante* genannt, und (2.13) heißt eine *Lipschitzbedingung* für die Funktion φ. Eine differenzierbare Funktion φ ist sicher lipschitzbeschränkt, wenn

$$|\varphi'(x)| \le L < 1 \qquad \text{für alle } x \in I \tag{2.14}$$

gilt; denn nach dem Mittelwertsatz der Differentialrechnung ist

$$\varphi(x) - \varphi(x') = \varphi'(\eta)(x - x'), \quad \eta \in (x, x') \subset I,$$

woraus beim Übergang zu den Beträgen und mit (2.14) folgt

$$|\varphi(x) - \varphi(x')| = |\varphi'(\eta)|\,|x - x'| \le L|x - x'|.$$

Abbildungen φ, für die eine Lipschitzbedingung (2.13) bzw. (2.14) gilt, werden auch als *kontrahierende Abbildungen* bezeichnet, weil der Abstand $|\varphi(x) - \varphi(x')|$ der Bilder kleiner ist als der Abstand $|x - x'|$ der Urbilder.

Satz 2.10. (*Eindeutigkeitssatz*)
Die Funktion $\varphi : I \to \mathbb{R}$ genüge im Intervall I einer Lipschitzbedingung (2.13) oder (2.14). Dann besitzt die Gleichung $x = \varphi(x)$ in I höchstens eine Lösung ξ.

Beweis. Angenommen, es gibt zwei Lösungen $\xi_1,\ \xi_2$ im Intervall I, so dass also

$$\xi_1 = \varphi(\xi_1) \quad \text{und} \quad \xi_2 = \varphi(\xi_2) \tag{2.15}$$

gelten. Dann folgt mit (2.13) und (2.15)

$$|\xi_1 - \xi_2| = |\varphi(\xi_1) - \varphi(\xi_2)| \le L|\xi_1 - \xi_2|$$

und daraus

$$(1 - L)\,|\xi_1 - \xi_2| \le 0\,.$$

Wegen $1 - L > 0$ ist $|\xi_1 - \xi_2| \le 0$, also kann nur $|\xi_1 - \xi_2| = 0$ sein und damit $\xi_1 = \xi_2$. Die Gleichung $x = \varphi(x)$ besitzt also höchstens eine Lösung $\xi \in I$. □

Da eine Funktion φ, die in I einer Lipschitzbedingung genügt, überall in I stetig ist (die Umkehrung gilt nicht, d. h. nicht jede stetige Funktion genügt einer Lipschitzbedingung) und da die Stetigkeit von φ zusammen mit der Bedingung (i) des Satzes 2.8 hinreichend ist für die Existenz einer Lösung ξ in I, gilt mit Satz 2.10 weiter der

Satz 2.11. (*Existenz- und Eindeutigkeitssatz*)
Die in dem endlichen, abgeschlossenen Intervall I definierte Funktion $\varphi : I \to \mathbb{R}$ erfülle die folgenden Bedingungen:

(i) $\varphi(x) \in I$ für alle $x \in I$.

(ii) φ ist in I lipschitzbeschränkt, d. h. φ genüge für alle $x, x' \in I$ einer Lipschitzbedingung $|\varphi(x)-\varphi(x')| \le L|x-x'|$ mit $0 \le L < 1$ oder, falls φ in I differenzierbar ist, einer Bedingung $|\varphi'(x)| \le L < 1$.

Dann besitzt die Gleichung $x = \varphi(x)$ in I genau eine Lösung ξ.

Beispiel 2.12. (Fortsetzung von Beispiel 2.9)
Die Funktion $\varphi(x) = x - \mathrm{e}^x$ genügt in $I = [-a, 0]$ einer Lipschitzbedingung $|\varphi'(x)| \le L < 1$, da wegen $0 < \mathrm{e}^x \le 1$ für $x \in I$ mit $\varphi'(x) = 1 - \mathrm{e}^x$ gilt $|\varphi'(x)| = |1 - \mathrm{e}^x| < 1$.

Falls eine Lösung existieren würde, wäre sie nach Satz 2.10 eindeutig bestimmt. Im vorliegenden Falle existiert aber keine Lösung, da die Bedingung (i) in Satz 2.8 bzw. 2.11 nicht erfüllt ist. □

2.3.3 Konvergenz eines Iterationsverfahrens

2.3.3.1 Heuristische Betrachtungen

Die Funktion $\varphi : I \to \mathbb{R}$ sei in I differenzierbar. Dann ergibt sich anschaulich, dass die durch die Iterationsvorschrift $x^{(\nu+1)} = \varphi(x^{(\nu)})$ definierte Folge $\{x^{(\nu)}\}$ konvergiert, wenn φ die Bedingungen des Satzes 2.11 für die Existenz und Eindeutigkeit einer Lösung der Gleichung $x = \varphi(x)$ erfüllt:

$$\begin{array}{lll} \text{(i)} & \varphi(x) \in I & \text{für alle } x \in I, \\ \text{(ii)} & |\varphi'(x)| \le L < 1 & \text{für alle } x \in I. \end{array}$$

Die Konvergenz ist für $0 \le \varphi'(x) < 1$ monoton (Abb. 2.4) und für $-1 < \varphi'(x) \le 0$ alternierend (Abb. 2.5).

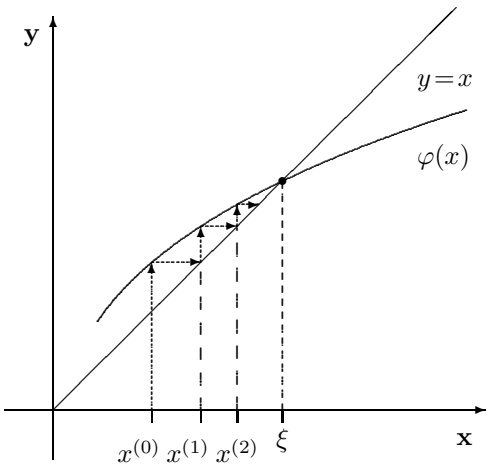


Abb. 2.4. Monotone Konvergenz, $0 \le \varphi'(x) < 1$

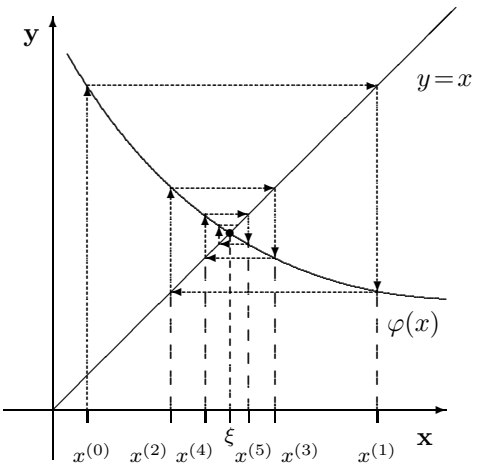


Abb. 2.5. Alternierende Konvergenz, $-1 < \varphi'(x) \le 0$

Je kleiner der Betrag der 1. Ableitung der Schrittfunktion φ ist, desto schneller konvergiert die Iterationsfolge gegen ξ.

Für $|\varphi'(x)| > 1$ divergiert das Verfahren, wie aus den Abbildungen 2.6 und 2.7 zu erkennen ist.

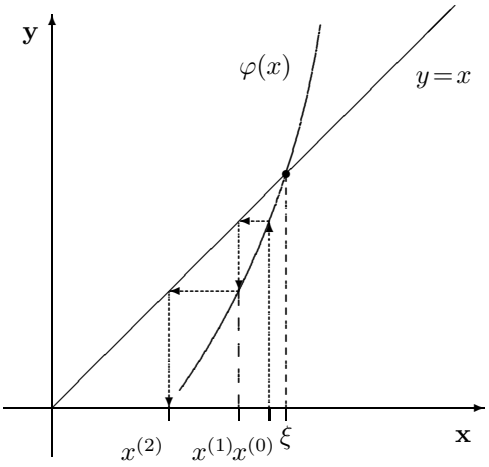


Abb. 2.6. Divergenz, $1 < \varphi'(x)$

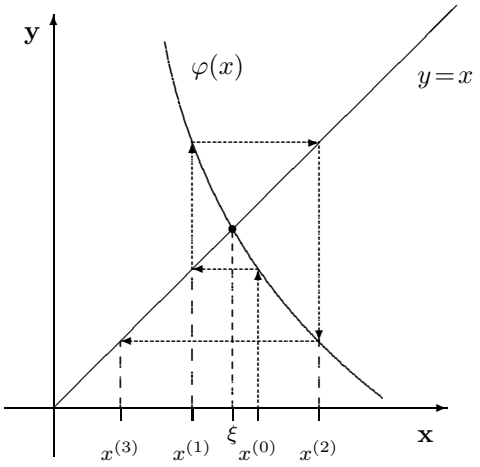


Abb. 2.7. Divergenz, $\varphi'(x) < -1$

Beispiel 2.13. (Fortsetzung von Beispiel 2.6)
Ausgehend von der Umformung (I): $x = 2 - x^2 = \varphi(x)$ der Gleichung $x^2 + x - 2 = 0$ wird $\varphi'(x) = -2x$ untersucht (vgl. Abb. 2.8). Für den Startwert $x^{(0)} = -3$ ist $\varphi'(-3) = 6 > 1$, d. h. die Bedingung $|\varphi'(x)| < 1$ ist bereits für den gewählten Startwert verletzt. Da die Bedingung $|\varphi'(x)| < 1$ nur für $|x| < \frac{1}{2}$ erfüllt werden kann, ist die bei der Umformung (I) entstehende Funktion φ als Schrittfunktion für die Iteration zur Bestimmung beider Lösungen 1 und -2 ungeeignet.

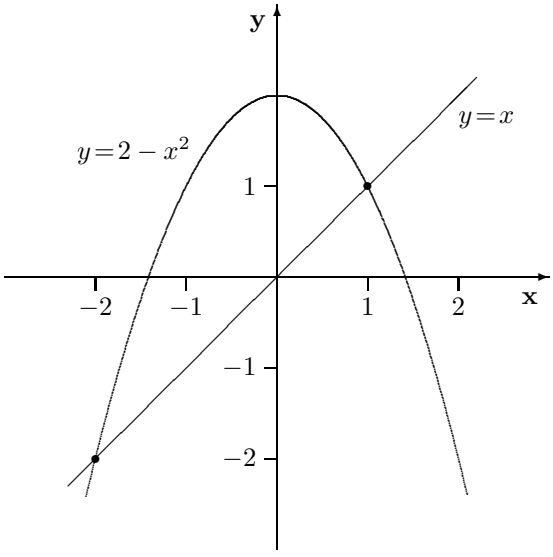


Abb. 2.8. (zu Beispiel 2.13)

□

2.3.3.2 Analytische Betrachtung

Satz 2.14. (*Fixpunktsatz*)
Die Funktion $\varphi : I \to \mathbb{R}$ erfülle die Voraussetzungen (i) und (ii) des Satzes 2.11. Dann konvergiert die mittels der Iterationsvorschrift

$$x^{(\nu+1)} = \varphi(x^{(\nu)}), \quad \nu = 0, 1, 2, \ldots$$

erzeugte Folge $\{x^{(\nu)}\}$ mit einem beliebigen Startwert $x^{(0)} \in I$ gegen den Fixpunkt ξ der Abbildung φ, d. h. es gilt

$$\lim_{\nu\to\infty} x^{(\nu+1)} = \xi\,.$$

Beweis. Nach Satz 2.11 hat die Gleichung $x = \varphi(x)$ in I genau eine Lösung ξ, also gilt $\xi = \varphi(\xi)$. Mit $x^{(\nu+1)} = \varphi(x^{(\nu)})$ und der Lipschitzbedingung (2.13) erhält man

$$|x^{(\nu+1)} - \xi| = |\varphi(x^{(\nu)}) - \varphi(\xi)| \overset{(LBed)}{\leq} L|x^{(\nu)} - \xi|$$

und in analoger Weise fortfahrend

$$|x^{(\nu+1)} - \xi| \leq L|x^{(\nu)} - \xi| \leq L^2|x^{(\nu-1)} - \xi| \leq \ldots \leq L^{\nu+1}|x^{(0)} - \xi|\,. \tag{2.16}$$

Wegen $0 \leq L < 1$ ist $\lim\limits_{\nu\to\infty} L^{\nu+1} = 0$, und aus (2.16) folgt

$$\lim_{\nu\to\infty} |x^{(\nu+1)} - \xi| \leq |x^{(0)} - \xi| \lim_{\nu\to\infty} L^{\nu+1} = 0, \quad \text{d. h. } \lim_{\nu\to\infty} x^{(\nu+1)} = \xi.$$

Damit folgt wegen der Stetigkeit von φ

$$\xi = \lim_{\nu\to\infty} x^{(\nu+1)} = \lim_{\nu\to\infty} \varphi(x^{(\nu)}) = \varphi(\lim_{\nu\to\infty} x^{(\nu)}) = \varphi(\xi)\,.$$

□

Beispiel 2.15. (Fortsetzung von Beispiel 2.6 und 2.7)
Zur Auflösung (III): $x = \frac{2}{x} - 1$ der Gleichung $x^2 + x - 2 = 0$ gehören die Schrittfunktion

$$\varphi(x) = \frac{2}{x} - 1 \qquad \text{für } x \neq 0$$

und mit der Fixpunktgleichung $x = \varphi(x)$ die Iterationsvorschrift

$$x^{(\nu+1)} = \varphi(x^{(\nu)}) = \frac{2}{x^{(\nu)}} - 1\,, \quad \nu = 0, 1, 2, \ldots$$

a) Wegen $\varphi'(x) = -\frac{2}{x^2} < 0$, $x \neq 0$, ist φ streng monoton fallend (Abb. 2.9). Als Intervall wird $I = [-3, -1]$ gewählt. Wegen $\varphi(-3) = -\frac{5}{3} \in I$ und $\varphi(-1) = -3 \in I$ gilt: φ erfüllt in I die Bedingung (i) des Satzes 2.11.

b) Lipschitzbeschränkung: φ' wird in $I = [-3, -1]$ abgeschätzt:

$$|\varphi'(x)| = \frac{2}{x^2} \quad \leq \quad \frac{2}{\min\limits_{x \in I}(x^2)} = \frac{2}{1} = 2 \quad > \quad 1\,.$$

Also ist φ in I nicht lipschitzbeschränkt.
Damit $|\varphi'(x)| = 2/x^2 < 1$ ist, muss $x^2 > 2$, also $|x| > \sqrt{2}$ sein. Der rechte Intervallrand muss modifiziert werden, um diese Bedingung zu erfüllen; das neue Intervall sei $I = [-3, -1.5]$.
$|\varphi'(x)|$ nimmt das Maximum am rechten Rand von I an, also gilt

$$|\varphi'(x)| \leq |\varphi'(-1.5)| = 0.\overline{8} < 0.89 = L < 1\,;$$

in I ist die Voraussetzung (ii) des Satzes 2.11 erfüllt. Damit kann die Lösung $\xi_2 = -2$ iterativ mit der Schrittfunktion aus der Umformung (III) berechnet werden.

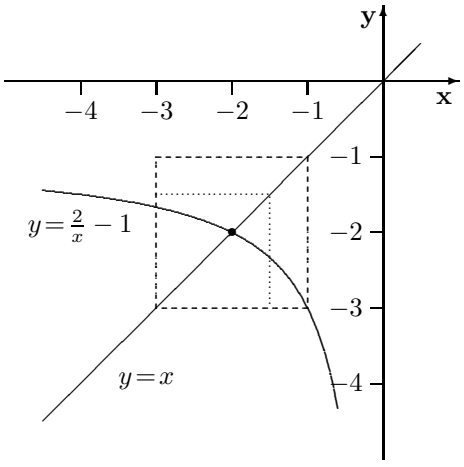


Abb. 2.9. Intervall $[-3, -1]$ wird verkleinert auf $[-3, -1.5]$

Zur Bestimmung der Lösung $\xi_1 = 1$ ist die Umformung (III) ungeeignet. Wählt man nämlich z. B. $I = [0.5, 1.2]$, so gilt dort $|\varphi'(x)| > 1$, so dass die Bedingung (ii) des Satzes 2.11 verletzt ist. □

2.3.4 Fehlerabschätzungen und Rechnungsfehler

Die nach ν Iterationsschritten erzeugte Näherungslösung $x^{(\nu)}$ unterscheidet sich von der exakten Lösung ξ um den Fehler $\Delta^{(\nu)} := x^{(\nu)} - \xi$ unter der Annahme, dass keine Rechnungsfehler gemacht wurden. Es wird nun für ein festes ν eine Schranke ε für den absoluten Fehler $|\Delta^{(\nu)}|$ gesucht. Ferner interessiert bei vorgegebener Schranke ε die Anzahl ν der Iterationsschritte, die erforderlich ist, damit $|\Delta^{(\nu)}| \leq \varepsilon$ gilt.

(i) Fehlerabschätzungen mit Verwendung der Lipschitzkonstante L

Mit (2.11) und (2.13) folgt

$$\left\{\begin{array}{lcl} |x^{(\nu+1)}-x^{(\nu)}| & \leq & L|x^{(\nu)}-x^{(\nu-1)}|, \\ |x^{(\nu+2)}-x^{(\nu+1)}| & \leq & L|x^{(\nu+1)}-x^{(\nu)}| \leq L^2|x^{(\nu)}-x^{(\nu-1)}|, \\ \quad\vdots & & \quad\vdots \\ |x^{(\nu+m)}-x^{(\nu+m-1)}| & \leq & L^m|x^{(\nu)}-x^{(\nu-1)}|. \end{array}\right. \tag{2.17}$$

Aus (2.17) erhält man weiter mit der Dreiecksungleichung

$$\begin{aligned} |x^{(\nu+m)}-x^{(\nu)}| &= |x^{(\nu+m)}-x^{(\nu+m-1)}+x^{(\nu+m-1)}-\ldots+x^{(\nu+1)}-x^{(\nu)}| \\ &\leq |x^{(\nu+m)}-x^{(\nu+m-1)}|+|x^{(\nu+m-1)}-x^{(\nu+m-2)}|+\ldots+|x^{(\nu+1)}-x^{(\nu)}| \\ &\leq (L^m+L^{m-1}+\ldots+L)\,|x^{(\nu)}-x^{(\nu-1)}| \\ &= L\sum_{j=0}^{m-1} L^j|x^{(\nu)}-x^{(\nu-1)}| \\ &= L\,\tfrac{1-L^m}{1-L}\,|x^{(\nu)}-x^{(\nu-1)}|\,. \end{aligned}$$

Wegen $0<1-L^m\leq 1$ folgt daraus

$$|x^{(\nu+m)}-x^{(\nu)}| \leq \frac{L}{1-L}|x^{(\nu)}-x^{(\nu-1)}|. \tag{2.18}$$

Setzt man in der letzten Zeile von (2.17) $\nu=1$, $m=\overline{\nu}-1$ und schreibt anschließend wieder ν statt $\overline{\nu}$, so erhält man $|x^{(\nu)}-x^{(\nu-1)}|\leq L^{\nu-1}|x^{(1)}-x^{(0)}|$. Geht man damit in (2.18) ein, so ergibt sich

$$|x^{(\nu+m)}-x^{(\nu)}| \leq \frac{L}{1-L}|x^{(\nu)}-x^{(\nu-1)}| \leq \frac{L^\nu}{1-L}|x^{(1)}-x^{(0)}|\,. \tag{2.19}$$

Für festes ν und $m\to\infty$ folgen aus (2.19) mit $\lim\limits_{m\to\infty} x^{(\nu+m)}=\xi$

1. die *a posteriori-Fehlerabschätzung*

$$|\Delta^{(\nu)}| = |x^{(\nu)}-\xi| \leq \frac{L}{1-L}|x^{(\nu)}-x^{(\nu-1)}| = \varepsilon_1 \tag{2.20}$$

und

2. die *a priori-Fehlerabschätzung*

$$|\Delta^{(\nu)}| = |x^{(\nu)}-\xi| \leq \frac{L^\nu}{1-L}|x^{(1)}-x^{(0)}| = \varepsilon_2 \tag{2.21}$$

mit $\varepsilon_1\leq\varepsilon_2$.

Die a priori-Fehlerabschätzung (2.21) kann bereits nach dem ersten Iterationsschritt vorgenommen werden. Sie dient vor allem dazu, bei vorgegebener Fehlerschranke ε die Anzahl ν der höchstens erforderlichen Iterationsschritte abzuschätzen, denn aus der Forderung

$$|x^{(\nu)} - \xi| \le \frac{L^\nu}{1-L}|x^{(1)} - x^{(0)}| \overset{!}{\le} \varepsilon$$

ergibt sich mit $L^\nu \le \frac{\varepsilon(1-L)}{|x^{(1)}-x^{(0)}|} =: K$ und mit $\log L < 0$ für $0 < L < 1$ die Ungleichung

$$\nu \ge \frac{\log K}{\log L}\,. \tag{2.22}$$

Die a posteriori-Fehlerabschätzung (2.20) kann erst im Verlauf oder nach Abschluss der Rechnung genutzt werden, da sie $x^{(\nu)}$ als bekannt voraussetzt; sie liefert eine bessere Schranke als die a priori-Fehlerabschätzung und wird deshalb vorzugsweise zur Abschätzung des Fehlers verwendet. Um rasche Konvergenz zu erreichen, sollten die Schrittfunktion φ und das zugehörige Intervall I so gewählt werden, dass $L < 0.2$ gilt.

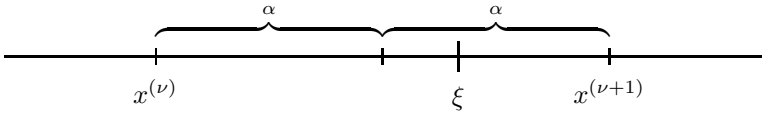
Wenn $\frac{L}{1-L} \le 1$ ist, also $L \le 1/2$, folgt aus (2.20) $|x^{(\nu)} - \xi| \le |x^{(\nu)} - x^{(\nu-1)}|$, d. h. der absolute Fehler von $x^{(\nu)}$ ist kleiner (für $L < 1/2$) oder höchstens gleich (für $L = 1/2$) der absoluten Differenz der letzten beiden Näherungen. Jeder Iterationsschritt bringt demnach eine bessere Annäherung der gesuchten Lösung. Für $1/2 < L < 1$ kann jedoch der absolute Fehler von $x^{(\nu)}$ größer sein als $|x^{(\nu)} - x^{(\nu-1)}|$, so dass hier die Iterationsfolge noch nichts über den Fehler aussagen kann.

Im Falle *monotoner Konvergenz* ($0 \le \varphi'(x) < 1$) folgt aus der Abschätzung $|x^{(\nu)} - \xi| \le \varepsilon$ (bzw. $x^{(\nu)} - \varepsilon \le \xi \le x^{(\nu)} + \varepsilon$) eine schärfere Eingrenzung der Lösung: $x^{(\nu)} \le \xi \le x^{(\nu)} + \varepsilon$, falls die $x^{(\nu)}$ von links gegen ξ streben, $x^{(\nu)} - \varepsilon \le \xi \le x^{(\nu)}$, falls die $x^{(\nu)}$ von rechts gegen ξ streben.

(ii) Praktikable Fehlerabschätzungen ohne Verwendung der Lipschitzkonstante L

(1) Bei alternierender Konvergenz

Im Falle alternierender Konvergenz ($-1 < \varphi'(x) \le 0$) schachteln die iterierten Werte die Lösung von links und rechts ein. Aus der anschließenden Skizze lässt sich deshalb leicht der folgende Zusammenhang erklären:



$$\left|x^{(\nu+1)} - \xi\right| \;\le\; \frac{1}{2}\,\left|x^{(\nu+1)} - x^{(\nu)}\right| = \alpha; \tag{2.23}$$

damit hat man eine Möglichkeit, den Fehler ohne Kenntnis der Lipschitzkonstante abzuschätzen.

(2) Fehlerabschätzung mit dem Satz von Bolzano (Zwischenwertsatz)

Für die Rechenpraxis empfiehlt sich eine in [KIOU1979] angegebene Methode, durch die man im Allgemeinen genauere Schranken erzielen kann als mit der a posteriori-Fehlerabschätzung. Zudem hat sie den großen Vorteil, ohne Lipschitzkonstante auszukommen und sowohl für monotone als auch für alternierende Konvergenz anwendbar zu sein; sie bezieht sich auf die gegebene Funktion f und nicht auf die Schrittfunktion φ.

Sei $x^{(\nu)}$ eine iterativ bestimmte Näherung für die Nullstelle ξ ungerader Ordnung von f (Abb. 2.10) und gelte für ein vorgegebenes $\varepsilon > 0$

$$f(x^{(\nu)} - \varepsilon) \cdot f(x^{(\nu)} + \varepsilon) < 0, \tag{2.24}$$

so folgt daraus nach dem Zwischenwertsatz von Bolzano (Satz 2.4), dass im Intervall $(x^{(\nu)} - \varepsilon,\ x^{(\nu)} + \varepsilon)$ eine Nullstelle ξ liegen muss. Damit gilt die Fehlerabschätzung $|x^{(\nu)} - \xi| < \varepsilon$.

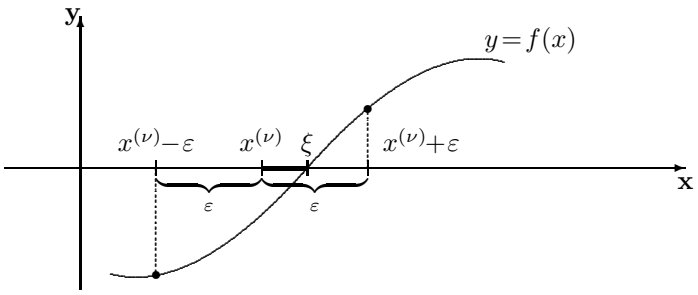


Abb. 2.10. $f(x^{(\nu)}-\varepsilon) \cdot f(x^{(\nu)}+\varepsilon) < 0 \ \Rightarrow\ |x^{(\nu)}-\xi| < \varepsilon$

Praktisch geht man bei der Fehlerabschätzung mit Bolzano wie folgt vor: Man setzt zunächst ein ε fest, welches sich über das Abbruchkriterium für die Iteration sinnvoll festlegen lässt (z. B. $\varepsilon = 10^{-k}, k \in \mathbb{N}$). Für dieses ε prüft man die Bedingung (2.24), wobei eine Rechnung mit doppelter Genauigkeit zu empfehlen ist. Ist (2.24) erfüllt, so ist ε eine obere Schranke für den absoluten Fehler. Um eine möglichst kleine obere Schranke zu erhalten, führt man die Rechnung noch einmal mit einem kleineren ε durch (z. B. mit $\varepsilon_1 = 10^{-k-1}$). Ist (2.24) auch für ε_1 erfüllt, so ist ε_1 für $|x^{(\nu)} - \xi|$ eine kleinere obere Schranke als ε. Analog fährt man so lange fort, bis sich (2.24) für ein ε_j nicht mehr erfüllen lässt (z. B. $\varepsilon_j = 10^{-k-j}$). Dann ist ε_{j-1} (z. B. $\varepsilon_{j-1} = 10^{-k-j+1}$) die genaueste Fehlerschranke, die man auf diese Weise erhalten hat.

Es wurde bisher vorausgesetzt, dass für diese Art der Fehlerabschätzung eine Nullstelle ξ ungerader Ordnung vorliegen muss. Unter Verwendung des Satzes 2.3 kann man sie aber auch für Nullstellen ξ gerader Ordnung einsetzen, indem man anstelle der Funktion f die Funktion $g = f/f'$ für die Fehlerabschätzung verwendet, da ξ dann eine einfache Nullstelle von g ist. Statt (2.24) ergibt sich hier die analoge Bedingung

$$g(x^{(\nu)} - \varepsilon) \cdot g(x^{(\nu)} + \varepsilon) < 0.$$

Rechnungsfehler

Es sei $\varepsilon^{(\nu)}$ der lokale Rechnungsfehler des ν-ten Iterationsschrittes, der bei der Berechnung von $x^{(\nu)} = \varphi(x^{(\nu-1)})$ entsteht. Gilt $|\varepsilon^{(\nu)}| \leq \varepsilon$ für $\nu = 0, 1, 2, \ldots$, so ergibt sich für den akkumulierten Rechnungsfehler des ν-ten Iterationsschrittes

$$|r^{(\nu)}| \leq \frac{\varepsilon}{1-L}, \qquad 0 \leq L < 1.$$

Die Fehlerschranke $\varepsilon/(1-L)$ ist also unabhängig von der Anzahl ν der Iterationsschritte; der Algorithmus (2.11) ist somit stabil (vgl. Anmerkung nach Definition 36).
Da sich der Gesamtfehler aus dem Verfahrensfehler und dem Rechnungsfehler zusammensetzt, sollten Rechnungsfehler und Verfahrensfehler von etwa gleicher Größenordnung sein. Dann ergibt sich aus (2.21) bei bekanntem L die Beziehung

$$\frac{L^\nu}{1-L}|x^{(1)} - x^{(0)}| \approx \frac{\varepsilon}{1-L}\,;$$

mit $L^\nu \approx \frac{\varepsilon}{|x^{(1)}-x^{(0)}|}$ und $0 < L < 1$ folgt für die Anzahl der höchstens auszuführenden Iterationsschritte

$$\nu \geq \left(\log \frac{\varepsilon}{|x^{(1)} - x^{(0)}|}\right) / \log L\,.$$

Beispiel 2.16. (Fortsetzung von Beispiel 2.15)

Mit der Iterationsvorschrift $x^{(\nu+1)} = -1 + 2/x^{(\nu)}$ und $I = [-3, -1.5]$ erhält man ausgehend vom Startwert $x^{(0)} = -3$ bei Rechnung mit 14 Stellen die folgenden Werte $x^{(\nu)}$, die mit 8-stelliger Mantisse angegeben sind. Es wird so lange iteriert, bis die Forderung $|x^{(\nu)} - x^{(\nu-1)}| \leq 0.5{\cdot}10^{-4}$ erfüllt ist.

ν	$x^{(\nu)}$	$\lvert x^{(\nu)} - x^{(\nu-1)}\rvert$
0	−3.0000000	
1	−1.6666666	1.3333333
2	−2.2000000	0.5333333
3	−1.9090909	0.2909091
4	−2.0476190	0.1385281
5	−1.9767442	0.0708749
6	−2.0117647	0.0350205
7	−1.9941520	0.0176127
8	−2.0029326	0.0087805
9	−1.9985359	0.0043967
10	−2.0007326	0.0021967
11	−1.9996338	0.0010988
12	−2.0001831	0.0005493
13	−1.9999085	0.0002747
14	−2.0000458	0.0001373
15	−1.9999771	0.0000687
16	−2.0000114	0.0000343

1. a posteriori-Fehlerabschätzung: Mit $L = 0.89$ (vgl. Beispiel 2.15) folgt

$$|x^{(16)} - \xi| \leq \frac{L}{1-L}|x^{(16)} - x^{(15)}| \leq \frac{0.89}{1-0.89} \cdot 0.35{\cdot}10^{-4} \leq 0.29{\cdot}10^{-3}$$

2. Die a priori-Fehlerabschätzung hätte man benutzen können, um im Voraus eine Aussage über die maximal notwendige Anzahl ν von Iterationsschritten zu erhalten. Mit $|x^{(\nu)} - \xi| \leq \varepsilon = 0.5{\cdot}10^{-3}$ und $|x^{(1)} - x^{(0)}| = 1.33$ ergeben sich $K = \big(0.5{\cdot}10^{-3}(1-0.89)\big)/1.33 = 4.135{\cdot}10^{-5}$ und mit (2.22)

$$\nu \geq \frac{\lg K}{\lg L} = \frac{-4.384}{-0.051} = 86.6\,.$$

   Es sind also höchstens $\nu = 87$ Iterationsschritte erforderlich, um die geforderte Genauigkeit zu erreichen. In vielen Fällen wird man aber mit weit weniger Schritten auskommen, denn die a priori-Fehlerabschätzung ist sehr grob. Die a posteriori-Fehlerabschätzung zeigt, dass diese Genauigkeitsforderung bereits nach 16 Iterationsschritten erfüllt ist. Um rasche Konvergenz zu erreichen, sollten die Schrittfunktion φ und das zugehörige Intervall I so gewählt werden, dass $L < 0.2$ gilt. Dann sind auch die Fehlerabschätzungen genauer.

3. Fehlerabschätzung bei alternierender Konvergenz:

   Wegen $\varphi'(x) = -2/x^2 < 0$ für alle $x \in I$ liegt der Fall der alternierenden Konvergenz vor. Mit $|x^{(16)} - x^{(15)}| = 0.35{\cdot}10^{-4}$ erhält man somit gemäß (2.23)

$$|x^{(16)} - \xi| \;\leq\; \frac{1}{2}|x^{(16)} - x^{(15)}| = \frac{0.35}{2}{\cdot}10^{-4} = 0.175{\cdot}10^{-4} \;<\; 0.18{\cdot}10^{-4},$$

   also eine sogar präzisere Abschätzung als die a posteriori-Fehlerabschätzung, weil $L > \frac{1}{2}$ ist. Für die Praxis ist wichtig, dass diese Fehlerabschätzung ohne Lipschitzkonstante einsetzbar ist.

4. Fehlerabschätzung unter Verwendung des Satzes von Bolzano, also ohne Verwendung einer Lipschitzkonstante. Gegeben ist die Gleichung $f(x) = x^2 + x - 2 = 0$. Untersucht wird der absolute Fehler von $x^{(16)} = -2.0000114$.

   1. Wahl: $\varepsilon_1 = 10^{-3}$. Dann folgt aus $f(x^{(16)} - \varepsilon_1) > 0$ und $f(x^{(16)} + \varepsilon_1) < 0$, dass für den absoluten Fehler gilt: $|x^{(16)} - \xi| < 10^{-3}$.

   2. Wahl: $\varepsilon_2 = 10^{-4}$. Hier ergibt sich ebenfalls $f(x^{(16)} - \varepsilon_2) > 0$, $f(x^{(16)} + \varepsilon_2) < 0$, also $|x^{(16)} - \xi| < 10^{-4}$.

   3. Wahl: $\varepsilon_3 = 10^{-5}$. Man erhält $f(x^{(16)} - \varepsilon_3) > 0$ und $f(x^{(16)} + \varepsilon_3) > 0$, also ist ε_3 zu klein gewählt. Ein weiterer Versuch wird unternommen mit der

   4. Wahl: $\varepsilon_4 = 0.5{\cdot}10^{-4}$. Man erhält $f(x^{(16)} - \varepsilon_4) > 0$ und $f(x^{(16)} + \varepsilon_4) < 0$; also gilt für den absoluten Fehler $|x^{(16)} - \xi| < 0.5{\cdot}10^{-4}$.

Tatsächlich ist $|x^{(16)} - \xi| = 0.0000114$. Man sieht, dass diese Fehlerabschätzung nach Bolzano am besten geeignet ist, um den absoluten Fehler möglichst genau anzugeben. Zudem ist nur sie praktikabel, denn meist ist es bedeutend schwieriger als beim vorliegenden Beispiel, eine möglichst kleine Lipschitzkonstante anzugeben. □

2.3.5 Praktische Durchführung

Bei der Ermittlung der Lösung einer Gleichung $f(x) = 0$ mit Hilfe des allgemeinen Iterationsverfahrens geht man wie folgt vor:

Algorithmus 2.17.
Gesucht ist eine Lösung ξ der Gleichung $f(x) = 0$.

1. Schritt: Festlegung eines Intervalls I, in welchem mindestens eine Nullstelle von f liegt.
2. Schritt: Äquivalente Umformung von $f(x) = 0$ in eine Gleichung der Gestalt $x = \varphi(x)$.
3. Schritt: Prüfung, ob die Funktion φ für alle $x \in I$ die Voraussetzungen des Satzes 2.11 erfüllt.
4. Schritt: Aufstellung der Iterationsvorschrift $x^{(\nu+1)} := \varphi(x^{(\nu)})$, $\nu = 0, 1, 2, \ldots$ und Wahl eines beliebigen Startwertes $x^{(0)} \in I$.
5. Schritt: Berechnung der Iterationsfolge $\{x^{(\nu)}\}$, $\nu = 1, 2, \ldots$ Die Iteration ist so lange fortzusetzen, bis mit einer Schranke $\delta_1 > 0$ für den relativen Fehler

$$|x^{(\nu+1)} - x^{(\nu)}| \leq \delta_1 |x^{(\nu+1)}| \tag{2.25}$$

oder mit einer Schranke $\delta_2 > 0$ für den absoluten Fehler

$$|x^{(\nu+1)} - x^{(\nu)}| \leq \delta_2 \tag{2.26}$$

und zusätzlich mit $\delta_3 > 0$

$$|f(x^{(\nu+1)})| \leq \delta_3$$

gilt. Dabei ist (2.25) im Allgemeinen der Abbruchbedingung (2.26) vorzuziehen. Beide Bedingungen sind im kombinierten Fehlertest (1.6) enthalten.

6. Schritt: Fehlerabschätzung (s. Abschnitt 2.3.4).

Beispiel 2.18.

Gegeben: $f(x) = \cos x + 1 - \sqrt{x}, \quad x \geq 0.$

Gesucht: Eine Lösung ξ der Gleichung $f(x) = 0$ mit dem allgemeinen Iterationsverfahren.

Lösung: (vgl. Algorithmus 2.17)

1. Schritt: Überblick über Lage und Anzahl der Lösungen von $f(x) = 0$. Mit

$$f_1(x) := \cos x + 1\,, \qquad f_2(x) := \sqrt{x}$$

gilt

$$f(x) = f_1(x) - f_2(x) = 0 \quad \Longleftrightarrow \quad f_1(x) = f_2(x)\,.$$

Daher liefert die Abszisse eines Schnittpunktes der Graphen von f_1 und f_2 eine Näherung für eine Lösung ξ der Gleichung $f(x) = 0$ (Abbildung 2.11).

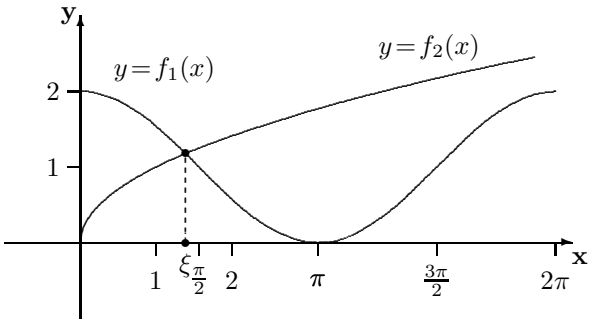


Abb. 2.11.

Im Intervall $[0, \pi]$ sind f_1 monoton fallend und f_2 monoton wachsend. Wegen $f_1(0) = 2 > 0 = f_2(0)$ und $f_1(\pi) = 0 < \sqrt{\pi} = f_2(\pi)$ haben die Graphen von f_1 und f_2 in $[0, \pi]$ genau einen Schnittpunkt.
Wegen $f_1(x) \leq 2$ und $f_2(x) = \sqrt{x} \geq 2$ für $x \geq 4$ gibt es keine weiteren Schnittpunkte.

Aus Abbildung 2.11 liest man ab $\xi \in [1, \frac{\pi}{2}]$. Mit $f(1.3) = 0.127$ und $f(1.5) = -0.154$ ergibt sich $I = [1.3, 1.5]$ als ein engeres Einschlussintervall für die Lösung ξ.

2. Schritt: Äquivalente Auflösung von $f(x) = 0$. Die naheliegende Auflösung

$$x = (\cos x + 1)^2 =: \varphi(x)$$

mit

$$\varphi'(x) = -2\,(\cos x + 1)\,\sin x$$

erweist sich wegen $|\varphi'(x)| > 2$ für $x \in I$ als unbrauchbar. Daher wird die Auflösung

$$x = \arccos(\sqrt{x} - 1) =: \varphi(x)$$

gewählt mit

$$\varphi'(x) = \frac{-1}{2\,\sqrt{x}\,\sqrt{1 - (\sqrt{x} - 1)^2}}\,, \quad x > 0\,.$$

3. Schritt: Genügt φ für $x \in I$ den Voraussetzungen (i) und (ii) des Satzes 2.11?

zu (i):
Wegen $\varphi'(x) < 0$ für $x > 0$ ist φ streng monoton fallend. Mit $\varphi(1.3) = 1.43 \in I$ und $\varphi(1.5) = 1.34 \in I$ gilt wegen der Monotonie auch $\varphi(x) \in I$ für alle $x \in I = [1.3, 1.5]$. Also ist die Bedingung (i) erfüllt.

zu (ii):
Um eine Lipschitzkonstante L zu ermitteln, wird $\varphi'(x)$ in I grob abgeschätzt:

$$\begin{aligned} |\varphi'(x)| &= \frac{1}{2\sqrt{x}\sqrt{1-(\sqrt{x}-1)^2}} \\ &\leq \frac{1}{2\min\limits_{x\in I}\sqrt{x}\cdot\min\limits_{x\in I}\sqrt{1-(\sqrt{x}-1)^2}} \\ &= \frac{1}{2\sqrt{1.3}\sqrt{1-(\sqrt{1.5}-1)^2}} = 0.450 = L < \tfrac{1}{2} < 1\,. \end{aligned}$$

φ erfüllt also die Voraussetzungen der Sätze 2.11 und 2.14.

4. Schritt: Iterationsvorschrift

$$x^{(\nu+1)} = \varphi(x^{(\nu)}) = \arccos(\sqrt{x^{(\nu)}} - 1)\,, \quad \nu = 0, 1, 2, \ldots$$

mit dem Startwert $x^{(0)} = 1.3 \in I$.

5. Schritt: Wegen $\varphi'(x) < 0$ in I liegt alternierende Konvergenz vor, so dass mit der zugehörigen Fehlerabschätzung eine Abbruchbedingung konstruiert werden kann, die eine bestimmte Anzahl genauer Dezimalen garantiert. Für drei genaue Dezimalen erhält man mit (2.23)

$$|x^{(\nu)} - \xi| \leq \frac{1}{2}\,|x^{(\nu)} - x^{(\nu-1)}| \overset{!}{\leq} 0.5{\cdot}10^{-3}$$

und damit die Abbruchbedingung

$$|x^{(\nu)} - x^{(\nu-1)}| \leq 1{\cdot}10^{-3}\,.$$

Iteration:

ν	$x^{(\nu)}$	$\lvert x^{(\nu)} - x^{(\nu-1)}\rvert$
0	1.3	.
1	1.430 157 740	.
2	1.373 629 308	.
3	1.397 917 137	.
4	1.387 435 119	
5	1.391 950 063	0.004 514 944
6	1.390 003 705	0.001 946 358
7	1.390 842 462	0.000 838 757

Bei $\nu = 7$ ist die Abbruchbedingung erfüllt, also ist

$$\xi = 1.391 \approx 1.390\,842\,462 = x^{(7)}\,.$$

6. Schritt: Fehlerabschätzungen

Die Abbruchbedingung war so gewählt, dass mit der Fehlerabschätzung für alternierende Konvergenz bereits 3 genaue Dezimalen erreicht wurden.

Fehlerabschätzung nach Bolzano: $|x^{(\nu)} - \xi| < \varepsilon$

Annahme: $\varepsilon_1 = 0.5{\cdot}10^{-3}$

$$\begin{aligned} f(x^{(7)} - \varepsilon_1) &= f(1.390\,342\,462) = 0.000\,348\ldots > 0 \\ f(x^{(7)} + \varepsilon_1) &= f(1.391\,342\,462) = -0.001\,059\ldots < 0 \\ \Longrightarrow \quad |x^{(7)} - \xi| &< 0.5{\cdot}10^{-3} \quad \text{d. h. 3 genaue Dezimalen} \end{aligned}$$

Annahme: $\varepsilon_2 = 0.5{\cdot}10^{-4}$

$$\begin{aligned} f(x^{(7)} - \varepsilon_2) &= f(1.390\,792\,462) = -0.000\,285\ldots < 0 \\ f(x^{(7)} + \varepsilon_2) &= f(1.390\,892\,462) = -0.000\,426\ldots < 0 \\ \Longrightarrow \quad & \varepsilon_2 \text{ zu klein!} \end{aligned}$$

□

2.4 Konvergenzordnung eines Iterationsverfahrens

Bei Iterationsverfahren können die Anzahl der erforderlichen Iterationsschritte und auch der Rechenaufwand im Allgemeinen nicht im Voraus ermittelt werden. Die Konvergenzordnung kann aber als Maßstab für den erforderlichen Rechenaufwand eines Verfahrens dienen.

Definition 2.19. (*Konvergenzordnung*)
Die Iterationsfolge $\{x^{(\nu)}\}$ konvergiert von mindestens p-ter Ordnung gegen ξ, wenn eine Konstante M, $0 \le M < \infty$ existiert, so dass für $p \in \mathbb{R}$, $p \ge 1$, gilt

$$\lim_{\nu\to\infty} \frac{|x^{(\nu+1)} - \xi|}{|x^{(\nu)} - \xi|^p} = M. \tag{2.27}$$

Das Iterationsverfahren $x^{(\nu+1)} = \varphi(x^{(\nu)})$ heißt dann ein Verfahren von mindestens p-ter Ordnung; es besitzt genau die Ordnung p, wenn $M \neq 0$ ist.

Durch (2.27) wird also ausgedrückt, dass der Fehler der $(\nu{+}1)$-ten Näherung ungefähr gleich M-mal der p-ten Potenz des Fehlers der ν-ten Näherung ist. Die Konvergenzgeschwindigkeit wächst mit der Konvergenzordnung. Bei $p = 1$ spricht man von *linearer Konvergenz*, bei $p = 2$ von *quadratischer Konvergenz* und allgemein bei $p > 1$ von *superlinearer Konvergenz*. Es gilt der

Satz 2.20.
Die Schrittfunktion φ sei für $x \in I$ p-mal stetig differenzierbar. Gilt dann mit $\lim\limits_{\nu\to\infty} x^{(\nu)} = \xi$

$$\varphi(\xi) = \xi, \varphi'(\xi) = \varphi''(\xi) = \ldots = \varphi^{(p-1)}(\xi) = 0, \varphi^{(p)}(\xi) \neq 0,$$

so ist $x^{(\nu+1)} = \varphi(x^{(\nu)})$ ein Iterationsverfahren der Ordnung p mit

$$M = \frac{1}{p!}|\varphi^{(p)}(\xi)| \leq \frac{1}{p!} \max_{x\in I} |\varphi^{(p)}(x)| \leq M_1.$$

Im Fall $p = 1$ gilt zusätzlich $M = |\varphi'(\xi)| < 1$.

Beweis. Die Taylorentwicklung der Schrittfunktion φ an der Stelle ξ lautet

$$\varphi(x^{(\nu)}) = \varphi(\xi) + (x^{(\nu)} - \xi)\varphi'(\xi) + \frac{1}{2}(x^{(\nu)} - \xi)^2\varphi''(\xi) + \ldots$$

und mit $x^{(\nu+1)} = \varphi(x^{(\nu)})$ und $\xi = \varphi(\xi)$ ergibt sich

$$x^{(\nu+1)} - \xi = \varphi(x^{(\nu)}) - \varphi(\xi) = (x^{(\nu)} - \xi)\varphi'(\xi) + \frac{1}{2}(x^{(\nu)} - \xi)^2\varphi''(\xi) + \ldots \tag{2.28}$$

Wegen $\varphi(\xi) = \xi, \varphi'(\xi) = \ldots = \varphi^{(p-1)}(\xi) = 0, \varphi^{(p)}(\xi) \neq 0$ erhält (2.28) die Form

$$x^{(\nu+1)} - \xi = \frac{(x^{(\nu)} - \xi)^p}{p!}\varphi^{(p)}(\xi) + \mathrm{O}\left((x^{(\nu)} - \xi)^{p+1}\right)$$

bzw.

$$\frac{x^{(\nu+1)} - \xi}{(x^{(\nu)} - \xi)^p} = \frac{1}{p!}\varphi^{(p)}(\xi) + \mathrm{O}\left((x^{(\nu)} - \xi)\right) .$$

Durch Grenzübergang folgt weiter

$$\lim_{\nu\to\infty} \left| \frac{x^{(\nu+1)} - \xi}{(x^{(\nu)} - \xi)^p} \right| = \frac{1}{p!}|\varphi^{(p)}(\xi)| = M$$

mit

$$M \leq \frac{1}{p!} \max_{x\in I} |\varphi^{(p)}(x)| \leq M_1 .$$

Im Fall $p = 1$ ist das Iterationsverfahren $x^{(\nu+1)} = \varphi(x^{(\nu)})$ mit $\varphi'(\xi) \neq 0$ ein Verfahren erster Ordnung, und wegen der Lipschitzbedingung (2.14) ist $M = |\varphi'(\xi)| \leq L < 1$. □

Es gilt außerdem der folgende in [COLL1968], S.231 bewiesene Satz, der zur Konstruktion von Iterationsverfahren beliebig hoher Konvergenzordnung eingesetzt werden kann.

Satz 2.21.
Sind $x^{(\nu+1)} = \varphi_1(x^{(\nu)})$ und $x^{(\nu+1)} = \varphi_2(x^{(\nu)})$ zwei Iterationsverfahren der Konvergenzordnungen p_1 bzw. p_2, so ist

$$x^{(\nu+1)} = \varphi_1\big(\varphi_2(x^{(\nu)})\big)$$

ein Iterationsverfahren, das mindestens die Konvergenzordnung $p_1 \cdot p_2$ besitzt.

Ist beispielsweise $x^{(\nu+1)} = \varphi(x^{(\nu)})$ ein Iterationsverfahren der Konvergenzordnung $p > 1$, so erhält man durch die Schrittfunktion $\varphi_s(x)$ mit

$$\begin{aligned} &\varphi_1(x) = \varphi(x), \\ &\varphi_s(x) = \varphi\big(\varphi_{s-1}(x)\big) \qquad \text{für} \qquad s = 2, 3, \ldots \end{aligned}$$

ein Iterationsverfahren der Konvergenzordnung p^s.

2.5 Newtonsche Verfahren

2.5.1 Das Newtonsche Verfahren für einfache Nullstellen

Die Funktion f sei im Intervall $[a, b]$ stetig differenzierbar und besitze in (a, b) eine *einfache Nullstelle* ξ; also sind $f(\xi) = 0$ und $f'(\xi) \neq 0$. Ferner sei $f'(x) \neq 0$ für alle $x \in [a, b]$.

Zum Newtonschen Iterationsverfahren führt eine einfache geometrische Überlegung.

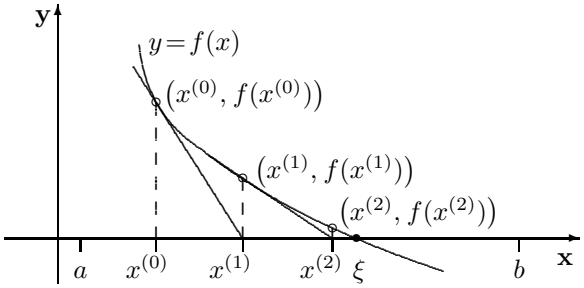


Abb. 2.12. Newtonsches Iterationsverfahren

Mit $x^{(0)} \in [a, b]$ wird im Punkt $\big(x^{(0)}, f(x^{(0)})\big)$ die Tangente des Graphen von f erzeugt und dann deren Schnittpunkt $(x^{(1)}, 0)$ mit der x-Achse bestimmt. Aus der Gleichung dieser Tangente

$$y = f'(x^{(0)})(x - x^{(0)}) + f(x^{(0)})$$

folgt mit $x = x^{(1)}$ und $y = 0$

$$x^{(1)} = x^{(0)} - \frac{f(x^{(0)})}{f'(x^{(0)})};$$

$x^{(1)}$ wird als eine gegenüber $x^{(0)}$ verbesserte Näherung für die Nullstelle ξ angesehen. Mit $x^{(1)}$ wird das Verfahren in derselben Weise fortgesetzt. Damit ergibt sich die nächste Näherung

$$x^{(2)} = x^{(1)} - \frac{f(x^{(1)})}{f'(x^{(1)})}\,.$$

Für dieses geometrisch plausible Verfahren werden im folgenden Satz hinreichende Bedingungen für die Konvergenz der Folge $x^{(0)}, x^{(1)}, x^{(2)}, \ldots$ gegen die Nullstelle ξ angegeben.

Satz 2.22.
Die Funktion f sei im Intervall $[a,b]$ zweimal stetig differenzierbar und besitze in (a,b) eine einfache Nullstelle ξ; also sind $f(\xi) = 0$ und $f'(\xi) \neq 0$.
Dann gibt es ein Intervall $I = (\xi - r, \xi + r) \subset [a,b]$, $r > 0$, so dass die Iterationsfolge

$$x^{(\nu+1)} = x^{(\nu)} - \frac{f(x^{(\nu)})}{f'(x^{(\nu)})}, \quad \nu = 0, 1, 2, \ldots, \tag{2.29}$$

für das Verfahren von Newton mit jedem Startwert $x^{(0)} \in I$ gegen die Nullstelle ξ konvergiert, und zwar von mindestens zweiter Ordnung.

Beweis. Vergleicht man (2.29) mit (2.4) und (2.2), so ergibt sich für die Schrittfunktion φ des Newton-Verfahrens

$$\varphi(x) = x - \frac{f(x)}{f'(x)}\,. \tag{2.30}$$

(2.3) zeigt, dass hier $g(x) = 1/f'(x)$ gesetzt wurde.

Um den Fixpunktsatz 2.14 auf die Schrittfunktion (2.30) anwenden zu können, muss ein Intervall I angegeben werden, so dass φ für alle $x \in I$ den Voraussetzungen (i) und (ii) des Satzes 2.11 genügt.

Da f' stetig ist, gibt es wegen $f'(\xi) \neq 0$ eine Umgebung der Nullstelle ξ, in der $f'(x) \neq 0$ ist. Diese Umgebung sei $I_s = (\xi - s, \xi + s) \subset [a,b]$, $s > 0$. Somit ist die Schrittfunktion φ $\big($vgl. (2.30)$\big)$ stetig im Intervall I_s.

Die Ableitung der Schrittfunktion (2.30) ist

$$\varphi'(x) = \frac{f(x)\,f''(x)}{{f'}^2(x)}\,. \tag{2.31}$$

Auch φ' ist stetig in I_s.

Mit $f(\xi) = 0$ und $f'(\xi) \neq 0$ ist

$$\varphi'(\xi) = 0\,. \tag{2.32}$$

Wegen der Stetigkeit von φ' in I_s gibt es eine Umgebung von ξ, in der $|\,\varphi'(x)\,| \leq L$ ist mit $0 < L < 1$. Diese Umgebung sei das Intervall

$$I = (\xi - r, \xi + r) \subseteq I_s \subset [a,b],\ 0 < r \leq s\,. \tag{2.33}$$

Aus

$$|\,\varphi'(x)\,| \le L \quad \text{für alle } x \in I \tag{2.34}$$

mit $0 < L < 1$ folgt nach (2.14), dass die Schrittfunktion φ lipschitzbeschränkt ist, und damit gilt (ii).

Um (i) nachzuweisen, muss gezeigt werden, dass mit $x \in I$, also $|\,x-\xi\,| < r$, auch $\varphi(x) \in I$ ist, d. h. $|\,\varphi(x) - \xi\,| < r$.
Wegen $\varphi(\xi) = \xi$ $\big($vgl. (2.30)$\big)$ ist mit (ii)

$$|\,\varphi(x) - \xi\,| = |\,\varphi(x) - \varphi(\xi)\,| \le L\,|\,x - \xi\,| < |\,x - \xi\,| < r\,.$$

Damit gilt auch (i).

Nach dem Fixpunktsatz 2.14 konvergiert also die Iterationsfolge (2.29) für jeden Startwert $x^{(0)} \in I$ gegen die Nullstelle ξ.

Wegen (2.32) konvergiert das Newtonsche Verfahren (2.29) nach Satz 2.20 von mindestens zweiter Ordnung, d. h. mindestens quadratisch. □

Aus dem Beweis des Satzes 2.22 geht hervor, dass das Newtonsche Verfahren nur für solche Startwerte $x^{(0)}$ konvergiert, die genügend nahe bei der Nullstelle ξ liegen.

Wegen der speziellen Schrittfunktion (2.30) kann eine gegenüber der a posteriori-Fehlerabschätzung (2.20) in Abschnitt 2.3.4 verbesserte Fehlerabschätzung für das Newtonsche Verfahren angegeben werden.

Satz 2.23.
Die Newtonsche Iterationsfolge (2.29) besitze gemäß Satz 2.22 das Konvergenzintervall I. Dann gilt unter Verwendung der a posteriori-Fehlerabschätzung (2.20) mit

$$\frac{1}{2}\,\frac{\max\limits_{x\in I}|f''(x)|}{\min\limits_{x\in I}|f'(x)|} \le M_1$$

die Fehlerabschätzung für $\nu = 1, 2, \ldots$

$$|\,x^{(\nu+1)} - \xi\,| \le M_1 \left(\frac{L}{1-L}\right)^2 \left|\,x^{(\nu)} - x^{(\nu-1)}\,\right|^2 \tag{2.35}$$

und allgemeiner mit $m = 1, 2, \ldots$

$$|\,x^{(\nu+m)} - \xi\,| \le \frac{1}{M_1}\left(M_1 \frac{L}{1-L}\left|\,x^{(\nu)} - x^{(\nu-1)}\,\right|\right)^{2^m}. \tag{2.36}$$

Beweis. Nach der Taylorschen Formel gilt

$$0 = f(\xi) = f(x^{(\nu)}) + f'(x^{(\nu)})(\xi - x^{(\nu)}) + \frac{1}{2}\, f''(\widetilde{x}^{(\nu)})(\xi - x^{(\nu)})^2, \tag{2.37}$$

wobei $\widetilde{x}^{(\nu)}$ zwischen $x^{(\nu)}$ und ξ liegt. Mit Division von (2.37) durch $f'(x^{(\nu)})$ $\big($für $x^{(\nu)} \in I$ ist $f'(x^{(\nu)}) \neq 0\big)$ erhält man

$$-\frac{f(x^{(\nu)})}{f'(x^{(\nu)})} - \xi + x^{(\nu)} = \frac{1}{2}\, \frac{f''(\widetilde{x}^{(\nu)})}{f'(x^{(\nu)})}(\xi - x^{(\nu)})^2$$

und mit (2.29)

$$x^{(\nu+1)} - \xi = \frac{1}{2}\, \frac{f''(\widetilde{x}^{(\nu)})}{f'(x^{(\nu)})}\, (x^{(\nu)} - \xi)^2. \tag{2.38}$$

Beim Übergang zu den Beträgen folgt aus (2.38)

$$\begin{aligned} |\, x^{(\nu+1)} - \xi\,| \;&\leq\; \tfrac{1}{2}\, \tfrac{\max\limits_{x\in I} |f''(x)|}{\min\limits_{x\in I} |f'(x)|}\, \big|\, x^{(\nu)} - \xi\,\big|^2 \\ &\leq\; M_1 |\, x^{(\nu)} - \xi\,|^2 \;=\; \tfrac{1}{M_1}\, \big(M_1 |\, x^{(\nu)} - \xi\,|\big)^2 . \end{aligned} \tag{2.39}$$

Wird (2.20)

$$\big|\, x^{(\nu)} - \xi\,\big| \;\leq\; \frac{L}{1-L}\, \big|\, x^{(\nu)} - x^{(\nu-1)}\,\big| \tag{2.40}$$

auf der rechten Seite der Ungleichung (2.39) eingesetzt, so ergibt sich die Fehlerabschätzung (2.35).

Die gegenüber (2.39) allgemeinere Ungleichung für $m = 1, 2, \ldots$

$$\big|\, x^{(\nu+m)} - \xi\,\big| \leq \frac{1}{M_1}\, \big(M_1 |\, x^{(\nu)} - \xi\,|\big)^{2^m} \tag{2.41}$$

gilt nach (2.39) für $m = 1$. Mit der Annahme, (2.41) gelte für m, wird (2.41) für $m+1$ bewiesen (vollständige Induktion). Aus (2.41) für $m+1$ folgt mit (2.39)

$$\begin{aligned} \big|\, x^{(\nu+1+m)} - \xi\,\big| \;&\leq\; \frac{1}{M_1}\, \big(M_1 |\, x^{(\nu+1)} - \xi\,|\big)^{2^m} \\ &\leq\; \frac{1}{M_1}\, \Big(M_1\, \frac{1}{M_1}\, \big(M_1 |\, x^{(\nu)} - \xi\,|\big)^2\Big)^{2^m} \\ &=\; \frac{1}{M_1}\, \big(M_1 |\, x^{(\nu)} - \xi\,|\big)^{2^{m+1}}. \end{aligned}$$

Wenn auf der rechten Seite der Ungleichung (2.41) $|\, x^{(\nu)} - \xi\,|$ mittels (2.40) ersetzt wird, entsteht die Fehlerabschätzung (2.36). □

Mit (2.35) kann mit zwei Näherungen $x^{(\nu-1)}$ und $x^{(\nu)}$ der absolute Fehler $|\,x^{(\nu+1)} - \xi\,|$ der nächsten (noch nicht berechneten) Näherung $x^{(\nu+1)}$ im Voraus abgeschätzt werden.

Eine Lipschitzkonstante L erhält man durch Abschätzung von (2.31) auf dem Intervall I:

$$|\varphi'(x)| = |\,f(x)\,f''(x)/{f'}^2(x)\,| \le L < 1\,.$$

Mit einer Lipschitzkonstante $L \le 0.5$ ist $L/(1-L) \le 1$ und (2.40) lautet

$$|\,x^{(\nu)} - \xi\,| \le |\,x^{(\nu)} - x^{(\nu-1)}\,|\,.$$

Gilt für die ν-te Näherung

$$|\,x^{(\nu)} - x^{(\nu-1)}\,| \le \varepsilon\,,$$

dann ist auch

$$|\,x^{(\nu)} - \xi\,| \le \varepsilon\,.$$

Der absolute Fehler von $x^{(\nu)}$ ist dann höchstens gleich ε. So erhält man nach Vorgabe von $\varepsilon > 0$ ein geeignetes Abbruchkriterium.

Beispiel 2.24.

Gegeben: Die Gleichung $f(x) = x^2 - a = 0$, $a > 0$.

Gesucht: Die Lösung $x = \sqrt{a}$ mit dem Newton-Verfahren.

Lösung: Wegen $f'(x) = 2x$ ist die Schrittfunktion (2.30)

$$\varphi(x) = x - \frac{x^2 - a}{2x} = \frac{1}{2}\left(x + \frac{a}{x}\right)\,,$$

und die Iterationsvorschrift (2.29) lautet

$$x^{(\nu+1)} = \frac{1}{2}\left(x^{(\nu)} + \frac{a}{x^{(\nu)}}\right)\,, \quad \nu = 0, 1, 2, \ldots \tag{2.42}$$

Ferner ist

$$\varphi'(x) = \frac{1}{2}\left(1 - \frac{a}{x^2}\right)\,.$$

Als numerisches Beispiel wird $a = 5$ gewählt. Wegen $2 < \sqrt{5} < 3$ sei $I = [2, 3]$. Dann ist für alle $x \in I$

$$\left|\varphi'(x)\right| = \frac{1}{2}\left|\,1 - \frac{5}{x^2}\,\right| \le \frac{1}{2}\left|\,1 - \frac{5}{9}\,\right| \le 0.23 = L < \frac{1}{2}\,.$$

Die Folge (2.42) mit $a = 5$ konvergiert also mit jedem Startwert $x^{(0)} \in I = [2, 3]$.

Mit der Lipschitzkonstante $L = 0.23$ ist $L/(1-L) = 0.2987$, und die a posteriori-Fehlerabschätzung (2.40) lautet

$$|\,x^{(\nu)} - \xi\,| \le 0.2987\,|\,x^{(\nu)} - x^{(\nu-1)}\,|\,, \quad \nu = 0, 1, 2, \ldots \tag{2.43}$$

Mit dem Abbruchkriterium

$$|\, x^{(\nu)} - x^{(\nu-1)} \,| \le 0.5 \cdot 10^{-7}$$

erhält man $\xi = \sqrt{5}$ mit 7 Dezimalen.

Für die Fehlerabschätzung (2.35) werden

$$\min_{x \in I} |f'(x)| = \min_{x \in I} |2x| = 4 \quad \text{und} \quad \max_{x \in I} |f''(x)| = 2$$

benötigt. Damit ist $M_1 = \frac{2}{2 \cdot 4} = 0.25$. Die Fehlerabschätzung (2.35) lautet

$$|\, x^{(\nu+1)} - \xi \,| \le 0.0223 \,|\, x^{(\nu)} - x^{(\nu-1)} \,|^2, \quad \nu = 0, 1, 2, \ldots \tag{2.44}$$

Aus der nachfolgenden Tabelle folgt wegen $|\, x^{(5)} - x^{(4)} \,| = 0$ für die Nullstelle $\xi = x^{(5)} = 2.236\,067\,98 \approx \sqrt{5}$.

ν	$x^{(\nu)}$	$\lvert\, x^{(\nu)} - x^{(\nu-1)} \,\rvert$	$\lvert\, x^{(\nu)} - \sqrt{5} \,\rvert$	(2.43)	(2.44)
0	3.00000000		0.76393202		
1	2.33333333	0.66666667	0.09726536	0.19913333	
2	2.23809524	0.09523810	0.00202726	0.02844762	0.00991352
3	2.23606890	0.00202634	0.00000092	0.00060527	0.00020232
4	2.23606798	0.00000092	0.00000000	0.00000027	0.00000009
5	2.23606798	0.00000000	0.00000000	0.00000000	$1.88 \cdot 10^{-14}$

Die mit den Fehlerabschätzungen (2.43) und (2.44) ermittelten Schranken für den absoluten Fehler zeigen im Vergleich mit dem exakten Fehler $|\, x^{(\nu)} - \sqrt{5} \,|$, dass die Abschätzung (2.44) kleinere Schranken liefert. □

Für die einfache Funktion f im vorangehenden Beispiel konnten L und M_1 unschwer bestimmt werden. Im Allgemeinen ist die Ermittlung von L und M_1 zu einem gewählten Intervall I mit einem erheblich größeren Aufwand verbunden.

Bemerkung. Da für jeden Iterationsschritt mit dem Newton-Verfahren ein Funktionswert und ein Ableitungswert zu berechnen sind, eignet sich dieses Verfahren in erster Linie für die Berechnung einzelner Nullstellen eines Polynoms, weil die benötigten Werte sich leicht mit dem Horner-Schema (siehe Kapitel 3) ermitteln lassen, weniger dagegen für die Lösung beliebiger transzendenter Gleichungen. In allen Fällen konvergiert das Newton-Verfahren nur für solche Startwerte, die genügend nahe bei der Nullstelle liegen.

Zur Berechnung von Nullstellen beliebiger transzendenter Funktionen eignet sich besser ein Einschlussverfahren. Siehe dazu Abschnitt 2.9 Entscheidungshilfen.

2.5.2 Gedämpftes Newton-Verfahren

Analog zum gedämpften Newton-Verfahren für nichtlineare Systeme (Abschnitt 6.3.1, Algorithmus 9) lässt sich das gedämpfte Newton-Verfahren für Einzelgleichungen angeben. Man führt für $\nu = 0, 1, 2, \ldots$ folgende Schritte durch:

(i) Berechne $\Delta x^{(\nu+1)} := x^{(\nu+1)} - x^{(\nu)} = -f(x^{(\nu)})/f'(x^{(\nu)})$ $\big($Newtonschritt (2.29)$\big)$.
(ii) Berechne für $i = 0, 1, \ldots$

$$x_i^{(\nu+1)} := x^{(\nu)} + \frac{1}{2^i}\Delta x^{(\nu+1)}\,.$$

Wenn $|f(x_i^{(\nu+1)})| < |f(x^{(\nu)})|$ gilt, wird $x^{(\nu+1)} := x_i^{(\nu+1)}$ gesetzt.
Andernfalls wird der Schritt mit dem nächsten i wiederholt.
Wenn mit einem vorgegebenen $i_{\max} \in \mathbb{N}$ für alle
$i \le i_{\max}$ die obige Bedingung nicht erfüllt ist, wird (mit $i = 0$)

$$x^{(\nu+1)} := x^{(\nu)} + \Delta x^{(\nu+1)}$$

gesetzt.

2.5.3 Das Newtonsche Verfahren für mehrfache Nullstellen – Das modifizierte Newtonsche Verfahren

Die Funktion f sei im Intervall $[a, b]$ genügend oft stetig differenzierbar und besitze in (a, b) eine Nullstelle ξ der Vielfachheit j, $j \ge 2$. Nach Satz 2.2 sind also

$$f(\xi) = f'(\xi) = \ldots = f^{(j-1)}(\xi) = 0,\ f^{(j)}(\xi) \neq 0\,.$$

Für die Ableitung (2.31) der Schrittfunktion (2.30) des Newtonschen Verfahrens für einfache Nullstellen ergibt sich mit (2.2), (2.4) und (2.6)

$$\varphi'(x) = \frac{f(x)\,f''(x)}{{f'}^2(x)} = \frac{h_0(x)\,h_2(x)}{h_1^2(x)}\,. \tag{2.45}$$

Mit (2.3), (2.5) und (2.7) ist für $j \ge 2$

$$\varphi'(\xi) = \frac{h_0(\xi)\,h_2(\xi)}{h_1^2(\xi)} = \frac{\big((j-1)!\,\big)^2}{j!\,(j-2)!} = \frac{j-1}{j} = 1 - \frac{1}{j} \ge \frac{1}{2}\,. \tag{2.46}$$

Das Newtonsche Verfahren, das für einfache Nullstellen mindestens quadratisch konvergiert (Satz 2.22), konvergiert nach Satz 2.20 in der Umgebung einer mehrfachen Nullstelle nur linear.

Mit der Schrittfunktion

$$\psi(x) = x - j\,\frac{f(x)}{f'(x)} \tag{2.47}$$

kann die quadratische Konvergenz auch in der Umgebung einer mehrfachen Nullstelle ξ ($j \geq 2$) beibehalten werden. Mit (2.2) und (2.4) lautet (2.47)

$$\psi(x) = x - j\,(x-\xi)\,\frac{h_0(x)}{h_1(x)}\,,$$

und es ist

$$\psi(\xi) = \xi\,.$$

Für die Ableitung der Schrittfunktion (2.47) ergibt sich mit (2.45)

$$\psi'(x) = 1 - j\,\left(1 - \frac{f(x)\,f''(x)}{{f'}^2(x)}\right) = 1 - j\,\left(1 - \frac{h_0(x)\,h_2(x)}{h_1^2(x)}\right)\,.$$

Mit (2.46) ist

$$\psi'(\xi) = 1 - j\left(1 - \left(1 - \frac{1}{j}\right)\right) = 1 - j\,\frac{1}{j} = 0\,,$$

und daraus folgt nach Satz 2.20 die mindestens quadratische Konvergenz mit der Schrittfunktion (2.47).

Wegen $\psi'(\xi) = 0$ und wegen der Stetigkeit von ψ' in einer Umgebung von ξ gilt: Es gibt ein Intervall $I = (\xi - r, \xi + r) \subset [a,b]$, $r > 0$, so dass für alle $x \in I$ $|\psi'(x)| \leq L < 1$ ist. Somit ist ψ in I lipschitzbeschränkt. Ferner ist wegen $\psi(\xi) = \xi$ für alle $x \in I$

$$|\,\psi(x) - \xi\,| = |\,\psi(x) - \psi(\xi)\,| \leq L\,|\,x - \xi\,| \leq |\,x - \xi\,| < r\,,$$

also $\psi(x) \in I$. Somit genügt die Schrittfunktion (2.47) den Voraussetzungen (i) und (ii) des Fixpunktsatzes 2.14. Damit folgt der

Satz 2.25. (*Newtonsches Verfahren für mehrfache Nullstellen*)
Die Funktion f sei im Intervall $[a,b]$ genügend oft stetig differenzierbar und besitze in (a,b) eine Nullstelle ξ der Vielfachheit $j \geq 2$. Dann gibt es ein Intervall $I = (\xi - r, \xi + r)$, $r > 0$, $I \subset [a,b]$, so dass die Iterationsfolge

$$x^{(\nu+1)} = x^{(\nu)} - j\frac{f(x^{(\nu)})}{f'(x^{(\nu)})}\,, \qquad \nu = 0, 1, 2, \ldots,$$

mit jedem Startwert $x^{(0)} \in I$ von mindestens zweiter Ordnung gegen die j-fache Nullstelle ξ konvergiert.

Die Anwendung des Newtonschen Verfahrens für mehrfache Nullstellen setzt die Kenntnis der Vielfachheit j der gesuchten Nullstelle ξ voraus; j wird allerdings nur in speziellen Fällen bekannt sein.

Nach Satz 2.3 ist eine j-fache Nullstelle ξ ($j \geq 2$) einer Funktion f eine einfache Nullstelle der Funktion g mit $g(x) = f(x)/f'(x)$. Insofern kann man die Nullstelle von g mit dem Newtonschen Verfahren für einfache Nullstellen ermitteln. Dieses Verfahren beschreibt der Satz

Satz 2.26. (*Modifiziertes Newtonsches Verfahren für mehrfache Nullstellen*)
Die Funktion f sei im Intervall $[a,b]$ genügend oft stetig differenzierbar und besitze in (a,b) eine Nullstelle ξ der Vielfachheit $j \geq 2$. Dann gibt es ein Intervall $I = (\xi - r, \xi + r)$, $r > 0$, $I \subset [a,b]$, so dass mit jedem Startwert $x^{(0)} \in I$ das Verfahren mit der Iterationsvorschrift

$$\begin{aligned} x^{(\nu+1)} &= x^{(\nu)} - J(x^{(\nu)}) \, \frac{f(x^{(\nu)})}{f'(x^{(\nu)})} \quad \text{mit} \\ J(x^{(\nu)}) &= \frac{1}{1 - \frac{f(x^{(\nu)}) f''(x^{(\nu)})}{f'^{\,2}(x^{(\nu)})}}, \quad \nu = 0, 1, 2, \ldots, \end{aligned}$$

quadratisch konvergiert. Es gelten zugleich

$$\lim_{\nu \to \infty} (x^{(\nu)}) = \xi \quad \text{und} \quad \lim_{\nu \to \infty} J(x^{(\nu)}) = j \, .$$

Beweis. Die Schrittfunktion des Newtonschen Verfahrens, angewendet auf die Funktion g mit

$$\begin{aligned} g(x) &= \frac{f(x)}{f'(x)}, \\ g'(x) &= 1 - \frac{f(x) f''(x)}{f'^{\,2}(x)} \quad \text{und} \\ J(x) &= \frac{1}{g'(x)} \end{aligned}$$

lautet

$$\varphi(x) = x - \frac{g(x)}{g'(x)} = x - J(x) \, \frac{f(x)}{f'(x)} \, .$$

Mit den Taylorentwicklungen (2.2), (2.4) und (2.6) an der Stelle ξ sind

$$\begin{aligned} \varphi(x) &= x - J(x) \, (x - \xi) \, \frac{h_0(x)}{h_1(x)} \quad \text{und} \\ J(x) &= \frac{1}{1 - \frac{h_0(x) \, h_2(x)}{h_1^2(x)}} \, . \end{aligned} \tag{2.48}$$

(1) Mit (2.46) erhält man

$$J(\xi) = \frac{1}{1 - (1 - \frac{1}{j})} = j \, . \tag{2.49}$$

(2) Zur Bestimmung der Konvergenzordnung wird $\varphi'(\xi)$ berechnet. Dabei werden (2.49) und (2.8) verwendet. Mit (2.48) und $\varphi(\xi) = \xi$ ist

$$\begin{aligned}\varphi'(\xi) &= \lim_{x\to\xi} \frac{\varphi(x)-\varphi(\xi)}{x-\xi} \\ &= \lim_{x\to\xi} \frac{1}{x-\xi}\left(x - J(x)\,(x-\xi)\,\frac{h_0(x)}{h_1(x)} - \xi\right) \\ &= \lim_{x\to\xi}\left(1 - J(x)\,\frac{h_0(x)}{h_1(x)}\right) = 1 - j\,\frac{1}{j} = 0\,.\end{aligned}$$

Nach Satz 2.20 hat die Schrittfunktion φ die Konvergenzordnung 2.

(3) Ähnlich wie beim Satz 2.25 für (2.47) kann gezeigt werden, dass die Schrittfunktion φ in einem Intervall $I = (\xi-r, \xi+r)$, $r > 0$, den Voraussetzungen des Fixpunktsatzes 2.14 genügt. □

Beispiel 2.27.

Gegeben: Die Funktion $f : f(x) = 1 - \sin x$. f besitzt bei $\xi = \frac{\pi}{2}$ eine doppelte Nullstelle.

Gesucht: Die Nullstelle ξ mit dem Newtonschen Verfahren für einfache Nullstellen, mit dem Newtonschen Verfahren für mehrfache Nullstellen und mit dem modifizierten Newtonschen Verfahren. Es wird mit 15-stelliger Mantisse gerechnet und die Rechnung abgebrochen, wenn die Bedingung $|x^{(\nu)} - x^{(\nu-1)}| \le 0.5{\cdot}10^{-14}$ erfüllt ist. Als Startwert wird $x^{(0)} = 2$ gewählt.

1. Newtonsches Verfahren für einfache Nullstellen mit der Iterationsvorschrift (2.29):

$$x^{(\nu+1)} = x^{(\nu)} - \frac{f(x^{(\nu)})}{f'(x^{(\nu)})} = x^{(\nu)} - \frac{1-\sin(x^{(\nu)})}{-\cos(x^{(\nu)})}$$

ν	$x^{(\nu)}$	$\lvert x^{(\nu)} - x^{(\nu-1)}\rvert$
0	2.00000000000000	
1	1.78204190153914	0.21795809846086
2	1.67602457140144	0.10601733013770
3	1.62336184567011	0.05266272573132
4	1.59707303266146	0.02628881300865
5	1.58393392371128	0.01313910895018
6	1.57736503077218	0.00656889293910
7	1.57408066697409	0.00328436379809
8	1.57243849540833	0.00164217156576
9	1.57161741091709	0.00082108449124
10	1.57120686883293	0.00041054208416
11	1.57100159781103	0.00020527102190
12	1.57089896230260	0.00010263550843

ν	$x^{(\nu)}$	$\lvert x^{(\nu)} - x^{(\nu-1)}\rvert$
13	1.57084764454871	0.00005131775390
14	1.57082198567180	0.00002565887691
15	1.57080915623335	0.00001282943845
16	1.57080274151412	0.00000641471923
17	1.57079953415451	0.00000320735961
18	1.57079793047470	0.00000160367980
19	1.57079712863480	0.00000080183990
20	1.57079672771482	0.00000040091998
21	1.57079652725488	0.00000020045994
22	1.57079642702485	0.00000010023003
23	1.57079637691009	0.00000005011476
24	1.57079635185228	0.00000002505781
25	1.57079633932381	0.00000001252847
26	1.57079633305860	0.00000000626521
27	1.57079632992561	0.00000000313298
28	1.57079632836721	0.00000000155840
29	1.57079632757422	0.00000000079299
30	1.57079632715686	0.00000000041736
31	1.57079632700710	0.00000000014977
32	1.57079632700710	0.00000000000000

Die exakte Lösung $\xi = \frac{\pi}{2}$ lautet auf 14 Dezimalen gerundet $\frac{\pi}{2} = 1.57079632679489$. Vergleicht man sie mit dem Näherungswert $x^{(32)}$, so gilt für den absoluten Fehler $|x^{(32)} - \xi| = |x^{(32)} - \frac{\pi}{2}| \leq 0.21{\cdot}10^{-9}$; obwohl also $|x^{(32)} - x^{(31)}| \leq 0.5{\cdot}10^{-14}$ ist, erhält man einen Näherungswert $x^{(32)}$, der nur auf 9 Dezimalen genau ist. Die Ursache dafür liegt darin, dass der Ausdruck f/f' in der Iterationsvorschrift für $x^{(\nu)} \to \xi$ unbestimmt $(\frac{0}{0})$ wird. Wird der Zähler in der mitgeführten Stellenzahl früher Null als der Nenner, so bleibt die Iteration dort stehen; dieser Fall liegt hier vor; andernfalls entfernen sich die iterierten Werte wieder von der Lösung. Wenn der Nenner Null wird, muss das Verfahren abgebrochen werden.

2. Newtonsches Verfahren für doppelte Nullstellen (Satz 2.25 mit Vielfachheit $j = 2$):

$$x^{(\nu+1)} = x^{(\nu)} - 2\frac{f(x^{(\nu)})}{f'(x^{(\nu)})} = x^{(\nu)} - 2\frac{1 - \sin(x^{(\nu)})}{-\cos(x^{(\nu)})}$$

ν	$x^{(\nu)}$	$\lvert x^{(\nu)} - x^{(\nu-1)}\rvert$
0	2.00000000000000	
1	1.56408380307828	0.43591619692172
2	1.57079635199940	0.00671254892113
3	1.57079632679621	0.00000002520319
4	1.57079632679621	0.00000000000000

Hier ist die Abfrage $|x^{(\nu)} - x^{(\nu-1)}| \leq 0.5{\cdot}10^{-14}$ bereits nach vier Iterationsschritten erfüllt, da das Verfahren quadratisch konvergiert. Aber auch hier gilt für den absoluten Fehler des Näherungswertes $x^{(4)}$ $|x^{(4)} - \frac{\pi}{2}| \leq 0.13{\cdot}10^{-11}$; die Ursache dafür ist dieselbe wie im ersten Falle.

3. Modifiziertes Newtonsches Verfahren für mehrfache Nullstellen (Satz 2.26): Rechnung mit doppelter Stellenzahl, Rundung auf einfache.

ν	$x^{(\nu)}$	$\lvert x^{(\nu)} - x^{(\nu-1)}\rvert$	$J(x^{(\nu)})$
0	2.0000000000000		1.9092974268257
1	1.5838531634529	0.4161468365471	1.9999147607192
2	1.5707966977821	0.0130564656707	2.0000003799090
3	1.5707963267948	0.0000003709874	1.0000000000000
4	1.5707963267948	0.0000000000000	

Für den absoluten Fehler des Näherungswertes $x^{(4)}$ gilt $|x^{(4)} - \frac{\pi}{2}| \leq 0.25{\cdot}10^{-12}$, obwohl für die iterierten Werte die Abfrage $|x^{(4)} - x^{(3)}| \leq 0.5{\cdot}10^{-14}$ erfüllt ist. Die Ursache liegt wie oben darin, dass $\sin(x^{(3)})$ in den mitgeführten Stellen exakt 1 ist, d. h. $1 - \sin(x^{(3)})$ verschwindet, aber $\cos(x^{(3)}) \neq 0$ ist, so dass $x^{(4)} = x^{(3)}$ gilt. Man sieht hier außerdem, dass $J(x^{(2)})$ der tatsächlichen Vielfachheit $j = 2$ schon sehr nahe ist, $J(x^{(3)})$ jedoch gleich 1 wird. Dies liegt daran, dass $f(x^{(3)}) = 1 - \sin(x^{(3)})$ in den mitgeführten Stellen verschwindet, $f'(x^{(3)}) \neq 0$ ist und damit $f\,f''/f'^2$ verschwindet. Würde hier f'^2 vor f Null, so würde $J(x^{(\nu)})$ von j in beliebiger Weise abwandern.

Wegen $f(x) = 1 - \sin x$ kann in diesem Fall $J(x^{(\nu)})$ vereinfacht dargestellt werden (Satz 2.26):

$$J(x^{(\nu)}) = \frac{1}{1 - \frac{\left(1-\sin(x^{(\nu)})\right)\sin(x^{(\nu)})}{\cos^2(x^{(\nu)})}} = 1 + \sin(x^{(\nu)}).$$

Damit ergibt sich $J(x^{(3)}) = 1 + 1 = 2$. □

Empfehlung zum modifizierten Newton-Verfahren

Wegen seiner geringen Effizienz (vgl. Abschnitt 2.9) sollte man mit dem modifizierten Newton-Verfahren nur so lange iterieren, bis die Vielfachheit der Nullstelle geklärt ist, dann aber mit dem Verfahren von Newton für mehrfache Nullstellen weiterrechnen. Die Vielfachheit ist dann geklärt, wenn sich von einem gewissen ν an entweder $J(x^{(\nu)}) = 1$ ergibt oder

$$|J(x^{(\nu)}) - J(x^{(\nu-1)})| > |J(x^{(\nu-1)}) - J(x^{(\nu-2)})|$$

gilt, d. h. $J(x^{(\nu)})$ sich wieder von der Vielfachheit j entfernt. In beiden Fällen ist die zu $J(x^{(\nu-1)})$ nächste ganze Zahl die gesuchte Vielfachheit j. Dieses Verhalten der $J(x^{(\nu)})$ ist bedingt durch die beschränkte Stellenzahl der Maschinenzahlen und den für $x^{(\nu)} \to \xi$ unbestimmten Ausdruck ff''/f'^2 im Nenner von $J(x^{(\nu)})$. Wird nämlich für ein $x^{(\nu)}$ wegen der beschränkten Stellenzahl $f(x^{(\nu)})$ identisch Null, während $f'(x^{(\nu)})$ noch verschieden von Null ist, so erhält man $J(x^{(\nu)}) = 1$, obwohl eine mehrfache Nullstelle vorliegt.

Generelle Empfehlung. Da eine doppelte Nullstelle ξ von f nach Satz 2.3 eine einfache Nullstelle ξ von $g(x) = f(x)/f'(x)$ ist, kann auch zur Berechnung einer doppelten Nullstelle ein Einschlussverfahren verwendet werden. Für Beispiel 2.27 würde das bedeuten, dass man anstelle der doppelten Nullstelle von $f(x) = 1 - \sin x$ die einfache Nullstelle der Funktion

$$g(x) = \frac{f(x)}{f'(x)} = \frac{1 - \sin x}{-\cos x}$$

mit einem Einschlussverfahren ermittelt, das mit größerer Effizienz arbeitet als das Newton-Verfahren (siehe Abschnitt 2.9). Die oben erwähnten numerischen Probleme wegen $g(\xi) = \frac{0}{0}$ treten natürlich auch hier auf.

Anmerkungen zu mehrfachen Nullstellen. Bei einer nichtlinearen Gleichung $f(x) = 0$, die im Rahmen eines technischen Problems gelöst werden muss, werden die im Funktionsterm auftretenden Daten im Allgemeinen mit Fehlern behaftet sein. Darum ist nicht zu erwarten, dass die Funktion f eine mehrfache Nullstelle hat.

Beispielsweise hat die quadratische Funktion $f(x) = x^2 + px + q$ genau dann eine doppelte Nullstelle, wenn exakt $p^2 - 4q = 0$ ist. Wenn p und q mit Messfehlern behaftet sind, wird $|\, p^2 - 4q \,| = \varepsilon > 0$ sein. Für $p^2 - 4q > 0$ gibt es dann zwei benachbarte einfache Nullstellen und für $p^2 - 4q < 0$ keine.

Eine doppelte Nullstelle ξ von f mit $f(\xi) = f'(\xi) = 0$, $f''(\xi) \neq 0$ ist eine Stelle, an der die Funktion f ein lokales Minimum oder Maximum besitzt. Wegen $f''(\xi) \neq 0$ ist ξ einfache Nullstelle von f' und kann mit einem Einschlussverfahren berechnet werden. Wenn dann $|f(\xi)| \leq \varepsilon$ ist, kann ξ als doppelte Nullstelle von f akzeptiert werden.

2.6 Das Sekantenverfahren

Das Sekantenverfahren wird hier behandelt, weil es in fast jeder Numerik-Vorlesung als Standardverfahren vorkommt. Für die praktische Anwendung ist es allerdings nicht zu empfehlen, man sollte dem Sekantenverfahren die Einschlussverfahren mit höherer Effizienz unbedingt vorziehen.

2.6.1 Das Sekantenverfahren für einfache Nullstellen

Die Funktion f sei in $I = [a, b]$ stetig und besitze in (a, b) eine einfache Nullstelle ξ. Zur näherungsweisen Bestimmung von ξ mit Hilfe des Verfahrens von Newton ist die Berechnung der Ableitung f' von f erforderlich, so dass die Differenzierbarkeit von f vorausgesetzt werden muss. Das Sekantenverfahren ist ein Iterationsverfahren, das ohne Ableitungen arbeitet und zwei Startwerte $x^{(0)}$, $x^{(1)}$ erfordert (siehe auch Abschnitt 2.7 Einschlussverfahren).

Durch die folgende Überlegung gelangt man zu der Iterationsvorschrift für das Verfahren (Abb. 2.13). Es seien $x^{(0)}$, $x^{(1)} \in I$. Durch die Punkte $\big(x^{(0)}, f(x^{(0)})\big)$ und $\big(x^{(1)}, f(x^{(1)})\big)$ legt man die Sekante des Funktionsgraphen und schneidet sie mit der x-Achse. Für die Abszisse $x^{(2)}$ des Schnittpunktes findet man

$$x^{(2)} = x^{(1)} - \frac{x^{(1)} - x^{(0)}}{f(x^{(1)}) - f(x^{(0)})} f(x^{(1)}).$$

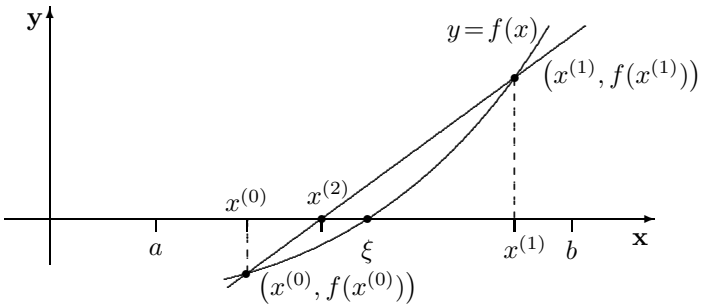


Abb. 2.13. Sekantenverfahren

Die Iterationsvorschrift für das Sekantenverfahren lautet

$$\begin{cases} x^{(\nu+1)} = x^{(\nu)} - \frac{x^{(\nu)} - x^{(\nu-1)}}{f(x^{(\nu)}) - f(x^{(\nu-1)})} f(x^{(\nu)}) & \text{wenn } f(x^{(\nu)}) \neq f(x^{(\nu-1)}), \\ x^{(\nu+1)} = x^{(\nu)} - \frac{x^{(\nu)} - x^{(\nu-1)}}{f(x^{(\nu)}) - \mathbf{2} \cdot f(x^{(\nu-1)})} f(x^{(\nu)}) & \text{sonst}, \\ \nu = 1, 2, \ldots \end{cases} \tag{2.50}$$

Falls $f(x^{(\nu+1)}) = 0$ ist, wird das Verfahren mit $\xi = x^{(\nu+1)}$ abgebrochen. Wesentlich für die Konvergenz des Verfahrens ist, dass die Startwerte $x^{(0)}$, $x^{(1)}$ hinreichend nahe an der Nullstelle ξ liegen. Es gilt der folgende *Konvergenzsatz*:

Satz 2.28.
Falls die Funktion f für alle $x \in (a, b)$ zweimal stetig differenzierbar ist und mit zwei positiven Zahlen m, M den Bedingungen

$$|f'(x)| \geq m, \quad |f''(x)| \leq M, \quad x \in (a, b),$$

genügt, gibt es immer eine Umgebung $I_r = [\xi - r, \xi + r] \subset (a, b)$, $r > 0$, so dass ξ in I_r die einzige Nullstelle von f ist und das Verfahren für jedes Paar von Startwerten $x^{(0)}$, $x^{(1)} \in I_r$, $x^{(0)} \neq x^{(1)}$, gegen die gesuchte Nullstelle ξ konvergiert.

Da die Überprüfung der Voraussetzungen dieses Satzes meist nicht praktikabel ist, bleibt nur die Empfehlung, die Startwerte $x^{(0)}$ und $x^{(1)}$ möglichst nahe bei der Nullstelle ξ zu wählen.
Wenn das Startintervall den Bedingungen des Satzes 2.28 nicht genügt, kann beim Sekantenverfahren die Konvergenz von der Bezeichnung der Startwerte $x^{(0)}$ und $x^{(1)}$ abhängen.

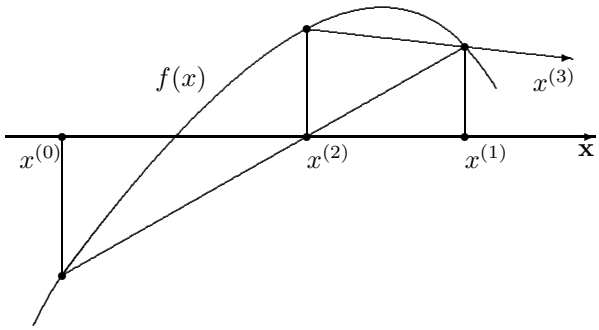


Abb. 2.14. Startwerte, für die das Sekantenverfahren divergiert

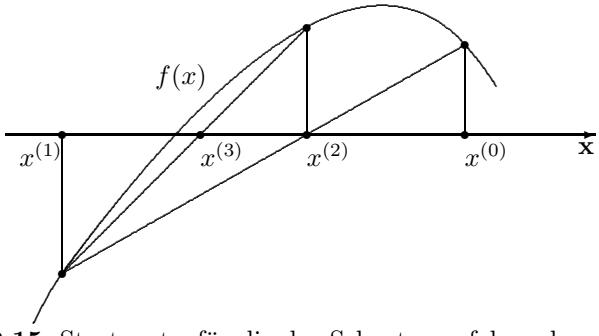


Abb. 2.15. Startwerte, für die das Sekantenverfahren konvergiert

Beispiel 2.29.

Gegeben: Die Funktion $f(x) = \frac{1}{8}x^2 - x + \frac{3}{2}$ und die Startwerte $x^{(0)} = 1$, $x^{(1)} = 5$, die wegen $f(1) = 0.625$, $f(5) = -0.375$ eine Nullstelle von f einschließen.

Gesucht: Die zwischen $x^{(0)}$ und $x^{(1)}$ liegende Nullstelle mit dem Sekantenverfahren.

Lösung: Mit den gegebenen Startwerten konvergiert das Sekantenverfahren gegen die Nullstelle 6, die nicht im Einschlussintervall $[1, 5]$ liegt. Mit den Startwerten $x^{(0)} = 5$, $x^{(1)} = 1$ liefert das Sekantenverfahren die Nullstelle 2.

Ein Einschlussverfahren (siehe Abschnitt 2.7) ermittelt in beiden Fällen die Nullstelle 2. □

Beispiel 2.30.

Gegeben: Die Funktion $f(x) = \ln x - \sqrt{x} + 1.5$, die für $x > 0$ reell und stetig ist, und die Startwerte $x^{(0)} = 0.2$, $x^{(1)} = 2$, die wegen $f(0.2) \approx -0.557$, $f(2) \approx 0.779$ eine Nullstelle von f einschließen.

Gesucht: Die im Intervall $[0.2, 2]$ liegende Nullstelle mit dem Sekantenverfahren.

Lösung: Mit den gegebenen Startwerten erzeugt das Sekantenverfahren

$$\begin{aligned} x^{(2)} &= 0.950213 \in [0.2, 2] \quad \text{und} \\ x^{(3)} &= -0.682864 \notin [0.2, 2]\,. \end{aligned}$$

Also ist $f(x^{(3)})$ nicht reell, und das Sekantenverfahren versagt.
Mit den Startwerten $x^{(0)} = 2$, $x^{(1)} = 0.2$ konvergiert das Sekantenverfahren gegen die Nullstelle $\xi = 0.429\,817\,028$.

Ein Einschlussverfahren (siehe Abschnitt 2.7) erzeugt dagegen Näherungen $x^{(\nu)}$, die stets im Einschlussintervall $[x^{(0)}, x^{(1)}]$ liegen. □

Die Konvergenzordnung des Sekantenverfahrens mit der Iterationsvorschrift (2.50) ist $p = (1 + \sqrt{5})/2 \approx 1.62$. Prinzipiell kann die Vorschrift (2.50) auch zur näherungsweisen Berechnung mehrfacher Nullstellen verwendet werden, dann geht jedoch die hohe Konvergenzordnung verloren. Das *modifizierte Sekantenverfahren* (Abschnitt 2.6.2) besitzt auch bei mehrfachen Nullstellen die Konvergenzordnung p ≈ 1.62. Zur Effizienz der Verfahren siehe Abschnitt 2.9.

2.6.2 Das modifizierte Sekantenverfahren für mehrfache Nullstellen

Ist ξ eine Nullstelle der Vielfachheit j, $j \geq 2$, von f und ist $|f^{(j+1)}(x)|$ in der Umgebung von ξ beschränkt, so ist ξ eine einfache Nullstelle der Funktion h

$$h(x) = \frac{f^2(x)}{f\big(x + f(x)\big) - f(x)}\,, \tag{2.51}$$

und $|h''(x)|$ ist in der Umgebung von ξ beschränkt. Verwendet man in der Iterationsvorschrift (2.50) statt f die durch (2.51) definierte Funktion h, so konvergiert dieses modifizierte Sekantenverfahren ebenfalls von der Ordnung $p = (1 + \sqrt{5})/2$ gegen die mehrfache Nullstelle ξ von f (Beweis s. [KIOU1979]). Die Effizienz ist wegen $H = 2$ allerdings nur $E = p^{1/2} = 1.272$ (siehe Abschnitt 2.9).

2.7 Einschlussverfahren

Eine stetige Funktion $f : [a, b] \to \mathbb{R}$ mit $f(a) \cdot f(b) < 0$ besitzt nach dem Zwischenwertsatz im Intervall (a, b) mindestens eine Nullstelle; ein solches Intervall $[a, b]$ nennt man ein Einschlussintervall. Im Folgenden sei das Intervall $[a, b]$ so gewählt, dass es genau eine Nullstelle ξ der Funktion f einschließt.

Bei den in diesem Abschnitt behandelten Einschlussverfahren wird ein gegebenes Einschlussintervall in zwei Teilintervalle zerlegt; eines von diesen ist wieder ein Einschlussintervall. So können fortlaufend kleinere Einschlussintervalle erzeugt werden (Abschnitt 2.7.1).

Das einfachste dieser Verfahren ist das Bisektionsverfahren (Abschnitt 2.7.2), das stets, aber nur linear, gegen die Nullstelle konvergiert.

Bei der Regula falsi (Abschnitt 2.7.3), die ebenfalls nur linear konvergiert, kann es vorkommen, dass die erzeugten Grenzen der Einschlussintervalle sich der Nullstelle nur von einer Seite her nähern (Abbildung 2.17).

Deshalb wurden einige Einschlussverfahren (Abschnitte 2.7.4 bis 2.7.6) entwickelt, die dieses Verhalten vermeiden und zudem eine höhere Konvergenzordnung besitzen. Allerdings wird diese höhere Konvergenzordnung erst in einer hinreichend kleinen Umgebung der Nullstelle wirksam.

Daher ist es zweckmäßig, ein gegebenes Einschlussintervall zunächst mit dem Bisektionsverfahren zu verkleinern und dann ein Verfahren höherer Ordnung einzusetzen (siehe dazu das Beispiel 2.41).

Falls eine genügend oft stetig differenzierbare Funktion f im Intervall $[a, b]$ eine Nullstelle ξ gerader Ordnung besitzt, ist diese wegen Satz 2.3 eine einfache Nullstelle der Funktion $g(x) = f(x)/f'(x)$ und kann daher mit einem Einschlussverfahren, angewandt auf g, ermittelt werden. Siehe dazu auch die Anmerkungen zu mehrfachen Nullstellen am Ende des Abschnitts 2.5.

2.7.1 Das Prinzip der Einschlussverfahren

Ausgehend von einem Einschlussintervall $[a, b]$ werden $x^{(1)} = a$, $f_1 = f(a)$, $x^{(2)} = b$, $f_2 = f(b)$ gesetzt; damit ist $f_1 \cdot f_2 < 0$. Dann wird eine Zahl

$$x^{(3)} = x^{(2)} + q\,(x^{(1)} - x^{(2)})\,, \quad 0 < q < 1\,, \tag{2.52}$$

erzeugt, die zwischen $x^{(1)}$ und $x^{(2)}$ liegt. Die Verfahren unterscheiden sich in der Wahl von q. Mit $x^{(3)}$ wird der Funktionswert $f_3 = f(x^{(3)})$ berechnet. Wenn $f_3 = 0$ ist, ist $\xi = x^{(3)}$ Nullstelle von f.

Wenn $f_3 \neq 0$ ist, hat f entweder zwischen $x^{(2)}$ und $x^{(3)}$ oder zwischen $x^{(1)}$ und $x^{(3)}$ einen Vorzeichenwechsel. Eines der beiden durch $x^{(3)}$ erzeugten Teilintervalle ist also wieder ein Einschlussintervall, mit dem das Verfahren fortgesetzt werden kann. Das neue Einschlussintervall wird wie folgt ermittelt.

Wenn $f_2 \cdot f_3 < 0$ ist, liegt ξ zwischen $x^{(2)}$ und $x^{(3)}$, und die Intervallgrenzen und Funktionswerte werden umbenannt:

$$\begin{aligned} x^{(1)} &:= x^{(2)}\,, \quad f_1 := f_2\,, \\ x^{(2)} &:= x^{(3)}\,, \quad f_2 := f_3\,. \end{aligned}$$

Wenn $f_2 \cdot f_3 > 0$ ist und somit $f_1 \cdot f_3 < 0$ ist, liegt ξ zwischen $x^{(1)}$ und $x^{(3)}$. Dann werden gesetzt:

$$x^{(2)} := x^{(3)}\,, \quad f_2 := f_3\,.$$

Nun gilt wieder $f_1 \cdot f_2 < 0$, und mit $x^{(1)}$ und $x^{(2)}$ kann das Verfahren fortgesetzt werden. Dabei ist $x^{(2)}$ die zuletzt berechnete Intervallgrenze.

Da ξ zwischen $x^{(1)}$ und $x^{(2)}$ liegt, gelten für die absoluten Fehler dieser Intervallgrenzen

$$|\,\xi - x^{(1)}\,| < |\,x^{(2)} - x^{(1)}\,|\,, \quad |\,\xi - x^{(2)}\,| < |\,x^{(2)} - x^{(1)}\,|\,.$$

Wenn mit einer positiven Schranke AbsErr für den absoluten Fehler für die Länge des Einschlussintervalls

$$|\,x^{(2)} - x^{(1)}\,| \leq \text{ AbsErr} \tag{2.53}$$

gilt, sind die absoluten Fehler der beiden Intervallgrenzen kleiner als AbsErr.

Wenn mit einer positiven Schranke RelErr für den relativen Fehler mit der zuletzt berechneten Intervallgrenze $x^{(2)}$ $(\neq 0)$

$$|\,x^{(2)} - x^{(1)}\,| \leq |\,x^{(2)}\,|\,\text{RelErr} \tag{2.54}$$

ist, gilt für den relativen Fehler dieser Intervallgrenze

$$\frac{|\,\xi - x^{(2)}\,|}{|\,x^{(2)}\,|} < \frac{|\,x^{(2)} - x^{(1)}\,|}{|\,x^{(2)}\,|} \leq \text{ RelErr.}$$

Ein Einschlussverfahren kann also außer mit $f_3 = 0$ auch abgebrochen werden, wenn eine der Abfragen (2.53), (2.54) erfüllt ist. Beide können kombiniert werden zu

$$|\,x^{(2)} - x^{(1)}\,| \leq |\,x^{(2)}\,|\,\text{RelErr} \;+\; \text{AbsErr}\,.$$

Mit AbsErr > 0 und RelErr $= 0$ entsteht (2.53), und mit AbsErr $= 0$ und RelErr > 0 ergibt sich (2.54).

Von den beiden Intervallgrenzen $x^{(1)}$, $x^{(2)}$ kann man diejenige als die beste Näherung für die Nullstelle ξ wählen, für die der Funktionswert dem Betrage nach kleiner ist.

Die Fehlerschranken AbsErr und RelErr müssen größer als die Maschinengenauigkeit ϱ sein (etwa $2\,\varrho$ bis $3\,\varrho$).

Bemerkung. Für die praktische Anwendung ist es sinnvoll, die Abbruchbedingung (2.54) für den relativen Fehler zu verwenden, weil so eine Aussage über die Anzahl der gültigen Ziffern einer Näherungszahl gemacht werden kann (vgl. Satz 1.24). Der absolute Fehler macht eine Aussage über die Anzahl der gültigen Dezimalen (vgl. Definition 17).

Das heißt, die Abfrage

$$|\,x^{(2)} - \xi\,| \leq \text{ AbsErr } = 0.5 \cdot 10^{-k}$$

liefert $x^{(2)}$ mit k genauen Dezimalen, die Abfrage

$$\frac{|\,x^{(2)} - \xi\,|}{|\,x^{(2)}\,|} \leq \text{ RelErr } = 5 \cdot 10^{-m}$$

liefert $x^{(2)}$ mit m genauen Ziffern, beginnend mit der ersten von 0 verschiedenen Ziffer von $x^{(2)}$.

2.7.2 Das Bisektionsverfahren

Bei diesem einfachsten Einschlussverfahren wird in (2.52) $q = 0.5$ gesetzt. Dann wird wegen

$$| x^{(3)} - x^{(2)} | = 0.5 \, | x^{(1)} - x^{(2)} |$$

die Länge des Einschlussintervalls halbiert. Mit fortgesetzter Intervallhalbierung konvergiert das Bisektionsverfahren linear gegen die Nullstelle.

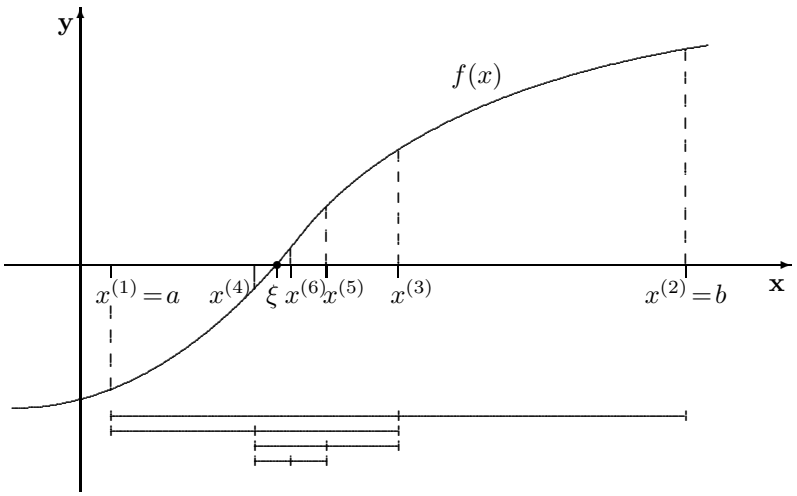


Abb. 2.16. Das Bisektionsverfahren

Algorithmus 2.31. (*Bisektionsverfahren*)

Gegeben: (i) $f \in \mathrm{C}[a,b]$ mit $f(a) \cdot f(b) < 0$,
(ii) Schranken AbsErr und RelErr für den absoluten bzw. den relativen Fehler mit AbsErr > 0 und RelErr $= 0$
oder mit AbsErr $= 0$ und RelErr > 0.

Gesucht: Eine Zahl $\xi \in (a,b)$, für die $f(\xi) = 0$ ist, oder ein Einschlussintervall $[x^{(1)}, x^{(2)}]$ bzw. $[x^{(2)}, x^{(1)}]$ für ξ mit
$| x^{(1)} - x^{(2)} | \leq | x^{(2)} | \,\mathrm{RelErr} + \mathrm{AbsErr}$.

Vorbereitung: $x^{(1)} := a, \quad x^{(2)} := b$
$f_1 := f(x^{(1)}), \; f_2 := f(x^{(2)})$.

Pro Iterationsschritt wird wie folgt vorgegangen:

1. *Halbierung des Einschlussintervalls* durch Ermittlung von

$$x^{(3)} := x^{(2)} + 0.5(x^{(1)} - x^{(2)}) \, .$$

2. *Berechnung des neuen Funktionswertes* $f_3 := f(x^{(3)})$.
   Falls $f_3 = 0$ ist, wird die Iteration mit $\xi := x^{(3)}$ abgebrochen, andernfalls geht es mit 3. weiter.
3. *Festlegung des neuen Einschlussintervalls:*
   Falls $f_2 \cdot f_3 < 0$ ist, liegt ξ zwischen $x^{(2)}$ und $x^{(3)}$, und es wird gesetzt
   $$x^{(1)} := x^{(2)},\ x^{(2)} := x^{(3)}, \quad f_1 := f_2,\ f_2 := f_3;$$
   falls $f_2 \cdot f_3 > 0$ ist, liegt ξ zwischen $x^{(1)}$ und $x^{(3)}$, und es wird gesetzt
   $$x^{(2)} := x^{(3)}, \quad f_2 := f_3.$$
   In beiden Fällen liegt jetzt ξ zwischen $x^{(1)}$ und $x^{(2)}$, und $x^{(2)}$ ist der zuletzt berechnete Näherungswert.
4. *Prüfung der Abbruchbedingung*:
   Falls
   $$|\, x^{(1)} - x^{(2)} \,| \le |\, x^{(2)} \,| \,\mathrm{RelErr} \;+\; \mathrm{AbsErr}$$
   ist, erfolgt Abbruch. Dann wird gesetzt
   $$\xi := x^{(2)}, \qquad \text{falls } |f_2| \le |f_1| \text{ ist}, \qquad \text{und sonst} \qquad \xi := x^{(1)}.$$
   Andernfalls wird die Iteration mit 1. fortgesetzt.

Bemerkung 2.32. Beim Bisektionsverfahren kann die Anzahl n der Intervallhalbierungen, die erforderlich ist, um eine vorgegebene Anzahl k gültiger Dezimalen zu erhalten, vorab bestimmt werden. Für n und k ($k \in \mathbb{N}$) gelte also

$$\frac{|b-a|}{2^n} \quad \le \quad \frac{1}{2}\cdot 10^{-k}\,.$$

Dabei sei $0.5 \cdot 10^{-k} < |b-a|$. Dann folgen

$$\begin{aligned} 2^n &\ge 2\,|b-a|\,10^k\,, \\ n \lg 2 &\ge \lg 2 + \lg|b-a| + k\,, \\ n &\ge 1 + \frac{k + \lg|b-a|}{\lg 2}\,. \end{aligned} \tag{2.55}$$

Beispiel 2.33.

Gegeben: Die Funktion $f(x) = \sin x + 1 - 1/x$ für $x \in [a,b]$ mit $a = 0.6$, $b = 0.7$ und die Schranke $\mathrm{AbsErr} = 0.5{\cdot}10^{-6}$ für den absoluten Fehler.

Gesucht: Eine Nullstelle ξ der Funktion f im Intervall $(0.6, 0.7)$ mit Hilfe des Bisektionsverfahrens auf 6 Dezimalen genau.

Lösung: Wegen $f(0.6) \approx -0.102$ und $f(0.7) \approx 0.216$ liegt mindestens eine Nullstelle in $(0.6, 0.7)$. Mit $k = 6$, $a = 0.6$ und $b = 0.7$ erhält man nach der Formel (2.55)

die Anzahl der erforderlichen Intervallhalbierungen:

$$n \;\geq\; 1 + \frac{6 + \lg 0.1}{\lg 2} \;=\; 17.6096\,.$$

Nach 18 Intervallhalbierungen ist die Länge des letzten Einschlussintervalls $\leq 0.5{\cdot}10^{-6}$.

ν	$x^{(3)}$	$f(x^{(3)})$	$x^{(1)}$	$f(x^{(1)})$	$x^{(2)}$	$f(x^{(2)})$	$\lvert x^{(2)} - x^{(1)}\rvert$
			0.6	$f < 0$	0.7	$f > 0$	0.1
1	0.65	+0.0667249	0.6	$f < 0$	0.65	$f > 0$	0.05
2	0.625	−0.0149027	0.65	$f > 0$	0.625	$f < 0$	0.025
3	0.6375	+0.0265609	0.625	$f < 0$	0.6375	$f > 0$	0.0125
⋮							
16	0.6294479	+0.0000048	0.6294464	$f < 0$	0.6294479	$f > 0$	0.0000015
17	0.6294472	+0.0000023	0.6294464	$f < 0$	0.6294472	$f > 0$	0.0000008
18	0.6294468	+0.0000010	0.6294464	$-2.43{\cdot}10^{-7}$	0.6294468	$1.03{\cdot}10^{-6}$	$0.0000004 < \varepsilon$

In dieser Tabelle steht jeder Schritt in einer Zeile. Die neue Näherung ist $x^{(3)}$. Nach Umspeichern ergeben sich rechts die Grenzen $x^{(1)}$ und $x^{(2)}$ des neuen Einschlussintervalls. In der letzten Spalte kann man die Abbruchbedingung prüfen.

Wegen $|f(x^{(1)})| < |f(x^{(2)})|$ für $\nu = 18$ wird gewählt

$$\xi \;=\; 0.629\,446 \;\approx\; x^{(1)}\,.$$

□

2.7.3 Die Regula falsi

Dieses Verfahren verwendet die Sekante, die die Punkte $(x^{(1)}, f_1)$ und $(x^{(2)}, f_2)$ verbindet, und bestimmt deren Schnittpunkt $(x^{(3)}, 0)$ mit der x-Achse (Sekantenschritt). Mit der Gleichung

$$y = f_2 + \frac{f_1 - f_2}{x^{(1)} - x^{(2)}}\,(x - x^{(2)})$$

dieser Sekante ergibt sich für $y = 0$

$$x^{(3)} = x^{(2)} + \frac{f_2}{f_2 - f_1}\,(x^{(1)} - x^{(2)})\,.$$

Hier ist also mit (2.52)

$$q = \frac{f_2}{f_2 - f_1}\,,$$

und wegen $f_1 \cdot f_2 < 0$ ist $0 < q < 1$.

Wie die Abbildung 2.17 zeigt, kann der Fall eintreten, dass die Grenzen der Einschlussintervalle sich der Nullstelle nur von einer Seite her nähern, während die Grenze $x^{(1)}$ auf der anderen Seite der Nullstelle unverändert bleibt.

Dabei wird die Korrektur

$$\Delta x = x^{(3)} - x^{(2)} = q\,(x^{(1)} - x^{(2)})$$

dem Betrage nach immer kleiner. Um von einer Intervallgrenze $x^{(2)}$ nahe der Nullstelle auf die andere Seite der Nullstelle zu gelangen und damit die Grenze $x^{(1)}$ loszuwerden, kann man wie folgt vorgehen:

Es sei

$$\text{tol} \;= |\, x^{(2)} \,|\, \text{RelErr} \;+\; \text{AbsErr}\,.$$

Wenn $|\,\Delta x\,| \leq \text{tol}$ ist, wird

$$\Delta x = 0.9 \cdot \text{tol} \;\cdot\; \text{sgn}\,(x^{(1)} - x^{(2)})$$

gesetzt. Wenn dann $x^{(2)}$ und $x^{(3)} = x^{(2)} + \Delta x$ die Nullstelle einschließen, ist wegen $|\,\Delta x\,| < \text{tol}$ auch die Abbruchbedingung

$$|\, x^{(3)} - x^{(2)} \,| = |\,\Delta x\,| < \;\text{tol}$$

erfüllt. Der Faktor 0.9 verhindert, dass die Abbruchbedingung infolge von Rundungsfehlern evtl. nicht erfüllt ist.

Bemerkung. Dieser zusätzliche Schritt wird auch in die folgenden Algorithmen zum Pegasus-Verfahren und zum Verfahren von Anderson-Björck aufgenommen.

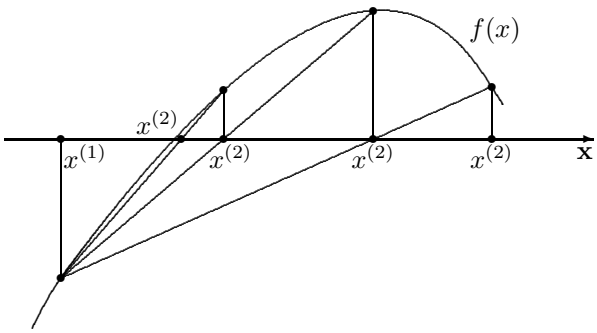


Abb. 2.17. Mit $x^{(2)} := x^{(3)}$ in allen Schritten

Algorithmus 2.34. (*Regula falsi*)

Gegeben: (i) $f \in \mathrm{C}[a,b]$ mit $f(a) \cdot f(b) < 0$,
(ii) Schranken AbsErr und RelErr für den absoluten bzw. den relativen Fehler mit AbsErr > 0 und RelErr $= 0$
oder mit AbsErr $= 0$ und RelErr > 0.

Gesucht: Eine Zahl $\xi \in (a,b)$, für die $f(\xi) = 0$ ist, oder ein Einschlussintervall $[x^{(1)}, x^{(2)}]$ bzw. $[x^{(2)}, x^{(1)}]$ für ξ mit
$| x^{(1)} - x^{(2)} | \leq | x^{(2)} |\,\mathrm{RelErr} \; + \; \mathrm{AbsErr}$.

Vorbereitung: $x^{(1)} := a, \qquad x^{(2)} := b$
$f_1 := f(x^{(1)}),\ f_2 := f(x^{(2)})$.

Pro Iterationsschritt wird wie folgt vorgegangen:

1. *Bestimmung der neuen Intervallgrenze*:

   Berechne
   $$\mathrm{tol} \; := | x^{(2)} |\,\mathrm{RelErr} \; + \; \mathrm{AbsErr}$$
   und
   $$\Delta x := \frac{f_2}{f_2 - f_1}\,(x^{(1)} - x^{(2)})\,.$$
   Wenn $| \Delta x | \leq \mathrm{tol}$ ist, setze
   $$\Delta x := 0.9 \cdot \mathrm{tol} \; \cdot \; \mathrm{sgn}\,(x^{(1)} - x^{(2)})\,.$$
   Berechne $x^{(3)} := x^{(2)} + \Delta x$.

2. *Berechnung des neuen Funktionswertes* $f_3 := f(x^{(3)})$.
   Falls $f_3 = 0$ ist, wird die Iteration mit $\xi := x^{(3)}$ abgebrochen, andernfalls geht es mit 3. weiter.

3. *Festlegung des neuen Einschlussintervalls:*
   Falls $f_2 \cdot f_3 < 0$ ist, liegt ξ zwischen $x^{(2)}$ und $x^{(3)}$, und es wird gesetzt
   $$x^{(1)} := x^{(2)},\ x^{(2)} := x^{(3)}, \quad f_1 := f_2,\ f_2 := f_3;$$
   falls $f_2 \cdot f_3 > 0$ ist, liegt ξ zwischen $x^{(1)}$ und $x^{(3)}$, und es wird gesetzt
   $$x^{(2)} := x^{(3)}, \quad f_2 := f_3.$$
   In beiden Fällen liegt jetzt ξ zwischen $x^{(1)}$ und $x^{(2)}$, und $x^{(2)}$ ist der zuletzt berechnete Näherungswert.

4. *Prüfung der Abbruchbedingung*:
   Falls
   $$| x^{(1)} - x^{(2)} | \leq \; \mathrm{tol}$$
   ist, erfolgt Abbruch. Dann wird gesetzt
   $$\xi := x^{(2)}, \qquad \text{falls } \; |f_2| \leq |f_1| \text{ ist}\,, \qquad \text{und sonst} \qquad \xi := x^{(1)}.$$
   Andernfalls, also mit $| x^{(2)} - x^{(1)} | > \mathrm{tol}$, wird die Iteration mit 1. fortgesetzt.

Beispiel 2.35. (vgl. Beispiel 2.33)

Gegeben: Die Funktion $f(x) = \sin x + 1 - 1/x$ für $x \in [a, b]$ mit $a = 0.6$, $b = 0.7$ sowie die Schranke RelErr $= 5{\cdot}10^{-7}$ für den relativen Fehler (AbsErr $= 0$).

Gesucht: Die Nullstelle ξ der Funktion f im Intervall $(0.6, 0.7)$ mit der Regula falsi mit 7 gültigen Ziffern.

Lösung:

ν	$x^{(3)}$	$f(x^{(3)})$	$x^{(1)}$	$f(x^{(1)})$	$x^{(2)}$	$f(x^{(2)})$	$\lvert x^{(2)} - x^{(1)}\rvert / \lvert x^{(2)}\rvert$
			0.6	< 0	0.7	> 0	
1	0.63211636	$9{\cdot}10^{-3}$	0.6	< 0	0.63211636	> 0	$5{\cdot}10^{-2}$
2	0.62954848	$3{\cdot}10^{-4}$	0.6	< 0	0.62954848	> 0	$4.69{\cdot}10^{-2}$
3	0.62945038	$1{\cdot}10^{-5}$	0.6	< 0	0.62945038	> 0	$4.68{\cdot}10^{-2}$
4	0.62944663	$5{\cdot}10^{-7}$	0.6	< 0	0.62944663	> 0	$4.68{\cdot}10^{-2}$
5	0.62944635	$-4.5{\cdot}10^{-7}$	0.62944663	$5{\cdot}10^{-7}$	0.62944635	$-4.5{\cdot}10^{-7}$	$4.5{\cdot}10^{-7}$

$\Rightarrow \xi \approx x^{(2)} = \underbrace{0.6294464}_{\text{7 genaue Ziffern}}$

□

2.7.4 Das Pegasus-Verfahren

Das Pegasus-Verfahren ist ein gegenüber der Regula falsi verbessertes Verfahren, das in einer hinreichend kleinen Umgebung einer einfachen Nullstelle die Konvergenzordnung $p = 1.642$ besitzt. Falls nach einem Sekantenschritt die älteste Intervallgrenze $x^{(1)}$ nicht ersetzt werden kann, wird der Funktionswert f_1 mittels $g \cdot f_1$, $0 < g < 1$, modifiziert. Damit verbessert sich die Chance, die Grenze $x^{(1)}$ nach dem nächsten Schritt ersetzen zu können. Siehe dazu die geometrische Interpretation und Abbildung 2.18.

Basierend auf den Originalarbeiten [DOWE1971] und [DOWE1972] wurde hier der folgende Algorithmus entwickelt.

Algorithmus 2.36. (*Pegasus-Verfahren*)

Gegeben: (i) $f \in \mathrm{C}[a,b]$ mit $f(a) \cdot f(b) < 0$,
(ii) Schranken AbsErr und RelErr für den absoluten bzw. den relativen Fehler mit AbsErr > 0 und RelErr $= 0$
oder mit AbsErr $= 0$ und RelErr > 0.

Gesucht: Eine Zahl $\xi \in (a,b)$, für die $f(\xi) = 0$ ist, oder ein Einschlussintervall $[x^{(1)}, x^{(2)}]$ bzw. $[x^{(2)}, x^{(1)}]$ für ξ mit
$| x^{(1)} - x^{(2)} | \leq | x^{(2)} |\,\mathrm{RelErr} \; + \; \mathrm{AbsErr}$.

Vorbereitung: $x^{(1)} := a, \qquad x^{(2)} := b$
$f_1 := f(x^{(1)}),\ f_2 := f(x^{(2)})$.

Pro Iterationsschritt ist wie folgt vorzugehen:

1. *Bestimmung der neuen Intervallgrenze*:

   Berechne $$\mathrm{tol} \; := | x^{(2)} |\,\mathrm{RelErr} \; + \; \mathrm{AbsErr}$$

   und $$\Delta x := \frac{f_2}{f_2 - f_1}\,(x^{(1)} - x^{(2)})\,.$$

   Wenn $| \Delta x | \leq \mathrm{tol}$ ist, setze
   $$\Delta x := 0.9 \cdot \mathrm{tol} \; \cdot \; \mathrm{sgn}\,(x^{(1)} - x^{(2)})\,.$$

   Berechne $x^{(3)} := x^{(2)} + \Delta x$.

2. *Berechnung des neuen Funktionswertes* $f_3 := f(x^{(3)})$.
   Falls $f_3 = 0$ ist, wird die Iteration mit $\xi := x^{(3)}$ abgebrochen, andernfalls geht es mit 3. weiter.

3. *Festlegung des neuen Einschlussintervalls:*
   Falls $f_2 \cdot f_3 < 0$ ist, wenn also die Nullstelle ξ zwischen $x^{(2)}$ und $x^{(3)}$ liegt, wird gesetzt
   $$x^{(1)} := x^{(2)},\ x^{(2)} := x^{(3)}, \quad f_1 := f_2,\ f_2 := f_3.$$

   Falls $f_2 \cdot f_3 > 0$ ist, wenn also die Nullstelle ξ zwischen $x^{(1)}$ und $x^{(3)}$ liegt, wird die Stelle $x^{(1)}$ beibehalten, jedoch mit einem mittels
   $$f_1 := g f_1, \quad g = \frac{f_2}{f_2 + f_3}, \quad 0 < g < 1$$
   abgeänderten Funktionswert (modifizierter Schritt). Dann wird gesetzt
   $$x^{(2)} := x^{(3)}, \qquad f_2 := f_3.$$

   ξ liegt jetzt zwischen $x^{(1)}$ und $x^{(2)}$, und $x^{(2)}$ ist die zuletzt berechnete Näherung.

4. *Prüfung der Abbruchbedingung*:
   Falls
   $$| x^{(1)} - x^{(2)} | \leq \; \mathrm{tol}$$
   ist, erfolgt Abbruch. Dann wird gesetzt
   $$\xi := x^{(2)}, \qquad \text{falls } \; |f_2| \leq |f_1| \text{ ist}\,, \qquad \text{und sonst} \qquad \xi := x^{(1)}.$$

   Andernfalls, also mit $| x^{(2)} - x^{(1)} | > \mathrm{tol}$, wird die Iteration mit 1. fortgesetzt.

Für den modifizierten Schritt wird nachfolgend eine geometrische Interpretation angegeben.

Geometrische Interpretation für den modifizierten Schritt

Die Konstruktion von

$$f_1^* := f_1 \cdot \frac{f_2}{f_2 + f_3} \quad \text{ergibt sich wegen} \quad \frac{f_1^*}{f_1} = \frac{f_2}{f_2 + f_3}$$

geometrisch wie folgt (Abbildung 2.18): Die Verbindungsgeraden der Punkte $(x^{(2)}, f_2+f_3)$ und $(x^{(1)}, f_1)$ sowie $(x^{(2)}, f_2)$ und $(x^{(1)}, f_1^*)$ schneiden sich in einem Punkt S der x-Achse. Mit dem Punkt $(x^{(1)}, f_1^*)$ verbessert sich die Chance, das Einschlussintervall auf der Seite von $x^{(1)}$ zu verkürzen. Das zeigt die nächste Näherung $x^{(4)}$ in Abbildung 2.18.

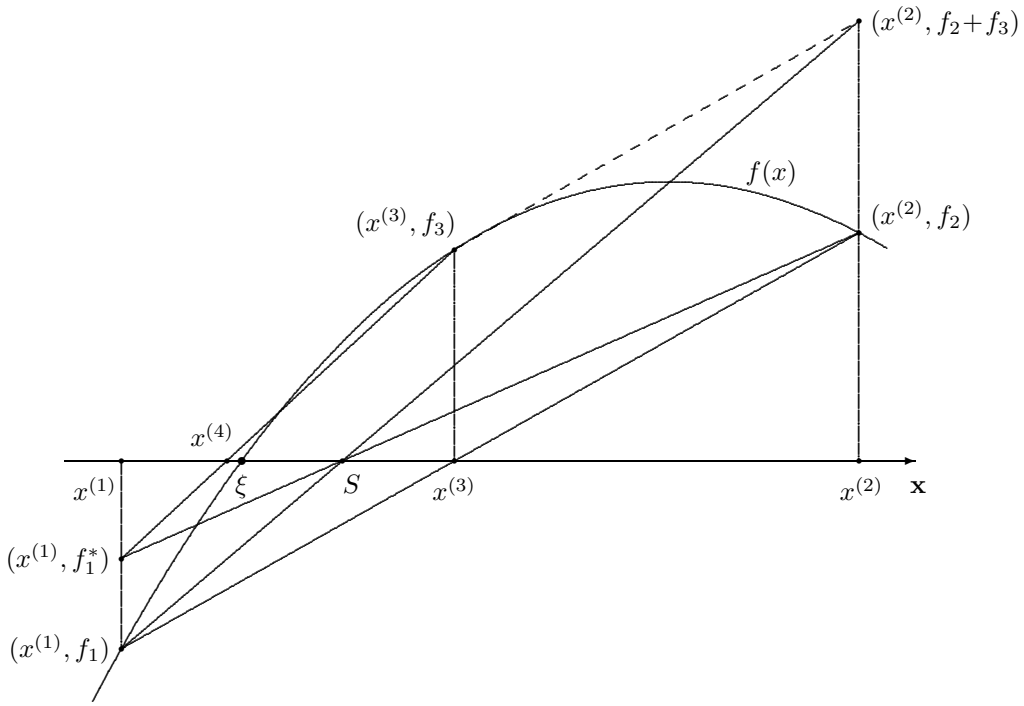


Abb. 2.18. Zum Pegasus-Verfahren, modifizierter Schritt

Beispiel 2.37. (vgl. Beispiel 2.35)

Gegeben: Die Funktion $f(x) = \sin x + 1 - 1/x$ und die Schranke RelErr $= 5 \cdot 10^{-7}$ für den relativen Fehler (AbsErr $= 0$).

Gesucht: Die Nullstelle ξ von f im Intervall $(0.6, 0.7)$ mit Hilfe des Pegasus-Verfahrens.

Lösung:

ν	$x^{(3)}$	$f(x^{(3)})$	$x^{(1)}$	$f(x^{(1)})$	$x^{(2)}$	$f(x^{(2)})$	$\lvert x^{(2)}-x^{(1)}\rvert/\lvert x^{(2)}\rvert$
			0.6	$f<0$	0.7	$f>0$	
1	0.6321164	$+8.9\cdot10^{-3}$	0.6	$f<0$	0.6321164	$f>0$	$5.08\cdot10^{-2}$
2	0.6294517	$+1.7\cdot10^{-5}$	0.6	$f<0$	0.6294517	$f>0$	$4.68\cdot10^{-2}$
3	0.6294465	$-5.6\cdot10^{-8}$	0.6294517	$f>0$	0.6294465	$f<0$	$8.26\cdot10^{-6}$
4	0.6294468	$+8.9\cdot10^{-7}$	0.6294465	$-6\cdot10^{-8}$	0.6294468	$9\cdot10^{-7}$	$4.45\cdot10^{-7}$

Zur Darstellung dieser Tabelle vgl. Beispiel 2.33. Wegen $|x^{(2)}-x^{(1)}|/|x^{(2)}|<5\cdot10^{-7}$ für $\nu=4$ ist die Nullstelle auf 7 Ziffern genau, also $\xi\approx x^{(1)}=0.629\,446\,5$. □

2.7.5 Das Verfahren von Anderson-Björck

Das Verfahren von Anderson-Björck arbeitet ähnlich wie das Pegasus-Verfahren; lediglich im 3. Schritt des Algorithmus (bei der Festlegung des neuen Einschlussintervalls) wird die Modifikation des Funktionswertes f_1 an der Stelle $x^{(1)}$ auf andere Weise vorgenommen. Die Konvergenzordnung des Verfahrens in der Umgebung einer einfachen Nullstelle liegt zwischen 1.682 und 1.710.

Algorithmus 2.38. (*Verfahren von Anderson-Björck*)

Gegeben: (i) $f\in C[a,b]$ mit $f(a)\cdot f(b)<0$,
(ii) Schranken AbsErr und RelErr für den absoluten bzw. den relativen Fehler mit AbsErr >0 und RelErr $=0$
oder mit AbsErr $=0$ und RelErr >0.

Gesucht: Eine Zahl $\xi\in(a,b)$, für die $f(\xi)=0$ ist, oder ein Einschlussintervall $[x^{(1)},x^{(2)}]$ bzw. $[x^{(2)},x^{(1)}]$ für ξ mit
$|\,x^{(1)}-x^{(2)}\,|\le|\,x^{(2)}\,|\,\text{RelErr}\ +\ \text{AbsErr}$.

Vorbereitung: $x^{(1)}:=a,\qquad x^{(2)}:=b$
$f_1:=f(x^{(1)}),\ f_2:=f(x^{(2)})$.

Pro Iterationsschritt ist wie folgt vorzugehen:

1. *Bestimmung der neuen Intervallgrenze*:

   Berechne
   $$\text{tol}\ :=|\,x^{(2)}\,|\,\text{RelErr}\ +\ \text{AbsErr}$$
   und
   $$\Delta x:=\frac{f_2}{f_2-f_1}\,(x^{(1)}-x^{(2)})\,.$$
   Wenn $|\,\Delta x\,|\le\text{tol}$ ist, setze
   $$\Delta x:=0.9\cdot\text{tol}\ \cdot\ \operatorname{sgn}(x^{(1)}-x^{(2)})\,.$$
   Berechne $x^{(3)}:=x^{(2)}+\Delta x$.

2. *Berechnung des neuen Funktionswertes* $f_3 := f(x^{(3)})$.
   Falls $f_3 = 0$ ist, wird die Iteration mit $\xi := x^{(3)}$ abgebrochen, andernfalls geht es mit 3. weiter.
3. *Festlegung des neuen Einschlussintervalls:*
   Falls $f_2 \cdot f_3 < 0$ gilt, wenn also die Nullstelle ξ zwischen $x^{(2)}$ und $x^{(3)}$ liegt, wird gesetzt
   $$x^{(1)} := x^{(2)}, \qquad x^{(2)} := x^{(3)}, \qquad f_1 := f_2, \qquad f_2 := f_3.$$
   Falls $f_2 \cdot f_3 > 0$ gilt, wenn also die Nullstelle ξ zwischen $x^{(1)}$ und $x^{(3)}$ liegt, wird die Stelle $x^{(1)}$ beibehalten und ihr ein abgeänderter Funktionswert
   $$f_1 := g f_1 \quad \text{mit} \quad g = 1 - f_3/f_2\,, \;\; g < 1$$
   oder, falls $g \leq 0$ ist, mit $g = 0.5$ zugeordnet (modifizierter Schritt). Dann wird gesetzt
   $$x^{(2)} := x^{(3)}, \qquad f_2 := f_3.$$
   Jetzt liegt ξ zwischen $x^{(1)}$ und $x^{(2)}$, und $x^{(2)}$ ist die zuletzt berechnete Näherung.
4. *Prüfung der Abbruchbedingung*:
   Falls
   $$|\, x^{(1)} - x^{(2)} \,| \leq \text{ tol}$$
   ist, erfolgt Abbruch. Dann wird gesetzt
   $$\xi := x^{(2)}, \qquad \text{falls } |f_2| \leq |f_1| \text{ ist}\,, \qquad \text{und sonst} \qquad \xi := x^{(1)}.$$
   Andernfalls, also mit $|\, x^{(2)} - x^{(1)} \,| > \text{tol}$, wird die Iteration mit 1. fortgesetzt.

Die Originalarbeit zu dem Verfahren von Anderson-Björck ist [ANDE1973]. Auf dieser Basis wurden hier zusätzlich der Algorithmus formuliert und die geometrische Interpretation entwickelt.

Geometrische Interpretation für den modifizierten Schritt

Die Ersetzung von f_1 durch gf_1 im 3. Schritt des Algorithmus lässt sich wie folgt geometrisch interpretieren (Abbildung 2.19): Durch die drei Punkte

$$P_1 = (x^{(1)}, f_1), \quad P_2 = (x^{(2)}, f_2), \quad P_3 = (x^{(3)}, f_3)$$

wird die interpolierende quadratische Parabel gelegt, und im (mittleren) Punkt P_3 wird die Parabeltangente konstruiert. Falls diese Tangente die x-Achse zwischen $x^{(3)}$ und $x^{(1)}$ schneidet, wird dieser Schnittpunkt $x^{(4)}$ als die nächste Näherung für die gesuchte Nullstelle genommen.
Die Parabeltangente kann wie folgt konstruiert werden. H_1 sei der Schnittpunkt der Gerade P_2P_3 mit der Gerade $x = x^{(1)}$ und H_2 der Schnittpunkt der Gerade P_1P_3 mit der Gerade $x = x^{(2)}$. Die Parabeltangente durch P_3 ist parallel zur Verbindungsgerade H_1H_2.

Im Algorithmus wird die Tangente durch den Punkt P_3 und ihren Schnittpunkt $(x^{(1)}, f_1^*)$ mit der Gerade $x = x^{(1)}$ festgelegt.

Nach Anderson-Björck ergibt sich

$$f_1^* = g f_1 \qquad \text{mit} \qquad g = \frac{s_{23}}{s_{12}}, \qquad s_{23} = \frac{f_2 - f_3}{x^{(2)} - x^{(3)}}.$$

Wenn $g > 0$ ist, liegen die Punkte $P_1 = (x^{(1)}, f_1)$ und $(x^{(1)}, f_1^*)$ auf derselben Seite der x-Achse, und der Schnittpunkt $x^{(4)}$ der Parabeltangente mit der x-Achse liegt zwischen $x^{(3)}$ und $x^{(1)}$.

Wegen des vorhergehenden Sekantenschritts sind die Punkte $(x^{(1)}, f_1)$, $(x^{(2)}, f_2)$ und $(x^{(3)}, 0)$ kollinear, und daher gilt

$$s_{12} = \frac{f_1 - f_2}{x^{(1)} - x^{(2)}} = \frac{f_2}{x^{(2)} - x^{(3)}}.$$

Damit ergibt sich für g wesentlich einfacher

$$g = \frac{s_{23}}{s_{12}} = \frac{f_2 - f_3}{x^{(2)} - x^{(3)}} \frac{x^{(1)} - x^{(2)}}{f_1 - f_2} = \frac{f_2 - f_3}{x^{(2)} - x^{(3)}} \frac{x^{(2)} - x^{(3)}}{f_2} = \frac{f_2 - f_3}{f_2} = 1 - \frac{f_3}{f_2}.$$

Wegen $f_3 \cdot f_2 > 0$ ist $g < 1$. Falls $g \leq 0$ ist, schneidet die Tangente die x-Achse nicht zwischen $x^{(3)}$ und $x^{(1)}$. In diesem Fall wird (wie beim Illinois-Verfahren) g = 0.5 gesetzt, also $f_1^* = 0.5 f_1$.

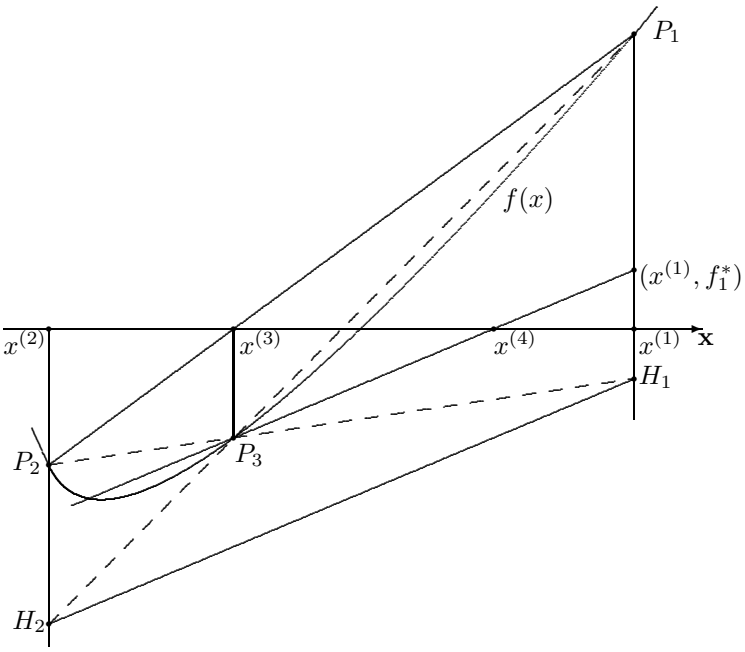


Abb. 2.19. Zum Verfahren von Anderson-Björck. Mit den Geraden P_2P_3 und P_1P_3 ergeben sich die Punkte H_1 und H_2. Die zu H_1H_2 parallele Gerade durch P_3 ist die Parabeltangente in diesem Punkt

Beispiel 2.39.

Gegeben: Die Funktion $f(x) = \sin x + 1 - 1/x$ und die Schranke RelErr $= 5 \cdot 10^{-7}$ für den relativen Fehler (AbsErr $= 0$).

Gesucht: Die Nullstelle ξ von f im Intervall $(0.6, 0.7)$ mit Hilfe des Verfahrens von Anderson–Björck.

Lösung:

ν	$x^{(3)}$	$f(x^{(3)})$	$x^{(1)}$	$f(x^{(1)})$	$x^{(2)}$	$f(x^{(2)})$	$\lvert x^{(2)} - x^{(1)}\rvert / \lvert x^{(2)}\rvert$
			0.6	$f < 0$	0.7	$f > 0$	
1	0.63211636	$+8.9{\cdot}10^{-3}$	0.6	$f < 0$	0.63211636	$f > 0$	$5.08{\cdot}10^{-2}$
2	0.62944753	$+3.5{\cdot}10^{-6}$	0.6	$f < 0$	0.62944753	$f > 0$	$4.68{\cdot}10^{-2}$
3	0.62944648	$-1.2{\cdot}10^{-8}$	0.62944753	$f > 0$	0.62944648	$f < 0$	$1.67{\cdot}10^{-6}$
4	0.62944676	$+9.3{\cdot}10^{-7}$	0.62944648	$-1.2{\cdot}10^{-8}$	0.62944676	$9.3{\cdot}10^{-7}$	$4.45{\cdot}10^{-7}$

Zur Darstellung dieser Tabelle vgl. Beispiel 2.33. Wegen $|x^{(2)} - x^{(1)}|/|x^{(2)}| < 5 \cdot 10^{-7}$ für $\nu = 4$ ist die Nullstelle auf 7 Ziffern genau, also $\xi \approx x^{(1)} = 0.629\,446\,5$. □

2.7.6 Die Verfahren von King und Anderson-Björck-King – Das Illinois-Verfahren

Das Verfahren von King (s. Originalarbeit [KING1973]) unterscheidet sich vom Pegasus-Verfahren nur dadurch, dass nie zwei Sekantenschritte nacheinander ausgeführt werden, sondern auf jeden Sekantenschritt ein modifizierter Schritt folgt. Es besitzt eine etwas höhere Konvergenzordnung (siehe Abschnitt 2.9).

Das Verfahren von Anderson-Björck-King verläuft ganz analog; es arbeitet nach dem Verfahren von Anderson-Björck mit der Zusatzbedingung von King, dass nie zwei Sekantenschritte nacheinander erfolgen dürfen. Hier geschieht der modifizierte Schritt nach der Anderson-Björck-Methode, beim Verfahren von King nach der Pegasus-Methode.

Gemeinsam ist den Einschlussverfahren Pegasus, Anderson-Björck und damit auch King der Sekantenschritt. Lediglich der modifizierte Schritt wird unterschiedlich realisiert; bei Pegasus wird $g = f_2/(f_2 + f_3)$ gesetzt, bei Anderson-Björck $g = 1 - f_3/f_2$ und, falls $g \leq 0$ ist, wird $g = 0.5$ gesetzt. Das Illinois-Verfahren verwendet stets $g = 0.5$.

2.7.7 Ein kombiniertes Einschlussverfahren

In dem folgenden Algorithmus 2.40 wird das gegebene Einschlussintervall $[a, b]$ zunächst mit dem Bisektionsverfahren so lange verkleinert, bis seine Länge nicht größer ist als eine vorgegebene Länge LB. Erst dann kommt ein Einschlussverfahren höherer Konvergenzordnung, das Pegasus-Verfahren, das Verfahren von Anderson-Björck oder auch das Illinois-Verfahren, zum Einsatz.

Die Formulierung der Algorithmen 2.36 und 2.38 ist günstig für die Herleitung und die geometrische Interpretation der Verfahren. Der folgende Algorithmus führt dagegen einen eventuell erforderlichen modifizierten Schritt erst nach einer nicht erfüllten Abbruchbedingung aus; daher weichen die Formeln für g von denen in Abschnitt 2.7.4 und 2.7.5 ab.

Der Algorithmus 2.40 kann in etwas erweiterter Form auch mit den Verfahren von King und Anderson-Björck-King arbeiten. Zugunsten einer einfachen und übersichtlichen Darstellung wird auf diese Verfahren verzichtet (siehe jedoch Beispiel 2.41).

Algorithmus 2.40.

Gegeben: $f \in C[a, b]$ mit $f(a) \cdot f(b) < 0$, Schranken AbsErr und RelErr für den absoluten bzw. den relativen Fehler mit AbsErr > 0 und RelErr $= 0$ oder mit AbsErr $= 0$ und RelErr > 0, die Länge $LB > 0$ für die Entscheidung zwischen Bisektions- und Sekantenschritt, die maximal zulässige Anzahl $nfmax$ der Funktionsauswertungen.

Gesucht: Eine Zahl $\xi \in (a, b)$, für die $f(\xi) = 0$ ist, oder ein Einschlussintervall $[x^{(1)}, x^{(2)}]$ bzw. $[x^{(2)}, x^{(1)}]$ für ξ mit

$$|x^{(1)} - x^{(2)}| \;\le\; |x^{(2)}| \text{ RelErr } + \text{ AbsErr}\,.$$

Vorbereitung: $x^{(1)} := a,\;\; x^{(2)} := b,\quad f_1 := f(x^{(1)}),\;\; f_2 := f(x^{(2)}),$
$nf := 2,\quad v := x^{(1)} - x^{(2)}.$

1. Wenn $nf \ge nfmax$ ist, konnte mit $nfmax$ Funktionsauswertungen eine Nullstelle von f nicht ermittelt werden. Die Iteration wird dann abgebrochen.

2. Berechnung der aktuellen Fehlertoleranz: tol $:= |x^{(2)}|$ RelErr + AbsErr.

3. Verfahrensschritt wählen und nächste Näherung berechnen:
   Wenn $|v| > LB$ ist, wird ein Bisektionsschritt ausgeführt mit

   $$\Delta x := v \cdot 0.5\,, \quad Bis := 1\,,$$

   andernfalls ein Sekantenschritt mit

   $$\Delta x := v \cdot f_2/(f_2 - f_1)\,, \quad Bis := 0\,.$$

   Falls $|\Delta x| \le$ tol ist, wird

   $$\Delta x := 0.9 \cdot \operatorname{sgn}(v) \cdot \text{tol}$$

   gesetzt. Die nächste Näherung ist

   $$x^{(3)} := x^{(2)} + \Delta x\,.$$

4. Berechnung des Funktionswertes

$$f_3 := f(x^{(3)}), \quad nf := nf + 1.$$

Falls $f_3 = 0$ ist, wird die Iteration mit $\xi := x^{(3)}$ abgebrochen.

5. Ermittlung des neuen Einschlussintervalls:
Berechne $e := \mathrm{sgn}(f_2) \cdot f_3$. Wenn $e < 0$ ist, werden $x^{(1)}$ mit $x^{(2)}$ und f_1 mit f_2 getauscht. Dann werden $x^{(2)} := x^{(3)}$ gesetzt, f_2 mit f_3 getauscht und $v := x^{(1)} - x^{(2)}$ berechnet. Jetzt liegt ξ zwischen $x^{(1)}$ und $x^{(2)}$, und $x^{(2)}$ ist die zuletzt berechnete Näherung.

6. Prüfung der Abbruchbedingung: Wenn $|v| \leq \mathrm{tol}$ ist, wird die Iteration abgebrochen mit

$$\xi := x^{(2)}, \quad \text{falls } |f_2| \leq |f_1| \text{ ist}, \text{ andernfalls mit } \quad \xi := x^{(1)}.$$

Wenn $|v| > \mathrm{tol}$ ist, weiter mit 7.

7. Vorbereitung des nächsten Iterationsschrittes. Im Fall $e > 0$ wird der Funktionswert f_1 mittels $f_1 := g \cdot f_1$, $0 < g < 1$, wie folgt modifiziert:
   a) Pegasus-Verfahren mit $g = f_3/(f_3 + f_2)$
   b) Verfahren von Anderson-Björck:
      - nach einem Bisektionsschritt ($Bis = 1$) mit $g = f_3/(f_3 + f_2)$
      - nach einem Sekantenschritt ($Bis = 0$) mit $g = 1 - f_2/f_3$
        oder, falls $g \leq 0$ ist, mit $g = 0.5$
   c) Illinois-Verfahren mit $g = 0.5$

Mit dem ermittelten g wird $f_1 := g \cdot f_1$ gesetzt.
Die Iteration wird mit 1. fortgesetzt.

Bemerkungen.

- Für das kombinierte Verfahren eignet sich eine Länge LB mit $0 < LB \leq 0.2$, beispielsweise $LB = 0.15$. Mit $LB > |b - a|$ werden nur Sekantenschritte, mit $LB = 0$ nur Bisektionsschritte ausgeführt. Siehe dazu auch Beispiel 2.41.

- Zu 1. Die Vorgabe von $nfmax$ (etwa $nfmax = 100$) schützt vor einer Endlosschleife, wenn AbsErr oder RelErr zu klein oder wenn LB zu groß gewählt worden ist.

- Zu 5. Wenn $e < 0$ ist, liegt ξ zwischen $x^{(2)}$ und $x^{(3)}$, andernfalls zwischen $x^{(1)}$ und $x^{(3)}$.

- Zu 7. Wegen der Vertauschung von f_2 mit f_3 in 5. sehen die Formeln für g anders aus als in den Abschnitten 2.7.4 und 2.7.5. Ein Anderson-Björck-Schritt mit $g = 1 - f_2/f_3$ setzt voraus, dass vorher ein Sekantenschritt ($Bis = 0$) ausgeführt wurde (siehe Abschnitt 2.7.5). Deshalb wird er nach einem Bisektionsschritt ($Bis = 1$) durch einen Pegasusschritt ersetzt.

2.7.8 Das Zeroin-Verfahren

Der als Zeroin-Verfahren bezeichnete programmierte Algorithmus entspricht dem bei [DEKK1969], [FORS1977], [BREN1971] angegebenen Verfahren. Es ist eine geschickte Kombination des Bisektionsverfahrens ($p = 1$), des Sekantenverfahrens ($p = 1.618$) sowie der inversen quadratischen Interpolation ($p = 2$). Es ist ein Einschlussverfahren, bei dem ständig mit drei iterierten Näherungswerten für die gesuchte Nullstelle operiert wird. Aufgrund geometrischer Überlegungen wird jeweils das für die momentane Situation sinnvollste Verfahren eingesetzt. Eine ausführliche Beschreibung findet man bei [QUAR2002], Abschnitt 6.2.3.

Beispiel 2.41.

Für einen Vergleich verschiedener Einschlussverfahren werden üblicherweise die Nullstellen von gewissen Testfunktionen berechnet und die dafür benötigten Funktionsauswertungen gezählt.

Für 12 solcher Funktionen (als eine Auswahl aus einem umfangreicheren Test) sind die Ergebnisse in der nachfolgenden Tabelle zusammengestellt. Verglichen werden das Bisektionsverfahren, das Zeroin-Verfahren, das Illinois- und das Pegasus-Verfahren sowie die Verfahren von Anderson-Björck, King und Anderson-Björck-King.

Für die letzten fünf Verfahren wird der Algorithmus 2.40 (mit einer Erweiterung für die letzten beiden Verfahren) angewendet. Sie werden jeweils einmal ohne eine vorangehende Verkleinerung des Startintervalls $[a, b]$ ($LB > |b - a|$) und einmal mit einer solchen Verkleinerung mittels Bisektion ($LB < |b-a|$) eingesetzt. Mit $LB = 0$ führt der Algorithmus das Bisektionsverfahren aus.

Damit beim Test alle Verfahren mit einem Intervall der Länge LB beginnen, haben die Startintervalle die Längen $4 \cdot LB$, $8 \cdot LB$ oder $16 \cdot LB$, so dass nach 2, 3 bzw. 4 Bisektionsschritten die Länge LB erreicht ist. Für die Tabelle ist $LB = 0.15$ gewählt.

Die Abbruchbedingungen bei ν Funktionsauswertungen sind $f(x^{(\nu)}) = 0$ oder $|x^{(\nu)} - x^{(\nu-1)}| \leq |x^{(\nu)}| \cdot \text{RelErr}$ mit $\text{RelErr} = 2{\cdot}10^{-11}$. Die angegebenen Nullstellen haben 10 gültige Ziffern.

Funktionen und Startintervalle:

1. $f(x) = x^2 \left(\frac{x^2}{3} + \sqrt{2}\,\sin x\right) - \frac{\sqrt{3}}{18}$, $[0, 1.2]$, $\xi = 0.399\,422\,2917$
2. $f(x) = 11x^{11} - 1$, $[0.4, 1.6]$, $\xi = 0.804\,133\,0975$
3. $f(x) = 35x^{35} - 1$, $[-0.5, 1.9]$, $\xi = 0.903\,407\,6632$
4. $f(x) = 2\,(xe^{-9} - \mathrm{e}^{-9x}) + 1$, $[-0.5, 0.7]$, $\xi = 0.077\,014\,241\,35$
5. $f(x) = x^2 - (1 - x)^9$, $[-1.4, 1]$, $\xi = 0.259\,204\,4937$
6. $f(x) = (x - 1)\,\mathrm{e}^{-9x} + x^9$, $[-0.8, 1.6]$, $\xi = 0.536\,741\,6626$

7. $f(x) = x^2 + \sin\left(\frac{x}{9}\right) - \frac{1}{4}$, $[-0.5, 1.9]$, $\xi = 0.447\,541\,7621$
8. $f(x) = \frac{1}{8}\left(9 - \frac{1}{x}\right)$, $[0.001, 1.201]$, $\xi = \frac{1}{9} = 0.111\,111\,1111$
9. $f(x) = \tan x - x - 0.046\,3025$, $[-0.9, 1.5]$, $\xi = 0.500\,000\,0340$
10. $f(x) = x^2 + x\sin(x\sqrt{75}) - 0.2$, $[0.4, 1]$, $\xi = 0.679\,808\,9215$
11. $f(x) = x^9 + 0.0001$, $[-1.2, 0]$, $\xi = -0.359\,381\,3664$
12. $f(x) = \ln x + \frac{x^2}{2\cdot \mathrm{e}} - 2\,\frac{x}{\sqrt{\mathrm{e}}} + 1$, $[1, 3.4]$, Die Verfahren liefern bei Abbruch mit $f(x^{(\nu)}) = 0$ unterschiedliche Werte für die Nullstelle ξ, die alle im Intervall $[1.6483, 1.6492]$ liegen. Die exakte Nullstelle ist $\xi = \sqrt{\mathrm{e}} = 1.648\,721\,270\,70\ldots$.

Die Tabelle zeigt, dass bei der Mehrzahl der Funktionen eine vorbereitende Intervallverkleinerung zweckmäßig oder sogar unerlässlich ist. — bedeutet, dass die Nullstelle mit 100 Funktionsauswertungen nicht gefunden wurde.

Anzahl der Funktionsauswertungen

Verfahren/Funktion	1	2	3	4	5	6	7	8	9	10	11	12
Bisektion	40	39	40	41	41	40	41	42	40	38	40	12
Illinois ohne Bisektion	14	21	63	18	22	22	14	18	19	13	28	25
Illinois mit Bisektion	12	13	19	14	14	14	13	15	13	12	15	21
Pegasus ohne Bisektion	12	19	63	17	20	28	11	18	18	12	26	36
Pegasus mit Bisektion	11	12	16	12	12	11	11	16	11	10	14	29
King ohne Bisektion	11	17	60	16	19	28	12	18	17	10	25	38
King mit Bisektion	11	12	15	12	12	11	11	15	11	10	13	29
Anderson-Björck ohne Bisektion	14	74	—	29	11	22	14	7	19	12	—	30
Anderson-Björck mit Bisektion	10	11	16	11	11	11	12	11	12	10	14	24
Anderson-Björck-King ohne Bisektion	16	67	—	26	11	22	13	7	18	11	—	29
Anderson-Björck-King mit Bisektion	10	11	15	11	11	11	11	10	11	10	13	24
Zeroin	12	14	17	10	11	11	13	13	15	12	14	28

□

2.8 Anwendungsbeispiele

Beispiel 2.42. (*Rohrleitungsbeispiel*)
Es geht um die Ermittlung des Rohrdurchmessers d zur Optimierung der in Abb. 2.20 skizzierten Druckrohrleitung einer Freistrahlturbine. Die Düse am Rohrleitungsende ist in Abb. 2.20 rechts noch einmal getrennt skizziert.

Eingabeparameter:

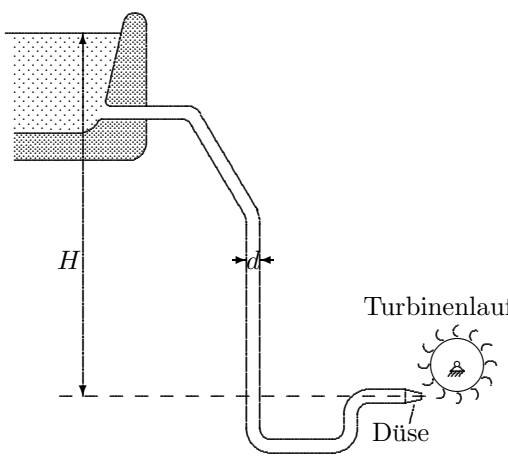
α_0	:	Geschätztes Flächenverhältnis Düse/Rohr, $\alpha \in [0.1,0.25]$
$\dot{V}$	:	Volumenstrom in $\frac{\mathrm{m}^3}{\mathrm{s}}$
H	:	Gesamtgefälle in m
L	:	Gesamtlänge der Rohrleitung in m
ζ_D	:	Widerstandsbeiwert der Düse bezogen auf die Strahlgeschwindigkeit
ζ_R	:	Summe aller Formwiderstände der Rohrleitung bezogen auf die Strömungsgeschwindigkeit im Rohr
g	:	Gravitationskonstante in $\frac{\mathrm{m}}{\mathrm{s}^2}$
ϱ	:	Dichte des Wassers in $\frac{\mathrm{kg}}{\mathrm{m}^3}$
λ	:	Rohrreibungsbeiwert

Zu den Eingabeparametern gehören noch Größen aus den Abbruchkriterien für die Iteration, z. B.

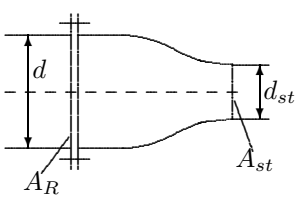
ε	:	$\lvert d^{(\nu)} - d^{(\nu-1)}\rvert \leq \varepsilon$, $\varepsilon > 0$, $\nu =$ Iterationsindex
$N_{\max}$	:	Maximale Anzahl der Iterationen

Ausgabeparameter:

d	:	Rohrdurchmesser in m.



Düse am Rohrleitungsende:



$$\alpha = \frac{A_{st}}{A_R} = \left(\frac{d_{st}}{d}\right)^2$$

Abb. 2.20.

Vorgehensweise:

(a) Zunächst berechnet man das optimale Flächenverhältnis der Düse α_{opt} aus der Forderung, dass die Strahlleistung P_{st} maximal wird.

(b) Für dieses optimierte Flächenverhältnis α_{opt} ist dann derjenige Rohrdurchmesser d zu bestimmen, der für den vorgegebenen Volumenstrom erforderlich ist.

Die *Energiegleichung* (erweiterte *Bernoulli-Gleichung*) lautet

$$\mathrm{g}H = \frac{v_{st}^2}{2} + (\lambda\frac{L}{d} + \zeta_R)\frac{v_R^2}{2} + \zeta_D\frac{v_{st}^2}{2}\,. \tag{2.56}$$

Aus der *Kontinuitätsgleichung*

$$\dot{V} = A_{st}\,v_{st} = A_R\,v_R \tag{2.57}$$

folgt mit $\alpha = A_{st}/A_R$ für die Strömungsgeschwindigkeit im Rohr

$$v_R = \alpha\,v_{st}\,.$$

Damit kann aus (2.56) die *Strahlgeschwindigkeit* v_{st} berechnet werden:

$$v_{st} = \sqrt{\frac{2\mathrm{g}H}{1+\zeta_D+\alpha^2(\lambda\frac{L}{d}+\zeta_R)}}\,.$$

Die *Strahlleistung* ist

$$\begin{aligned} P_{st} &= \tfrac{1}{2}\varrho\dot{V}v_{st}^2 = \tfrac{1}{2}\varrho A_R\,v_R\,v_{st}^2 = \tfrac{1}{2}\varrho A_R\,\alpha\,v_{st}^3 \\ &= \tfrac{1}{2}\varrho A_R\alpha\left[\tfrac{2\mathrm{g}H}{1+\zeta_D+\alpha^2(\lambda\frac{L}{d}+\zeta_R)}\right]^{3/2}. \end{aligned}$$

Zu (a): Mit der notwendigen Bedingung $\frac{\partial P_{st}}{\partial\alpha} = 0$ für eine maximale Strahlleistung ergibt sich das optimale Flächenverhältnis

$$\alpha_{opt} = \sqrt{\frac{1+\zeta_D}{2(\lambda\frac{L}{d}+\zeta_R)}}\,.$$

(Wegen $\frac{\partial^2 P_{st}}{\partial\alpha^2}\,(\alpha_{opt}) < 0$ liegt tatsächlich ein Maximum vor.)

Setzt man α_{opt} für α in die Formel für die Strahlgeschwindigkeit v_{st} ein, so ergibt sich die optimale Strahlgeschwindigkeit

$$v_{st}(\alpha_{opt}) = \sqrt{\frac{4\mathrm{g}H}{3(1+\zeta_D)}}\,.$$

Damit folgt für die optimale Strömungsgeschwindigkeit im Rohr

$$v_R(\alpha_{opt}) = \alpha_{opt} \cdot v_{st}(\alpha_{opt}) = \sqrt{\frac{2gH}{3(\lambda\frac{L}{d} + \zeta_R)}}\,.$$

Zu (b): Mit der Querschnittsfläche des Rohres

$$A_R = \pi\,\frac{d^2}{4}$$

ist nach (2.57) der optimale Volumenstrom

$$\dot{V} = \pi\,\frac{d^2}{4}\,v_R(\alpha_{opt}) = \pi\,\frac{d^2}{4}\sqrt{\frac{2gH}{3(\lambda\frac{L}{d} + \zeta_R)}}\,,$$

und daraus folgt für den optimalen Rohrdurchmesser d die nichtlineare Gleichung

$$d = \sqrt{\frac{4\dot{V}}{\pi}\sqrt{\frac{3}{2gH}}\sqrt{\lambda\frac{L}{d} + \zeta_R}} = \varphi(d)\,. \tag{2.58}$$

Zweimaliges Quadrieren liefert

$$f(d) = d^4 - \frac{24\dot{V}^2}{\pi^2 gH}\,(\lambda\frac{L}{d} + \zeta_R) = 0\,. \tag{2.59}$$

Zahlenbeispiel.
$\dot{V} = 6.2\,\mathrm{m}^3/\mathrm{s}$, $H = 1130\,\mathrm{m}$, $L = 1300\,\mathrm{m}$, $\zeta_D = 0.04$, $\zeta_R = 2.5$, $g = 9.81\ \mathrm{m/s^2}$, $\varrho = 10^3\,\mathrm{kg/m^3}$, $\lambda = 0.02$.
Schranke für den absoluten Fehler $\varepsilon = 0.5 \cdot 10^{-6}$.
Mit den Startwerten $d^{(1)} = 0.7$ und $d^{(2)} = 0.8$ sind $f(0.7) < 0$ und $f(0.8) > 0$; also ist $0.7 < d < 0.8$. Für die Lösung der Gleichung (2.59)

$$d^4 - 0.008\,432\,327\left(\frac{26}{d} + 2.5\right) = 0$$

liefert das Pegasus-Verfahren mit 5 Iterationsschritten

$$d = 0.748\,551.$$

Somit ergibt sich für den gesuchten optimalen Rohrdurchmesser bei Rundung auf 3-stellige Mantisse $d \approx 0.749$ m.
Mit dem Quadrat der optimalen Strahlgeschwindigkeit

$$v_{st}^2(\alpha_{opt}) = \frac{4gH}{3(1+\zeta_D)} = \frac{4 \cdot 9.81 \cdot 1130}{3 \cdot 1.04}\,\frac{\mathrm{m}^2}{\mathrm{s}^2} = 14211.9\,\mathrm{m}^2/\mathrm{s}^2$$

erhält man die maximale Strahlleistung

$$P_{st} = \frac{1}{2}\varrho\dot{V}v_{st}^2(\alpha_{opt}) = 44\,057\,\mathrm{kW}.$$

Das allgemeine Iterationsverfahren, angewendet auf die Gleichung (2.58)

$$d = \sqrt{0.091\,827\,70\sqrt{\frac{26}{d} + 2.5}} = \varphi(d)$$

benötigt mit dem Startwert $d^{(0)} = 0.7$ und mit derselben Fehlerschranke 10 Iterationsschritte. Es konvergiert wesentlich schlechter als das Pegasus-Verfahren. □

Beispiel 2.43. (*Turbinenbeispiel*)
Eine Turbine arbeitet zwischen zwei Wasserspeichern, deren Spiegel die Höhendifferenz h besitzen. Stromabwärts von der Turbine wird das Wasser über einen Diffusor, dessen Endquerschnitt die Fläche A hat, in das untere Becken eingeleitet. An der Turbinenwelle wird die Leistung P angenommen. Frage: Welcher Volumenstrom $\dot{V}$ fließt durch die Turbine, wenn von den Verlusten nur der Austrittsverlust am Diffusor berücksichtigt wird?

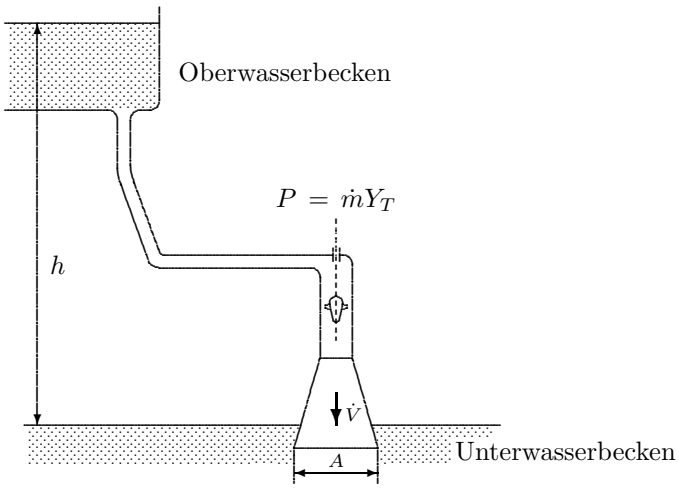


Abb. 2.21.

Ausgehend von der erweiterten Bernoulli-Gleichung erhält man

$$\mathrm{g} \cdot h - Y_T = \frac{1}{2}\left(\frac{\dot{V}}{A}\right)^2 ;$$

dabei sind Y_T die Stutzenarbeit der Turbine und $0.5(\dot{V}/A)^2$ der Austrittsverlust am Diffusor. Durch Auflösung nach Y_T und wegen $Y_T = P/(\varrho\dot{V})$ ergibt sich die Beziehung

$$\mathrm{g} \cdot h - \frac{1}{2}\left(\frac{\dot{V}}{A}\right)^2 = \frac{P}{\varrho\dot{V}} . \tag{2.60}$$

Durch einfache Umstellung folgt aus (2.60) die kubische Gleichung für $\dot{V}$

$$f(\dot{V}) := \dot{V}^3 - 2\mathrm{g}hA^2\dot{V} + \frac{2PA^2}{\varrho} = 0 .$$

Die praktisch interessierende Lösung dieser Gleichung $f(\dot{V}) = 0$ kann z. B. mit dem Newton-Verfahren ermittelt werden. Ein Startwert $\dot{V}^{(0)}$ für die Iteration ergibt sich aus (2.60) bei Vernachlässigung des Austrittsverlustes und anschließender Auflösung nach $\dot{V}$ zu

$$\dot{V}^{(0)} = \frac{P}{\varrho \cdot h \cdot \mathrm{g}}$$

mit $\varrho = 10^3$ kg/m^3, g = 9.81 m/s^2. Man kann aber auch mit jedem Einschlussverfahren arbeiten.

Zur Lösung der kubischen Gleichung $f(\dot{V}) = 0$ könnte aber ebenso ein Verfahren zur Bestimmung sämtlicher Lösungen einer algebraischen Gleichung ohne Kenntnis von Startwerten (z. B. das Verfahren von Muller) verwendet werden (s. Abschnitt 3.3); hier ist dann aus den drei Lösungen die für den Anwendungsfall sinnvolle Lösung auszuwählen (siehe dazu das Beispiel 11). □

2.9 Effizienz der Verfahren und Entscheidungshilfen

Mit Hilfe des *Effizienzindex E* von Traub (s. [TRAU1984], App. C) lassen sich die Iterationsverfahren gut vergleichen. Seien H die *Hornerzahl* (Anzahl der erforderlichen Funktionsauswertungen pro Iterationsschritt) und p die Konvergenzordnung eines Iterationsverfahrens, so ist

$$E := p^{1/H}$$

der Effizienzindex.

Die folgende Tabelle gibt eine Übersicht über die Konvergenzordnung und den Effizienzindex bei Verfahren zur Berechnung einfacher und mehrfacher Nullstellen; *je größer E, desto effizienter ist das Verfahren in der Umgebung der Nullstelle.*

Verfahren	Konvergenz-ordnung p	Horner-zahl H	Effizienz E	Einschluss-verfahren
Newton, einfache Nullstelle	2	2	1.414	nein
Newton, mehrfache Nullstelle	2	2	1.414	nein
Newton, modifiziert	2	3	1.260	nein
Bisektion	1	1	1	ja
Regula falsi	1	1	1	ja

Verfahren	Konvergenz-ordnung p	Horner-zahl H	Effizienz E	Einschluss-verfahren
Sekantenverfahren, einfache Nullstelle	1.618	1	1.618	nein
Illinois	1.442	1	1.442	ja
Pegasus	1.642	1	1.642	ja
Anderson-Björck	1.682...1.710	1	1.682...1.710	ja
King	1.710...1.732	1	1.710...1.732	ja
Anderson-Björck-King	1.710...1.732	1	1.710...1.732	ja

Es zeigt sich, dass man (nicht zuletzt wegen der sicheren Konvergenz) am effektivsten mit dem Pegasus-Verfahren, dem Verfahren von Anderson-Björck oder den Verfahren von King und Anderson-Björck-King arbeitet. Das Bisektionsverfahren wird benutzt, um ein Startintervall mit Einschluss zu verkleinern, bevor eines der genannten Verfahren eingesetzt wird.

Will man einzelne *Polynom*-Nullstellen berechnen, so lässt sich auch das Newton-Verfahren zusammen mit dem Hornerschema zur Berechnung der Funktions- und Ableitungswerte effektiv einsetzen. Für transzendente Gleichungen sind weder das Sekantenverfahren noch das Newton-Verfahren zu empfehlen.

Die Ermittlung der Lösung einer Gleichung $f(x) = 0$ mit dem Algorithmus 2.17 für das allgemeine Iterationsverfahren erfordert umfangreiche Vorbereitungen. Der größte Aufwand steckt im 3. Schritt, in dem zu prüfen ist, ob bei der zu $f(x) = 0$ äquivalenten Gleichung $\varphi(x) = x$ die Funktion φ den Voraussetzungen des Existenz- und Eindeutigkeitssatzes 2.11 genügt. Dagegen kann schon nach dem 1. Schritt – Festlegung eines Intervalls I, in dem mindestens eine Nullstelle von f liegt – ein Einschlussverfahren zur Bestimmung dieser Nullstelle eingesetzt werden. Darum hat das allgemeine Iterationsverfahren für die praktische Ermittlung von Nullstellen keine Bedeutung.

Für den Fall mehrfacher Nullstellen sind die „Anmerkungen zu mehrfachen Nullstellen“ in Abschnitt 2.5.3 zu beachten.

Ergänzende Literatur zu Kapitel 2

[HAMM1994], 8.; [HERM2001], Kap.4; [HILD1987], 10.; [MAES1988], 5.; [PREU2001], Kap.2; [RALS1979], V.; [SCHW1997], 5.; [STOE1989], 5.; [STUM1982], 2; [WERN1993], II §§1-5.

Kapitel 3

Verfahren zur Lösung algebraischer Gleichungen

3.1 Vorbemerkungen

Im Folgenden werden Polynome P_n mit

$$P_n(x) = \sum_{j=0}^{n} a_j x^j, \qquad a_j \in \mathbb{C}, \qquad n \in \mathbb{N}, \qquad a_n \neq 0 \tag{3.1}$$

vom Grad n betrachtet. Gesucht sind Nullstellen eines Polynoms (3.1) und somit Lösungen einer algebraischen Gleichung n-ten Grades

$$P_n(x) = \sum_{j=0}^{n} a_j x^j = 0. \tag{3.2}$$

Der Fundamentalsatz der Algebra besagt, dass eine algebraische Gleichung (3.2) genau n komplexe Lösungen x_k besitzt, die entsprechend ihrer Vielfachheit α_k gezählt werden. Jedes algebraische Polynom (3.1) lässt sich daher in n Linearfaktoren zerlegen:

$$P_n(x) = a_n(x - x_1)(x - x_2)\ldots(x - x_n).$$

Kommt der Linearfaktor $(x - x_k)$ genau α_k-fach vor, so heißt x_k eine α_k-fache Lösung von (3.2), und es ergibt sich die Zerlegung

$$P_n(x) = a_n(x - x_1)^{\alpha_1}(x - x_2)^{\alpha_2}\ldots(x - x_m)^{\alpha_m} \quad \text{mit} \quad \alpha_1 + \alpha_2 + \ldots + \alpha_m = n\,.$$

Wenn alle Koeffizienten des Polynoms (3.1) reell sind, $a_j \in \mathbb{R}$, können komplexe Lösungen von (3.2) nur als Paare konjugiert komplexer Lösungen auftreten, d. h. mit $x = \alpha + \mathrm{i}\beta$ ist auch $\overline{x} = \alpha - \mathrm{i}\beta$ eine Lösung von (3.2), und zwar mit derselben Vielfachheit. Der Grad n einer Gleichung (3.2) mit reellen Koeffizienten, die keine reellen Wurzeln besitzt, muss somit gerade sein, und jede derartige Gleichung ungeraden Grades besitzt mindestens eine reelle Lösung.

Wenn eine Nullstelle x_0 des Polynoms P_n bekannt ist, dann ist

$$P_n(x) = (x - x_0)\, P_{n-1}(x),$$

und das Polynom P_{n-1} vom Grad $n-1$ besitzt die $n-1$ restlichen Nullstellen von P_n. P_{n-1} ergibt sich mittels Division von P_n durch $x - x_0$

$$P_{n-1}(x) = \frac{P_n(x)}{x - x_0}$$

und heißt daher *abdividiertes Polynom* oder *Deflationspolynom*. Wenn x_0 keine Nullstelle von P_n ist, ist P_n nicht ohne Rest durch $x - x_0$ teilbar, und aus

$$P_n(x) = (x - x_0)\, P_{n-1}(x) + R_n$$

folgt wegen $P_n(x_0) = R_n$

$$P_n(x) = (x - x_0)\, P_{n-1}(x) + P_n(x_0).$$

Außerdem gelten

$$P_{n-1}(x) = \frac{P_n(x) - P_n(x_0)}{x - x_0}, \quad x \neq x_0,$$

und

$$P_{n-1}(x_0) = \lim_{x \to x_0} \frac{P_n(x) - P_n(x_0)}{x - x_0} = {P_n}'(x_0).$$

Zur Berechnung von Funktions- und Ableitungswerten eines Polynoms und zur Berechnung von abdividierten Polynomen bzw. Deflationspolynomen dient das im Folgenden erläuterte Horner-Schema. Es wird bei allen Verfahren zur Berechnung von Polynomnullstellen als Hilfsmittel eingesetzt, beispielsweise wenn Polynomwerte mit den in Kapitel 2 behandelten Newton-Verfahren ermittelt werden.

Will man **sämtliche** Lösungen einer algebraischen Gleichung (3.2) berechnen, so verwendet man z. B. das Muller-Verfahren, das Verfahren von Bauhuber oder das Verfahren von Jenkins und Traub (siehe Abschnitt 3.3).

3.2 Das Horner-Schema

Mit dem Horner-Schema werden Funktions- und Ableitungswerte eines Polynoms P_n (3.1) an einer festen Stelle x_0 berechnet; es arbeitet übersichtlich, rundungsfehlergünstig und erfordert einen geringeren Rechenaufwand als die Auswertung des Polynoms durch die Berechnung aller auftretenden Potenzen von x_0.

3.2.1 Das einfache Horner-Schema für reelle Argumentwerte

Der Wert des Polynoms (3.1) an der Stelle x_0, $x_0 \in \mathbb{R}$, ist

$$P_n(x_0) = a_n x_0^n + a_{n-1} x_0^{n-1} + \ldots + a_1 x_0 + a_0; \qquad a_j \in \mathbb{R}.$$

$P_n(x_0)$, kann auch in der folgenden Form geschrieben werden:

$$P_n(x_0) = \Big(\ldots \big(\underbrace{(a_n x_0 + a_{n-1})}_{a_{n-1}^{(1)}} x_0 + a_{n-2} \big) x_0 + \ldots + a_1 \Big) x_0 + a_0 \; .$$

$a_{n-2}^{(1)}$ (über $a_{n-1}^{(1)} x_0 + a_{n-2}$), $a_1^{(1)}$ (über $a_{n-2}^{(1)} x_0 + a_1$), $a_0^{(1)}$ (über $a_1^{(1)} x_0 + a_0$)

In dieser Darstellung sind außer Additionen nur Multiplikationen mit dem festen, reellen Faktor x_0 auszuführen. Im Einzelnen sind zu berechnen:

$$\left\{ \begin{array}{lcll} a_n^{(1)} & := & a_n & \\ a_{n-1}^{(1)} & := & a_n^{(1)} x_0 + a_{n-1} & \\ a_{n-2}^{(1)} & := & a_{n-1}^{(1)} x_0 + a_{n-2} & \\ \vdots & & & \\ a_1^{(1)} & := & a_2^{(1)} x_0 + a_1 & \\ a_0^{(1)} & := & a_1^{(1)} x_0 + a_0 & = P_n(x_0). \end{array} \right. \tag{3.3}$$

Diese Rechenoperationen (3.3) werden zweckmäßig in der folgenden Anordnung, dem einfachen Horner-Schema, durchgeführt:

P_n	a_n	a_{n-1}	a_{n-2}	$\ldots$	a_1	a_0
x_0	0	$a_n^{(1)} x_0$	$a_{n-1}^{(1)} x_0$	$\ldots$	$a_2^{(1)} x_0$	$a_1^{(1)} x_0$
$\sum$	$a_n^{(1)}$	$a_{n-1}^{(1)}$	$a_{n-2}^{(1)}$	$\ldots$	$a_1^{(1)}$	$\boxed{a_0^{(1)} = P_n(x_0)}$

In der ersten Zeile stehen, geordnet nach absteigenden Potenzen von x, alle Koeffizienten (auch die mit dem Wert 0) des Polynoms P_n. Die Summe der untereinander stehenden Elemente der ersten und der zweiten Zeile wird in der dritten Zeile notiert. Dann wird diese Summe mit x_0 multipliziert und das Produkt in der zweiten Zeile rechts daneben aufgeschrieben usw.
Bei komplexen Koeffizienten sind sowohl der Realteil als auch der Imaginärteil mit x_0 zu multiplizieren:

$$a_j^{(1)} x_0 = (\alpha_j^{(1)} + \mathrm{i}\, \beta_j^{(1)})\, x_0 = \alpha_j^{(1)} x_0 + \mathrm{i}\, \beta_j^{(1)} x_0.$$

Beispiel 3.1.

Gegeben: Das Polynom $P_3 : P_3(x) = 2x^3 - 4x^2 + 3x + 15$ mit reellen Koeffizienten.

Gesucht: Der Funktionswert $P_3(2)$.

Lösung: Man erhält mit dem einfachen Horner-Schema

P_3	2	-4	3	15
$x_0 = 2$	0	4	0	6
$\sum$	2	0	3	$\boxed{21 = P_3(2)}$

Für das Polynom P_3 mit komplexen Koeffizienten

$$P_3(x) = (2 + 3\,\mathrm{i})x^3 + (-4 + 2\,\mathrm{i})x^2 + (3 - 5\,\mathrm{i})x + 15 + 4\,\mathrm{i}$$

lautet das Horner-Schema für $x_0 = 2$

P_3	$2 + 3\,\mathrm{i}$	$-4 + 2\,\mathrm{i}$	$3 - 5\,\mathrm{i}$	$15 + 4\,\mathrm{i}$
$x_0 = 2$	0	$4 + 6\,\mathrm{i}$	$0 + 16\,\mathrm{i}$	$6 + 22\,\mathrm{i}$
$\sum$	$2 + 3\,\mathrm{i}$	$0 + 8\,\mathrm{i}$	$3 + 11\,\mathrm{i}$	$\boxed{21 + 26\,\mathrm{i} = P_3(2)}$

□

Bei der Division von $P_n(x)$ durch $(x - x_0)$ mit $x_0 \in \mathbb{R}$ entsteht als Quotient ein Polynom $P_{n-1}(x)$ vom Grad $n-1$ und als Rest eine Zahl R_n, dividiert durch $(x - x_0)$:

$$\frac{P_n(x)}{x - x_0} = P_{n-1}(x) + \frac{R_n}{x - x_0}, \qquad x \neq x_0, \qquad \text{bzw.}$$
$$P_n(x) = (x - x_0)P_{n-1}(x) + R_n. \tag{3.4}$$

Für $x = x_0$ ist $P_n(x_0) = R_n$, so dass sich für (3.4)

$$P_n(x) = (x - x_0)P_{n-1}(x) + P_n(x_0) \tag{3.5}$$

ergibt. Nach (3.3) ist

$$P_n(x_0) = a_0^{(1)},$$

und es zeigt sich, dass gilt:

$$P_{n-1}(x) = a_n^{(1)}x^{n-1} + a_{n-1}^{(1)}x^{n-2} + \ldots + a_2^{(1)}x + a_1^{(1)}.$$

Setzt man dies in (3.5) ein, so ergibt sich unter Verwendung von (3.3):

$$\begin{aligned}(x-x_0)P_{n-1}(x)+P_n(x_0) &= \left(a_n^{(1)}x^{n-1} + a_{n-1}^{(1)}x^{n-2} + \ldots + a_2^{(1)}x + a_1^{(1)}\right)(x - x_0) + a_0^{(1)} \\ &= a_n^{(1)}x^n + \left(a_{n-1}^{(1)} - a_n^{(1)}x_0\right)x^{n-1} + \ldots + \left(a_k^{(1)} - a_{k+1}^{(1)}x_0\right)x^k \\ &\quad + \ldots + \left(a_1^{(1)} - a_2^{(1)}x_0\right)x + \left(a_0^{(1)} - a_1^{(1)}x_0\right) = P_n(x).\end{aligned}$$

Abdividieren von Nullstellen (Deflation)

Ist x_0 eine Nullstelle von P_n, so gilt wegen $P_n(x_0) = 0$ gemäß (3.5)

$$P_n(x) = (x - x_0)P_{n-1}(x).$$

Die Koeffizienten des sogenannten abdividierten Polynoms bzw. Deflationspolynoms P_{n-1} sind die $a_j^{(1)}$ in der dritten Zeile des einfachen Horner-Schemas für x_0.

3.2.2 Das einfache Horner-Schema für komplexe Argumentwerte

Besitzt das Polynom P_n komplexe Koeffizienten und ist x_0 ein komplexer Argumentwert, so kann man zur Berechnung des Funktionswertes $P_n(x_0)$ das einfache Horner-Schema verwenden. Man hat dann lediglich für jeden Koeffizienten eine reelle und eine imaginäre Spalte zu berechnen; siehe Beispiel 3.1.

Besitzt das Polynom P_n jedoch nur reelle Koeffizienten, so kann man zur Berechnung des Funktionswertes $P_n(x_0)$ zu einem komplexen Argumentwert x_0 mit dem Horner-Schema ganz im Reellen bleiben, wenn man das sogenannte *doppelreihige* Horner-Schema verwendet. Zunächst nimmt man den zu x_0 konjugiert komplexen Argumentwert $\overline{x}_0$ hinzu und bildet

$$(x - x_0)(x - \overline{x}_0) = x^2 - px - q$$

mit reellen Zahlen p und q; es gilt $p = x_0 + \overline{x}_0$, $q = -x_0 \cdot \overline{x}_0$.

Dividiert man jetzt P_n durch $(x^2 - px - q)$, so erhält man die Beziehung

$$\left\{\begin{array}{lcll} P_n(x) & = & (x^2 - px - q)P_{n-2}(x) + b_1^{(1)}x_0 + b_0^{(1)} & \text{mit} \\ P_{n-2}(x) & = & b_n^{(1)}x^{n-2} + b_{n-1}^{(1)}x^{n-3} + \ldots + b_3^{(1)}x + b_2^{(1)}. & \end{array}\right. \tag{3.6}$$

Für die Koeffizienten $b_k^{(1)}$ von P_{n-2} gelten

$$\left\{\begin{array}{lcll} b_n^{(1)} & = & a_n^{(0)}, & \\ b_{n-1}^{(1)} & = & a_{n-1}^{(0)} + pb_n^{(1)}, & \\ \quad\vdots & & & \\ b_k^{(1)} & = & a_k^{(0)} + pb_{k+1}^{(1)} + qb_{k+2}^{(1)}, & k = (n{-}2)(-1)1, \\ \quad\vdots & & & \\ b_0^{(1)} & = & a_0^{(0)} + qb_2^{(1)}. & \end{array}\right. \tag{3.7}$$

Die Rechenoperationen (3.7) werden in dem folgenden *doppelreihigen Horner-Schema* durchgeführt:

P_n	$a_n^{(0)}$	$a_{n-1}^{(0)}$	$a_{n-2}^{(0)}$	$\dots$	$a_2^{(0)}$	$a_1^{(0)}$	$a_0^{(0)}$
q	0	0	$qb_n^{(1)}$	$\dots$	$qb_4^{(1)}$	$qb_3^{(1)}$	$qb_2^{(1)}$
p	0	$pb_n^{(1)}$	$pb_{n-1}^{(1)}$	$\dots$	$pb_3^{(1)}$	$pb_2^{(1)}$	0
$\sum$	$b_n^{(1)}$	$b_{n-1}^{(1)}$	$b_{n-2}^{(1)}$	$\dots$	$b_2^{(1)}$	$b_1^{(1)}$	$b_0^{(1)}$

Für $x = x_0$ folgt aus (3.6) wegen $x_0^2 - px_0 - q = 0$

$$P_n(x_0) = b_1^{(1)} x_0 + b_0^{(1)} \tag{3.8}$$

als gesuchter Funktionswert. Ist x_0 Nullstelle von P_n, so folgt wegen $P_n(x_0) = 0$ aus (3.8):

$$b_0^{(1)} = 0, \qquad b_1^{(1)} = 0.$$

Abdividieren von komplexen Nullstellen bei Polynomen mit reellen Koeffizienten

Ist x_0 Nullstelle von P_n, so gilt wegen $P_n(x_0) = 0$ (d. h. $b_0^{(1)} = 0$, $b_1^{(1)} = 0$) gemäß (3.6)

$$P_n(x) = (x^2 - px - q) P_{n-2}(x).$$

Die Koeffizienten des Deflationspolynoms P_{n-2} sind die $b_k^{(1)}$ im doppelreihigen Horner-Schema.

Beispiel 3.2.

Gegeben: $P_5(x) = x^5 - 3x^4 - x + 3$.

Gesucht: Das Deflationspolynom $P_{5-2}(x)$ zu den Nullstellen $x_0 = \mathrm{i}$ und $\overline{x}_0 = -\mathrm{i}$.

Lösung: Zunächst berechnet man die Werte p und q aus der Beziehung

$$\begin{aligned} p &= x_0 + \overline{x}_0 \\ q &= -x_0\,\overline{x}_0\,. \end{aligned}$$

Daraus ergeben sich: $p = \mathrm{i} - \mathrm{i} = 0$ und $q = (-\mathrm{i})(-\mathrm{i}) = -1$.

Man erhält mit dem doppelreihigen Horner-Schema

P_5	1	-3	0	0	-1	3
$q = -1$	0	0	$(-1)1$	$(-1)(-3)$	$(-1)(-1)$	$(-1)3$
$p = 0$	0	$0{\cdot}1$	$0{\cdot}(-3)$	$0{\cdot}(-1)$	$0{\cdot}3$	$0{\cdot}0$
$\sum$	1	-3	-1	3	0	0

Das Deflationspolynom lautet: $P_3(x) = x^3 - 3x^2 - x + 3$ □

3.2.3 Das vollständige Horner-Schema für reelle Argumentwerte

Da das Horner-Schema neben dem Funktionswert $P_n(x_0)$ auch die Koeffizienten $a_j^{(1)}$ des abdividierten Polynoms P_{n-1} liefert, ergibt sich die Möglichkeit, die k-ten Ableitungen $P_n^{(k)}$ des Polynoms P_n für $k = 1(1)n$ an der Stelle $x_0 \in \mathbb{R}$ zu berechnen. Aus (3.5) ergibt sich für die 1. Ableitung

$$P_n'(x) = P_{n-1}(x) + (x - x_0)P_{n-1}'(x),$$

und für $x = x_0$

$$P_n'(x_0) = P_{n-1}(x_0).$$

$P_n'(x_0)$ erhält man also, indem man an die 3. Zeile des Horner-Schemas ein weiteres einfaches Horner-Schema anschließt.

Beispiel 3.3. (Fortsetzung von Beispiel 3.1)

P_3	$\ldots$			
$x_0 = 2$	$\ldots$			
P_2	2	0	3	$\boxed{21 = P_3(2)}$
$x_0 = 2$	0	4	8	
	2	4		$\boxed{11 = P_2(2) = P_3'(2)}$

□

Verfährt man nun mit $P_{n-1}(x)$ analog wie mit $P_n(x)$, so erhält man

$$P_{n-1}'(x_0) = P_{n-2}(x_0) = \frac{1}{2}P_n''(x_0).$$

So fortfahrend erhält man schließlich für die abdividierten Polynome

$$P_{n-k}(x_0) = \frac{1}{k!}P_n^{(k)}(x_0) \quad \text{für} \quad k = 1(1)n.$$

Beweis. Durch mehrfache Anwendung des einfachen Horner-Schemas erhält man gemäß (3.6)

$$\left\{\begin{array}{lcll} P_n(x) & = & (x - x_0)P_{n-1}(x) & + \quad P_n(x_0), \\ P_{n-1}(x) & = & (x - x_0)P_{n-2}(x) & + \quad P_{n-1}(x_0), \\ \vdots & & & \\ P_1(x) & = & (x - x_0)P_0(x) & + \quad P_1(x_0). \end{array}\right. \tag{3.9}$$

Mit

$$c_k := P_{n-k}(x_0) \tag{3.10}$$

ergibt sich durch sukzessives Einsetzen der Gleichungen in (3.9) für P_n die Darstellung

$$\begin{aligned} P_n(x) &= (x-x_0)P_{n-1}(x)+c_0 \\ &= (x-x_0)[(x-x_0)P_{n-2}(x)+c_1]+c_0 \\ &= (x-x_0)^2P_{n-2}(x)+(x-x_0)c_1+c_0 \\ &\vdots \\ &= (x-x_0)^n c_n+(x-x_0)^{n-1}c_{n-1}+\ldots+(x-x_0)c_1+c_0. \end{aligned}$$

Durch Koeffizientenvergleich mit der Taylorentwicklung von P_n um x_0:

$$P_n(x)=\sum_{k=0}^{n}(x-x_0)^k\frac{1}{k!}P_n^{(k)}(x_0)\overset{!}{=}\sum_{k=0}^{n}(x-x_0)^k c_k \tag{3.11}$$

erhält man mit (3.10) und aus (3.11)

$$c_k=P_{n-k}(x_0)=\frac{1}{k!}P_n^{(k)}(x_0).$$

Durch Fortsetzung des Horner-Schemas erhält man also mit dem abdividierten Polynom in x_0 $P_{n-k}(x_0)$ die Taylorentwicklung von P_n an der Stelle $x=x_0$

$$P_n(x)=\sum_{k=0}^{n}(x-x_0)^k P_{n-k}(x_0)=\sum_{k=0}^{n}(x-x_0)^k\frac{1}{k!}P_n^{(k)}(x_0). \tag{3.12}$$

Rechenschema 3.4. (*Vollständiges Horner-Schema*)

P_n	$a_n^{(0)}$	$a_{n-1}^{(0)}$	$a_{n-2}^{(0)}$	$\ldots$	$a_1^{(0)}$	$a_0^{(0)}$
$x=x_0$	0	$a_n^{(1)}x_0$	$a_{n-1}^{(1)}x_0$	$\ldots$	$a_2^{(1)}x_0$	$a_1^{(1)}x_0$
P_{n-1}	$a_n^{(1)}$	$a_{n-1}^{(1)}$	$a_{n-2}^{(1)}$	$\ldots$	$a_1^{(1)}$	$\boxed{a_0^{(1)}=P_n(x_0)}$
$x=x_0$	0	$a_n^{(2)}x_0$	$a_{n-1}^{(2)}x_0$	$\ldots$	$a_2^{(2)}x_0$	
P_{n-2}	$a_n^{(2)}$	$a_{n-1}^{(2)}$	$a_{n-2}^{(2)}$	$\ldots$	$\boxed{a_1^{(2)}=\frac{1}{1!}P_n'(x_0)}$	$=P_{n-1}(x_0)$
$\ldots$	$\ldots$	$\ldots$	$\ldots$			
P_1	$a_n^{(n-1)}$	$a_{n-1}^{(n-1)}$	$\ldots$			
$x=x_0$	0	$a_n^{(n)}x_0$				
P_0	$a_n^{(n)}$	$\boxed{a_{n-1}^{(n)}=\frac{1}{(n-1)!}P_n^{(n-1)}(x_0)}$				$=P_1(x_0)$
$x=x_0$	0					
	$\boxed{a_n^{(n+1)}=\frac{1}{n!}P_n^{(n)}(x_0)}$					$=P_0(x_0)$

mit $a_n^{(k)}=a_n^{(k-1)}$, $a_j^{(k)}=a_{j+1}^{(k)}x_0+a_j^{(k-1)}$ für $j=n{-}1(-1)k{-}1$, $k=1(1)n{+}1$.

Anzahl der Punktoperationen

Die Aufstellung der Taylorentwicklung von P_n an einer Stelle x_0 mit Hilfe des vollständigen Horner-Schemas erfordert $(n^2+n)/2$ Punktoperationen, während der übliche Weg (Differenzieren, Berechnen der Werte der Ableitungen, Dividieren durch $k!$, wobei $k!$ als bekannter Wert vorausgesetzt wird) n^2+2n-2, also für $n \geq 3$ mehr als doppelt so viele Punktoperationen erfordert. Durch das Einsparen von Punktoperationen wird das rundungsfehlergünstige Arbeiten ermöglicht, denn durch hohe Potenzen häufen sich systematische Rundungsfehler an.

Beispiel 3.5. (Fortsetzung von Beispiel 3.1)

Gegeben: Das Polynom P_3 mit $P_3(x) = 2x^3 - 4x^2 + 3x + 15$.

Gesucht: Die Taylorentwicklung für P_3 an der Stelle $x_0 = 2$.

Lösung: Die Vorgehensweise erfolgt laut Rechenschema 3.4. Da nur die Koeffizienten der Taylorentwicklung interessieren, werden die Bezeichnungen P_{n-k} der Polynome niedrigeren Grades weggelassen.

P_3	2	-4	3	15	
$x_0 = 2$	0	4	0	6	
P_2	2	0	3	$\boxed{21 = P_3(2)}$	
$x_0 = 2$	0	4	8		
P_1	2	4	$\boxed{11 = P_3'(2)}$		
$x_0 = 2$	0	4			
P_0	2	$\boxed{8 = \frac{1}{2!}P_3''(2)}$			
$x_0 = 2$	0				
	$\boxed{2 = \frac{1}{3!}P_3'''(2)}$				

Die Taylorentwicklung (3.12) für P_3 um $x_0 = 2$ lautet hiermit

$$\begin{aligned} P_3(x) &= P_3(x_0) + (x-x_0)P_3'(x_0) + (x-x_0)^2\tfrac{1}{2!}P_3''(x_0) + (x-x_0)^3\tfrac{1}{3!}P_3'''(x_0) \\ &= 21 + 11(x-2) + 8(x-2)^2 + 2(x-2)^3. \end{aligned}$$

□

3.2.4 Anwendungen

Das Horner-Schema wird verwendet

(1) zur bequemen, schnellen und rundungsfehlergünstigen Berechnung der Funktionswerte und Ableitungswerte eines Polynoms P_n

(2) zur Aufstellung der Taylorentwicklung eines Polynoms P_n an einer Stelle x_0

(3) zum Abdividieren von Nullstellen (Deflation von Polynomen)

Man wird z. B. bei der iterativen Bestimmung einer Nullstelle eines Polynoms nach einem der Newton-Verfahren (siehe Abschnitt 2.5) P_n, P_n' bzw. P_n, P_n', P_n'' mit dem Horner-Schema berechnen.

Hat man für eine Nullstelle x_1 von P_n iterativ eine hinreichend gute Näherung erhalten, so dividiert man P_n durch $(x - x_1)$ und wendet das Iterationsverfahren auf das Deflationspolynom P_{n-1} an. So erhält man nacheinander alle Nullstellen von P_n und schließt aus, eine Nullstelle zweimal zu berechnen. Dabei könnten sich aber die Nullstellen der abdividierten Polynome immer weiter von den Nullstellen des Ausgangspolynoms P_n entfernen, so dass die Genauigkeit mehr und mehr abnimmt. Wilkinson empfiehlt deshalb in [WILK1969], S.70-83, das Abdividieren von Nullstellen grundsätzlich mit der betragskleinsten Nullstelle zu beginnen, d. h. mit einer Methode zu arbeiten, die für das jeweilige Polynom eine Anfangsnäherung so auswählt, dass die Iteration gegen die betragskleinste Nullstelle konvergiert (s. Verfahren von Muller, Abschnitt 3.3.2). Wird diese Forderung erfüllt, so ergeben sich alle Nullstellen mit einer Genauigkeit, die im Wesentlichen von ihrer Kondition bestimmt ist und nicht von der Genauigkeit der vorher bestimmten Nullstelle. Wilkinson empfiehlt außerdem, nachdem man alle Nullstellen mittels Iteration und Abdividieren gefunden hat, die berechneten Näherungswerte als Startwerte für eine Nachiteration mit dem ursprünglichen Polynom zu verwenden. Man erreicht damit eine Erhöhung der Genauigkeit, besonders in den Fällen, in denen das Abdividieren die Kondition verschlechtert hat.

Beispiel 3.6.

Gegeben: Das Polynom $P_3 : P_3(x) = x^3 - 1$ und die Nullstelle $x_1 = 1$ von P_3.

Gesucht: Die beiden anderen Nullstellen von P_3.

Lösung:

P_3	1	0	0	-1
$x_1 = 1$	0	1	1	1
P_2	1	1	1	$\boxed{0 = P_3(1)}$

Das Horner-Schema liefert die Koeffizienten des Deflationspolynoms P_2 :
$P_2(x) = x^2 + x + 1$.
Wegen $P_3(x) = (x - 1)P_2(x)$ ergeben sich aus $P_2(x) = 0$ die beiden anderen Nullstellen $x_{2,3} = (1/2)(-1 \pm \mathrm{i}\sqrt{3})$. □

3.3 Methoden zur Bestimmung sämtlicher Lösungen algebraischer Gleichungen

3.3.1 Vorbemerkungen und Überblick

Wenn hinreichend genaue Anfangsnäherungen für die Nullstellen eines Polynoms vorliegen, kann man mit Iterationsverfahren Folgen von Näherungswerten konstruieren, die gegen die Nullstellen konvergieren. Das Problem liegt in der Beschaffung der Startwerte.

Will man z. B. sämtliche reellen Nullstellen eines Polynoms P_n mit reellen Koeffizienten mit Hilfe eines der bisher angegebenen Iterationsverfahren berechnen, so muss man:

1. ein Intervall I ermitteln, in dem alle Nullstellen liegen. Das kann z. B. nach dem folgenden Satz geschehen:

   Ist $P_n(x) = x^n + a_{n-1}x^{n-1} + \ldots + a_1x + a_0$ das gegebene Polynom und $A = \max\limits_{k=0(1)n-1} |a_k|$, so liegen alle Nullstellen von P_n in einem Kreis um den Nullpunkt der komplexen Zahlenebene mit dem Radius $r = A+1$. Also ist $I = [-r, +r]$.

   Ist P_n ein Polynom mit lauter reellen Nullstellen (z. B. ein Orthogonalpolynom, s. Abschnitt 8.2.2, Sonderfälle 2.), so kann der *Satz von Laguerre* angewandt werden:

   Die Nullstellen liegen alle in einem Intervall, dessen Endpunkte durch die beiden Lösungen der quadratischen Gleichung

   $$nx^2 + 2a_{n-1}x + 2(n-1)a_{n-2} - (n-2)a_{n-1}^2 = 0$$

   gegeben sind.

2. die Anzahl reeller Nullstellen nach den Vorzeichenregeln von Sturm und Descartes berechnen,

3. die Lage der Nullstellen durch Intervallteilung, Berechnung der Funktionswerte und Abzählung der Anzahl der Vorzeichenwechsel ermitteln.

Mit 3. ist es möglich, Intervalle $I_k \subset I$ anzugeben, in denen jeweils nur eine Nullstelle x_k ungerader Ordnung liegt. Dann lässt sich z. B. das Newtonsche Verfahren zur näherungsweisen Berechnung der x_k anwenden. Dabei sind P_n und P_n' (bzw. P_n, P_n', P_n'') mit Hilfe des Horner-Schemas zu berechnen.

Der soeben beschriebene Weg ist mühsam und für die Praxis uninteressant. Hier braucht man Verfahren, die in kürzester Zeit und ohne Kenntnis von Startwerten sämtliche reellen und komplexen Nullstellen eines Polynoms mit reellen bzw. komplexen Koeffizienten liefern.

Für Polynome mit reellen Koeffizienten werden diese Anforderungen mühelos vom *Verfahren von Muller* erfüllt, s. Abschnitt 3.3.2. Das Muller-Verfahren lässt sich auch auf Polynome mit komplexen Koeffizienten erweitern.

Für Polynome mit komplexen Koeffizienten werden hier zwei Verfahren genannt, das *Verfahren von Jenkins und Traub* und das *Verfahren von Bauhuber*. Das Verfahren von Jenkins-Traub wird hier nur kurz ohne Formulierung eines Algorithmus beschrieben.

3.3.2 Das Verfahren von Muller

Das Verfahren von Muller [MULL1956] liefert ohne vorherige Kenntnis von Startwerten sämtliche reellen und konjugiert komplexen Nullstellen eines Polynoms

$$P_n : P_n(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n = \sum_{j=0}^{n} a_j x^j \quad \text{für} \quad a_j \in \mathbb{R}, \quad a_n \neq 0.$$

Dazu wird wie folgt vorgegangen:

Zunächst wird durch Muller-Iteration (s. Muller-Iteration) ein Näherungswert $x_1^{(N)}$ für die betragskleinste Nullstelle x_1 von P_n bestimmt. Nach Division $P_n(x)/(x - x_1^{(N)})$ mit Horner und Vernachlässigung des Restes erhält man ein Polynom P_{n-1} vom Grad $n-1$, das im Rahmen der erzielten Genauigkeit ungefähr gleich dem Deflationspolynom $P_n(x)/(x - x_1)$ ist. Von P_{n-1} wird wiederum durch Muller-Iteration mit P_{n-1} statt P_n und x_2 statt x_1 ein Näherungswert $x_2^{(N)}$ für die betragskleinste Nullstelle x_2 von P_{n-1} bestimmt. Mit $x_2^{(N)}$ wird analog verfahren. Man erhält so Näherungswerte sämtlicher Nullstellen von P_n ungefähr dem Betrage nach geordnet (möglicherweise erhält man auch die im Betrag zweitkleinste Nullstelle zuerst).

In den meisten Testbeispielen ergab sich die Anordnung

$$|x_1| \leq |x_2| \leq \ldots \leq |x_n|.$$

Hat man durch Abdividieren und Anwendung des Muller-Verfahrens auf das jeweilige Deflationspolynom alle Nullstellen von P_n näherungsweise gefunden, so empfiehlt es sich grundsätzlich, die gewonnenen Näherungswerte als Startwerte für eine Nachbesserung mit dem Newton-Verfahren, angewandt auf das ursprüngliche Polynom P_n, zu verwenden. Man sollte aber auf jeden Fall erst sämtliche Nullstellen von P_n auf dem beschriebenen Weg näherungsweise berechnen, bevor man sie verbessert. Nach Untersuchungen von Wilkinson ist es nicht notwendig, direkt jede Nullstelle noch vor dem Abdividieren mit dem ursprünglichen Polynom zu verbessern.

Prinzip der Muller-Iteration

Das Verfahren von Muller arbeitet mit quadratischen Parabeln (siehe Abbildung 3.1): Zur Bestimmung der (im Allgemeinen betragskleinsten) Nullstelle x_1 von P_n wird durch die drei Punkte

$$\left(x^{(0)}, P_n(x^{(0)})\right), \left(x^{(1)}, P_n(x^{(1)})\right), \left(x^{(2)}, P_n(x^{(2)})\right)$$

die quadratische Parabel mit der Gleichung $y = \Phi(x)$ gelegt; sie ist eindeutig bestimmt. Als Startwerte werden $x^{(0)} = -1$, $x^{(1)} = 1$, $x^{(2)} = 0$ genommen. Dann werden die beiden Nullstellen von Φ berechnet und die zu $x^{(2)}$ nächstliegende als neue Näherung $x^{(3)}$ für x_1

verwendet. In einem nächsten Schritt wird eine neue Parabel $y = \Phi(x)$ durch die Punkte $\big(x^{(k)}, P_n(x^{(k)})\big)$ für $k = 1, 2, 3$ gelegt und die zu $x^{(3)}$ nächstliegende Nullstelle von Φ als neue Näherung $x^{(4)}$ für x_1 bestimmt. Analog fortfahrend bricht man das Verfahren ab, wenn sich für einen festen Index N die Näherungen $x^{(N)}$ und $x^{(N-1)}$ hinreichend wenig voneinander unterscheiden, $x^{(N)}$ ist dann der gesuchte Näherungswert für die Nullstelle x_1 von P_n.

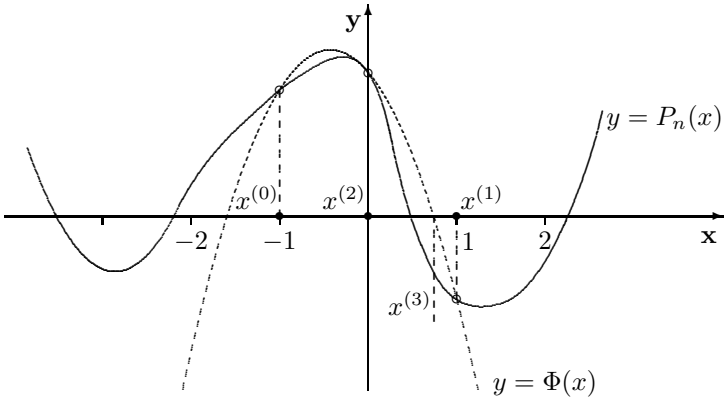


Abb. 3.1. Zum Muller-Verfahren

Um die Nullstelle x_2 von P_n zu erhalten, wird der Linearfaktor $(x - x^{(N)})$ von P_n mit Hilfe des Horner-Schemas abdividiert, der Rest vernachlässigt und das Muller-Verfahren nunmehr auf P_{n-1} angewandt. Man erhält so im Allgemeinen die betragskleinste Nullstelle von P_{n-1}, d. h. die Nullstelle x_2 von P_n mit $|x_1| \le |x_2|$.

Herleitung der Iterationsvorschrift für das Muller-Verfahren

Eine quadratische Parabel ist durch die Vorgabe von drei Punkten eindeutig bestimmt, ihre Gleichung $y = \Phi(x)$ ergibt sich als Interpolationspolynom Φ zu den Wertepaaren $\big(x^{(k)}, f(x^{(k)})\big) = \big(x^{(k)}, P_n(x^{(k)})\big)$ für $k = \nu{-}2,\ \nu{-}1,\ \nu$ in Lagrangescher Form wie folgt:

$$\begin{aligned}\Phi(x) &= \frac{(x - x^{(\nu-1)})(x - x^{(\nu-2)})}{(x^{(\nu)} - x^{(\nu-1)})(x^{(\nu)} - x^{(\nu-2)})} f(x^{(\nu)}) \\ &+ \frac{(x - x^{(\nu)})(x - x^{(\nu-2)})}{(x^{(\nu-1)} - x^{(\nu)})(x^{(\nu-1)} - x^{(\nu-2)})} f(x^{(\nu-1)}) \\ &+ \frac{(x - x^{(\nu)})(x - x^{(\nu-1)})}{(x^{(\nu-2)} - x^{(\nu)})(x^{(\nu-2)} - x^{(\nu-1)})} f(x^{(\nu-2)}).\end{aligned} \tag{3.13}$$

Mit $h_\nu := x^{(\nu)} - x^{(\nu-1)}$, $h := x - x^{(\nu)}$ erhält man für (3.13) die Darstellung

$$\begin{aligned}\Phi(x) = \Phi(x^{(\nu)} + h) &= \frac{(h + h_\nu)(h + h_\nu + h_{\nu-1})}{h_\nu(h_\nu + h_{\nu-1})} f(x^{(\nu)}) \\ &- \frac{h(h + h_\nu + h_{\nu-1})}{h_\nu h_{\nu-1}} f(x^{(\nu-1)}) \\ &+ \frac{h(h + h_\nu)}{(h_\nu + h_{\nu-1})h_{\nu-1}} f(x^{(\nu-2)}).\end{aligned}$$

Nach Zusammenfassung gleicher Potenzen in h und mit den Abkürzungen

$$q_\nu := \frac{h_\nu}{h_{\nu-1}}, \quad q := \frac{h}{h_\nu}, \quad f_\nu := f(x^{(\nu)})$$

erhält man die Parabelgleichung $y = \Phi(x)$ mit

$$\Phi(x) = \Phi(x^{(\nu)} + qh_\nu) = \frac{A_\nu q^2 + B_\nu q + C_\nu}{1 + q_\nu}$$

wobei die folgenden Abkürzungen verwendet werden:

$$\begin{aligned} A_\nu &:= q_\nu f_\nu - q_\nu(1+q_\nu)f_{\nu-1} + q_\nu^2 f_{\nu-2}, \\ B_\nu &:= (2q_\nu + 1)f_\nu - (1+q_\nu)^2 f_{\nu-1} + q_\nu^2 f_{\nu-2}, \\ C_\nu &:= (1+q_\nu)f_\nu. \end{aligned} \tag{3.14}$$

Zur Bestimmung der Schnittpunkte der Parabel mit der x-Achse setzt man

$$A_\nu q^2 + B_\nu q + C_\nu = 0.$$

Dies führt auf die beiden Werte für q:

$$\begin{aligned} q_{1/2} &= -\frac{B_\nu}{2A_\nu} \pm \sqrt{\frac{B_\nu^2}{4A_\nu^2} - \frac{C_\nu}{A_\nu}} = \frac{-B_\nu \pm \sqrt{B_\nu^2 - 4A_\nu C_\nu}}{2A_\nu} \\ &= \frac{-2C_\nu}{B_\nu \pm \sqrt{B_\nu^2 - 4A_\nu C_\nu}} \end{aligned} \tag{3.15}$$

und damit auf die Lösungen $x_{1/2}$ der Gleichung $\Phi(x^{(\nu)} + qh_\nu) = 0$:

$$x_{1/2} = x^{(\nu)} + h_\nu q_{1/2}. \tag{3.16}$$

Eine der beiden Lösungen wird als neue Näherung $x^{(\nu+1)}$ verwendet, es wird diejenige ausgewählt, die näher an $x^{(\nu)}$ liegt. Dies wird dadurch realisiert, dass im Nenner von (3.15) das Vorzeichen der Wurzel $\sqrt{B_\nu^2 - 4A_\nu C_\nu}$ so gewählt wird, dass der Nenner den größeren Betrag erhält.

Falls der Nenner von (3.15) verschwindet (dies ist dann der Fall, wenn $f(x^{(\nu)}) = f(x^{(\nu-1)}) = f(x^{(\nu-2)})$ gilt), schlägt Muller vor, statt (3.15) für $q_{1/2} = 1$ zu setzen und damit weiterzurechnen. In diesem Fall entartet die Parabel in eine zur x-Achse parallele Gerade, die keinen Schnittpunkt mit der x-Achse hat.

Der ausgewählte Wert von $q_{1/2}$ wird dann $q_{\nu+1}$ genannt, und es wird gesetzt

$$x^{(\nu+1)} := x^{(\nu)} + h_\nu q_{\nu+1} \quad \text{für} \quad \nu = 2, 3, 4, \ldots \tag{3.17}$$

Automatischer Startprozess

Als Startwerte für die Iteration werden fest vorgegeben

$$x^{(0)} = -1, \qquad x^{(1)} = 1, \qquad x^{(2)} = 0. \tag{3.18}$$

Als Funktionswerte an den Stellen $x^{(0)}$, $x^{(1)}$, $x^{(2)}$ (und nur an diesen!) werden im Allgemeinen nicht die Funktionswerte des jeweiligen Polynoms P_n genommen, sondern die Werte

$$\begin{array}{lcl} a_0 - a_1 + a_2 & \text{statt} & f_0 := P_n(x^{(0)}), \\ a_0 + a_1 + a_2 & \text{statt} & f_1 := P_n(x^{(1)}), \\ a_0 & \text{für} & f_2 := P_n(x^{(2)}). \end{array} \tag{3.19}$$

Die Verwendung dieser künstlichen Funktionswerte wurde von Muller selbst empfohlen (vgl. Algorithmus 3.7). Man kann aber ebenso an den Startstellen $x^{(0)}, x^{(1)}, x^{(2)}$ die wirklichen Polynomwerte $f_0 := P_n(x^{(0)})$, $f_1 := P_n(x^{(1)})$, $f_2 := P_n(x^{(2)})$ benutzen, wie es in Abbildung 3.1 gemacht wurde.

Abbruchbedingung

Die Iteration (3.17) wird abgebrochen, falls zu vorgegebenem $\varepsilon > 0$ die Bedingung

$$\left| \frac{x^{(\nu+1)} - x^{(\nu)}}{x^{(\nu+1)}} \right| < \varepsilon \tag{3.20}$$

erfüllt ist. Ist dies für ein $\nu = N-1$ der Fall, so ist $x^{(N)} = x_1^{(N)}$ der gesuchte Näherungswert für x_1.

Auftreten konjugiert komplexer Nullstellen

Falls der Radikand der Wurzel in (3.15) negativ ausfällt, so kann dies zwei Ursachen haben:

(1) Eine reelle Lösung der Gleichung $f(x) \equiv P_n(x) = 0$ wird durch eine Folge konjugiert komplexer Zahlen approximiert. Die Imaginärteile der Folge $\{x^{(\nu)}\}$ sowie die Imaginärteile der zugehörigen Polynomwerte streben dann gegen Null.

(2) x_1 ist eine komplexe Nullstelle. Mit x_1 ist dann auch $\overline{x}_1$ Nullstelle von P_n. In diesem Fall liefert die Division $P_n/[(x-x_1^{(N)})(x-\overline{x}_1^{(N)})]$ unter Vernachlässigung des Restes ein Polynom P_{n-2} vom Grad $n-2$ (s. Abschnitt 3.2.2), für das wiederum die Muller-Iteration eine Näherung für die im Allgemeinen betragskleinste Nullstelle liefert.

Zur Konvergenz des Verfahrens

Konvergenz im Großen konnte nicht nachgewiesen werden. Es konnte aber gezeigt werden, dass Konvergenz eintritt, wenn der Prozess hinreichend nahe an einer einfachen bzw. doppelten Nullstelle beginnt. Jedoch erreichte Muller mit der in [MULL1956], S.210, angegebenen Modifikation Konvergenz in allen getesteten Fällen zu dem angegebenen Startprozess. Die Modifikation besteht darin, dass man jeweils die Bedingung

$$|f(x^{(\nu+1)})/f(x^{(\nu)})| \leq 10 \tag{3.21}$$

prüfen muss. Ist der Wert > 10, so wird $q_{\nu+1}$ halbiert, es werden $h_{\nu+1}$, $x^{(\nu+1)}$ und $f(x^{(\nu+1)})$ neu berechnet.

Algorithmus 3.7. (*Muller-Verfahren*)
Gegeben ist ein Polynom P_n, gesucht ist die betragskleinste Nullstelle x_1 von P_n.

1. Schritt: Benutzung der Startwerte (3.18) $x^{(0)} = -1, x^{(1)} = 1, x^{(2)} = 0$ und Berechnung von $f_\nu := P_n(x^{(\nu)})$ für $\nu = 0, 1, 2$ bzw. der künstlichen Funktionswerte (3.19).

2. Schritt: Berechnung der Näherungen $x^{(\nu+1)}$, $\nu = 2, 3, 4$ ... für x_1 gemäß Iterationsvorschrift (3.17) mit (3.16),(3.15),(3.14).

   Nach jedem Iterationsschritt ist die Bedingung (3.21) zu prüfen. Ist sie nicht erfüllt, so ist $q_{\nu+1}$ zu halbieren, und es sind $h_{\nu+1}$, $x^{(\nu+1)}$ und $f_{\nu+1}$ neu zu berechnen und damit die Iteration fortzusetzen. Die Iteration ist abzubrechen, falls (3.20) für ein $\nu = N$–1 erfüllt ist. Dann ist $x^{(N)} = x_1^{(N)}$ der gesuchte Näherungswert für x_1.

3. Schritt:
   (a) Berechnung des abdividierten Polynoms P_{n-1} im Falle einer reellen Näherung $x_1^{(N)}$ mit dem Horner-Schema (siehe Abschnitt 3.2.1) unter Vernachlässigung des Restes, der entstehen kann, weil $x_1^{(N)}$ ein Näherungswert für x_1 ist.

   (b) Berechnung des abdividierten Polynoms P_{n-2} im Falle einer komplexen Lösung $x_1^{(N)}$ mit dem doppelreihigen Horner-Schema (siehe Abschnitt 3.2.2); es gilt $P_{n-2}(x) = P_n(x)/[(x - x_1^{(N)})(x - \overline{x}_1^{(N)})]$, wobei $\overline{x}_1^{(N)}$ die zu $x_1^{(N)}$ konjugiert komplexe Lösung von P_n ist. Auch hier ist der Rest zu vernachlässigen.

Alle weiteren Nullstellen werden nach dem gleichen Algorithmus berechnet, jeweils angewandt auf die abdividierten Polynome $f := P_{n-1}$, $f := P_{n-2}$,... . Man erhält so sämtliche Lösungen in etwa dem Betrage nach geordnet.

Konvergenzordnung

Sei ξ eine einfache Nullstelle des Polynoms $f(x) := P_n(x)$, so lässt sich zeigen, dass zwischen den Fehlern

$$\Delta x^{(\nu)} := x^{(\nu)} - \xi$$

aufeinander folgender Muller-Iterationen die Beziehung

$$\left\{ \begin{array}{lcl} \Delta x^{(\nu+1)} & = & \Delta x^{(\nu)} \cdot \Delta x^{(\nu-1)} \cdot \Delta x^{(\nu-2)} \cdot \left(\dfrac{f^{(3)}(\xi)}{6f'(\xi)} + O(\varepsilon) \right) \\ \text{mit} \quad \xi & = & \max\limits_{i=\nu-2,\nu-1,\nu} |\Delta x^{(i)}| \end{array} \right. \tag{3.22}$$

besteht. Heuristisch betrachtet ergibt sich aus (3.22) eine Beziehung der Form

$$|\Delta x^{(\nu+1)}| \approx K \cdot |\Delta x^{(\nu)}| \cdot |\Delta x^{(\nu-1)}| \cdot |\Delta x^{(\nu-2)}| \,,$$

in die man gemäß Definition der Konvergenzordnung p einsetzen kann

$$|\Delta x^{(k+1)}| \approx M |\Delta x^{(k)}|^p \quad \text{für} \quad k = \nu-1, \nu, \nu+1,$$

so dass sich ergibt

$$M|\Delta x^{(\nu)}|^p \approx KM^{-\frac{2}{p}-\frac{1}{p^2}}|\Delta x^{(\nu)}|^{1+\frac{1}{p}+\frac{1}{p^2}} \ . \tag{3.23}$$

Durch Vergleich zugehöriger Exponenten in (3.23) kann man wiederum folgern, dass die Gleichungen

$$M = KM^{-\frac{2}{p}-\frac{1}{p^2}} \Rightarrow K = M^{1+\frac{2}{p}+\frac{1}{p^2}}$$

und

$$p = 1 + \frac{1}{p} + \frac{1}{p^2} \Rightarrow p^3 - p^2 - p - 1 = 0 \tag{3.24}$$

gelten müssen. Als einzige reelle Nullstelle der kubischen Gleichung (3.24) erhält man nun für die Konvergenzordnung des Muller-Verfahrens im Falle einfacher Nullstellen den Wert

$$p \approx 1.84 \ .$$

Im Falle doppelter Nullstellen ermittelt man analog p ≈ 1.23.

Beispiel 3.8.

Gegeben: Das Polynom P_3 mit

$$f(x) := P_3(x) = x^3 - 7x^2 - 36x + 252 \ .$$

Gesucht: Eine (im Allgemeinen die betragskleinste) Nullstelle mit dem Muller-Verfahren. Für die Abbruchbedingung wähle man die sehr grobe Schranke $\varepsilon = 0.1$, damit nur wenige, mit dem Taschenrechner nachvollziehbare Schritte durchgeführt werden müssen.

Lösung: Die exakten Nullstellen liegen hier bei $x_1 = 6, x_2 = -6, x_3 = 7$, voraussichtlich wird mit Muller eine der Nullstellen -6 oder $+6$ zuerst ermittelt. Es wird nach Algorithmus 3.7 vorgegangen.

1. Schritt. (Bestimmung der Startwerte): Es werden $a_0 = 252, a_1 = -36, a_2 = -7, a_3 = 1$ gesetzt und die künstlichen Funktionswerte (3.19) verwendet:

$$\begin{array}{lcl} f_0 & := & a_0 - a_1 + a_2 = 281 \ , \\ f_1 & := & a_0 + a_1 + a_2 = 209 \ , \\ f_2 & := & a_0 = 252 \ . \end{array}$$

2. Schritt. (Muller-Iteration): Mit der Abbruchbedingung

$$\frac{|x^{(\nu+1)} - x^{(\nu)}|}{|x^{(\nu+1)}|} = K^{(\nu+1)} < \varepsilon = 0.1$$

erhält man folgende Werte in der Tabelle bei 6-stelliger Mantisse:

	$\nu=2$	$\nu=3$	$\nu=4$	$\nu=5$
$x^{(\nu)}$	0.00000	3.95638	5.04950	5.64275
$x^{(\nu-1)}$	1.00000	0.00000	3.95638	5.04950
$x^{(\nu-2)}$	−1.00000	1.00000	0.00000	3.95638
f_ν	252.000	61.9288	20.4851	5.64525
$f_{\nu-1}$	209.000	252.000	61.9288	20.4851
$f_{\nu-2}$	281.000	209.000	252.000	61.9288
h_ν	−1.00000	3.95638	1.09313	.593250
$h_{\nu-1}$	2.00000	−1.00000	3.95638	1.09313
q_ν	−.500000	−3.95638	.276295	.542710
A_ν	−3.50000	78.9176	3.05912	4.15283
B_ν	18.0000	640.840	−49.8353	−18.7408
C_ν	126.000	−183.085	26.1451	8.70899
$q_{\nu+1}$	−3.95638	.276295	.542710	.526021
$x^{(\nu+1)}$	3.95638	5.04950	5.64275	5.95482
$K^{(\nu+1)}$	1.00000	.216482	.105135	.0524050

$\Rightarrow x^{(6)} = x_1^{(6)} = 5.95482 \approx x_1$ (die exakte Lösung ist $x_1 = 6$).

3. Schritt. (Berechnung des Deflationspolynoms mit Horner):

	P_3	1.00000	−7.00000	−36.0000	252.000	
$5.95482 = x_1^{(6)}$		0.00000	5.95482	−6.22386	−251.435	
	$\widetilde{P}_2$	1.00000	−1.04518	−42.2224	0.564652	$= P_3(x_1^{(6)})$

Der Rest wird vernachlässigt, im Allgemeinen ist er vernachlässigbar klein, wenn die Abbruchschranke ε entsprechend klein gesetzt wird, also mehr Iterationsschritte durchgeführt werden. Das Deflationspolynom $\widetilde{P}_2$ lautet hier:

$$\widetilde{P}_2(x) = x^2 - 1.04518x - 42.2224.$$

Die Nullstellen von P_2 würde man hier nicht mehr mit der Muller-Iteration, sondern mit der Lösungsformel für die quadratische Gleichung bestimmen.
Da jedoch das Deflationspolynom $\widetilde{P}_2$ hier wegen der geringen Anzahl der Iterationen in der Muller-Iteration zur Bestimmung von x_1 so ungenau ist, wird hier auf die Lösung der quadratischen Gleichung $\widetilde{P}_2(x) = 0$ verzichtet. Stattdessen wird das (exakte) Deflationspolynom P_2 zu $x_1 = 6$ bestimmt, es ergibt sich aus

P_3	1	−7	−36	252
$x_1 = 6$	0	6	−6	−252
P_2	1	−1	−42	0

zu $P_2(x) = x^2 - x - 42$.

Aus $P_2(x) = 0 \Rightarrow x_{2,3} = \frac{1}{2} \pm \sqrt{\frac{1}{4} + 42} = \frac{1}{2} \pm \frac{13}{2} = \begin{cases} 7 \\ -6 \end{cases} \Rightarrow x_2 = -6, \quad x_3 = 7$ (exakte Lösungen). □

Beispiel 3.9.

Gegeben: Tschebyscheff-Polynom 20. Grades. Alle ungeraden Koeffizienten (a_i mit ungeradem i) sind Null, die geraden haben folgende Werte:

i	a_i	i	a_i	i	a_i
0	1	8	549120	16	5570560
2	−200	10	−2050048	18	−2621440
4	6600	12	4659200	20	524288
6	−84480	14	−6553600		

Gesucht: Die Nullstellen mit Hilfe des Muller-Verfahrens.

Lösung: Alle Nullstellen sind reell (d. h. die Imaginäranteile sind alle Null):

Nr.	Realteil	Funktionswert
1	$7.8459095727844944 \cdot 10^{-2}$	$4.499439 \cdot 10^{-18}$
2	$-7.8459095727844944 \cdot 10^{-2}$	$4.499439 \cdot 10^{-18}$
3	$2.3344536385590542 \cdot 10^{-1}$	$-1.640398 \cdot 10^{-16}$
4	$-2.3344536385590531 \cdot 10^{-1}$	$-2.591243 \cdot 10^{-15}$
5	$3.8268343236508895 \cdot 10^{-1}$	$3.134358 \cdot 10^{-14}$
6	$-5.2249856471592970 \cdot 10^{-1}$	$-7.623839 \cdot 10^{-14}$
7	$-3.8268343236509078 \cdot 10^{-1}$	$-4.363090 \cdot 10^{-14}$
8	$6.4944804833011016 \cdot 10^{-1}$	$5.602117 \cdot 10^{-13}$
9	$5.2249856471595912 \cdot 10^{-1}$	$-5.736676 \cdot 10^{-14}$
10	$-8.5264016435600176 \cdot 10^{-1}$	$-1.822139 \cdot 10^{-11}$
11	$-7.6040596559929752 \cdot 10^{-1}$	$-1.313971 \cdot 10^{-11}$
12	$9.2387953251541677 \cdot 10^{-1}$	$1.544246 \cdot 10^{-10}$
13	$-6.4944804833034364 \cdot 10^{-1}$	$-5.771170 \cdot 10^{-12}$
14	$8.5264016435246492 \cdot 10^{-1}$	$6.954797 \cdot 10^{-11}$
15	$7.6040596560043328 \cdot 10^{-1}$	$1.432173 \cdot 10^{-11}$
16	$-9.7236992039993275 \cdot 10^{-1}$	$-4.494755 \cdot 10^{-11}$
17	$-9.2387953250848343 \cdot 10^{-1}$	$-2.059382 \cdot 10^{-10}$
18	$9.7236992039193171 \cdot 10^{-1}$	$3.899448 \cdot 10^{-10}$
19	$-9.9691733373235725 \cdot 10^{-1}$	$-9.982099 \cdot 10^{-11}$
20	$9.9691733373603186 \cdot 10^{-1}$	$1.128625 \cdot 10^{-09}$

□

3.3.3 Das Verfahren von Bauhuber

Das Verfahren von Bauhuber [BAUH1970] liefert sämtliche reellen und komplexen Nullstellen eines Polynoms P_n mit komplexen Koeffizienten.

Prinzip des Verfahrens

Zu einem beliebigen Startwert $x^{(0)}$ soll eine Folge von Näherungen $\{x^{(\nu)}\}$, $\nu = 1,2,\ldots$, so konstruiert werden, dass die zugehörige Folge der Beträge von P_n monoton fällt

$$|P_n(x^{(0)})| > |P_n(x^{(1)})| > \ldots .$$

Als Iterationsverfahren wird das Verfahren von Newton verwendet. Die Iteration wird abgebrochen, wenn z. B. die Abfrage $|P_n(x^{(\nu+1)})| < \varepsilon$ zu vorgegebenem $\varepsilon > 0$ erfüllt ist. Gilt für ein festes ν

$$|P_n(x^{(\nu)})| \le |P_n(x^{(\nu+1)})|, \tag{3.25}$$

so muss $x^{(\nu+1)}$ aus der Folge der $\{x^{(\nu)}\}$ ausgeschlossen werden. Mit einem zweidimensionalen Suchprozess, der als *Spiralisierung* bezeichnet und komplex durchgeführt wird, wird dann ein neues $x^{(\nu+1)}$ ermittelt, für das

$$|P_n(x^{(\nu+1)})| < |P_n(x^{(\nu)})|$$

gilt; damit wird die Iteration fortgesetzt. Die Folgen der Näherungswerte werden durch Extrapolation verbessert. Ist $x^{(N)}$ der ermittelte Näherungswert, so wird er als Nullstelle von P_n bezeichnet; man berechnet das Deflationspolynom $P_n(x)/(x - x^{(N)})$ mit dem Horner-Schema, vernachlässigt den Rest und wendet das eben beschriebene Verfahren auf das Restpolynom P_{n-1} vom Grad $n-1$ an. Analog fortfahrend erhält man alle Nullstellen des Polynoms P_n.

Grundgedanke der Spiralisierung

Es sei $x^{(\nu+1)}$ derjenige Wert der Folge $\{x^{(\nu)}\}$, für den erstmals (3.25) gilt. Dann muss innerhalb eines Kreises um $x^{(\nu)}$ mit dem Radius $r = |x^{(\nu+1)} - x^{(\nu)}|$ ein x_{s+1} existieren mit

$$|P_n(x_{s+1})| < |P_n(x^{(\nu)})|,$$

welches durch Absuchen des Kreisgebietes von außen nach innen mit einer Polygonspirale ermittelt wird. Dazu wird mit einem komplexen Faktor $q = q_1 + \mathrm{i}q_2$, q_1, q_2 reell, $|q| < 1$ gearbeitet, den Bauhuber $q = 0.1 + 0.9\mathrm{i}$ wählt (diese Wahl ist nicht bindend).

Algorithmus für die Spiralisierung

Mit den Startwerten

$$\begin{aligned} x_0 &:= x^{(\nu+1)} && \text{mit (3.25)} \\ \Delta x_0 &:= x^{(\nu+1)} - x^{(\nu)} \\ q &:= 0.1 + 0.9\mathrm{i}, \end{aligned}$$

werden zunächst für $k = 0$ nacheinander die folgenden Größen berechnet:

$$\left\{ \begin{aligned} \Delta x_{k+1} &= q\,\Delta x_k \\ x_{k+1} &= x^{(\nu)} + \Delta x_{k+1} \\ \Delta P_n &= |P_n(x^{(\nu)})| - |P_n(x_{k+1})|. \end{aligned} \right. \tag{3.26}$$

Im Falle $\Delta P_n \le 0$ wird k um eins erhöht und (3.26) erneut berechnet. Analog wird so lange fortgefahren, bis erstmals für ein $k = s$ gilt $\Delta P_n > 0$. Dann wird $x^{(\nu+1)}$ ersetzt durch x_{s+1} und damit nach Newton weiter iteriert.

Beispiel 3.10.

Gegeben: Polynom 20. Grades mit reellen Koeffizienten: $a_i = 2^i, i = 0(1)20$, also: $a_0 = 1, a_1 = 2, a_2 = 4, a_3 = 8, \ldots, a_{20} = 1\,048\,576$.

Gesucht: Die Nullstellen mit Hilfe des Bauhuber-Verfahrens.

Lösung: Die Nullstellen sind alle komplex:

Nr.	Realteil	Imaginärteil	Funktionswert
1	$3.7365046793212141 \cdot 10^{-2}$	$4.9860189859059006 \cdot 10^{-1}$	$3.255883 \cdot 10^{-16}$
2	$3.7365046793212134 \cdot 10^{-2}$	$-4.9860189859059006 \cdot 10^{-1}$	$2.053316 \cdot 10^{-16}$
3	$-4.9441541311256426 \cdot 10^{-1}$	$7.4521133088087207 \cdot 10^{-2}$	$3.318920 \cdot 10^{-16}$
4	$-2.5000000000000000 \cdot 10^{-1}$	$-4.3301270189221935 \cdot 10^{-1}$	$7.412270 \cdot 10^{-16}$
5	$4.1311938715799745 \cdot 10^{-1}$	$-2.8166002903181098 \cdot 10^{-1}$	$2.094378 \cdot 10^{-15}$
6	$1.8267051218319749 \cdot 10^{-1}$	$4.6543687432210207 \cdot 10^{-1}$	$2.147101 \cdot 10^{-15}$
7	$-4.9441541311256426 \cdot 10^{-1}$	$-7.4521133088087207 \cdot 10^{-2}$	$3.318920 \cdot 10^{-16}$
8	$4.1311938715799745 \cdot 10^{-1}$	$2.8166002903181109 \cdot 10^{-1}$	$5.020949 \cdot 10^{-15}$
9	$-3.6652593591491317 \cdot 10^{-1}$	$3.4008636888545973 \cdot 10^{-1}$	$9.204744 \cdot 10^{-16}$
10	$3.1174490092936680 \cdot 10^{-1}$	$-3.9091574123401490 \cdot 10^{-1}$	$1.915832 \cdot 10^{-15}$
11	$-2.4999999999999997 \cdot 10^{-1}$	$4.3301270189221935 \cdot 10^{-1}$	$7.478286 \cdot 10^{-16}$
12	$-3.6652593591491323 \cdot 10^{-1}$	$-3.4008636888545973 \cdot 10^{-1}$	$1.429672 \cdot 10^{-15}$
13	$4.7778640289307039 \cdot 10^{-1}$	$1.4737758720545208 \cdot 10^{-1}$	$5.395137 \cdot 10^{-15}$
14	$-4.5048443395120957 \cdot 10^{-1}$	$-2.1694186955877898 \cdot 10^{-1}$	$1.904520 \cdot 10^{-15}$
15	$1.8267051218319749 \cdot 10^{-1}$	$-4.6543687432210218 \cdot 10^{-1}$	$2.276995 \cdot 10^{-15}$
16	$-1.1126046697815717 \cdot 10^{-1}$	$4.8746395609091175 \cdot 10^{-1}$	$1.536479 \cdot 10^{-15}$
17	$-1.1126046697815721 \cdot 10^{-1}$	$-4.8746395609091181 \cdot 10^{-1}$	$1.417072 \cdot 10^{-16}$
18	$3.1174490092936669 \cdot 10^{-1}$	$3.9091574123401496 \cdot 10^{-1}$	$4.364151 \cdot 10^{-15}$
19	$4.7778640289307039 \cdot 10^{-1}$	$-1.4737758720545216 \cdot 10^{-1}$	$9.020717 \cdot 10^{-15}$
20	$-4.5048443395120963 \cdot 10^{-1}$	$2.1694186955877906 \cdot 10^{-1}$	$1.427738 \cdot 10^{-15}$

□

3.3.4 Das Verfahren von Jenkins und Traub

Das Verfahren von Jenkins und Traub ([JENK1970], [TRAU1966]) ist ein Iterationsverfahren zur Ermittlung der betragskleinsten Nullstelle eines Polynoms P_n mit komplexen Koeffizienten. Es ist für alle Startwerte $x^{(0)} \in [-\infty,\ |x_i|_{\min}]$ global konvergent von mindestens zweiter Ordnung. Es behandelt auch den Fall von zwei oder mehr betragsgleichen Nullstellen. Je nachdem, ob die betragskleinste Nullstelle einfach, zweifach oder mehr als zweifach ist, wird der vom Computer auszuführende Algorithmus automatisch durch entsprechend eingebaute logische Entscheidungen modifiziert. Nachdem die betragskleinste(n) Nullstelle(n) näherungsweise ermittelt ist (sind), wird durch Abdividieren der Nullstelle(n) das Restpolynom bestimmt. Hiervon liefert das gleiche Verfahren eine Näherung für die nächste(n) Nullstelle(n).

3.4 Anwendungsbeispiel

Beispiel 3.11.

Gegeben: Für die im Beispiel 43 (Turbinenbeispiel) hergeleitete kubische Gleichung ergibt sich mit den Daten für h, A, P in einem konkreten Fall

$$P_3(\dot{V}) = \dot{V}^3 - 11144.16 \cdot \dot{V} + 44233.6 = 0\,.$$

Gesucht: Der optimale Volumenstrom $\dot{V}$, der mit Hilfe der Muller-Iteration berechnet werden soll. Für die Abbruchbedingung wähle man die Schranke $\epsilon = 0.00001$, damit nur wenige, mit dem Taschenrechner nachvollziehbare Schritte, durchgeführt werden müssen.

Lösung:

1. Schritt. (Bestimmung der Startwerte): Man setzt $a_0 = 44233.6$, $a_1 = -11144.16$ und $a_2 = 0$ und verwendet die künstlichen Funktionswerte (3.19), d. h.

$$\begin{array}{lcl} f_0 & := & a_0 - a_1 = 55377.76 \\ f_1 & := & a_0 + a_1 = 33089.44 \\ f_2 & := & a_0 = 44233.6 \end{array}$$

2. Schritt (Muller-Iteration): Mit der Abbruchbedingung

$$\frac{|\dot{V}^{(\nu+1)} - \dot{V}^{(\nu)}|}{|\dot{V}^{(\nu+1)}|} = K^{(\nu+1)} < \epsilon = 0.00001$$

erhält man folgende Werte in der Tabelle bei 10-stelliger Mantisse

	$\nu = 2$	$\nu = 3$	$\nu = 4$
$\dot{V}^{(\nu)}$	0.000000000	3.969165082	3.974847874
$\dot{V}^{(\nu-1)}$	1.000000000	0.000000000	3.969165082
$\dot{V}^{(\nu-2)}$	−1.000000000	1.000000000	0.000000000
f_ν	44233.60000	−63.12056400	0.059588900
$f_{\nu-1}$	33089.44000	44233.60000	63.12056400
$f_{\nu-2}$	55377.76000	33089.44000	44233.60000
h_ν	−1.000000000	3.969165082	0.00568279211
$h_{\nu-1}$	2.000000000	−1.000000000	3.969165082
q_ν	−0.500000000	−3.969165082	0.00143173488
A_ν	−0.025000000	−248.1970680	0.00025691218
B_ν	5572.055000	130901.2481	−63.15100478
C_ν	22116.80000	−187.4153746	0.05967421551
$q_{\nu+1}$	−3.969165082	0.00143173488	0.00094494483
$\dot{V}^{(\nu+1)}$	3.969165082	3.974847874	3.974853244
$K^{(\nu+1)}$	1.000000000	0.001442968792	0.00000135097

Die kleinste Nullstelle und somit der optimale Volumenstrom ist erreicht bei $\dot{V}^{(5)} = 3.974853244$. □

3.5 Entscheidungshilfen

Die Entscheidungshilfen für die Auswahl eines geeigneten Verfahrens zur Nullstellenbestimmung bei algebraischen Polynomen sind mit Vorbemerkungen und einem Überblick in Abschnitt 3.3.1 zusammengefasst.

Ergänzende Literatur zu Kapitel 3

[BART2001] 4.3.6; [BOHM1985]; [BRON1991], 2.4.2; [CARN1990], 3.2-3.4, 3.9; [COLL1973], I, 1.3-1.4; [CONT1987], 3; [FORD1977]; [HAMM1994], Kap.8; [ISAA1973], 3.3-3.4; [OPFE2002], Kap.2; [QUAR2002] Bd.1, Kap.6; [RALS2001], 8; [STOE1999]; [WERN1993], II §§3,6-8.

Kapitel 4

Direkte Verfahren zur Lösung linearer Gleichungssysteme

4.1 Aufgabenstellung und Motivation

Man unterscheidet *direkte* und *iterative* Methoden zur numerischen Lösung linearer Gleichungssysteme. Die direkten Methoden liefern die exakte Lösung, sofern man von Rundungsfehlern absieht. Die iterativen Methoden gehen von einer Anfangsnäherung für die Lösung (dem sogenannten Startvektor) aus und verbessern diese schrittweise; sie werden in Kapitel 5 behandelt.

Zu den direkten Methoden gehören der Gaußsche Algorithmus, das Gauß-Jordan-Verfahren, das Verfahren von Cholesky, die Verfahren für Systeme mit Bandmatrizen, die Methode des Pivotisierens und andere.

Zu den iterativen Methoden gehören das Iterationsverfahren in Gesamtschritten, das Iterationsverfahren in Einzelschritten und die Relaxationsverfahren.

Gegeben sei ein System von m linearen Gleichungen mit n Unbekannten x_i der Form

$$\left\{\begin{array}{lll} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &=& a_1, \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &=& a_2, \\ \vdots & & \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n &=& a_m, \end{array}\right. \tag{4.1}$$

wobei die Koeffizienten $a_{ik} \in \mathbb{R}$ und die rechten Seiten $a_i \in \mathbb{R}$, $i = 1(1)m$, $k = 1(1)n$, vorgegebene Zahlen sind. In Matrizenschreibweise lautet (4.1)

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}, \quad \boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{a} \in \mathbb{R}^m \tag{4.2}$$

mit

$$\boldsymbol{A} = (a_{ik}) = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}, \quad \boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \quad \boldsymbol{a} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix}.$$

Ein Vektor $\boldsymbol{x}$, dessen Komponenten x_i, $i = 1(1)n$, jede der m Gleichungen des Systems (4.1) zu einer Identität machen, heißt Lösungsvektor oder kurz Lösung von (4.1) bzw. (4.2). In Abschnitt 4.13 werden überbestimmte Systeme $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}$ mit (m, n)-Matrix $\boldsymbol{A}$, $\boldsymbol{a} \in \mathbb{R}^m$, $\boldsymbol{x} \in \mathbb{R}^n$, $m > n$ behandelt, sonst nur Systeme aus n Gleichungen mit n Unbekannten, d. h. der Fall $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}$ mit (n, n)-Matrix $\boldsymbol{A}$, $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{a} \in \mathbb{R}^n$.

Systeme linearer Gleichungen (im Folgenden kurz *lineare Gleichungssysteme* genannt) treten überall in den Anwendungen in Physik, Technik, Betriebswirtschaftslehre usw. auf; man begegnet ihnen z. B. in der Statik bei der Berechnung statisch unbestimmter Systeme, in der Elektrotechnik bei der Berechnung von Netzwerken (z. B. Bestimmung zugehöriger Ströme zu vorgegebenen Gleichspannungen und vorgegebenen Ohmschen Widerständen) und bei der Behandlung von Eigenwertproblemen der mathematischen Physik. Wie sich noch zeigen wird, gibt es auch eine Reihe von Problemen in der numerischen Mathematik, deren Behandlung auf Systeme linearer Gleichungen führt. So treten z. B. beim Newtonschen Verfahren für nichtlineare Gleichungssysteme bei jedem Schritt lineare Gleichungssysteme auf; die Methode der kleinsten Quadrate von Gauß (Ausgleichsrechnung) und die numerische Lösung von Randwertproblemen bei gewöhnlichen und partiellen Differentialgleichungen mit Hilfe von Differenzenverfahren führen auf lineare Gleichungssysteme. In der Betriebswirtschaftslehre spielen lineare Gleichungssysteme im Zusammenhang mit linearer Programmierung eine große Rolle.

Theoretisch ist das Problem der Auflösung linearer Gleichungssysteme vollständig geklärt. So lassen sich die Lösungen z. B. mit der Cramerschen Regel explizit berechnen; dieser Weg erfordert jedoch einen immensen Rechenaufwand. Zur Lösung eines Systems von n Gleichungen mit n Unbekannten erfordert die Cramersche Regel $(n^2-1)\,n! + n$ Punktoperationen; dies sind 359 251 210 Punktoperationen für $n = 10$ und bereits 10^{21} für $n = 20$. Daher ist die Cramersche Regel (schon ab $n = 3$) für praktische Rechnungen total unbrauchbar. Man muss also nach anderen, effektiveren Lösungsmöglichkeiten suchen. Ein solches Verfahren ist z. B. der Gaußsche Algorithmus, der für die Lösung von n Gleichungen mit n Unbekannten nur $\frac{n}{3}(n^2 + 3n - 1)$ Punktoperationen benötigt, d. h. im Fall $n = 10$ nur 430, im Fall $n = 20$ nur 3060 (gegenüber 10^{21} bei der Cramerschen Regel).

Anzahl der erforderlichen Punktoperationen
(Multiplikationen, Divisionen)

Zahl der Unbekannten	Cramersche Regel	Gaußscher Algorithmus
n	$(n^2-1)\,n! + n$	$n\,(n^2 + 3n - 1)/3$
2	8	6
3	51	17
4	364	36
5	2885	65
10	359 251 210	430
20	10^{21}	3060

Rechenzeit
(3 ns im Durchschnitt pro Multiplikation/Division)

Zahl der Unbekannten	Cramersche Regel	Gaußscher Algorithmus
10	1.08 sec	$\approx 10^{-6}$ sec
20	95 000 Jahre	$\approx 10^{-5}$ sec

Es gibt also brauchbare numerische Methoden. Trotz dieser Tatsache ist die praktische Bestimmung der Lösung für sehr große n eine problematische numerische Aufgabe; die Gründe hierfür sind:

1. Arbeitsaufwand (Rechenzeit) und Speicherplatzbedarf bei sehr großen Systemen, siehe Beispiele
2. Verfälschung der Ergebnisse durch Rundungsfehler (numerische Instabilität)
3. Schlecht konditionierte Probleme (mathematische Instabilität)

Im Folgenden sollen nicht nur Lösungsmethoden angegeben, sondern auch auf numerische Effekte, anfallenden Arbeitsaufwand u.a.m. aufmerksam gemacht werden. Von einem Lösungsverfahren wird gefordert, dass es Attribute wie Bandstruktur und Symmetrie berücksichtigt, rechenzeitoptimal ist, Rundungsfehler unter Kontrolle gehalten werden u.a.m.

Beispiele, bei denen große lineare Gleichungssysteme auftreten

1. LKW-Lagerbock (Beispiel der Fa. MEC Alsdorf):
   Pro LKW gibt es zwei Lagerböcke; die Blattfeder wird mit beiden Lagerböcken am Rahmen befestigt. Wenn sich die Feder z. B. bei einer Bodenwelle bewegt, tritt ein Moment auf (Abb. 4.1).
   Bei FEM-Berechnungen, die die mechanische Belastung und Temperaturberechnung betreffen, treten bei linearem Ansatz und bei Gleichungssystemen z. B. 4320 Gleichungen auf (siehe Finites Netz Abb. 4.2).

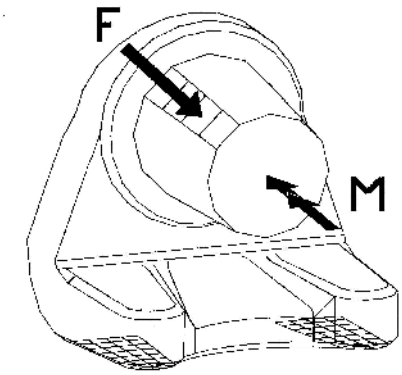


Abb. 4.1.

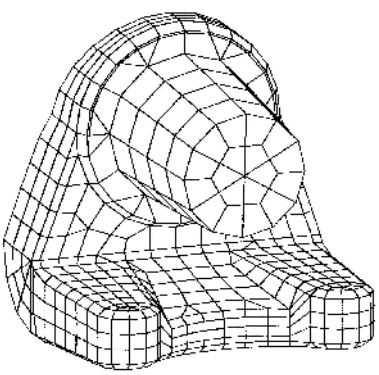


Abb. 4.2.

2. Parabolantenne der Firma Krupp mit 100 m Durchmesser am oberen Rand:

   Es handelt sich dabei um einen räumlichen Verbund aus Stäben und Balken, die geometrisch ein Rotationsparaboloid bilden. Die Berechnung muss so erfolgen, dass bei Verformung durch Neigung und Eigengewicht wegen der Richtgenauigkeit der Antenne immer wieder ein Rotationsparaboloid entsteht. Es sind jeweils ca. 5000 Gleichungen mit 5000 Unbekannten zu lösen.

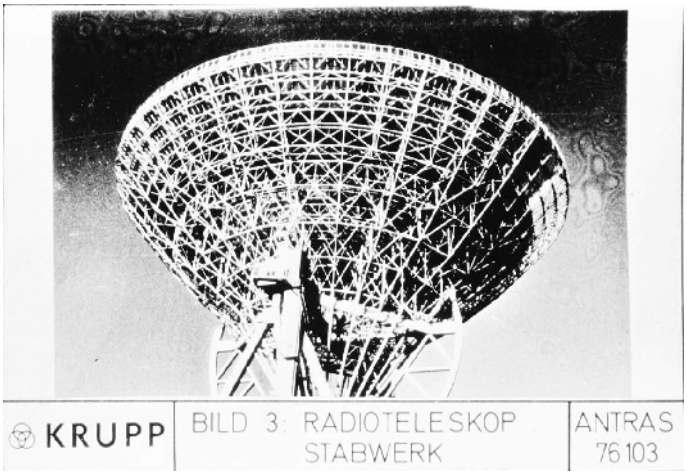


Abb. 4.3.

3. Beispiel des Aerodynamischen Instituts der RWTH Aachen. Numerische Simulation einer ablösenden Strömung um Tragflügelprofile, gerechnet mit den Navier-Stokes-Gleichungen:

   Wenn 3-dimensional gerechnet wird und ein $(31 \times 31 \times 51)$-Gitter mit je 4 Gleichungen verwendet wird, so erhält man nichtlineare Systeme aus 196 044 Gleichungen mit 196 044 Unbekannten, die iterativ (etwa mit 5 Iterationen) gelöst werden. Rechnet man bis zum Wirbelablösen 10 000 Zeitschritte, so ergeben sich $5 \times 10\,000 = 50\,000$ lineare Gleichungssysteme aus rund ca. 200 000 Gleichungen, die zu lösen sind.

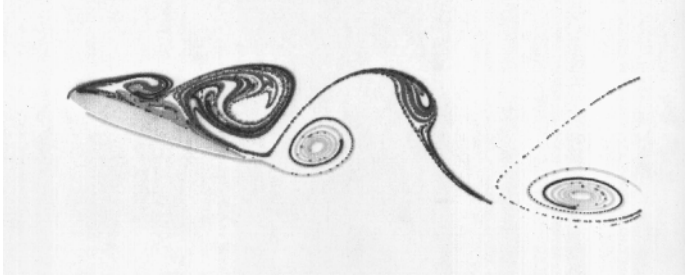


Abb. 4.4.

4. Beispiel des Instituts für Bildsame Formgebung der RWTH Aachen zum Freiformschmieden: Ein 8-kantiger Schmiedeblock aus Stahl wird mit 4 Hämmern bearbeitet, dann um 45° gedreht und wieder bearbeitet usw. Ziel ist die Berechnung des Temperaturfeldes. Hier entstehen bei 1400 Knoten Gleichungssysteme von 4200 Gleichungen. Bei 50 Zeitschritten und 5 Iterationen pro Zeitschritt hat man also 250 lineare Gleichungssysteme mit je 4200 Gleichungen zu lösen.

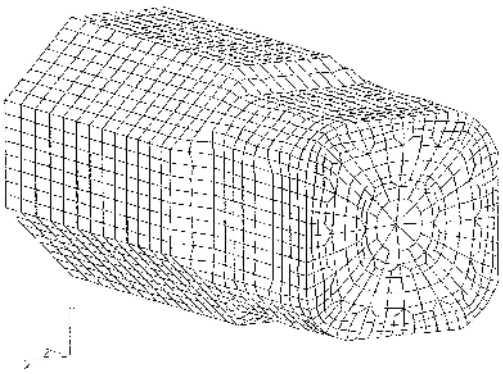


Abb. 4.5.

Betrachtet man einen Rundblock mit 200 mm Durchmesser und 250 mm Länge, so entstehen bei 3000 Knoten 9000 nichtlineare Gleichungen, d. h. bei 100 Zeitschritten mit 5 Iterationen pro Zeitschritt 500 lineare Systeme mit je 9000 Gleichungen.

5. Ein weiteres Finite-Element-Beispiel aus dem Institut für Bildsame Formgebung der RWTH Aachen: Bei der Simulation des Fließpress-Verfahrens zur Herstellung eines Zahnrades mit zwölf Zähnen wird unter Ausnutzung der Symmetriebedingungen mit dem Modell eines halben Zahnes gerechnet. In diesem Beispiel wird dazu ein Netz mit 2911 Knoten erstellt. Man erhält unter Berücksichtigung aller Randbedingungen insgesamt 7560 nichtlineare Gleichungen, die iterativ gelöst werden. Dabei tritt eine Bandmatrix auf.

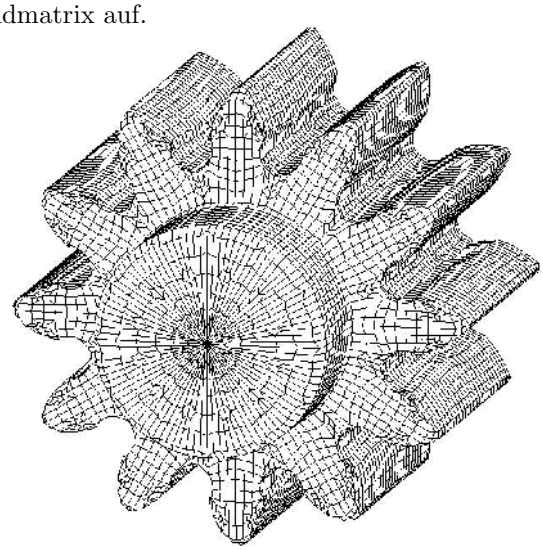


Abb. 4.6.

4.2 Definitionen und Sätze

In diesem Abschnitt werden einige Definitionen und Sätze über Matrizen und Determinanten zusammengestellt, die im weiteren Verlauf dieses Kapitels für numerische Verfahren benötigt werden.

Definition 4.1. (*Hauptabschnittsmatrix, Hauptabschnittsdeterminante*)
Die (k,k)-Matrix $\boldsymbol{A}_k$, die aus den ersten k Zeilen und k Spalten von $\boldsymbol{A} = (a_{ik})$, $i,k = 1(1)n$, gebildet wird, heißt *Hauptabschnittsmatrix*. Ihre Determinante $\det(\boldsymbol{A}_k)$ heißt *Hauptabschnittsdeterminante* der Ordnung k.

Beispiel 4.2.

$$\boldsymbol{A} = \begin{pmatrix} 1 & -2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & -9 \end{pmatrix} \quad \text{vom Typ } (3,3)$$

$$\boldsymbol{A}_1 = (1) \quad \text{mit } \det(\boldsymbol{A}_1) = 1, \text{ Hauptabschnittsdeterminante der Ordnung 1}$$

$$\boldsymbol{A}_2 = \begin{pmatrix} 1 & -2 \\ 4 & 5 \end{pmatrix} \quad \text{mit } \det(\boldsymbol{A}_2) = 13, \text{ Hauptabschnittsdeterminante der Ordnung 2}$$

$$\boldsymbol{A}_3 = \boldsymbol{A} \quad \text{mit } \det(\boldsymbol{A}_3) = -258, \text{ Hauptabschnittsdeterminante der Ordnung 3 gleich Determinante der Matrix } \boldsymbol{A}$$

□

Definition 4.3. (*Regulär, streng regulär*)
Eine Matrix $\boldsymbol{A} = (a_{ik})$, $i,k = 1(1)n$, heißt *regulär*, falls $\det(\boldsymbol{A}) \neq 0$ gilt; sie heißt *streng regulär*, wenn alle Hauptabschnittsdeterminanten von Null verschieden sind:

$$\det(\boldsymbol{A}_k) \neq 0 \quad \text{für alle} \quad k = 1(1)n.$$

Beispiel 4.4.
Die Matrix $\boldsymbol{A}$ aus dem Beispiel 4.2 ist regulär und zugleich streng regulär, während

$$\boldsymbol{A} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \text{ mit } \det(\boldsymbol{A}) = 0$$

nicht regulär ist. □

Definition 4.5. (*Untere Dreiecksmatrix, obere Dreiecksmatrix*)
Eine (n,n)-Matrix $\boldsymbol{L} = (l_{ik})$ heißt *untere Dreiecksmatrix* (Subdiagonalmatrix), wenn $l_{ik} = 0$ für $k > i$ gilt; sie heißt *normierte untere Dreiecksmatrix*, wenn außerdem $l_{ii} = 1$ für alle i ist.
Eine (n,n)-Matrix $\boldsymbol{R} = (r_{ik})$ heißt *obere Dreiecksmatrix* (Superdiagonalmatrix), wenn $r_{ik} = 0$ für $i > k$ gilt; sie heißt *normierte obere Dreiecksmatrix*, wenn außerdem $r_{ii} = 1$ für alle i ist.

Beispiel 4.6.

$$\boldsymbol{L}_1 = \begin{pmatrix} 1 & 0 & 0 \\ 3 & 2 & 0 \\ -3 & 5 & 7 \end{pmatrix} \qquad \boldsymbol{R}_1 = \begin{pmatrix} 5 & 7 & -8 \\ 0 & 1 & 3 \\ 0 & 0 & 4 \end{pmatrix}$$

untere Dreiecksmatrix — obere Dreiecksmatrix

$$\boldsymbol{L}_2 = \begin{pmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ -3 & 5 & 1 \end{pmatrix} \qquad \boldsymbol{R}_2 = \begin{pmatrix} 1 & 7 & -8 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{pmatrix}$$

normierte untere Dreiecksmatrix — normierte obere Dreiecksmatrix

□

Definition 4.7. (*Permutationsmatrix*)
Eine (n,n)-Matrix $\boldsymbol{P}$ heißt *Permutationsmatrix*, wenn in jeder Zeile und Spalte genau eine Eins und $n-1$ Nullen vorkommen.

Eine (n,n)-Permutationsmatrix $\boldsymbol{P}$ entsteht aus der (n,n)-Einheitsmatrix $\boldsymbol{E}$ dadurch, dass man in $\boldsymbol{E}$ die i-te und k-te Zeile vertauscht. Dann ist $\boldsymbol{PA}$ diejenige Matrix, die aus $\boldsymbol{A}$ durch Vertauschung der i-ten und k-ten Zeile hervorgeht.

Beispiel 4.8.

$$\boldsymbol{E} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \qquad \boldsymbol{P} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

aus $\boldsymbol{E}$ durch Vertauschung der 1. und 2. Zeile

$$\boldsymbol{PA} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & -2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & -9 \end{pmatrix} = \begin{pmatrix} 4 & 5 & 6 \\ 1 & -2 & 3 \\ 7 & 8 & -9 \end{pmatrix}$$

□

Satz 4.9.
Jede (n,n)-Matrix $\boldsymbol{A}$ mit $\det(\boldsymbol{A}_k) \neq 0$ für $k = 1(1)n-1$ kann ohne Zeilenvertauschungen eindeutig in das Produkt $\boldsymbol{LR}$ aus einer normierten unteren Dreiecksmatrix $\boldsymbol{L}$ und einer oberen Dreiecksmatrix $\boldsymbol{R}$ zerlegt werden:

$$\boldsymbol{L} = \begin{pmatrix} 1 & & & \\ l_{21} & 1 & & \\ \vdots & & \ddots & \\ l_{n1} & l_{n2} & \cdots & 1 \end{pmatrix}, \quad \boldsymbol{R} = \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ & r_{22} & \cdots & r_{2n} \\ & & \ddots & \vdots \\ & & & r_{nn} \end{pmatrix}.$$

Diese Dreieckszerlegung (Faktorisierung) wird als LR-Zerlegung bezeichnet.

Beispiel 4.10.

Gegeben: $\boldsymbol{A} = \begin{pmatrix} 1 & -1 & -1 \\ -2 & 6 & 3 \\ -1 & 13 & 6 \end{pmatrix}$ vom Typ $(3,3)$.

Es gilt $\det(\boldsymbol{A}_1) = 1 \neq 0$, $\det(\boldsymbol{A}_2) = 4 \neq 0 \Rightarrow$ die LR-Zerlegung existiert eindeutig.

$\boldsymbol{LR}$-Zerlegung von $\boldsymbol{A}$:

$$\boldsymbol{A} = \begin{pmatrix} 1 & -1 & -1 \\ -2 & 6 & 3 \\ -1 & 13 & 6 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -1 & 3 & 1 \end{pmatrix} \begin{pmatrix} 1 & -1 & -1 \\ 0 & 4 & 1 \\ 0 & 0 & 2 \end{pmatrix} = \boldsymbol{LR}$$

Wie macht man das schrittweise?

$$\boldsymbol{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} = \begin{pmatrix} 1 & -1 & -1 \\ -2 & 6 & 3 \\ -1 & 13 & 6 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ 0 & r_{22} & r_{23} \\ 0 & 0 & r_{33} \end{pmatrix} = \boldsymbol{LR}$$

Die Multiplikation der beiden Matrizen $\boldsymbol{L}$ und $\boldsymbol{R}$ rechts und ein Koeffizientenvergleich mit der Matrix $\boldsymbol{A}$ links ergibt **zeilenweise** die zu berechnenden Koeffizienten der Dreiecksmatrizen $\boldsymbol{L}$ und $\boldsymbol{R}$.

1. Zeile:	$a_{11} =$	1	$\stackrel{!}{=}$	$1 \cdot r_{11}$	$\Rightarrow$	r_{11}	$=$	1
	$a_{12} =$	-1	$=$	$1 \cdot r_{12}$	$\Rightarrow$	r_{12}	$=$	-1
	$a_{13} =$	-1	$=$	$1 \cdot r_{13}$	$\Rightarrow$	r_{13}	$=$	-1
2. Zeile:	$a_{21} =$	-2	$=$	$l_{21} \cdot r_{11} + 1 \cdot 0$	$\Rightarrow$	l_{21}	$=$	-2
	$a_{22} =$	6	$=$	$l_{21} \cdot r_{12} + 1 \cdot r_{22}$	$\Rightarrow$	r_{22}	$=$	4
	$a_{23} =$	3	$=$	$l_{21} \cdot r_{13} + 1 \cdot r_{23} + 0 \cdot r_{33}$	$\Rightarrow$	r_{23}	$=$	1
3. Zeile:	$a_{31} =$	-1	$=$	$l_{31} \cdot r_{11}$	$\Rightarrow$	l_{31}	$=$	-1
	$a_{32} =$	13	$=$	$l_{31} \cdot r_{12} + l_{32} \cdot r_{22}$	$\Rightarrow$	l_{32}	$=$	$\frac{1}{4}(13-1) = 3$
	$a_{33} =$	6	$=$	$l_{31} \cdot r_{13} + l_{32} \cdot r_{23} + r_{33}$	$\Rightarrow$	r_{33}	$=$	$6 - 1 - 3 = 2$

□

Analog ergeben sich die Formeln im folgenden

Algorithmus 4.11.
Die Elemente der Matrizen $\boldsymbol{L}$ und $\boldsymbol{R}$ gemäß Dreieckszerlegung nach Satz 4.9 werden wie folgt berechnet:

1. Für $k = 1(1)n$: $\quad r_{1k} := a_{1k}$
2. Für jedes $i = 2(1)n$

   2.1 Für $k = 1$: $\quad l_{i1} := \dfrac{a_{i1}}{a_{11}}$

   2.2 Für $k = 2(1)i{-}1$: $\quad l_{ik} := (a_{ik} - \sum\limits_{j=1}^{k-1} l_{ij}\, r_{jk})/r_{kk}$

   2.3 Für $k = i(1)n$: $\quad r_{ik} := a_{ik} - \sum\limits_{j=1}^{i-1} l_{ij}\, r_{jk}$

Werden für eine Dreieckszerlegung Zeilenvertauschungen zugelassen, so kann die Voraussetzung $\det(\boldsymbol{A}_k) \neq 0$ für $k = 1(1)n{-}1$ in Satz 4.9 entfallen; es gilt dann der

Satz 4.12.
Für eine (n,n)-Matrix $\boldsymbol{A}$ mit $\det \boldsymbol{A} \neq 0$ gilt mit einer (n,n)-Permutationsmatrix $\boldsymbol{P}$ die Zerlegung

$$\boldsymbol{PA} = \boldsymbol{LR},$$

wobei $\boldsymbol{L}$ und $\boldsymbol{R}$ durch $\boldsymbol{P}$ und $\boldsymbol{A}$ eindeutig bestimmt sind.
In $\boldsymbol{PA}$ sind die Zeilen von $\boldsymbol{A}$ permutiert. Es gilt mit $\det(\boldsymbol{P}) = (-1)^k$, k = Anzahl der Zeilenvertauschungen,

$$\det \boldsymbol{A} = (-1)^k \det \boldsymbol{R} = (-1)^k r_{11} r_{22} \ldots r_{nn}.$$

Jede reguläre Matrix $\boldsymbol{A}$ kann somit durch Linksmultiplikation mit einer Permutationsmatrix $\boldsymbol{P}$ (also durch Zeilenvertauschungen) in eine streng reguläre Matrix $\boldsymbol{PA}$ transformiert werden.

Beispiel 4.13.
Für die Matrix

$$\boldsymbol{A} = \begin{pmatrix} 1 & -1 & -1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \end{pmatrix}, \quad \det(\boldsymbol{A}_2) = 0$$

existiert keine LR-Zerlegung ohne Zeilenvertauschung nach Satz 4.9. Nach Vertauschung

der 2. und der 3. Zeile von $\boldsymbol{A}$ lässt sich eine eindeutige LR-Zerlegung angeben, denn mit

$$\boldsymbol{PA} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & -1 & -1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & -1 & -1 \\ -1 & -1 & 1 \\ -1 & 1 & -1 \end{pmatrix},$$

gilt

$$\boldsymbol{PA} = \boldsymbol{LR} = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & -1 & -1 \\ 0 & -2 & 0 \\ 0 & 0 & -2 \end{pmatrix}$$

□

Definition 4.14. (*Transponierte Matrix*)
Es sei $\boldsymbol{A} = (a_{ik})$, $i = 1(1)m$, $k = 1(1)n$, eine Matrix vom Typ (m, n). Dann heißt die Matrix $\boldsymbol{A}^\mathsf{T} = (a_{ki})$, $k = 1(1)n$, $i = 1(1)m$, die aus $\boldsymbol{A}$ durch Vertauschung von Zeilen und Spalten entsteht, die zu $\boldsymbol{A}$ *transponierte* Matrix.

Beispiel 4.15.

$$\boldsymbol{A} = \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 4 & 6 \end{pmatrix} \text{ vom Typ } (3,2) \quad \Rightarrow \quad \boldsymbol{A}^\mathsf{T} = \begin{pmatrix} 1 & 3 & 4 \\ 2 & 4 & 6 \end{pmatrix} \text{ vom Typ } (2,3)$$

□

Definition 4.16. (*Symmetrische Matrix*)
Gilt für eine (n, n)-Matrix $\boldsymbol{A} = (a_{ik})$, $i, k = 1(1)n$,

$$a_{ik} = a_{ki}$$

für alle i, k, so heißt $\boldsymbol{A}$ *symmetrisch*; es gilt $\boldsymbol{A} = \boldsymbol{A}^\mathsf{T}$, wobei $\boldsymbol{A}^\mathsf{T}$ die zu $\boldsymbol{A}$ transponierte Matrix ist.

Beispiel 4.17.

$$\boldsymbol{A} = \begin{pmatrix} 1 & 2 & -3 \\ 2 & 4 & 5 \\ -3 & 5 & 6 \end{pmatrix} = \boldsymbol{A}^\mathsf{T} \quad \Rightarrow \quad \boldsymbol{A} \text{ ist symmetrisch}$$

□

Definition 4.18. (*Orthogonale Matrix*)
Eine reelle (n, n)-Matrix $\boldsymbol{Q}$ heißt *orthogonal*, falls mit der Einheitsmatrix $\boldsymbol{E}$ vom Typ (n, n) gilt

$$\boldsymbol{Q}^\mathsf{T}\boldsymbol{Q} = \boldsymbol{Q}\boldsymbol{Q}^\mathsf{T} = \boldsymbol{E} \quad \text{bzw.} \quad \boldsymbol{Q}^{-1} = \boldsymbol{Q}^\mathsf{T}.$$

Beispiel 4.19.

$$\boldsymbol{Q}_1 = \begin{pmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{pmatrix}, \qquad \boldsymbol{Q}_1^{\mathsf{T}} = \begin{pmatrix} \cos\alpha & 0 & -\sin\alpha \\ 0 & 1 & 0 \\ \sin\alpha & 0 & \cos\alpha \end{pmatrix}$$

$$\boldsymbol{Q}_1 \cdot \boldsymbol{Q}_1^{\mathsf{T}} = \boldsymbol{Q}_1^{\mathsf{T}} \cdot \boldsymbol{Q}_1 = \boldsymbol{E} \quad \Rightarrow \quad \boldsymbol{Q}_1 \text{ orthogonal}$$

$$\boldsymbol{Q}_2 = \frac{1}{\sqrt{5}} \begin{pmatrix} 2 & -1 \\ 1 & 2 \end{pmatrix}, \qquad \boldsymbol{Q}_2^{\mathsf{T}} = \frac{1}{\sqrt{5}} \begin{pmatrix} 2 & 1 \\ -1 & 2 \end{pmatrix},$$

$$\Rightarrow \quad \boldsymbol{Q}_2 \cdot \boldsymbol{Q}_2^{\mathsf{T}} = \frac{1}{5} \begin{pmatrix} 2 & -1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} 2 & 1 \\ -1 & 2 \end{pmatrix} = \frac{1}{5} \begin{pmatrix} 5 & 0 \\ 0 & 5 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \boldsymbol{E}$$

□

Satz 4.20.
Sei $\boldsymbol{v} \in \mathbb{R}^n$ ein Vektor und $\boldsymbol{E}$ die (n,n)-Einheitsmatrix. Dann ist

$$\boldsymbol{H} := \boldsymbol{E} - \frac{2}{\|\boldsymbol{v}\|^2} \boldsymbol{v}\boldsymbol{v}^{\mathsf{T}}$$

eine symmetrische, orthogonale (n,n)-Matrix (Householder-Matrix), d. h. es gilt $\boldsymbol{H}^{\mathsf{T}}\boldsymbol{H} = \boldsymbol{H}^2 = \boldsymbol{E}$.

Beispiel 4.21.

$$\boldsymbol{v} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \quad \|\boldsymbol{v}\|^2 = \boldsymbol{v}^{\mathsf{T}}\,\boldsymbol{v} = (2,1) \begin{pmatrix} 2 \\ 1 \end{pmatrix} = 5$$

$$\boldsymbol{v}\,\boldsymbol{v}^{\mathsf{T}} = \begin{pmatrix} 2 \\ 1 \end{pmatrix} (2,1) = \begin{pmatrix} 4 & 2 \\ 2 & 1 \end{pmatrix}$$

$$\boldsymbol{H} := \boldsymbol{E} - \frac{2}{\|\boldsymbol{v}\|^2} \boldsymbol{v}\,\boldsymbol{v}^{\mathsf{T}} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \tfrac{2}{5} \begin{pmatrix} 4 & 2 \\ 2 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} \frac{8}{5} & \frac{4}{5} \\ \frac{4}{5} & \frac{2}{5} \end{pmatrix}$$

$$= \begin{pmatrix} 1-\frac{8}{5} & -\frac{4}{5} \\ -\frac{4}{5} & 1-\frac{2}{5} \end{pmatrix} = \begin{pmatrix} -\frac{3}{5} & -\frac{4}{5} \\ -\frac{4}{5} & \frac{3}{5} \end{pmatrix}$$

$$\boldsymbol{H}^{\mathsf{T}}\boldsymbol{H} = \begin{pmatrix} -\frac{3}{5} & -\frac{4}{5} \\ -\frac{4}{5} & \frac{3}{5} \end{pmatrix} \begin{pmatrix} -\frac{3}{5} & -\frac{4}{5} \\ -\frac{4}{5} & \frac{3}{5} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

d. h. $\boldsymbol{H}^{\mathsf{T}} = \boldsymbol{H}^{-1}$ ist eine orthogonale Matrix. □

Definition 4.22. (*Bandmatrix*)
Eine (n,n)-Matrix $\boldsymbol{A} = (a_{ik})$, $i,k = 1(1)n$, heißt *Bandmatrix*, wenn ihre Elemente außerhalb eines Bandes längs der Hauptdiagonale verschwinden. Sei m_ℓ die Anzahl der unteren Nebendiagonalen und m_r die Anzahl der oberen, dann gilt für die Nullelemente: $a_{ik} = 0$ für

$$i - k > m_\ell \quad \text{mit} \quad 0 \le m_\ell \le n-2 \quad \text{und} \quad k - i > m_r \quad \text{mit} \quad 0 \le m_r \le n-2.$$

Die Größe $m = m_\ell + m_r + 1$ heißt *Bandbreite*; es können höchstens m Nichtnull-Elemente in einer Zeile auftreten. Spezielle Bandmatrizen sind:

Diagonalmatrizen mit	$m_\ell = m_r = 0$,	
bidiagonale Matrizen mit	$m_\ell = 1,\ m_r = 0$	oder $\quad m_\ell = 0,\ m_r = 1$,
tridiagonale Matrizen mit	$m_\ell = m_r = 1$,	
fünfdiagonale Matrizen mit	$m_\ell = m_r = 2$.	

Beispiel 4.23.

$$\boldsymbol{A} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & -6 \end{pmatrix} \qquad \boldsymbol{B} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 0 & -1 & 2 & 0 \\ 0 & 0 & 3 & 4 \end{pmatrix}$$

Diagonalmatrix — untere Bidiagonalmatrix

$$\boldsymbol{C} = \begin{pmatrix} 1 & 2 & 0 & 0 \\ 0 & 1 & 3 & 0 \\ 0 & 0 & 4 & 5 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad \boldsymbol{D} = \begin{pmatrix} 1 & 2 & 0 & 0 \\ -1 & 2 & 3 & 0 \\ 0 & 4 & 5 & 6 \\ 0 & 0 & 1 & 3 \end{pmatrix}$$

obere Bidiagonalmatrix — tridiagonale Matrix

$$\boldsymbol{F} = \begin{pmatrix} 1 & 2 & 1 & 0 & 0 \\ 3 & 4 & 5 & 2 & 0 \\ 3 & 1 & -2 & 4 & 3 \\ 0 & 5 & 3 & 4 & 5 \\ 0 & 0 & 7 & 1 & 2 \end{pmatrix}$$

fünfdiagonale Matrix

□

Definition 4.24. (*Zyklisch tridiagonale Matrix*)
Eine (n,n)-Matrix $\boldsymbol{A} = (a_{ik})$, $i,k = 1(1)n$, heißt *zyklisch tridiagonal*, falls gilt

$$a_{ik} = 0 \quad \text{für} \quad 2 \le |i-k| \le n-2\,, \quad (a_{1n}, a_{n1}) \ne (0,0)\,.$$

Dies bedeutet, dass bei einer zyklisch tridiagonalen Matrix die Hauptdiagonale und die beiden Nebendiagonalen mit Nichtnull-Elementen besetzt sein können und mindestens eines der beiden Elemente links unten (a_{n1}) und rechts oben (a_{1n}) von Null verschieden ist.

Beispiel 4.25.

$$\boldsymbol{A} = \begin{pmatrix} 1 & 2 & 0 & 1 \\ -3 & 4 & 1 & 0 \\ 0 & 1 & 2 & 1 \\ -1 & 0 & 2 & 3 \end{pmatrix}, \quad \boldsymbol{B} = \begin{pmatrix} 1 & 2 & 0 & 0 & 4 \\ -3 & 4 & 1 & 0 & 0 \\ 0 & -1 & 5 & -1 & 0 \\ 0 & 0 & -3 & 6 & 3 \\ -5 & 0 & 0 & 2 & 1 \end{pmatrix}$$

sind zyklisch tridiagonale Matrizen. □

Definition 4.26. (*Diagonaldominante und stark diagonaldominante Matrix*)
Eine Matrix $\boldsymbol{A} = (a_{ik})$, $i, k = 1(1)n$, heißt *diagonaldominant*, falls

$$|a_{ii}| \geq \sum_{\substack{k=1 \\ k \neq i}}^{n} |a_{ik}| \quad \text{für} \quad i = 1(1)n\,;$$

für mindestens ein i muss das Zeichen „echt größer“ gelten (d. h. der Betrag des Diagonalelements ist größer oder gleich der Summe der Beträge der anderen Elemente in derselben Zeile; für mindestens eine Zeile muß dabei „echt größer“ gelten).

Die Matrix heißt *stark diagonaldominant*, wenn gilt

$$|a_{ii}| > \sum_{\substack{k=1 \\ k \neq i}}^{n} |a_{ik}| \quad \text{für} \quad i = 1(1)n\,.$$

Das heißt, dass für alle Zeilen der Betrag des Diagonalelements echt größer als die Summe der Beträge der übrigen Elemente der Zeile ist.

Beispiel 4.27.

$$\boldsymbol{A} = \begin{pmatrix} 10 & -1 & 2 \\ 5 & 8 & 1 \\ -1 & 2 & \textcircled{3} \end{pmatrix} \quad \text{ist diagonaldominant}$$

$$\boldsymbol{A} = \begin{pmatrix} 10 & -1 & 2 \\ 5 & 8 & 1 \\ -1 & 2 & \textcircled{5} \end{pmatrix} \quad \text{ist stark diagonaldominant}$$

□

Definition 4.28. (*Positiv definite und positiv semidefinite Matrix*)
Eine symmetrische Matrix $\boldsymbol{A} = (a_{ik})$, $i, k = 1(1)n$, heißt *positiv definit*, wenn für ihre quadratische Form Q gilt

$$Q(\boldsymbol{x}) := \boldsymbol{x}^{\mathsf{T}} \boldsymbol{A} \boldsymbol{x} > 0 \quad \text{für alle} \quad \boldsymbol{x} \neq \boldsymbol{0},\ \boldsymbol{x} \in \mathbb{R}^n.$$

Sie heißt *positiv semidefinit*, wenn gilt

$$Q(\boldsymbol{x}) \geq 0 \quad \text{für alle} \quad \boldsymbol{x} \in \mathbb{R}^n.$$

Notwendige Bedingung für positive Definitheit einer symmetrischen Matrix $\boldsymbol{A}=(a_{ik})$, $i,k=1(1)n$, ist:

$$a_{ii}>0 \quad \text{für alle } i.$$

Notwendige und hinreichende Kriterien für positive Definitheit einer symmetrischen Matrix $\boldsymbol{A}=(a_{ik})$, $i,k=1(1)n$:

1. $\boldsymbol{A}=\boldsymbol{A}^{\mathsf{T}}$ ist genau dann positiv definit, wenn sämtliche Hauptabschnittsdeterminanten positiv sind:

$$\det(\boldsymbol{A}_k)>0 \quad \text{für} \quad k=1(1)n.$$

Beispiel 4.29.

$$\boldsymbol{A}=\begin{pmatrix} 5 & -1 & -2 \\ -1 & 6 & -1 \\ -2 & -1 & 5 \end{pmatrix} \quad = \quad \boldsymbol{A}^{\mathsf{T}} \text{ symmetrisch}$$

ist positiv definit, da sämtliche Hauptabschnittsdeterminanten positiv sind

$$\begin{aligned} \det(\boldsymbol{A}_1) &= |5| &&= 5 &&> 0 \\ \det(\boldsymbol{A}_2) &= \begin{vmatrix} 5 & -1 \\ -1 & 6 \end{vmatrix} &&= 29 &&> 0 \\ \det(\boldsymbol{A}_3) &= \det(\boldsymbol{A}) &&= 112 &&> 0 \end{aligned}$$

□

2. Die Zerlegung $\boldsymbol{A}=\boldsymbol{L}\boldsymbol{R}$ mit $\boldsymbol{A}=\boldsymbol{A}^{\mathsf{T}}$ gemäß Satz 4.9 führt auf ein $\boldsymbol{R}$ mit $r_{ii}>0$ für alle i.

Beispiel 4.30.

$$\boldsymbol{A}=\begin{pmatrix} 4 & -2 & 0 \\ -2 & 5 & -2 \\ 0 & -2 & 5 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 \\ 0 & -\frac{1}{2} & 1 \end{pmatrix} \begin{pmatrix} \textcircled{4} & -2 & 0 \\ 0 & \textcircled{4} & -2 \\ 0 & 0 & \textcircled{4} \end{pmatrix} = \boldsymbol{L}\boldsymbol{R}$$

mit $r_{ii}>0$ für alle i □

3. Die Zerlegung $\boldsymbol{A}=\boldsymbol{R}^{\mathsf{T}}\boldsymbol{D}\boldsymbol{R}$ mit $\boldsymbol{A}=\boldsymbol{A}^{\mathsf{T}}$, $\boldsymbol{R}=$ normierte Superdiagonalmatrix, $\boldsymbol{D}=(d_{ik})=$ Diagonalmatrix, führt auf $d_{ii}>0$ für alle i.

Hinreichende Kriterien für positive Definitheit einer symmetrischen Matrix $\boldsymbol{A}=(a_{ik})$, $i,k=1(1)n$:

1. Jede symmetrische, stark diagonaldominante Matrix mit positiven Hauptdiagonalelementen ($a_{ii}>0$ für alle i) ist positiv definit.

2. Jede symmetrische, diagonaldominante Matrix $\boldsymbol{A}$ mit positiven Hauptdiagonalelementen ($a_{ii}>0$ für alle i) und $a_{ik}<0$ für alle $i\neq k$ ist positiv definit.

3. Jede symmetrische, tridiagonale, diagonaldominante Matrix $\boldsymbol{A}=(a_{ik})$, $i,k=1(1)n$, mit $a_{ii}>0$ für alle i, $a_{ik}\neq 0$ für $|i-k|=1$ ist positiv definit.

Beispiel 4.31.

$$\text{zu 1.} \quad \boldsymbol{A} = \begin{pmatrix} 16 & 0 & 4 \\ 0 & 9 & 0 \\ 4 & 0 & 17 \end{pmatrix}, \qquad \text{zu 2.} \quad \boldsymbol{B} = \begin{pmatrix} 3 & -1 & -2 \\ -1 & 6 & -3 \\ -2 & -3 & 8 \end{pmatrix}$$

$$\text{zu 3.} \quad \boldsymbol{C} = \begin{pmatrix} 10 & 1 & 0 & 0 \\ 1 & 10 & -2 & 0 \\ 0 & -2 & 5 & -3 \\ 0 & 0 & -3 & 6 \end{pmatrix}$$

□

Satz 4.32.
Eine stark diagonaldominante Matrix $\boldsymbol{A} = (a_{ik})$, $i, k = 1(1)n$, ist streng regulär.

Beispiel 4.33.

$$\boldsymbol{A} = \begin{pmatrix} 10 & -1 & 2 \\ 5 & 8 & 1 \\ -1 & 2 & 5 \end{pmatrix} \quad \text{streng regulär, da} \quad \det(\boldsymbol{A}_1) = |10| = 10 \neq 0\,,$$

$$\det(\boldsymbol{A}_2) = \begin{vmatrix} 10 & -1 \\ 5 & 8 \end{vmatrix} = 85 \neq 0\,, \quad \det(\boldsymbol{A}_3) = \det(\boldsymbol{A}) = 442 \neq 0$$

□

Satz 4.34.
Jede symmetrische, streng reguläre Matrix $\boldsymbol{A} = (a_{ik})$, $i, k = 1(1)n$, kann eindeutig in das Produkt $\boldsymbol{R}^{\mathsf{T}}\boldsymbol{D}\boldsymbol{R}$ mit einer normierten oberen Dreiecksmatrix $\boldsymbol{R}$, ihrer Transponierten $\boldsymbol{R}^{\mathsf{T}}$ und einer Diagonalmatrix $\boldsymbol{D}$ zerlegt werden.

Zerlegung für $n = 3$: Mit $\boldsymbol{A} = \boldsymbol{R}^{\mathsf{T}}\boldsymbol{D}\boldsymbol{R}$ gilt

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{12} & a_{22} & a_{23} \\ a_{13} & a_{23} & a_{33} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ r_{12} & 1 & 0 \\ r_{13} & r_{23} & 1 \end{pmatrix} \begin{pmatrix} d_{11} & 0 & 0 \\ 0 & d_{22} & 0 \\ 0 & 0 & d_{33} \end{pmatrix} \begin{pmatrix} 1 & r_{12} & r_{13} \\ 0 & 1 & r_{23} \\ 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 0 \\ r_{12} & 1 & 0 \\ r_{13} & r_{23} & 1 \end{pmatrix} \begin{pmatrix} d_{11} & d_{11}r_{12} & d_{11}r_{13} \\ 0 & d_{22} & d_{22}r_{23} \\ 0 & 0 & d_{33} \end{pmatrix}$$

$$= \begin{pmatrix} d_{11} & d_{11}r_{12} & d_{11}r_{13} \\ r_{12}d_{11} & d_{11}r_{12}^2 + d_{22} & d_{11}r_{13}r_{12} + d_{22}r_{23} \\ r_{13}d_{11} & d_{11}r_{12}r_{13} + d_{22}r_{23} & d_{11}r_{13}^2 + d_{22}r_{23}^2 + d_{33} \end{pmatrix}.$$

Aus dem Koeffizientenvergleich zwischen Ausgangsmatrix $\boldsymbol{A}$ und Produktmatrix ergibt sich zeilenweise für das obere Dreieck von $\boldsymbol{A}$

$$\begin{array}{lcll} a_{11} & = & d_{11} & \Rightarrow \quad d_{11} = a_{11} \\ a_{12} & = & d_{11} r_{12} & \Rightarrow \quad r_{12} = a_{12} \,/\, d_{11} \\ a_{13} & = & d_{11} r_{13} & \Rightarrow \quad r_{13} = a_{13} \,/\, d_{11} \\ \hline a_{22} & = & d_{11} r_{12}^2 + d_{22} & \Rightarrow \quad d_{22} = a_{22} - d_{11} r_{12}^2 \\ a_{23} & = & d_{11} r_{13} r_{12} + d_{22} r_{23} & \Rightarrow \quad r_{23} = (a_{23} - d_{11} r_{13} r_{12}) \,/\, d_{22} \\ \hline a_{33} & = & d_{11} r_{13}^2 + d_{22} r_{23}^2 + d_{33} & \Rightarrow \quad d_{33} = a_{33} - d_{11} r_{13}^2 - d_{22} r_{23}^2 \end{array}$$

Algorithmus 4.35.
Gegeben: $\boldsymbol{A} = (a_{ik})$, $i,k = 1(1)n$, symmetrisch, streng regulär
Gesucht: Elemente von $\boldsymbol{D}$ und $\boldsymbol{R}$ bei der Zerlegung $\boldsymbol{A} = \boldsymbol{R}^{\mathsf{T}} \boldsymbol{D} \boldsymbol{R}$

1. $d_{11} = a_{11}$
   $r_{1k} = \frac{a_{1k}}{d_{11}}\,, \quad k = 2(1)n$

2. Für jedes $i = 2(1)n{-}1$
   2.1 $d_{ii} = a_{ii} - \sum\limits_{j=1}^{i-1} d_{jj}\, r_{ji}^2$
   2.2 $r_{i,i+1} = \frac{1}{d_{ii}} \left(a_{i,i+1} - \sum\limits_{j=1}^{i-1} d_{jj}\, r_{j,j+1}\, r_{j,j+2} \right)$

3. $d_{nn} = a_{nn} - \sum\limits_{j=1}^{n-1} d_{jj}\, r_{jn}^2$

Beispiel 4.36. (vgl. Beispiel 4.30)
$d_{11} = 4 \qquad r_{12} = -\frac{1}{2} \qquad r_{13} = 0\,; \qquad d_{22} = 5 - 4 \cdot \left(\frac{1}{4}\right) = 4$
$r_{23} = \frac{1}{4}\big(-2 - 4 \cdot 0 \cdot \left(-\frac{1}{2}\right) \big) = -\frac{1}{2}\,; \qquad d_{33} = 5 - 4 \cdot 0 - 4 \cdot \left(-\frac{1}{2}\right)^2 = 5 - 1 = 4\,.$

$$\begin{array}{ccccc} \boldsymbol{A} & & \boldsymbol{R}^{\mathsf{T}} & \boldsymbol{D} & \boldsymbol{R} \\ \begin{pmatrix} 4 & -2 & 0 \\ -2 & 5 & -2 \\ 0 & -2 & 5 \end{pmatrix} & = & \begin{pmatrix} 1 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 \\ 0 & -\frac{1}{2} & 1 \end{pmatrix} & \begin{pmatrix} 4 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 4 \end{pmatrix} & \begin{pmatrix} 1 & -\frac{1}{2} & 0 \\ 0 & 1 & -\frac{1}{2} \\ 0 & 0 & 1 \end{pmatrix} \end{array}$$

□

Satz 4.37.
Jede symmetrische, positiv definite Matrix $\boldsymbol{A} = (a_{ik})$, $i,k = 1(1)n$, ist in das Produkt $\boldsymbol{R}^{\mathsf{T}} \boldsymbol{D} \boldsymbol{R}$ gemäß Satz 4.34 zerlegbar, wobei alle Diagonalelemente von $\boldsymbol{D}$ (Pivotelemente) positiv sind und $\boldsymbol{A}$ streng regulär ist.

Satz 4.38.
Jede symmetrische, positiv definite Matrix $\boldsymbol{A} = (a_{ik})$, $i,k = 1(1)n$, kann eindeutig in das Produkt $\boldsymbol{R}^\mathsf{T}\boldsymbol{R}$ mit der oberen Dreiecksmatrix $\boldsymbol{R} = (r_{ik})$, $r_{ii} > 0$ für alle i, und ihrer Transponierten $\boldsymbol{R}^\mathsf{T}$ zerlegt werden. Die Zerlegung heißt *Cholesky-Zerlegung*.

Cholesky-Zerlegung für $n = 3$: $\boldsymbol{A} = \boldsymbol{R}^\mathsf{T}\boldsymbol{R}$

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{12} & a_{22} & a_{23} \\ a_{13} & a_{23} & a_{33} \end{pmatrix} = \begin{pmatrix} r_{11} & 0 & 0 \\ r_{12} & r_{22} & 0 \\ r_{13} & r_{23} & r_{33} \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ 0 & r_{22} & r_{23} \\ 0 & 0 & r_{33} \end{pmatrix}$$

$$= \begin{pmatrix} r_{11}^2 & r_{11}\,r_{12} & r_{11}\,r_{13} \\ r_{11}\,r_{12} & r_{12}^2 + r_{22}^2 & r_{12}\,r_{13} + r_{22}\,r_{23} \\ r_{11}\,r_{13} & r_{12}\,r_{13} + r_{22}\,r_{23} & r_{13}^2 + r_{23}^2 + r_{33}^2 \end{pmatrix}$$

Koeffizientenvergleich liefert zeilenweise:

$$\begin{aligned} a_{11} &= r_{11}^2 && \Rightarrow\ r_{11} = \sqrt{a_{11}} \\ a_{12} &= r_{11}\,r_{12} && \Rightarrow\ r_{12} = a_{12}/r_{11} \\ a_{13} &= r_{11}\,r_{13} && \Rightarrow\ r_{13} = a_{13}/r_{11} \\ a_{22} &= r_{12}^2 + r_{22}^2 && \Rightarrow\ r_{22} = \sqrt{a_{22} - r_{12}^2} \\ a_{23} &= r_{12}\,r_{13} + r_{22}\,r_{23} && \Rightarrow\ r_{23} = (a_{23} - r_{12}r_{13})/r_{22} \\ a_{33} &= r_{13}^2 + r_{23}^2 + r_{33}^2 && \Rightarrow\ r_{33} = \sqrt{a_{33} - r_{13}^2 - r_{23}^2} \end{aligned}$$

Die Verallgemeinerung liefert den folgenden Algorithmus:

Algorithmus 4.39.

Gegeben: $\boldsymbol{A} = (a_{ik})$, $i,k = 1(1)n$, symmetrisch, positiv definit
Gesucht: Elemente r_{ik} von $\boldsymbol{R}$ in der Cholesky-Zerlegung $\boldsymbol{A} = \boldsymbol{R}^\mathsf{T}\boldsymbol{R}$

1. $r_{11} = \sqrt{a_{11}}$
2. Für jedes $j = 2(1)n{-}1$
   2.1 $r_{jj} = \sqrt{a_{jj} - \sum_{i=1}^{j-1} r_{ij}^2}$
   2.2 Für jedes $k = j{+}1(1)n$
   $$r_{jk} = \frac{1}{r_{jj}}\left(a_{jk} - \sum_{i=1}^{j-1} r_{ik}\,r_{ij}\right)$$
3. $r_{nn} = \sqrt{a_{nn} - \sum_{i=1}^{n-1} r_{in}^2}$

Beispiel 4.40.

Gegeben: Die Matrix $\boldsymbol{A} = \begin{pmatrix} 4 & -2 & 0 \\ -2 & 5 & -2 \\ 0 & -2 & 5 \end{pmatrix}$.

Gesucht: Die Matrix $\boldsymbol{R}$ mit $\boldsymbol{A} = \boldsymbol{R}^{\mathsf{T}}\boldsymbol{R}$ nach Algorithmus 4.39.

Lösung:
$$\begin{array}{llllll} r_{11} = \sqrt{4} = 2\,; & & r_{22} = \sqrt{5-(-1)^2} = 2 \\ r_{12} = -2/2 = -1\,; & & r_{23} = (-2-(-1)\cdot 0)/2 = -1 \\ r_{13} = 0/2 = 0\,; & & r_{33} = \sqrt{5-0^2-(-1)^2} = 2 \end{array}$$

$$\begin{array}{ccccc} \boldsymbol{A} & = & \boldsymbol{R}^{\mathsf{T}} & & \boldsymbol{R} \\ \begin{pmatrix} 4 & -2 & 0 \\ -2 & 5 & -2 \\ 0 & -2 & 5 \end{pmatrix} & = & \begin{pmatrix} 2 & 0 & 0 \\ -1 & 2 & 0 \\ 0 & -1 & 2 \end{pmatrix} & \begin{pmatrix} 2 & -1 & 0 \\ 0 & 2 & -1 \\ 0 & 0 & 2 \end{pmatrix} \end{array}$$

□

4.3 Lösbarkeitsbedingungen für ein lineares Gleichungssystem

Ein lineares Gleichungssystem $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}$ mit einer (m,n)-Matrix $\boldsymbol{A}$, $\boldsymbol{x} \in \mathbb{R}^m$, $\mathrm{Rg}(\boldsymbol{A}) = r$, $m \geq n$, heißt homogen, wenn $\boldsymbol{a} = \boldsymbol{0}$ ist, andernfalls heißt es inhomogen.

1. Das homogene System $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}$ ist stets lösbar; es besitzt $n-r$ linear unabhängige Lösungen $\boldsymbol{y}_i$. Die Gesamtheit der Lösungen lässt sich als Linearkombination
$$\boldsymbol{x} = c_1\boldsymbol{y}_1 + c_2\boldsymbol{y}_2 + \ldots + c_{n-r}\boldsymbol{y}_{n-r}$$
darstellen. Im Falle $r = n$ existiert nur die triviale Lösung $\boldsymbol{x} = \boldsymbol{0}$.
Für Systeme (4.2) aus n Gleichungen mit n Unbekannten gilt in der Formulierung über die Determinante:
   a) $\det \boldsymbol{A} \neq 0$: Es existiert nur die triviale Lösung $\boldsymbol{x} = \boldsymbol{0}$.
   b) $\det \boldsymbol{A} = 0$: Die Matrix $\boldsymbol{A}$ habe den Rang r: $\mathrm{Rg}(\boldsymbol{A}) = r$. Dann besitzt das homogene System genau $n-r$ linear unabhängige Lösungen.
2. Das inhomogene Gleichungssystem: $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}$ mit $\boldsymbol{a} \neq \boldsymbol{0}$. Es gilt der

Satz 4.41.
Ein inhomogenes Gleichungssystem $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a} \neq \boldsymbol{0}$ ist genau dann auflösbar, wenn der Rang der erweiterten Matrix $(\boldsymbol{A}, \boldsymbol{a})$ gleich dem Rang der Matrix $\boldsymbol{A}$ ist: $\mathrm{Rg}(\boldsymbol{A}, \boldsymbol{a}) = \mathrm{Rg}(\boldsymbol{A})$. Die Gesamtheit der Lösungen setzt sich aus der Lösung $\boldsymbol{x}_h$ des homogenen Systems und einer speziellen Lösung des inhomogenen Systems zusammen.

Für Systeme (4.2) mit $m = n$ gilt in der Formulierung über die Determinante

a) $\det \boldsymbol{A} \neq 0$: Es existiert genau eine Lösung, sie lautet $\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{a}$.
(Hier gilt: $\mathrm{Rg}(\boldsymbol{A}, \boldsymbol{a}) = \mathrm{Rg}(\boldsymbol{A}) = n$)

b) $\det \boldsymbol{A} = 0$: Ist das System auflösbar, d. h. $\mathrm{Rg}(\boldsymbol{A}, \boldsymbol{a}) = \mathrm{Rg}(\boldsymbol{A}) < n$, so ist die Lösung nicht eindeutig bestimmt. Sie ergibt sich als Summe aus einer Linearkombination der $n-r$ linear unabhängigen Lösungen des homogenen Systems und einer speziellen Lösung des inhomogenen Systems.

4.4 Prinzip der direkten Methoden zur Lösung linearer Gleichungssysteme

Das Prinzip der direkten Methoden besteht in einer Dreieckszerlegung (Faktorisierung) der Matrix $\boldsymbol{A}$ des zu lösenden Gleichungssystems $\boldsymbol{Ax} = \boldsymbol{a}$ aus n Gleichungen mit n Unbekannten. Die (n, n)-Matrix $\boldsymbol{A}$ wird im Allgemeinen in das Produkt $\boldsymbol{LR}$ einer unteren Dreiecksmatrix $\boldsymbol{L}$ und einer oberen Dreiecksmatrix $\boldsymbol{R}$ zerlegt (sofern die Zerlegung existiert), wobei eine der beiden Dreiecksmatrizen normiert sein muss, um eine eindeutige Zerlegung zu erreichen. Die Dreieckszerlegung bewirkt eine Überführung des Systems $\boldsymbol{Ax} = \boldsymbol{a}$ in ein äquivalentes System $\boldsymbol{Rx} = \boldsymbol{r}$, aus dem sich wegen der oberen Dreiecksform von $\boldsymbol{R}$ rekursiv die Lösung gewinnen lässt. Sind für die Dreieckszerlegung Zeilenvertauschungen erforderlich, so wird statt $\boldsymbol{A}$ eine aus $\boldsymbol{A}$ durch die gleichen Zeilenvertauschungen hervorgegangene Matrix $\boldsymbol{PA}$ ($\boldsymbol{P}$ Permutationsmatrix) in die beiden Dreiecksmatrizen $\boldsymbol{L}$ und $\boldsymbol{R}$ zerlegt.

Aus den Sätzen 4.9 bzw. 4.12 ergeben sich die folgenden Algorithmen für die Lösung linearer Systeme durch Dreieckszerlegung (Faktorisierung).

Algorithmus 4.42. (*Elimination ohne Zeilenvertauschungen*)

Gegeben: $\boldsymbol{Ax} = \boldsymbol{a}$ mit $\det(\boldsymbol{A}_k) \neq 0$ für $k = 1(1)n-1$.

Gesucht: Lösung $\boldsymbol{x}$.

1. Schritt: Dreieckszerlegung $\boldsymbol{A} = \boldsymbol{LR}$ zur Ermittlung von $\boldsymbol{L}$ und $\boldsymbol{R}$ gemäß Satz 4.9 und Algorithmus 4.11.

2. Schritt: Vorwärtselimination $\boldsymbol{a} = \boldsymbol{Lr}$ zur Bestimmung von $\boldsymbol{r}$. Es gilt

$$\begin{aligned} r_1 &= a_1 \\ r_i &= a_i - \sum_{k=1}^{i-1} l_{ik}\, r_i \qquad \text{für } i = 2(1)n \end{aligned}$$

3. Schritt: Rückwärtselimination $\boldsymbol{Rx} = \boldsymbol{r}$ zur Berechnung der Lösung $\boldsymbol{x}$. Es gilt

$$\begin{aligned} x_n &= \tfrac{r_n}{a_{nn}} \\ x_i &= \tfrac{1}{a_{ii}}\left(r_i - \sum_{k=i+1}^{n} a_{ik}\, x_k\right) \qquad \text{für } i = n-1(-1)1 \end{aligned}$$

Beweis. Aus $\boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{a}$ folgt nach Satz 4.9 mit $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{R}$ die Beziehung $\boldsymbol{L}\boldsymbol{R}\,\boldsymbol{x} = \boldsymbol{a}$. Ziel ist die Herstellung der oberen Dreiecksform $\boldsymbol{R}\,\boldsymbol{x} = \boldsymbol{r}$, so dass gilt

$$\boldsymbol{L}\underbrace{\boldsymbol{R}\,\boldsymbol{x}}_{:=\boldsymbol{r}} = \boldsymbol{a} \quad \Longrightarrow \quad \boldsymbol{L}\,\boldsymbol{r} = \boldsymbol{a}\,.$$

Das heißt man muss nach der Faktorisierung $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{R}$ zuerst $\boldsymbol{L}\,\boldsymbol{r} = \boldsymbol{a}$ durch Vorwärtselimination lösen, bevor man aus $\boldsymbol{R}\,\boldsymbol{x} = \boldsymbol{r}$ rekursiv die Lösung $\boldsymbol{x}$ ermitteln kann. Daraus ergibt sich die Reihenfolge der Schritte im Algorithmus.

Beispiel 4.43.

Gegeben: $\boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{a}$ mit

$$\boldsymbol{A} = \begin{pmatrix} 4 & -1 & -1 \\ 8 & 0 & -1 \\ 4 & 1 & 4 \end{pmatrix} \quad \boldsymbol{a} = \begin{pmatrix} 2 \\ 7 \\ 9 \end{pmatrix} \quad \boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

Gesucht: Der Lösungsvektor $\boldsymbol{x}$.

Lösung:

1. Schritt: Dreieckszerlegung $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{R} \Longrightarrow \boldsymbol{L}, \boldsymbol{R}$

$$\boldsymbol{A} = \begin{pmatrix} 4 & -1 & -1 \\ 8 & 0 & -1 \\ 4 & 1 & 4 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 4 & -1 & -1 \\ 0 & 2 & 1 \\ 0 & 0 & 4 \end{pmatrix}$$

2. Schritt: Vorwärtselimination $\boldsymbol{L}\,\boldsymbol{r} = \boldsymbol{a}$

$$\begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 7 \\ 9 \end{pmatrix}$$

$$\left.\begin{array}{lcl} r_1 & = & 2 \\ 2r_1 + r_2 & = & 7 \;\Rightarrow\; r_2 = 7 - 4 = 3 \\ r_1 + r_2 + r_3 & = & 9 \;\Rightarrow\; r_3 = 9 - 3 - 2 = 4 \end{array}\right\} \Longrightarrow \quad \boldsymbol{r} = \begin{pmatrix} 2 \\ 3 \\ 4 \end{pmatrix}.$$

3. Schritt: Rückwärtselimination

$$\begin{pmatrix} 4 & -1 & -1 \\ 0 & 2 & 1 \\ 0 & 0 & 4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \\ 4 \end{pmatrix}$$

$$\left.\begin{array}{lcl} 4x_3 & = & 4 \;\Rightarrow\; x_3 = 1 \\ 2x_2 + x_3 & = & 3 \;\Rightarrow\; x_2 = (3-1)/2 = 1 \\ 4x_1 - x_2 - x_3 & = & 2 \;\Rightarrow\; x_1 = (2+1+1)/4 = 1 \end{array}\right\} \Longrightarrow \quad \boldsymbol{x} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

□

Algorithmus 4.44. (*Elimination mit Zeilenvertauschungen*)

Gegeben: $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}$ mit $\det \boldsymbol{A} \neq 0$, d. h. mit regulärer Matrix $\boldsymbol{A}$.

Gesucht: Lösung $\boldsymbol{x}$.

1. Schritt: Dreieckszerlegung $\boldsymbol{PA} = \boldsymbol{LR}$ zur Ermittlung von $\boldsymbol{L}$ und $\boldsymbol{R}$ gemäß Satz 4.12.

2. Schritt: Vorwärtselimination $\boldsymbol{P}\boldsymbol{a} = \boldsymbol{L}\boldsymbol{r}$ zur Bestimmung von $\boldsymbol{r}$.

3. Schritt: Rückwärtselimination $\boldsymbol{R}\boldsymbol{x} = \boldsymbol{r}$ zur Berechnung der Lösung $\boldsymbol{x}$.

Aus Satz 4.12 ergibt sich für $\boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{a}$ durch Linksmultiplikation mit der Permutationsmatrix $\boldsymbol{P}$

$$\boldsymbol{PA}\,\boldsymbol{x} = \boldsymbol{P}\,\boldsymbol{a}$$

und mit $\boldsymbol{PA} = \boldsymbol{LR}$

$$\boldsymbol{L}\underbrace{\boldsymbol{R}\,\boldsymbol{x}} = \boldsymbol{P}\,\boldsymbol{a}\,,$$

woraus sich für $\boldsymbol{R}\,\boldsymbol{x} =: \boldsymbol{r}$ ergibt

$$\boldsymbol{L}\,\boldsymbol{r} = \boldsymbol{P}\,\boldsymbol{a} \quad \text{(Vorwärtselimination)}.$$

Die Vorwärtselimination muss nach der Faktorisierung aber vor der Rückwärtseliminationen $\boldsymbol{R}\,\boldsymbol{x} = \boldsymbol{r}$ durchgeführt werden, so dass sich die Reihenfolge in Algorithmus 4.44 ergibt.

Beispiel 4.45.

Gegeben: $\boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{a}$ mit

$$\boldsymbol{A} = \begin{pmatrix} 3 & 3 & 6 \\ 2 & 2 & 3 \\ 1 & 0 & 1 \end{pmatrix} \quad \boldsymbol{a} = \begin{pmatrix} -3 \\ -1 \\ 0 \end{pmatrix} \quad \boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

Gesucht: $\boldsymbol{x}$, $\det(\boldsymbol{A})$ über LR-Zerlegung mit Zeilenvertauschung.

Lösung:

1. Schritt: $\boldsymbol{PA} = \boldsymbol{LR}$

$$\boldsymbol{PA} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 3 & 3 & 6 \\ 2 & 2 & 3 \\ 1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 3 & 3 & 6 \\ 1 & 0 & 1 \\ 2 & 2 & 3 \end{pmatrix} = \boldsymbol{LR} = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{3} & 1 & 0 \\ \frac{2}{3} & 0 & 1 \end{pmatrix} \begin{pmatrix} 3 & 3 & 6 \\ 0 & -1 & -1 \\ 0 & 0 & -1 \end{pmatrix}$$

2. Schritt: $\boldsymbol{P}\,\boldsymbol{a} = \boldsymbol{L}\,\boldsymbol{r}$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} -3 \\ -1 \\ 0 \end{pmatrix} = \begin{pmatrix} -3 \\ 0 \\ -1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{3} & 1 & 0 \\ \frac{2}{3} & 0 & 1 \end{pmatrix} \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix} \quad \Longrightarrow \quad \boldsymbol{r} = \begin{pmatrix} -3 \\ 1 \\ 1 \end{pmatrix}$$

3. Schritt: $\boldsymbol{R}\,\boldsymbol{x} = \boldsymbol{r}$ durch Rückwärtselimination

$$\boldsymbol{A} = \begin{pmatrix} 3 & 3 & 6 \\ 0 & -1 & -1 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} -3 \\ 1 \\ 1 \end{pmatrix} \quad \Longrightarrow \quad \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}$$

Aus $\boldsymbol{PA} = \boldsymbol{LR}$ folgt

$$\begin{array}{rccccccc} & \det(\boldsymbol{PA}) & = & \det(\boldsymbol{P}) \cdot \det(\boldsymbol{A}) & = & \det(\boldsymbol{L}) \cdot \det(\boldsymbol{R}) & = & \det(\boldsymbol{R}) \\ \Longrightarrow & \det(\boldsymbol{A}) & = & \frac{1}{\det(\boldsymbol{P})} \cdot \det(\boldsymbol{R}) & = & (-1)\,(3)\,(-1)\,(-1) & = & -3 \end{array}$$

□

Die folgenden direkten Eliminationsverfahren arbeiten nach den angegebenen Algorithmen. Sie unterscheiden sich lediglich dadurch, dass spezielle Eigenschaften der Matrix $\boldsymbol{A}$ in $\boldsymbol{Ax} = \boldsymbol{a}$ ausgenutzt werden, wodurch eine zum Teil erhebliche Ersparnis an Rechenaufwand erreicht werden kann. Im Wesentlichen wird hier nur mit Systemen aus n Gleichungen für n Unbekannte gearbeitet, lediglich in Abschnitt 4.13 mit überbestimmten Systemen. Bevor aber spezielle Lösungsverfahren für lineare Systeme mit speziellen Matrizen entwickelt werden, soll der Gauß-Algorithmus in der Form des klassischen Gauß-Schemas hergeleitet werden, und danach wird der Zusammenhang zur Dreieckszerlegung hergestellt.

4.5 Der Gauß-Algorithmus

4.5.1 Gauß-Algorithmus mit Spaltenpivotsuche als Rechenschema

Das *Prinzip des Gaußschen Algorithmus* ist die Überführung eines Gleichungssystems der Form (4.1) mit $m = n$ in ein gestaffeltes System in oberer Dreiecksform

$$\left\{ \begin{array}{lllllll} r_{11}x_1 & + & r_{12}x_2 & + \cdots + & r_{1n}x_n & = & r_1, \\ & & r_{22}x_2 & + \cdots + & r_{2n}x_n & = & r_2, \\ & & & \ddots & \vdots & & \vdots \\ & & & & r_{nn}x_n & = & r_n, \end{array} \right. \tag{4.3}$$

aus dem sich die x_i, $i = 1(1)n$, durch Rückwärtselimination berechnen lassen, falls $r_{11}r_{22}\ldots r_{nn} \neq 0$ ist.

Zunächst wird der Algorithmus hergeleitet und anschließend wird der Zusammenhang mit den Algorithmen in Abschnitt 4.4 hergestellt.

Konstruktion des Verfahrens

Bekanntlich ist die Lösung eines Gleichungssystems (4.1) unabhängig von der Anordnung der Gleichungen. Man kann also o. B. d. A. eine Zeilenvertauschung derart vornehmen,

dass das betragsgrößte Element der ersten Spalte von $\boldsymbol{A}$ in die erste Zeile kommt (Spaltenpivotsuche vgl. Abschnitt 4.5.2). Die durch die Umordnung entstandene Matrix heiße $\boldsymbol{A}^{(0)}$, ihre Elemente $a_{ik}^{(0)}$ und die Komponenten der rechten Seite $a_i^{(0)}$, so dass (4.1) in das äquivalente System

$$\sum_{k=1}^{n} a_{ik}^{(0)} x_k = a_i^{(0)}, \quad i = 1(1)n, \tag{4.4}$$

übergeht. Ist det $\boldsymbol{A} \neq 0$, so gilt für das betragsgrößte Element (Pivotelement) der ersten Spalte $a_{11}^{(0)} \neq 0$. Zur Elimination von x_1 aus den Gleichungen $i = 2(1)n$ multipliziert man die 1. Gleichung von (4.4) mit $-a_{i1}^{(0)}/a_{11}^{(0)}$ und addiert sie jeweils zur i-ten Gleichung, so dass sich für $i = 2(1)n$ zusammen mit der unveränderten 1. Zeile ergibt (1. Eliminationsschritt):

$$\left\{\begin{array}{llll} a_{11}^{(0)} x_1 + & a_{12}^{(0)} x_2 & + \ldots & + a_{1n}^{(0)} x_n = \quad a_1^{(0)}, \\ & \widetilde{a}_{22}^{(1)} x_2 & + \ldots & + \widetilde{a}_{2n}^{(1)} x_n = \quad \widetilde{a}_2^{(1)}, \\ & \vdots & & \qquad\qquad\quad \vdots \\ & \widetilde{a}_{n2}^{(1)} x_2 & + \ldots & + \widetilde{a}_{nn}^{(1)} x_n = \quad \widetilde{a}_n^{(1)}, \end{array}\right. \tag{4.5}$$

mit

$$\widetilde{a}_{ik}^{(1)} = \left\{\begin{array}{ll} 0 \quad \text{für} \quad k = 1, & i = 2(1)n, \\ a_{ik}^{(0)} - a_{1k}^{(0)} \dfrac{a_{i1}^{(0)}}{a_{11}^{(0)}} & \text{sonst,} \end{array}\right. \qquad \widetilde{a}_i^{(1)} = a_i^{(0)} - a_1^{(0)} \frac{a_{i1}^{(0)}}{a_{11}^{(0)}}, \quad i = 2(1)n.$$

Das System (4.5) besteht also aus einer Gleichung mit den n Unbekannten $x_1, x_2, \ldots, x_n$ und $n-1$ Gleichungen mit den $n-1$ Unbekannten $x_2, \ldots, x_n$.

Auf die $n-1$ Gleichungen $i = 2(1)n$ von (4.5) wendet man das Eliminationsverfahren erneut an. Dazu muss man zunächst wieder eine Zeilenvertauschung durchführen, so dass das betragsgrößte Element der $\widetilde{a}_{i2}^{(1)}$ für $i = 2(1)n$ in der 2. Gleichung erscheint; nach der Zeilenvertauschung werden die Elemente der neu entstandenen Zeilen 2 bis n mit $a_{ik}^{(1)}$ bzw. $a_i^{(1)}$ bezeichnet:

$$\left\{\begin{array}{llll} a_{11}^{(0)} x_1 + & a_{12}^{(0)} x_2 & + \ldots & + a_{1n}^{(0)} x_n = \quad a_1^{(0)}, \\ & a_{22}^{(1)} x_2 & + \ldots & + a_{2n}^{(1)} x_n = \quad a_2^{(1)}, \\ & \vdots & & \qquad\qquad\quad \vdots \\ & a_{n2}^{(1)} x_2 & + \ldots & + a_{nn}^{(1)} x_n = \quad a_n^{(1)}, \end{array}\right. \tag{4.6}$$

wobei wegen det $\boldsymbol{A} \neq 0$ gelten muss $a_{22}^{(1)} \neq 0$.
Verfährt man nun analog mit der 2. bis n-ten Gleichung von (4.6), so sind für jeden

weiteren Eliminationsschritt j mit $j = 2(1)n{-}1$ die Elemente

$$\widetilde{a}_{ik}^{(j)} = \begin{cases} 0 \quad \text{für} \quad k = 1(1)j, \quad i = (j+1)(1)n, \\ a_{ik}^{(j-1)} - a_{jk}^{(j-1)} \frac{a_{ij}^{(j-1)}}{a_{jj}^{(j-1)}} \quad \text{sonst}, \end{cases}$$

$$\widetilde{a}_{i}^{(j)} = a_{i}^{(j-1)} - a_{j}^{(j-1)} \frac{a_{ij}^{(j-1)}}{a_{jj}^{(j-1)}}, \quad i = (j+1)(1)n,$$

zu berechnen. Nach jedem Eliminationsschritt j sind die Gleichungen $j+1$ bis n so umzuordnen, dass das betragsgrößte Element der $\widetilde{a}_{i,j+1}^{(j)}$ für $j+1 \le i \le n$ in der $(j+1)$-ten Gleichung steht; die Elemente der neu entstandenen Gleichungen $j+1$ bis n werden mit $a_{ik}^{(j)}$ bzw. $a_{i}^{(j)}$ bezeichnet. Man erhält so nach $n{-}1$ Eliminationsschritten das gestaffelte Gleichungssystem

$$\left\{ \begin{array}{lllll} a_{11}^{(0)} x_1 + & a_{12}^{(0)} x_2 + & a_{13}^{(0)} x_3 & + \ldots + & a_{1n}^{(0)} x_n \\ & a_{22}^{(1)} x_2 + & a_{23}^{(1)} x_3 & + \ldots + & a_{2n}^{(1)} x_n \\ & & a_{33}^{(2)} x_3 & + \ldots + & a_{3n}^{(2)} x_n \\ & & & & \vdots \\ & & & & a_{nn}^{(n-1)} x_n \end{array} \begin{array}{l} = a_1^{(0)}, \\ = a_2^{(1)}, \\ = a_3^{(2)}, \\ \quad \vdots \\ = a_n^{(n-1)}. \end{array} \right. \tag{4.7}$$

Mit $r_{ik} = a_{ik}^{(i-1)}$, $r_i = a_i^{(i-1)}$ besitzt (4.7) die Gestalt (4.3). Aus dem zu (4.1) äquivalenten System (4.7) berechnet man rekursiv die x_i gemäß

$$x_n = \frac{a_n^{(n-1)}}{a_{nn}^{(n-1)}}, \qquad x_j = \frac{a_j^{(j-1)}}{a_{jj}^{(j-1)}} - \sum_{k=j+1}^{n} \frac{a_{jk}^{(j-1)}}{a_{jj}^{(j-1)}} x_k, \quad j = n{-}1, n{-}2, \ldots, 1\,. \tag{4.8}$$

Im Fall $\det \boldsymbol{A} \neq 0$ darf keines der Diagonalelemente $a_{jj}^{(j-1)}$ verschwinden. Ist es nach irgendeinem Eliminationsschritt nicht mehr möglich, ein Element $a_{jj}^{(j-1)} \neq 0$ zu finden, so bedeutet dies, dass $\det \boldsymbol{A} = 0$ ist. Ob dann überhaupt eine Lösung existiert und wenn ja, wieviele Parameter sie besitzt, folgt automatisch aus der Rechnung. Für die Determinante von $\boldsymbol{A}$ gilt $\det \boldsymbol{A} = (-1)^k r_{11} r_{22} \ldots r_{nn}$, wobei k die Anzahl der Zeilenvertauschungen ist.

Da der Rang r von $\boldsymbol{A}$ gleich der Anzahl der nicht verschwindenden Diagonalelemente $r_{jj} = a_{jj}^{(j-1)}$ der Superdiagonalmatrix $\boldsymbol{R}$ (gegebenenfalls unter Spaltenvertauschungen) ist, lässt sich die Anzahl $n{-}r$ der Parameter nach Durchführung der $n{-}1$ Eliminationsschritte sofort angeben.

Rechenschema 4.46. (*Gaußscher Algorithmus für $n = 3$*)

Bezeichnung der Zeilen	$\boldsymbol{A}$			$\boldsymbol{a}$	erfolgte Operationen
$1^{(0)}$	$a_{11}^{(0)}$	$a_{12}^{(0)}$	$a_{13}^{(0)}$	$a_1^{(0)}$	
$2^{(0)}$	$a_{21}^{(0)}$	$a_{22}^{(0)}$	$a_{23}^{(0)}$	$a_2^{(0)}$	—
$3^{(0)}$	$a_{31}^{(0)}$	$a_{32}^{(0)}$	$a_{33}^{(0)}$	$a_3^{(0)}$	
$\widetilde{2}^{(1)}$	0	$\widetilde{a}_{22}^{(1)}$	$\widetilde{a}_{23}^{(1)}$	$\widetilde{a}_2^{(1)}$	$-\frac{a_{21}^{(0)}}{a_{11}^{(0)}}1^{(0)} + 2^{(0)}$
$\widetilde{3}^{(1)}$	0	$\widetilde{a}_{32}^{(1)}$	$\widetilde{a}_{33}^{(1)}$	$\widetilde{a}_3^{(1)}$	$-\frac{a_{31}^{(0)}}{a_{11}^{(0)}}1^{(0)} + 3^{(0)}$
$2^{(1)}$	0	$a_{22}^{(1)}$	$a_{23}^{(1)}$	$a_2^{(1)}$	Zeilenvertauschung von $\widetilde{2}^{(1)}, \widetilde{3}^{(1)}$ in $2^{(1)}, 3^{(1)}$,
$3^{(1)}$	0	$a_{32}^{(1)}$	$a_{33}^{(1)}$	$a_3^{(1)}$	so dass gilt $\lvert a_{22}^{(1)}\rvert = \max(\lvert\widetilde{a}_{22}^{(1)}\rvert, \lvert\widetilde{a}_{32}^{(1)}\rvert)$
$\widetilde{3}^{(2)} = 3^{(2)}$	0	0	$\widetilde{a}_{33}^{(2)} = a_{33}^{(2)}$	$\widetilde{a}_3^{(2)} = a_3^{(2)}$	$-\frac{a_{32}^{(1)}}{a_{22}^{(1)}}2^{(1)} + 3^{(1)}$

Die Zeilen $1^{(0)}$, $2^{(1)}$, $3^{(2)}$ bilden das gesuchte gestaffelte System (4.7), aus dem die Lösungen x_i rekursiv gemäß (4.8) bestimmt werden. Die Zeilenvertauschung der Zeilen $\widetilde{2}^{(1)}$, $\widetilde{3}^{(1)}$ erübrigt sich, falls $|\widetilde{a}_{22}^{(1)}| \geq |\widetilde{a}_{32}^{(1)}|$ ist; dann ist $\widetilde{a}_{2i}^{(1)} = a_{2i}^{(1)}$ und $\widetilde{a}_{3i}^{(1)} = a_{3i}^{(1)}$ für $i = 2, 3$ zu setzen.

Beispiel 4.47.

Gegeben: Das Gleichungssystem $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}$ mit

$$\boldsymbol{A} = \begin{pmatrix} 3 & 3 & 6 \\ 2 & 2 & 3 \\ 1 & 0 & 1 \end{pmatrix}, \qquad \boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}, \qquad \boldsymbol{a} = \begin{pmatrix} -3 \\ 1 \\ 0 \end{pmatrix}$$

Gesucht: Die Lösung $\boldsymbol{x}$ mit Hilfe des Gaußschen Algorithmus.

Lösung:

Bezeichnung der Zeilen	$\boldsymbol{A}$			$\boldsymbol{a}$	erfolgte Operationen
$1^{(0)}$	$\textcircled{3}$	3	6	-3	
$2^{(0)}$	$\boxed{2}$	2	3	1	—
$3^{(0)}$	$\boxed{1}$	0	1	0	
$\widetilde{2}^{(1)}$	0	0	-1	3	$-\left(\frac{\boxed{2}}{\textcircled{3}}\right)\cdot 1^{(0)}+2^{(0)}$
$\widetilde{3}^{(1)}$	0	-1	-1	1	$-\left(\frac{\boxed{1}}{\textcircled{3}}\right)\cdot 1^{(0)}+3^{(0)}$
$2^{(1)}$	0	-1	-1	1	Vertauschung der
$3^{(1)}$	0	0	-1	3	Zeilen $\widetilde{2}^{(1)}$ und $\widetilde{3}^{(1)}$

Aus $3^{(1)}$ folgt $x_3 = \frac{3}{-1} = -3$, mit $2^{(1)}$ folgt $x_2 = 2$, mit $1^{(0)}$ folgt $x_1 = 3$, also: $\boldsymbol{x} = (3, 2, -3)^{\mathsf{T}}$.

Bemerkung: Würde man hier ohne Zeilenvertauschung von $\widetilde{2}^{(1)}$ und $\widetilde{3}^{(1)}$ arbeiten, so käme es beim 2. Eliminationsschritt zur Division durch Null; dies wird durch die Zeilenvertauschung vermieden. □

Bemerkung. (***Homogene Systeme***)

Die Lösung *homogener Systeme* $\boldsymbol{Ax} = \boldsymbol{0}$ mit $\mathrm{Rg}(\boldsymbol{A}) = r$ erfolgt so, dass mit dem Gaußschen Algorithmus die Dreiecksmatrix $\boldsymbol{R}$ hergestellt wird. Das System reduziert sich auf r linear unabhängige Gleichungen (d. h. man erhält r Diagonalelemente $r_{ii} \neq 0$, $i = 1(1)r$). Für die restlichen $n-r$ Unbekannten setzt man beliebige Parameter ein, so dass sich damit die ersten r Unbekannten aus $\boldsymbol{Rx} = \boldsymbol{0}$ ermitteln lassen.

Beispiel 4.48.

Gegeben: Ein homogenes System $\boldsymbol{Ax} = \boldsymbol{0}$.

Gesucht: Die Lösungen des homogenen Systems mit dem Gaußschen Algorithmus.

Lösung:

Bezeichnung der Zeilen	$\boldsymbol{A}$			$\boldsymbol{a}$	erfolgte Operationen
$1^{(0)}$	2	-2	4	0	
$2^{(0)}$	-1	2	3	0	—
$3^{(0)}$	1	-1	2	0	
$2^{(1)}$	0	1	5	0	$-\left(-\frac{1}{2}\right) \cdot 1^{(0)} + 2^{(0)}$
$3^{(1)}$	0	0	0	0	$-\left(\frac{1}{2}\right) \cdot 1^{(0)} + 3^{(0)}$

Das gestaffelte System $\boldsymbol{Rx} = \boldsymbol{0}$ lautet:

$$\begin{aligned} 1^{(0)} &: \quad 2x_1 - 2x_2 + 4x_3 = 0 \\ 2^{(1)} &: \quad x_2 + 5x_3 = 0 \\ 3^{(1)} &: \quad 0 \cdot x_3 = 0 \end{aligned}$$

Wegen $\det \boldsymbol{A} = 0$ existiert eine nichttriviale Lösung. Aus $3^{(1)}$ folgt: $x_3 = t$, $t \in \mathbb{R}$, aus $2^{(1)}$: $x_2 = -5t$ und aus $1^{(0)}$: $x_1 = -7t$, so dass sich der vom Parameter t abhängige Lösungsvektor ergibt

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = t \begin{pmatrix} -7 \\ -5 \\ 1 \end{pmatrix}$$

d. h. alle Punkte einer Gerade durch den Nullpunkt sind Lösung des homogenen Systems. □

4.5.2 Spaltenpivotsuche

Wenn die Koeffizienten gerundete Zahlen sind oder im Verlaufe der Rechnung gerundet werden muss, sind Zeilenvertauschungen unerlässlich, um Verfälschungen des Ergebnisses durch Rundungsfehler möglichst zu dämpfen. Man bezeichnet diese Strategie als *Spaltenpivotsuche* oder *teilweise Pivotsuche* und die Diagonalelemente $\mathrm{r}_{jj} = \mathrm{a}_{jj}^{(j-1)}$ als *Pivotelemente*. Unter Verwendung der Spaltenpivotsuche wird der Gauß-Algorithmus in den meisten Fällen stabil. Vollkommen stabil wird er, wenn man als Pivotelement jeweils das betragsgrößte Element der gesamten Restmatrix verwendet, man spricht dann von *vollständiger Pivotsuche*. Hierfür ist der Aufwand sehr groß; für die Praxis ist die Spaltenpivotsuche bzw. die weiter unten erläuterte skalierte Pivotsuche in vielen Anwendungsfällen ausreichend.

Beispiel 4.49.

Gegeben: Das lineare Gleichungssystem

$$\left\{\begin{array}{lrcrcl} (1) & 0.2420\cdot 10^{-3}x_1 & + & 0.6004\cdot 10^{-2}x_2 & = & 0.1743\cdot 10^{-2}, \\ (2) & 0.4000\cdot 10^{0}x_1 & + & 0.9824\cdot 10^{1}x_2 & = & 0.2856\cdot 10^{1}. \end{array}\right.$$

Gesucht: Zu bestimmen sind x_1 und x_2 nach dem Gaußschen Algorithmus; die exakten Lösungen sind $x_1 = 1$, $x_2 = 0.25$.

Lösung:

1. Fall: Die Gleichungen werden in der obigen Reihenfolge benutzt. Zur Elimination von x_1 aus Gleichung (2) wird (1) mit

$$-\frac{0.4000}{0.2420\cdot 10^{-3}} = -0.1653\cdot 10^{4} = c_1$$

multipliziert und $c_1\cdot(1)+(2)$ gebildet. Man erhält das gestaffelte System

$$\left\{\begin{array}{rcr} 0.2420\cdot 10^{-3}x_1 + 0.6004\cdot 10^{-2}x_2 & = & 0.1743\cdot 10^{-2} \\ -0.1010\cdot 10^{0}x_2 & = & -0.2500\cdot 10^{-1}. \end{array}\right.$$

Als Lösungen ergeben sich daraus: $x_1 = 0.1062\cdot 10^{1}$; $x_2 = 0.2475$, sie weichen stark von den exakten Lösungen ab.

2. Fall: Die Reihenfolge der Gleichungen (1) und (2) wird vertauscht, so dass jetzt das betragsgrößte Element der ersten Spalte in der ersten Gleichung steht

$$\left\{\begin{array}{lrcrcl} 1^{(0)} & 0.4000\cdot 10^{0}x_1 & + & 0.9824\cdot 10^{1}x_2 & = & 0.2856\cdot 10^{1}, \\ 2^{(0)} & 0.2420\cdot 10^{-3}x_1 & + & 0.6004\cdot 10^{-2}x_2 & = & 0.1743\cdot 10^{-2}. \end{array}\right.$$

Zur Elimination von x_1 aus $2^{(0)}$ wird $1^{(0)}$ mit

$$-\frac{0.2420\cdot 10^{-3}}{0.4000} = -0.6050\cdot 10^{-3} = c_2$$

multipliziert und $c_2\cdot 1^{(0)} + 2^{(0)}$ gebildet. Man erhält das gestaffelte System

$$\left\{\begin{array}{rcr} 0.4000\cdot 10^{0}x_1 + 0.9824\cdot 10^{1}x_2 & = & 0.2856\cdot 10^{1}, \\ 0.6000\cdot 10^{-4}x_2 & = & 0.15000\cdot 10^{-4}. \end{array}\right.$$

Hieraus ergeben sich als Lösungen $x_1 = 0.1000\cdot 10^{1}$ und $x_2 = 0.2500$, sie stimmen mit den exakten Lösungen überein.

Die Abweichungen der Lösungen im 1. Fall sind dadurch entstanden, dass mit dem sehr großen Faktor c_1 multipliziert wurde und die Rundungsfehler dadurch entsprechend angewachsen sind. □

Beispiel 4.50.

Gegeben: Das lineare Gleichungssystem

$$\begin{pmatrix} 7 & 8 & 9 \\ 8 & 9 & 10 \\ 9 & 10 & 8 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 24 \\ 27 \\ 27 \end{pmatrix}.$$

Gesucht: Eine näherungsweise Lösung mit dem Gaußschen Algorithmus unter Verwendung der Gleitpunktarithmetik mit dreistelliger Mantisse

(i) ohne Pivotisierung

(ii) mit Spaltenpivotsuche.

Die exakte Lösung ist $\boldsymbol{x}_{ex} = (1, 1, 1)^{\mathsf{T}}$.

Lösung:

zu (i): Gaußscher Algorithmus ohne Pivotisierung

Bezeichnung der Zeilen	$\boldsymbol{A}$			$\boldsymbol{a}$	erfolgte Operationen
$1^{(0)}$	7.00	8.00	9.00	24.0	
$2^{(0)}$	8.00	9.00	10.0	27.0	—
$3^{(0)}$	9.00	10.0	8.00	27.0	
$2^{(1)}$	0.00	−0.12	−0.30	−0.400	$-\frac{8.00}{7.00} \cdot 1^{(0)} + 2^{(0)}$
$3^{(1)}$	0.00	−0.30	−3.60	−4.00	$-\frac{9.00}{7.00} \cdot 1^{(0)} + 3^{(0)}$
$3^{(2)}$	0.00	0.00	−2.85	−3.00	$-\frac{-0.300}{-0.120} \cdot 2^{(1)} + 3^{(1)}$

$\Rightarrow$ Lösungskomponenten: x_3=1.05, x_2=0.708, x_1=1.27

$$\boldsymbol{x} = \begin{pmatrix} 1.27 \\ 0.708 \\ 1.05 \end{pmatrix} \quad \Rightarrow \quad \boldsymbol{r} = \boldsymbol{x} - \boldsymbol{x}_{ex} = \begin{pmatrix} 1.27 \\ 0.708 \\ 1.05 \end{pmatrix} - \begin{pmatrix} 1.00 \\ 1.00 \\ 1.00 \end{pmatrix} = \begin{pmatrix} 0.270 \\ -0.292 \\ 0.0500 \end{pmatrix}$$

$|\boldsymbol{r}| = 0.401$ (Abstand der Lösung $\boldsymbol{x}$ von der exakten Lösung $\boldsymbol{x}_{ex}$).

zu (ii): Gaußscher Algorithmus mit teilweiser Pivotisierung

Bezeichnung der Zeilen	$\boldsymbol{A}$			$\boldsymbol{a}$	Operationen
$\widetilde{1}^{(0)}$	7.00	8.00	9.00	24.0	
$\widetilde{2}^{(0)}$	8.00	9.00	10.0	27.0	—
$\widetilde{3}^{(0)}$	9.00	10.0	8.00	27.0	
$1^{(0)}$	9.00	10.0	8.00	27.0	
$2^{(0)}$	8.00	9.00	10.0	27.0	—
$3^{(0)}$	7.00	8.00	9.00	24.0	
$\widetilde{2}^{(1)}$	0.00	0.110	2.89	3.00	$-\frac{8.00}{9.00}\cdot 1^{(0)} + 2^{(0)}$
$\widetilde{3}^{(1)}$	0.00	0.220	2.78	3.00	$-\frac{7.00}{9.00}\cdot 1^{(0)} + 3^{(0)}$
$2^{(1)}$	0.00	0.220	2.78	3.00	
$3^{(1)}$	0.00	0.110	2.89	3.00	—
$3^{(2)}$	0.00	0.00	1.50	1.50	$-\frac{0.110}{0.220}\cdot 2^{(1)} + 3^{(1)}$

$\Rightarrow$ Lösungskomponenten: $x_3 = 1.00$, $x_2 = 1.00$, $x_1 = 1.00$

$$\boldsymbol{x} = \begin{pmatrix} 1.00 \\ 1.00 \\ 1.00 \end{pmatrix} = \boldsymbol{x}_{ex},$$

d. h. hier erhält man die exakte Lösung auf 3 Mantissenstellen genau. □

Bemerkung. (Pivotsuche ist nur entscheidend wirkungsvoll, wenn alle Zeilen- und Spaltenbetragssummen annähernd gleich groß sind)

$$z_i := \sum_{j=1}^{n} |a_{ij}| \quad \approx \quad s_k := \sum_{j=1}^{n} |a_{jk}| \quad \text{für} \quad i,k = 1(1)n. \tag{4.9}$$

Matrizen, für die (4.9) gilt, heißen *äquilibriert* (vgl. [MAES1985], 2.2.2). Da sich aber der Rechenaufwand bei einer Äquilibrierung (vgl. Abschnitt 4.14.3) beträchtlich erhöhen würde, führt man bei nicht äquilibrierten Matrizen statt der Spaltenpivotsuche eine sogenannte *skalierte Spaltenpivotsuche* durch. Man ersetzt bei der Elimination das Glied

$$|a_{jj}^{(j-1)}| = \max_{i=j(1)n} \left(|\widetilde{a}_{ij}^{(j-1)}| \right)$$

in Abschnitt 4.5.1 durch

$$\frac{|a_{jj}^{(j-1)}|}{z_j} := \max_{i=j(1)n} \left(\frac{|\widetilde{a}_{ij}^{(j-1)}|}{z_i} \right). \tag{4.10}$$

4.5.3 Gauß-Algorithmus als Dreieckszerlegung

Die Vorgehensweise beim Gaußschen Algorithmus entspricht genau dem Algorithmus 4.42, wenn ohne Zeilenvertauschung gearbeitet wird.
Dann besteht der folgende Zusammenhang: Für die Elemente der Zerlegungsmatrizen $\boldsymbol{L} = (l_{ij})$, $\boldsymbol{R} = (r_{ij})$ und den Vektor $\boldsymbol{r} = (r_1, r_2, \ldots, r_n)^{\mathsf{T}}$ gilt

$$r_{ij} = \begin{cases} a_{ij}^{(i-1)} & , \quad i \le j \\ 0 & , \quad i > j \end{cases}, \quad l_{ij} = \begin{cases} a_{ij}^{(j-1)} / a_{jj}^{(j-1)} & , \quad i > j \\ 1 & , \quad i = j \\ 0 & , \quad i < j \end{cases},$$
$$r_i = a_i^{(i-1)}, \quad i = 1(1)n,$$

und die Lösungen x_i ergeben sich rekursiv aus

$$x_n = \frac{r_n}{r_{nn}},$$
$$x_i = \frac{1}{r_{ii}} \left(r_i - \sum_{j=i+1}^{n} r_{ij} x_j \right), \quad i = n{-}1, n{-}2, \ldots, 1.$$

Während des Eliminationsprozesses können die Elemente von $\boldsymbol{A}$ zeilenweise mit den Elementen der Zerlegungsmatrizen überspeichert werden; die l_{ij} für $i > j$ stehen dann unterhalb der Hauptdiagonale, die $l_{ii} = 1$ werden nicht abgespeichert und die r_{ij} $(i \le j)$ stehen in und über der Hauptdiagonale. Ebenso können die Elemente von $\boldsymbol{a}$ durch die von $\boldsymbol{r}$ überspeichert werden.

Beispiel 4.51.

Gesucht: Die Lösung $\boldsymbol{x}$ des Gleichungssystems $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}$ mit

$$\boldsymbol{A} = \begin{pmatrix} 2 & 1 & 1 \\ 2 & 3 & 0 \\ 4 & -4 & 7 \end{pmatrix}, \qquad \boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}, \qquad \boldsymbol{a} = \begin{pmatrix} 4 \\ 5 \\ 7 \end{pmatrix}$$

Lösung *(ohne Zeilenvertauschung)*: Wegen

$$\det(\boldsymbol{A}_1) = |2| = 2 \neq 0, \quad \det(\boldsymbol{A}_2) = \begin{vmatrix} 2 & 1 \\ 2 & 3 \end{vmatrix} = 4 \neq 0$$

sind die Voraussetzungen des Satzes 4.9 erfüllt. Mit dem vorgenannten Algorithmus 4.42 erhält man

1. Zerlegung $\boldsymbol{A} = \boldsymbol{LR}$: Durch Koeffizientenvergleich erhält man zeilenweise (von oben nach unten)

$$\boldsymbol{A} = \begin{pmatrix} 2 & 1 & 1 \\ 2 & 3 & 0 \\ 4 & -4 & 7 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 2 & -3 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 & 1 \\ 0 & 2 & -1 \\ 0 & 0 & 2 \end{pmatrix} = \boldsymbol{LR}$$

2. Vorwärtselimination $\boldsymbol{Lr} = \boldsymbol{a}$:

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 2 & -3 & 1 \end{pmatrix} \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix} = \begin{pmatrix} 4 \\ 5 \\ 7 \end{pmatrix} \Rightarrow \boldsymbol{r} = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix} = \begin{pmatrix} 4 \\ 1 \\ 2 \end{pmatrix} \quad \downarrow$$

3. Rückwärtselimination $\boldsymbol{Rx} = \boldsymbol{r}$:

$$\begin{pmatrix} 2 & 1 & 1 \\ 0 & 2 & -1 \\ 0 & 0 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 4 \\ 1 \\ 2 \end{pmatrix} \Rightarrow \boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \quad \uparrow$$

Zum Vergleich wird der Gaußsche Algorithmus als Rechenschema angegeben:

	Bezeichnung der Zeilen	$\boldsymbol{A}$			$\boldsymbol{a}$	erfolgte Operationen	Koeffizienten von $\boldsymbol{L}$
$\Rightarrow$	$1^{(0)}$	$\boxed{2}$	1	1	4		
	$2^{(0)}$	2	3	0	5	—	—
	$3^{(0)}$	4	−4	7	7		
$\Rightarrow$	$2^{(1)} = \widetilde{2}^{(1)}$	0	$\textcircled{2}$	−1	1	$-\left(\frac{2}{\boxed{2}}\right) \cdot 1^{(0)} + 2^{(0)}$	$l_{21} = \left(\frac{2}{2}\right) = 1$
	$3^{(1)} = \widetilde{3}^{(1)}$	0	−6	5	−1	$-\left(\frac{4}{\boxed{2}}\right) \cdot 1^{(0)} + 3^{(0)}$	$l_{31} = \left(\frac{4}{2}\right) = 2$
$\Rightarrow$	$3^{(2)}$	0	0	2	2	$-\left(\frac{-6}{\textcircled{2}}\right) \cdot 2^{(1)} + 3^{(1)}$	$l_{32} = \left(\frac{-6}{2}\right) = -3$

Aus den Zeilen $1^{(0)}$, $2^{(1)}$ und $3^{(2)}$ ergeben sich die obere Dreiecksmatrix $\boldsymbol{R}$ sowie die rechte Seite $\boldsymbol{r}$ des Systems $\boldsymbol{R}\,\boldsymbol{x} = \boldsymbol{r}$.

Die Faktoren der Operationen in den Klammern sind die l_{ij}. Man erhält für die Zerlegungsmatrizen $\boldsymbol{L}$ und $\boldsymbol{R}$ und den Vektor $\boldsymbol{r}$:

$$\boldsymbol{L} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 2 & -3 & 1 \end{pmatrix}, \qquad \boldsymbol{R} = \begin{pmatrix} 2 & 1 & 1 \\ 0 & 2 & -1 \\ 0 & 0 & 2 \end{pmatrix}, \qquad \boldsymbol{r} = \begin{pmatrix} 4 \\ 1 \\ 2 \end{pmatrix}$$

so dass aus $\boldsymbol{Rx} = \boldsymbol{r}$ rekursiv die Lösung $\boldsymbol{x} = (1,1,1)^{\mathsf{T}}$ folgt. □

Im folgenden Algorithmus soll die Vorgehensweise der Dreieckszerlegung mit Spaltenpivotsuche unter Verwendung der Überspeicherung der Elemente von $\boldsymbol{A}$ durch die Elemente von $\boldsymbol{L}$ und $\boldsymbol{R}$ formuliert werden.

Algorithmus 4.52. (*Dreieckszerlegung mit Spaltenpivotsuche*)

Gegeben: $\boldsymbol{A} = (a_{ij})$, $i, j = 1(1)n$, $\det \boldsymbol{A} \neq 0$, d. h. $\boldsymbol{A}$ ist regulär.

Gesucht: Dreieckszerlegung $\boldsymbol{PA} = \boldsymbol{LR}$, wobei $\boldsymbol{L}$ und $\boldsymbol{R}$ auf $\boldsymbol{A}$ überspeichert werden.

Dann sind nacheinander folgende Schritte auszuführen:

1. Vorbesetzen des Pivotvektors $\boldsymbol{p} = (p_1, p_2, \ldots, p_n)^\mathsf{T}$ mit $p_i = i$ für alle i.
2. Für jeden Wert $j = 1(1)n-1$ ist durchzuführen:

   2.1 Bestimme $i_0 \geq j$ mit $|a_{i_0 j}| = \max\{|a_{ij}|,\ i = j(1)n\}$ (Pivotsuche) und vertausche p_{i_0} mit p_j und die i_0-te Zeile in $\boldsymbol{A}$ mit der j-ten Zeile. Gilt $a_{jj} = 0$, dann ist $\boldsymbol{A}$ singulär und das Verfahren ist abzubrechen. Andernfalls:

   2.2 Für jedes $i = j + 1(1)n$ ist durchzuführen:

   2.2.1 Ersetze a_{ij} durch a_{ij}/a_{jj} .

   2.2.2 Führe für $k = j + 1(1)n$ durch:
   Ersetze a_{ik} durch $a_{ik} - a_{jk}a_{ij}$.

Dann ist

$$\boldsymbol{P} = (\boldsymbol{e}_{p_1}, \boldsymbol{e}_{p_2}, \ldots, \boldsymbol{e}_{p_n})^\mathsf{T},$$

$$\boldsymbol{L} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ a_{21} & 1 & & 0 \\ \vdots & & \ddots & \vdots \\ a_{n1} & \cdots & a_{n,n-1} & 1 \end{pmatrix}, \quad \boldsymbol{R} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a_{22} & \cdots & a_{2n} \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & & a_{nn} \end{pmatrix};$$

$\boldsymbol{e}_{p_j}$ ist der p_j-te Standard-Einheitsvektor mit einer 1 in der p_j-ten Komponente.

Algorithmus 4.53. (*Gauß-Algorithmus mit Spaltenpivotsuche*)

Gegeben: $\boldsymbol{Ax} = \boldsymbol{a}$, $\det \boldsymbol{A} \neq 0$.

Gesucht: Lösung $\boldsymbol{x}$.

1. Schritt: Bestimmung des Pivotvektors $\boldsymbol{p}$ und der Dreiecksmatrizen $\boldsymbol{L}$ und $\boldsymbol{R}$ nach Algorithmus 4.52.

2. Schritt: Berechnung von $\boldsymbol{r} = (r_1, r_2, \ldots, r_n)^\mathsf{T}$ durch Vorwärtselimination aus $\boldsymbol{Pa} = \boldsymbol{Lr}$ mit

$$\begin{aligned} \boldsymbol{Pa} &= (a_{p_1}, a_{p_2}, \ldots, a_{p_n})^\mathsf{T} \quad \text{nach der Vorschrift} \\ r_1 &= a_{p_1}, \quad r_i = a_{p_i} - \sum_{j=1}^{i-1} a_{ij} r_j \quad \text{für} \quad i = 2(1)n. \end{aligned}$$

3. Schritt: Berechnung der Lösung $\boldsymbol{x}$ aus $\boldsymbol{Rx} = \boldsymbol{r}$ durch Rückwärtselimination mit

$$\begin{aligned} x_n &= \tfrac{r_n}{a_{nn}}, \\ x_i &= \tfrac{1}{a_{ii}} \left(r_i - \sum_{j=i+1}^{n} a_{ij} x_j \right) \quad \text{für} \quad i = n{-}1, n{-}2, \ldots, 1. \end{aligned}$$

Ganz analog lassen sich die letzten beiden Algorithmen unter Verwendung der skalierten Spaltenpivotsuche formulieren. Dann ist lediglich noch die in Abschnitt 4.5.2 angegebene Skalierung gemäß Formel (4.10) zu beachten.

Beispiel 4.54.

Gesucht: Wiederum die Lösung $\boldsymbol{x}$ des Gleichungssystems $\boldsymbol{Ax} = \boldsymbol{a}$ mit

$$\boldsymbol{A} = \begin{pmatrix} 3 & 3 & 6 \\ 2 & 2 & 3 \\ 1 & 0 & 1 \end{pmatrix}, \qquad \boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}, \qquad \boldsymbol{a} = \begin{pmatrix} -3 \\ 1 \\ 0 \end{pmatrix}$$

Lösung durch den Gauß-Algorithmus mit Spaltenpivotsuche.

	Bezeichnung der Zeilen	$\boldsymbol{A}$			$\boldsymbol{a}$	erfolgte Operationen	Koeffizienten von $\boldsymbol{L}$
$\Rightarrow$	$1^{(0)}$	$\textcircled{3}$	3	6	-3		
	$2^{(0)}$	$\boxed{2}$	2	3	1	—	—
	$3^{(0)}$	$\boxed{1}$	0	1	0		
	$\widetilde{2}^{(1)}$	0	0	-1	3	$-\left(\frac{\boxed{2}}{\textcircled{3}}\right) \cdot 1^{(0)} + 2^{(0)}$	
	$\widetilde{3}^{(1)}$	0	-1	-1	1	$-\left(\frac{\boxed{1}}{\textcircled{3}}\right) \cdot 1^{(0)} + 3^{(0)}$	—
$\Rightarrow$	$2^{(1)}$	0	$\textcircled{-1}$	-1	1	Vertauschung von	$l_{21} = \left(\frac{1}{3}\right) = \frac{1}{3}$
	$3^{(1)}$	0	0	-1	3	2. und 3. Zeile	$l_{31} = \left(\frac{2}{3}\right) = \frac{2}{3}$
$\Rightarrow$	$3^{(2)}$	0	0	-1	3	$-\left(\frac{0}{\textcircled{-1}}\right) \cdot 2^{(1)} + 3^{(1)}$	$l_{32} = \left(\frac{0}{-1}\right) = 0$

Daraus ergeben sich die Zerlegungsmatrizen $\boldsymbol{L}$ und $\boldsymbol{R}$ sowie der Vektor $\boldsymbol{r}$

$$\boldsymbol{L} = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{3} & 1 & 0 \\ \frac{2}{3} & 0 & 1 \end{pmatrix}, \qquad \boldsymbol{R} = \begin{pmatrix} 3 & 3 & 6 \\ 0 & -1 & -1 \\ 0 & 0 & -1 \end{pmatrix}, \qquad \boldsymbol{r} = \begin{pmatrix} -3 \\ 1 \\ 3 \end{pmatrix}$$

so dass sich aus $\boldsymbol{R}\,\boldsymbol{x} = \boldsymbol{r}$ rekursiv die Lösung $\boldsymbol{x}^{\mathsf{T}} = (3, 2, -3)$ errechnen lässt. Wegen der Zeilenvertauschung gilt auch die Zerlegung

$$\begin{aligned} \boldsymbol{LR} &= \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{3} & 1 & 0 \\ \frac{2}{3} & 0 & 1 \end{pmatrix} \begin{pmatrix} 3 & 3 & 6 \\ 0 & -1 & -1 \\ 0 & 0 & -1 \end{pmatrix} = \begin{pmatrix} 3 & 3 & 6 \\ 1 & 0 & 1 \\ 2 & 2 & 3 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 3 & 3 & 6 \\ 2 & 2 & 3 \\ 1 & 0 & 1 \end{pmatrix} = \boldsymbol{PA} \end{aligned}$$

mit einer Permutationsmatrix $\boldsymbol{P}$, die aus der Einheitsmatrix $\boldsymbol{E}$ durch Vertauschung der 2. und 3. Zeile entsteht.

Der Vektor $\boldsymbol{r}$ ergibt sich aus

$$\boldsymbol{P}\,\boldsymbol{a} = \begin{pmatrix} -3 \\ 0 \\ 1 \end{pmatrix} = \boldsymbol{L}\,\boldsymbol{r} = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{3} & 1 & 0 \\ \frac{2}{3} & 0 & 1 \end{pmatrix} \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix} \quad \downarrow$$

durch Vorwärtselimination: $\boldsymbol{r}^{\mathsf{T}} = (-3, 1, 3)$. □

4.5.4 Gauß-Algorithmus für Systeme mit mehreren rechten Seiten

Liegen Systeme mit gleicher Matrix $\boldsymbol{A}$ und m rechten Seiten $\boldsymbol{a}_j$, $j = 1(1)m$, vor, so kann man statt $\boldsymbol{A}\boldsymbol{x}_j = \boldsymbol{a}_j$ schreiben

$$\boldsymbol{A}\boldsymbol{X} = \boldsymbol{A}^* \quad \text{mit} \quad \boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m) \text{ und } \boldsymbol{A}^* = (\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_m),$$

wobei $\boldsymbol{A} = (\boldsymbol{a}_{ik})$, $i, k = 1(1)n$, die gemeinsame Matrix der m Systeme ist, $\boldsymbol{X}$ die (n, m)-Matrix, die spaltenweise aus den m Lösungsvektoren $\boldsymbol{x}_j$, $j = 1(1)m$, aufgebaut ist und $\boldsymbol{A}^*$ die (n, m)-Matrix, deren m Spalten die m rechten Seiten $\boldsymbol{a}_j$ sind. Die Dreieckszerlegung der Matrix $\boldsymbol{A}$ braucht also nur einmal für alle m Systeme durchgeführt zu werden, während Vorwärts- und Rückwärtselimination m-mal zu machen sind. Es sind $n^3/3 - n/3 + mn^2$ Punktoperationen erforderlich. Zusammengefasst ergibt sich damit folgender

Algorithmus 4.55. (*Ohne Spaltenpivotsuche*)

Gegeben: $\boldsymbol{A}\boldsymbol{X} = \boldsymbol{A}^*$, $\det(\boldsymbol{A}_k) \neq 0$ für $k = 1(1)n{-}1$, $\boldsymbol{A}^* = (\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_m)$.
Gesucht: $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m)$

1. Schritt: Faktorisierung $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{R}$ gemäß Satz 4.9.
2. Schritt: Vorwärtselimination $\boldsymbol{A}^* = \boldsymbol{L}\boldsymbol{R}^*$ zur Berechnung von $\boldsymbol{R}^*$.
3. Schritt: Rückwärtselimination $\boldsymbol{R}\boldsymbol{X} = \boldsymbol{R}^*$ zur Berechnung von $\boldsymbol{X}$.

Mit Spaltenpivotsuche muss analog mit der Permutationsmatrix $\boldsymbol{P}$ (vgl. Satz 4.12 bzw. Algorithmus 4.44) multipliziert werden.

Das folgende Beispiel wird mit Spaltenpivotsuche im Gauß-Schema gerechnet.

Beispiel 4.56.

Gegeben sind $\boldsymbol{A}\,\boldsymbol{x}_1 = \boldsymbol{a}_1$ und $\boldsymbol{A}\,\boldsymbol{x}_2 = \boldsymbol{a}_2$ mit

$$\boldsymbol{A} = \begin{pmatrix} 3 & 3 & 6 \\ 2 & 2 & 3 \\ 1 & 0 & 1 \end{pmatrix}, \quad \boldsymbol{a}_1 = \begin{pmatrix} -3 \\ 1 \\ 0 \end{pmatrix}, \quad \boldsymbol{a}_2 = \begin{pmatrix} -6 \\ -3 \\ -2 \end{pmatrix}$$

	Bezeichnung der Zeilen	$\boldsymbol{A}$			$\boldsymbol{a}_1$	$\boldsymbol{a}_2$	erfolgte Operationen
$\Rightarrow$	$1^{(0)}$	3	3	6	-3	-6	
	$2^{(0)}$	2	2	3	1	-3	—
	$3^{(0)}$	1	0	1	0	-2	
	$\widetilde{2}^{(1)}$	0	0	-1	3	1	$-\left(\frac{2}{3}\right) \cdot 1^{(0)} + 2^{(0)}$
	$\widetilde{3}^{(1)}$	0	-1	-1	1	0	$-\left(\frac{1}{3}\right) \cdot 1^{(0)} + 3^{(0)}$
$\Rightarrow$	$2^{(1)}$	0	-1	-1	1	0	$-\left(\frac{1}{3}\right) \cdot 1^{(0)} + 3^{(0)}$
	$3^{(1)}$	0	0	-1	3	1	$-\left(\frac{2}{3}\right) \cdot 1^{(0)} + 2^{(0)}$
$\Rightarrow$	$3^{(2)}$	0	0	-1	3	1	$-\left(\frac{0}{-1}\right) \cdot 2^{(1)} + 3^{(1)}$

Es ergeben sich aus $1^{(0)}$, $2^{(1)}$ und $3^{(2)}$:

$$\boldsymbol{R} = \begin{pmatrix} 3 & 3 & 6 \\ 0 & -1 & -1 \\ 0 & 0 & -1 \end{pmatrix}, \quad \boldsymbol{r}_1 = \begin{pmatrix} -3 \\ 1 \\ 3 \end{pmatrix}, \quad \boldsymbol{r}_2 = \begin{pmatrix} -6 \\ 0 \\ 1 \end{pmatrix}$$

Die Lösungen ergeben sich aus

$$\boldsymbol{R}\,\boldsymbol{x}_1 = \boldsymbol{r}_1 \quad \text{und} \quad \boldsymbol{R}\,\boldsymbol{x}_2 = \boldsymbol{r}_2$$

bzw. verkürzt geschrieben

$$\boldsymbol{R}(\boldsymbol{x}_1, \boldsymbol{x}_2) = (\boldsymbol{r}_1, \boldsymbol{r}_2) \quad \text{bzw.} \quad \boldsymbol{R}\boldsymbol{X} = \boldsymbol{R}^*\,,$$

also

$$\begin{pmatrix} 3 & 3 & 6 \\ 0 & -1 & -1 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} x_{1,1} & x_{1,2} \\ x_{2,1} & x_{2,2} \\ \underbrace{x_{3,1}}_{\boldsymbol{x}_1} & \underbrace{x_{3,2}}_{\boldsymbol{x}_2} \end{pmatrix} = \begin{pmatrix} -3 & -6 \\ 1 & 0 \\ \underbrace{3}_{\boldsymbol{r}_1} & \underbrace{1}_{\boldsymbol{r}_2} \end{pmatrix}$$

Daraus ergeben sich rekursiv die Lösungen

$$\boldsymbol{x}_1 = \begin{pmatrix} 3 \\ 2 \\ -3 \end{pmatrix}, \quad \boldsymbol{x}_2 = \begin{pmatrix} -1 \\ 1 \\ -1 \end{pmatrix}$$

□

4.6 Matrizeninversion mit dem Gauß-Algorithmus

Gegeben seien n lineare Gleichungssysteme aus n Gleichungen mit n Unbekannten

$$\boldsymbol{A}\boldsymbol{x}_i = \boldsymbol{e}_i, \qquad i = 1(1)n,$$

mit $\det \boldsymbol{A} \neq 0$, $\boldsymbol{e}_i$ i-ter Einheitsvektor. Fasst man die n rechten Seiten $\boldsymbol{e}_i$ zu der Einheitsmatrix $\boldsymbol{E}$ zusammen und die n Lösungsvektoren $\boldsymbol{x}_i$ zu einer Matrix $\boldsymbol{X}$, so lassen sich die n Systeme kompakt in der Form $\boldsymbol{A}\boldsymbol{X} = \boldsymbol{E}$ schreiben.

Daraus resultiert gemäß Definition der Inversen: $\boldsymbol{X} = \boldsymbol{A}^{-1}$, d. h. die n Lösungsvektoren $\boldsymbol{x}_i$ der n Systeme $\boldsymbol{A}\boldsymbol{x}_i = \boldsymbol{e}_i$ bauen spaltenweise die inverse Matrix $\boldsymbol{A}^{-1}$ auf. Man gewinnt $\boldsymbol{A}^{-1}$, indem man die n Systeme mit dem Gaußschen Algorithmus löst; auch hier ist die Spaltenpivotsuche unerlässlich. Es sind $4n^3/3 - n/3$ Punktoperationen erforderlich.

Algorithmus 4.57. (*Ohne Spaltenpivotsuche*)

Gegeben: $\boldsymbol{A}\boldsymbol{X} = \boldsymbol{E}$, $\boldsymbol{A} = (a_{ik})$, $i, k = 1(1)n$, streng regulär.

Gesucht: $\boldsymbol{X} = \boldsymbol{A}^{-1}$.

1. Schritt: Faktorisierung $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{R}$ gemäß Satz 4.9 liefert $\boldsymbol{L}$ und $\boldsymbol{R}$.
2. Schritt: Vorwärtselimination $\boldsymbol{E} = \boldsymbol{L}\boldsymbol{R}^*$ liefert $\boldsymbol{R}^*$.
3. Schritt: Rückwärtselimination $\boldsymbol{R}\boldsymbol{X} = \boldsymbol{R}^*$ liefert $\boldsymbol{X} = \boldsymbol{A}^{-1}$.

Algorithmus 4.58. (*Mit Spaltenpivotsuche*)

Gegeben: $\boldsymbol{AX} = \boldsymbol{E}$, $\boldsymbol{A} = (a_{ik})$, $i, k = 1(1)n$, $\boldsymbol{A}$ regulär.

Gesucht: $\boldsymbol{X} = \boldsymbol{A}^{-1}$.

1. Schritt: Faktorisierung $\boldsymbol{PA} = \boldsymbol{LR}$ gemäß Satz 4.12 bzw. Algorithmus 4.52 liefert $\boldsymbol{L}$ und $\boldsymbol{R}$.
2. Schritt: Vorwärtselimination $\boldsymbol{PE} = \boldsymbol{LR}^*$ liefert $\boldsymbol{R}^*$.
3. Schritt: Rückwärtselimination $\boldsymbol{RX} = \boldsymbol{R}^*$ liefert $\boldsymbol{X} = \boldsymbol{A}^{-1}$.

Man sollte dieses Verfahren nur anwenden, wenn $\boldsymbol{A}^{-1}$ explizit gebraucht wird. Auf keinen Fall sollte es zur Lösung von m Systemen $\boldsymbol{Ax}_i = \boldsymbol{y}_i$, $i = 1(1)m$, durch $\boldsymbol{x}_i = \boldsymbol{A}^{-1}\boldsymbol{y}_i$ verwendet werden, weil dann $4n^3/3 - n/3 + mn^2$ Punktoperationen benötigt werden im Gegensatz zu $n^3/3 - n/3 + mn^2$ bei Anwendung des in Abschnitt 4.5.4 angegebenen Verfahrens.

Beispiel 4.59.

Gegeben: Die Matrix $\boldsymbol{A} = \begin{pmatrix} 2 & 3 \\ 1 & 4 \end{pmatrix}$.

Gesucht: Die Inverse $\boldsymbol{A}^{-1}$.

Lösung:

Rechenschema

Bezeichnung der Zeilen	$\boldsymbol{A}$	$\boldsymbol{E}$	erfolgte Operationen
$1^{(0)}$	$2 \quad 3$	$1 \quad 0$	
$2^{(0)}$	$1 \quad 4$	$0 \quad 1$	—
$2^{(1)}$	$0 \quad +\frac{5}{2}$	$-\frac{1}{2} \quad 1$	$-\frac{1}{2} \cdot 1^{(0)} + 2^{(0)}$

Gestaffeltes System

$$\underbrace{\begin{pmatrix} 2 & 3 \\ 0 & \frac{5}{2} \end{pmatrix}}_{\boldsymbol{R}} \underbrace{\begin{pmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{pmatrix}}_{\boldsymbol{A}^{-1}} = \underbrace{\begin{pmatrix} 1 & 0 \\ -\frac{1}{2} & 1 \end{pmatrix}}_{\boldsymbol{R}^*} \Rightarrow \boldsymbol{A}^{-1} = \frac{1}{5}\begin{pmatrix} 4 & -3 \\ -1 & 2 \end{pmatrix}.$$

□

Beispiel 4.60.

Gegeben: Die Matrix

$$\boldsymbol{A} = \begin{pmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{pmatrix}.$$

Gesucht: Die Inverse $\boldsymbol{A}^{-1}$.

Lösung: **Rechenschema**

Bezeichnung der Zeilen	$\boldsymbol{A}$				$\boldsymbol{E}$				erfolgte Operationen
$1^{(0)}$	2	-1	0	0	1	0	0	0	
$2^{(0)}$	-1	2	-1	0	0	1	0	0	
$3^{(0)}$	0	-1	2	-1	0	0	1	0	—
$4^{(0)}$	0	0	-1	2	0	0	0	1	
$2^{(1)}$	0	$\frac{3}{2}$	-1	0	$\frac{1}{2}$	1	0	0	$-\left(\frac{-1}{2}\right)\cdot 1^{(0)}+2^{(0)}$
$3^{(1)}$	0	-1	2	-1	0	0	1	0	
$4^{(1)}$	0	0	-1	2	0	0	0	1	
$3^{(2)}$	0	0	$\frac{4}{3}$	-1	$\frac{1}{3}$	$\frac{2}{3}$	1	0	$-\left(\frac{-2}{3}\right)\cdot 2^{(1)}+3^{(1)}$
$4^{(2)}$	0	0	-1	2	0	0	0	1	
$4^{(3)}$	0	0	0	$\frac{5}{4}$	$\frac{1}{4}$	$\frac{2}{4}$	$\frac{3}{4}$	1	$-\left(\frac{-3}{4}\right)\cdot 3^{(2)}+4^{(2)}$

Gestaffeltes System

$$\underbrace{\begin{pmatrix} 2 & -1 & 0 & 0 \\ 0 & \frac{3}{2} & -1 & 0 \\ 0 & 0 & \frac{4}{3} & -1 \\ 0 & 0 & 0 & \frac{5}{4} \end{pmatrix}}_{\boldsymbol{R}} \underbrace{\begin{pmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \\ x_{41} & x_{42} & x_{43} & x_{44} \end{pmatrix}}_{\boldsymbol{A}^{-1}} = \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{2} & 1 & 0 & 0 \\ \frac{1}{3} & \frac{2}{3} & 1 & 0 \\ \frac{1}{4} & \frac{2}{4} & \frac{3}{4} & 1 \end{pmatrix}}_{\boldsymbol{R}^*}$$

$$\Rightarrow \boldsymbol{A}^{-1} = \frac{1}{5}\begin{pmatrix} 4 & 3 & 2 & 1 \\ 3 & 6 & 4 & 2 \\ 2 & 4 & 6 & 3 \\ 1 & 2 & 3 & 4 \end{pmatrix}.$$

□

4.7 Verfahren für Systeme mit symmetrischen Matrizen

Ist die Matrix $\boldsymbol{A} = (a_{ik})$, $i, k = 1(1)n$, in $\boldsymbol{Ax} = \boldsymbol{a}$ symmetrisch ($\boldsymbol{A} = \boldsymbol{A}^{\mathsf{T}}$), so genügt es, die Elemente des oberen Dreiecks einschließlich der Diagonale zu speichern. Für Systeme mit symmetrischen, streng regulären Matrizen ($\det \boldsymbol{A}_k \neq 0$ für alle k) und symmetrischen, positiv definiten (und damit streng regulären) Matrizen, für die $\det(\boldsymbol{A}_k) > 0$ für alle k gilt, werden im Folgenden Lösungsalgorithmen angegeben, die auf den Sätzen 4.34,

4.37 und 4.38 aufbauen. Die Verfahren haben nur Sinn ohne Spaltenpivotsuche. Deshalb ist strenge Regularität stets Voraussetzung.

4.7.1 Systeme mit symmetrischer, streng regulärer Matrix

Algorithmus 4.61. (*Ohne Spaltenpivotsuche*)

Gegeben: $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}$ mit symmetrischer, streng regulärer Matrix $\boldsymbol{A} = (a_{ik})$, $i, k = 1(1)n$, und rechter Seite $\boldsymbol{a} = (a_i)$, $i = 1(1)n$.

Gesucht: $\boldsymbol{x} = (x_i)$, $i = 1(1)n$.

1. Schritt: (Faktorisierung $\boldsymbol{A} = \boldsymbol{R}^{\mathsf{T}}\boldsymbol{D}\boldsymbol{R}$ mit normierter oberer Dreiecksmatrix $\boldsymbol{R}$ und Diagonalmatrix $\boldsymbol{D}$; es ergeben sich $\boldsymbol{D}$ und $\boldsymbol{R}$ bzw. $\boldsymbol{R}^{\mathsf{T}}$)

1.1 $d_{11} = a_{11}$

$r_{1k} = \frac{a_{1k}}{d_{11}}\,, \quad k = 2(1)n$

1.2 Für jedes $i = 2(1)n{-}1$

1.2.1 $d_{ii} = a_{ii} - \sum\limits_{j=1}^{i-1} d_{jj}\, r_{ji}^2$

1.2.2 $r_{i,i+1} = \frac{1}{d_{ii}}\left(a_{i,i+1} - \sum\limits_{j=1}^{i-1} d_{jj}\, r_{j,j+1}\, r_{j,j+2}\right)$

1.3 $d_{nn} = a_{nn} - \sum\limits_{j=1}^{n-1} d_{jj}\, r_{jn}^2$

2. Schritt: (Vorwärtselimination $\boldsymbol{R}^{\mathsf{T}}\boldsymbol{z} = \boldsymbol{a} \Rightarrow \boldsymbol{z}$ und $\boldsymbol{D}\boldsymbol{r} = \boldsymbol{z} \Rightarrow \boldsymbol{r}$)

2.1 $z_1 = \frac{a_1}{r_{11}}$

2.2 Für jedes $j = 2(1)n$

$z_j = \left(a_j - \sum\limits_{i=1}^{j-1} r_{ij}\, z_i\right)\frac{1}{r_{jj}}$

2.3 Für jedes $j = 1(1)n$

$r_j = \frac{z_j}{d_{jj}}$

3. Schritt: (Rückwärtselimination $\boldsymbol{R}\boldsymbol{x} = \boldsymbol{r}$)

3.1 $x_n = \frac{r_n}{r_{nn}}$

3.2 Für jedes $i = n{-}1(-1)1$

$x_i = \frac{1}{r_{ii}}\left(r_i - \sum\limits_{k=i+1}^{n} r_{ik} x_k\right)$

Die Durchführung des Verfahrens verläuft analog zu Algorithmus 4.42. Hier ergeben sich auch negative Hauptdiagonalelemente der Matrix $\boldsymbol{D}$ (siehe auch Beispiel 4.43).

4.7.2 Systeme mit symmetrischer, positiv definiter Matrix – Cholesky-Verfahren

Ist die Matrix $\boldsymbol{A}$ in $\boldsymbol{Ax} = \boldsymbol{a}$ symmetrisch ($a_{ik} = a_{ki}$) und positiv definit ($\boldsymbol{x}^\mathsf{T}\boldsymbol{Ax} > 0$ für alle $\boldsymbol{x} \neq \boldsymbol{0}$), so kann das Cholesky-Verfahren angewandt werden; es benötigt asymptotisch nur halb so viele Punktoperationen wie der Gaußsche Algorithmus und ca. halb so viel Speicherplatz. Im Anschluss werden zwei Darstellungsformen angegeben, die erste mit der Zerlegung $\boldsymbol{A} = \boldsymbol{R}^\mathsf{T}\boldsymbol{R}$, die zweite mit der Zerlegung $\boldsymbol{A} = \boldsymbol{R}^\mathsf{T}\boldsymbol{DR}$. Die zweite Form ist numerisch günstiger, da die Berechnung von Quadratwurzeln vermieden wird.

Prinzip des Verfahrens für die 1. Darstellungsform mit der Cholesky-Zerlegung $\boldsymbol{A} = \boldsymbol{R}^\mathsf{T}\boldsymbol{R}$

Mit der Zerlegung $\boldsymbol{A} = \boldsymbol{R}^\mathsf{T}\boldsymbol{R}$, wo $\boldsymbol{R} = (r_{ik})$ eine obere Dreiecksmatrix mit $r_{ii} > 0$ ist, wird das System $\boldsymbol{Ax} = \boldsymbol{a}$ in ein äquivalentes System $\boldsymbol{Rx} = \boldsymbol{r}$ überführt in folgenden Schritten:

1. (Faktorisierung) $\boldsymbol{A} = \boldsymbol{R}^\mathsf{T}\boldsymbol{R} \Rightarrow \boldsymbol{R}$,
2. (Vorwärtselimination) $\boldsymbol{a} = \boldsymbol{R}^\mathsf{T}\boldsymbol{r} \Rightarrow \boldsymbol{r}$,
3. (Rückwärtselimination) $\boldsymbol{Rx} = \boldsymbol{r} \Rightarrow \boldsymbol{x}$.

Durchführung des Verfahrens

Die Elemente der Matrizen $\boldsymbol{A}, \boldsymbol{R}, \boldsymbol{R}^\mathsf{T}$ und des Vektors $\boldsymbol{r}$ werden wie folgt bezeichnet

$$\boldsymbol{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & \ddots & \vdots \\ a_{n1} & \cdots & a_{n,n-1} & a_{nn} \end{pmatrix} = \boldsymbol{A}^\mathsf{T}, \text{ d. h. } a_{ik} = a_{ki}$$

$$\boldsymbol{R} = \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ 0 & r_{22} & \cdots & r_{2n} \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & & r_{nn} \end{pmatrix}, \; r_{ii} > 0$$

$$\boldsymbol{R}^\mathsf{T} = \begin{pmatrix} r_{11} & 0 & \cdots & 0 \\ r_{12} & r_{22} & & 0 \\ \vdots & & \ddots & \vdots \\ r_{1n} & \cdots & r_{1,n-1} & r_{nn} \end{pmatrix}, \; r_{ii} > 0 \qquad \boldsymbol{r} = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{pmatrix}.$$

Aus dem Koeffizientenvergleich $\boldsymbol{A} = \boldsymbol{R}^\mathsf{T}\boldsymbol{R}$ ergeben sich zeilenweise die Formeln für die r_{ik}. Aus dem Koeffizientenvergleich $\boldsymbol{a} = \boldsymbol{R}^\mathsf{T}\boldsymbol{r}$ ergeben sich die r_i durch Vorwärtselimination und aus dem Koeffizientenvergleich $\boldsymbol{Rx} = \boldsymbol{r}$ ergeben sich die x_i durch Rückwärtselimination. Man erhält den

Algorithmus 4.62.

Gegeben: $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}$ mit symmetrischer, positiv definiter Matrix $\boldsymbol{A} = (a_{ik})$, $i, k = 1(1)n$, $n \geq 2$, $\boldsymbol{a} = (a_i)$, $i = 1(1)n$.

Gesucht: $\boldsymbol{x} = (x_i)$, $i = 1(1)n$.

Dann sind nacheinander folgende Schritte auszuführen:

1. Schritt: (Zerlegung $\boldsymbol{A} = \boldsymbol{R}^{\mathsf{T}}\boldsymbol{R}$)

1.1 $r_{11} = \sqrt{a_{11}}$

1.2 Für jedes $j = 2(1)n-1$

1.2.1 $r_{jj} = \sqrt{a_{jj} - \sum\limits_{i=1}^{j-1} r_{ij}^2}$

1.2.2 Für jedes $k = j+1(1)n$

$$r_{jk} = \frac{1}{r_{jj}}\left(a_{jk} - \sum_{i=1}^{j-1} r_{ik}\, r_{ij}\right)$$

1.3 $r_{nn} = \sqrt{a_{nn} - \sum\limits_{i=1}^{n-1} r_{in}^2}$

2. Schritt: (Vorwärtselimination $\boldsymbol{a} = \boldsymbol{R}^{\mathsf{T}}\boldsymbol{r}$)

2.1 $r_1 = \frac{a_1}{r_{11}}$

2.2 Für jedes $j = 2(1)n$

$$r_j = \left(a_j - \sum_{i=1}^{j-1} r_{ij}\, r_i\right)\frac{1}{r_{jj}}$$

3. Schritt: (Rückwärtselimination $\boldsymbol{R}\boldsymbol{x} = \boldsymbol{r}$)

3.1 $x_n = \frac{r_n}{r_{nn}}$

3.2 Für jedes $i = n-1(-1)1$

$$x_i = \frac{1}{r_{ii}}\left(r_i - \sum_{k=i+1}^{n} r_{ik}x_k\right)$$

Für die Determinante von $\boldsymbol{A}$ gilt:

$$\det \boldsymbol{A} = \det(\boldsymbol{R}^{\mathsf{T}}) \det \boldsymbol{R} = (r_{11}r_{22}\ldots r_{nn})^2.$$

Beispiel 4.63.

Gegeben: Das System $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}$ mit

$$\boldsymbol{A} = \begin{pmatrix} 2 & 0 & -1 \\ 0 & 2 & -1 \\ -1 & -1 & 2 \end{pmatrix}, \qquad \boldsymbol{a} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}.$$

Prüfung der Voraussetzungen für die Anwendung des Cholesky-Verfahrens:
Wegen $a_{ik} = a_{ki}$ ist $\boldsymbol{A} = \boldsymbol{A}^{\mathsf{T}}$, also $\boldsymbol{A}$ symmetrisch. Außerdem sind sämtliche Hauptabschnittsdeterminanten (vgl. Definition 4.1)

$$\det(\boldsymbol{A}_1) = |2| = 2 > 0\,, \qquad \det(\boldsymbol{A}_2) = \begin{vmatrix} 2 & 0 \\ 0 & 2 \end{vmatrix} = 4 > 0\,, \qquad \det \boldsymbol{A} = 4 > 0$$

positiv, d. h. $\boldsymbol{A}$ ist positiv definit.

1. Weg: Lösung des Systems $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}$ mit positiv definiter Matrix $\boldsymbol{A}$ gemäß Alg. 4.62

1. Schritt: $\boldsymbol{A} = \boldsymbol{R}^{\mathsf{T}}\boldsymbol{R}$ (Zerlegung)

$$\begin{pmatrix} 2 & 0 & -1 \\ 0 & 2 & -1 \\ -1 & -1 & 2 \end{pmatrix} = \begin{pmatrix} \sqrt{2} & 0 & 0 \\ 0 & \sqrt{2} & 0 \\ -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 1 \end{pmatrix} \begin{pmatrix} \sqrt{2} & 0 & -\frac{1}{\sqrt{2}} \\ 0 & \sqrt{2} & -\frac{1}{\sqrt{2}} \\ 0 & 0 & 1 \end{pmatrix}.$$

2. Schritt: $\boldsymbol{a} = \boldsymbol{R}^{\mathsf{T}}\boldsymbol{r}$ (Vorwärtselimination)

$$\begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \sqrt{2} & 0 & -\frac{1}{\sqrt{2}} \\ 0 & \sqrt{2} & -\frac{1}{\sqrt{2}} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 1 \end{pmatrix}.$$

3. Schritt: $\boldsymbol{R}\boldsymbol{x} = \boldsymbol{r}$ (Rückwärtselimination)

$$\begin{pmatrix} \sqrt{2} & 0 & -\frac{1}{2}\sqrt{2} \\ 0 & \sqrt{2} & -\frac{1}{2}\sqrt{2} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} \frac{1}{2}\sqrt{2} \\ \frac{1}{2}\sqrt{2} \\ 1 \end{pmatrix},$$

woraus sich rekursiv die Lösung $x_3 = x_2 = x_1 = 1$ ergibt.

Die Lösung des Systems $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}$ mit dem Cholesky-Verfahren gemäß Algorithmus 4.62 kann verkürzt wie folgt geschrieben werden

$$(\boldsymbol{A}, \boldsymbol{a}) = (\boldsymbol{R}^{\mathsf{T}}\boldsymbol{R}, \boldsymbol{R}^{\mathsf{T}}\boldsymbol{r}) = \boldsymbol{R}^{\mathsf{T}}(\boldsymbol{R}, \boldsymbol{r})\,.$$

Durch zeilenweisen Koeffizientenvergleich der Matrix $(\boldsymbol{A}, \boldsymbol{a})$ mit dem Matrizenprodukt $\boldsymbol{R}^{\mathsf{T}}(\boldsymbol{R}, \boldsymbol{r})$ können die Matrix $\boldsymbol{R}$ und der Vektor $\boldsymbol{r}$ berechnet werden, danach aus $\boldsymbol{R}\boldsymbol{x} = \boldsymbol{r}$ die Lösung.

$$\underbrace{\left(\begin{array}{ccc|c} 2 & 0 & -1 & 1 \\ 0 & 2 & -1 & 1 \\ -1 & -1 & 2 & 0 \end{array}\right)}_{\boldsymbol{A}\quad\ \boldsymbol{a}} = \underbrace{\begin{pmatrix} \sqrt{2} & 0 & 0 \\ 0 & \sqrt{2} & 0 \\ -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 1 \end{pmatrix}}_{\boldsymbol{R}^{\mathsf{T}}} \underbrace{\left(\begin{array}{ccc|c} \sqrt{2} & 0 & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & \sqrt{2} & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & 0 & 1 & 1 \end{array}\right)}_{\boldsymbol{R}\quad\ \boldsymbol{r}}.$$

Das gestaffelte System $\boldsymbol{R}\boldsymbol{x} = \boldsymbol{r}$ lautet

$$\begin{pmatrix} \sqrt{2} & 0 & -\frac{1}{2}\sqrt{2} \\ 0 & \sqrt{2} & -\frac{1}{2}\sqrt{2} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} \frac{1}{2}\sqrt{2} \\ \frac{1}{2}\sqrt{2} \\ 1 \end{pmatrix}.$$

□

2. Weg: Prinzip des Verfahrens in der 2. Darstellungsform mit der Zerlegung $\boldsymbol{A} = \boldsymbol{R}^{\mathsf{T}} \boldsymbol{D} \boldsymbol{R}$

Mit der Zerlegung $\boldsymbol{A} = \boldsymbol{R}^{\mathsf{T}} \boldsymbol{D} \boldsymbol{R}$, wobei $\boldsymbol{D}$ eine Diagonalmatrix und $\boldsymbol{R}$ eine normierte obere Dreiecksmatrix ist, wird das System $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}$ gemäß Algorithmus 4.61 in ein äquivalentes System $\boldsymbol{R}\boldsymbol{x} = \boldsymbol{r}$ übergeführt. Die Zerlegung wird so vorgenommen, dass die Anzahl der Punktoperationen wie in der 1. Darstellung $n^3/6 + O(n^2)$ ist.

Durchführung des Verfahrens

Es werden folgende Bezeichnungen benutzt:

$$\boldsymbol{D} = \begin{pmatrix} d_1 & & & & \\ & d_2 & & & \\ & & \ddots & & \\ & & & d_{n-1} & \\ & & & & d_n \end{pmatrix}, \quad \boldsymbol{R} = \begin{pmatrix} 1 & r_{12} & & \cdots & r_{1n} \\ & 1 & r_{23} & \cdots & r_{2n} \\ & & \ddots & \ddots & \vdots \\ & & & 1 & r_{n-1,n} \\ & & & & 1 \end{pmatrix}, \quad \boldsymbol{r} = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{pmatrix}, \quad \boldsymbol{z} = \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{pmatrix}.$$

Algorithmus 4.64.

Gegeben: $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}$ mit symmetrischer, positiv definiter Matrix $\boldsymbol{A} = (a_{ik})$, $i, k = 1(1)n$, $n \geq 2$, $\boldsymbol{a} = (a_i)$, $i = 1(1)n$.

Gesucht: $\boldsymbol{x} = (x_i)$, $i = 1(1)n$.

Dann sind nacheinander folgende Schritte auszuführen:

1. (Faktorisierung $\boldsymbol{A} = \boldsymbol{R}^{\mathsf{T}} \boldsymbol{D} \boldsymbol{R}$)

   1.1 $d_1 = a_{11}$

   1.2 Für jedes $j = 1(1)n$:

   1.2.1 Für jedes $i = 1(1)j{-}1$
   $h_i = r_{ij} d_i$ (Zwischenspeicher) für $j > 1$

   1.2.2 $d_j = a_{jj} - \sum_{i=1}^{j-1} h_i r_{ij}$ für $j = 2(1)n$

   1.2.3 Für jedes $k = j{+}1(1)n$
   $r_{jk} = \frac{1}{d_j} \left(a_{jk} - \sum_{i=1}^{j-1} h_i r_{ik} \right)$ für $j \leq n{-}1$

2. (Vorwärtselimination $\boldsymbol{R}^{\mathsf{T}} \boldsymbol{z} = \boldsymbol{a}, \boldsymbol{D}\boldsymbol{r} = \boldsymbol{z}$)

   2.1 $z_1 = a_1$

   2.2 $z_j = a_j - \sum_{i=1}^{j-1} r_{ij} z_i$ für $j = 2(1)n$

   2.3 $r_j = z_j / d_j$ für $j = 1(1)n$

3. (Rückwärtselimination $\boldsymbol{Rx} = \boldsymbol{r}$)

   3.1 $x_n = r_n$

   3.2 $x_j = r_j - \sum_{i=j+1}^{n} r_{ji} x_i$ für $j = n{-}1(-1)1$

Wenn man bei der Faktorisierung den Koeffizientenvergleich zwischen $\boldsymbol{A}$ und $\boldsymbol{R}^{\mathsf{T}}\boldsymbol{DR}$ in etwas anderer Form ausführt, können gegenüber Algorithmus 4.64 noch $n(n-1)/2$ Punktoperationen eingespart werden, vgl. dazu [MAES1985], S. 77. Dann ergibt sich der folgende Algorithmus 4.65.

Algorithmus 4.65.

Gegeben: $\boldsymbol{Ax} = \boldsymbol{a}$ mit symmetrischer, positiv definiter Matrix $\boldsymbol{A}$.

Gesucht: $\boldsymbol{x} = (x_i)$, $i = 1(1)n$.

1. (Faktorisierung $\boldsymbol{A} = \boldsymbol{R}^{\mathsf{T}}\boldsymbol{DR}$)
   Für jedes $j = 1(1)n$

   1.1 Für jedes $i = 1(1)j{-}1$

      1.1.1 $h = a_{ij}$

      1.1.2 $r_{ij} = h/d_i$

      1.1.3 Für jedes $k = i{+}1(1)j$
      a_{kj} wird durch $a_{kj} - hr_{ik}$ ersetzt

   1.2 $d_j = a_{jj}$

2. (Vorwärtselimination $\boldsymbol{R}^{\mathsf{T}}\boldsymbol{z} = \boldsymbol{a}\,, \boldsymbol{Dr} = \boldsymbol{z}$)
   Für jedes $j = 1(1)n$

   2.1 $z_j = a_j$

   2.2 Für jedes $i = 1(1)j{-}1$
   $z_j := z_j - r_{ij} z_i$

   2.3 $r_j = z_j/d_j$

3. (Rückwärtselimination $\boldsymbol{Rx} = \boldsymbol{r}$)
   Für jedes $j = n(-1)1$

   3.1 $x_j = r_j$

   3.2 Für jedes $i = j{+}1(1)n$
   $x_j := x_j - r_{ji} x_i$

Für die Determinante von $\boldsymbol{A}$ gilt:

$$\det \boldsymbol{A} = \det(\boldsymbol{R}^{\mathsf{T}}) \det \boldsymbol{D} \det \boldsymbol{R} = \det \boldsymbol{D} = d_1 d_2 \ldots d_n.$$

4.7.3 Systeme mit symmetrischer, positiv definiter Matrix – Verfahren der konjugierten Gradienten (CG-Verfahren)

Ist die Matrix $\boldsymbol{A}$ symmetrisch und positiv definit, so kann auch das Verfahren der konjugierten Gradienten (CG-Verfahren) angewandt werden. Anstelle des linearen Gleichungssystems $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}$ wird hier iterativ die äquivalente Minimierungsaufgabe gelöst. Es gilt der

Satz 4.66.
Die folgenden Aufgaben sind äquivalent:
(i) löse $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}$ und (ii) minimiere $F(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^{\mathsf{T}}\boldsymbol{A}\boldsymbol{x} - \boldsymbol{x}^{\mathsf{T}}\boldsymbol{a}$.

Der Beweis ergibt sich sofort mit der Hilfsfunktion

$$E(\boldsymbol{x}) = \frac{1}{2}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{a})^{\mathsf{T}}\boldsymbol{A}^{-1}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{a})\,.$$

Da auch $\boldsymbol{A}^{-1}$ positiv definit ist, ist $E(\boldsymbol{x}) \geq 0$. Somit ist $E(\boldsymbol{x})$ genau dann minimal, wenn gilt $\boldsymbol{A}\boldsymbol{x} - \boldsymbol{a} = \boldsymbol{0}$. Die Berechnung von $E(\boldsymbol{x})$ ergibt unter Verwendung von $\boldsymbol{A}^{\mathsf{T}} = \boldsymbol{A}$

$$E(\boldsymbol{x}) = F(\boldsymbol{x}) + \frac{1}{2}\underbrace{\boldsymbol{a}^{\mathsf{T}}\boldsymbol{A}^{-1}\boldsymbol{a}}_{\geq 0}\,.$$

Daraus folgt, dass $E(\boldsymbol{x})$ und $F(\boldsymbol{x})$ an derselben Stelle minimal sind, d. h.

$$F(\boldsymbol{x}) \stackrel{!}{=} \text{ Min. } \Longleftrightarrow \boldsymbol{A}\boldsymbol{x} - \boldsymbol{a} = \boldsymbol{0}\,.$$

Die Richtung des stärksten Abstiegs einer Funktion ist durch ihren negativen Gradienten gegeben.
Für F gilt : $\qquad \text{grad } F(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} - \boldsymbol{a}.$

Es besteht also die Aufgabe, jenen Punkt zu suchen, in dem der Gradient verschwindet.

Algorithmus 4.67. (*CG-Verfahren*)
Gegeben ist eine symmetrische, positiv definite (n, n)-Matrix $\boldsymbol{A}$.
Gesucht ist die Lösung $\boldsymbol{x}$ des linearen Gleichungssystems $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}$.

1. Startpunkt $\boldsymbol{x}_0 \in \mathbb{R}^n$ (beliebig)
   $\boldsymbol{d}_0 = -\boldsymbol{g}_0 = -(\boldsymbol{A}\boldsymbol{x}_0 - \boldsymbol{a})$. Falls $\boldsymbol{g}_0 = \boldsymbol{0}$, kann abgebrochen werden, $\boldsymbol{x}_0$ ist dann Lösung.
2. Für $k = 0, 1, \ldots, n-1$ werden nacheinander berechnet:

   2.1 $\alpha_k = -\dfrac{\boldsymbol{d}_k^{\mathsf{T}}\boldsymbol{g}_k}{\boldsymbol{d}_k^{\mathsf{T}}\boldsymbol{A}\boldsymbol{d}_k}$

   2.2 $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k\boldsymbol{d}_k \qquad$ mit $\quad \{\boldsymbol{x}_{k+1} \in \boldsymbol{x}_0 + U_{k+1}(\boldsymbol{d}_0, \ldots, \boldsymbol{d}_k)\}$

2.3 $\boldsymbol{g}_{k+1} = \boldsymbol{g}_k + \alpha_k \boldsymbol{A}\,\boldsymbol{d}_k$
mit $\{\boldsymbol{g}_{k+1} \perp \boldsymbol{d}_0, \ldots, \boldsymbol{d}_k,\ \boldsymbol{g}_{k+1} \notin U_{k+1}(\boldsymbol{d}_0, \ldots, \boldsymbol{d}_k)\}$

Gilt zu vorgegebenem $\varepsilon > 0$ $\|\boldsymbol{g}_{k+1}\|_\infty < \varepsilon$, dann kann abgebrochen werden mit $\boldsymbol{x}_{k+1}$ als Lösung.

2.4 $\beta_k = \frac{\boldsymbol{g}_{k+1}^{\mathsf{T}} \boldsymbol{A}\boldsymbol{d}_k}{\boldsymbol{d}_k^{\mathsf{T}} \boldsymbol{A}\boldsymbol{d}_k}$

2.5 $\boldsymbol{d}_{k+1} = -\boldsymbol{g}_{k+1} + \beta_k \boldsymbol{d}_k$
mit $\{\boldsymbol{d}_{k+1} \perp \boldsymbol{A}\,\boldsymbol{d}_k,\ \boldsymbol{d}_{k+1} \notin U_{k+1}(\boldsymbol{d}_0, \ldots, \boldsymbol{d}_k)\}$

Für $k = n-1$ (d. h. im n-ten Schritt) erhält man $U_n(\boldsymbol{d}_0, \boldsymbol{d}_1, \ldots, \boldsymbol{d}_{n-1}) = \mathbb{R}^n$, denn $\boldsymbol{d}_n \neq \boldsymbol{0}$ mit $\boldsymbol{d}_n \notin U_n = \mathbb{R}^n$ kann nicht existieren!

Das CG-Verfahren kann sowohl den direkten Verfahren zugerechnet werden, weil es nach genau n Schritten bis auf Rundungsfehler die exakte Lösung liefert, als auch den iterativen Verfahren, weil es im Allgemeinen wegen der raschen Konvergenz bereits nach wenigen Schritten eine ausreichend gute Lösung liefert.

Vorteile des Verfahrens sind

- die leichte Vektorisierbarkeit und Parallelisierbarkeit
- die rasche Konvergenz; ihre Geschwindigkeit hängt allerdings von der Kondition der Matrix $\boldsymbol{A}$ ab; je besser die Kondition um so geringer die benötigte Schrittzahl bei vorgegebener Genauigkeit.

Ein Nachteil des Verfahrens ist die große Empfindlichkeit gegen Rundungsfehler.
Dass das Verfahren im Gegensatz zum Gauß-Algorithmus leicht vektorisierbar und parallelisierbar ist, ergibt sich aus der Art der Operationen, die pro Iterationsschritt auszuführen sind. Es sind pro Schritt drei Skalarprodukte zu berechnen und eine Matrix-Vektor-Multiplikation ($\boldsymbol{A}\,\boldsymbol{d}_k$). Da diese Operationen bei herkömmlicher Verarbeitung ca. 97 % der Rechenzeit ausmachen, muss hier auf besonders effektive Berechnung geachtet werden.

In der Literatur ist eine Variante des CG-Verfahrens, das sogenannte CG-Verfahren mit Vorkonditionierung zu finden. Man erreicht damit eine Verringerung der Zahl der Iterationsschritte bei erhöhtem Rechenaufwand pro Schritt. Das Verfahren ist z. B. in [BUNS1995], S. 156 ff., [SCHW1991], [SCHW1988], [MAES1985], S. 132-133 beschrieben.

Geometrische Interpretation des CG-Verfahrens für den Fall n = 2:

$F(x, y)$ ist dann ein elliptisches Paraboloid, die Höhenlinien sind Ellipsen, die Grundrisse der Höhenlinien sind ähnliche Ellipsen mit demselben Mittelpunkt.
Für diesen Fall beinhaltet das Verfahren der konjugierten Gradienten folgende Konstruktionsschritte, die in der Abbildung erkennbar sind.

1. Wähle x_0 auf einer Ellipse
2. Konstruiere die Tangente in x_0 : $\boldsymbol{t}_0$
3. Konstruiere die Normale in x_0 : $\boldsymbol{n}_0 = t_0^{\perp}$
4. Halbiere die auf $\boldsymbol{n}_0$ liegende Sehne der Ellipse: x_1
5. Konstruiere den zu x_0x_1 konjugierten Durchmesser durch x_1; er verbindet die Punkte der Ellipse, deren Tangenten zu x_0x_1 parallel sind.
6. Halbiere den Durchmesser: x_2

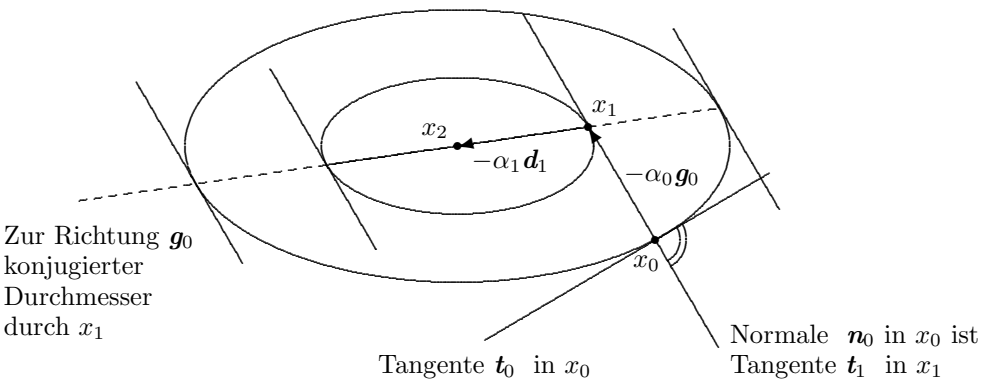


Abb. 4.7. Geometrische Interpretation des CG-Verfahrens

Beispiel 4.68.

Gegeben: Das Gleichungssystem $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ mit der symmetrischen und positiv definiten Matrix $\boldsymbol{A}$ sowie der rechte Seite $\boldsymbol{b}$

$$\boldsymbol{A} = \begin{pmatrix} 5.0000 & -2.0000 \\ -2.0000 & 10.0000 \end{pmatrix}, \qquad \boldsymbol{b} = \begin{pmatrix} 2.0000 \\ 2.0000 \end{pmatrix}$$

Gesucht: Die Lösung $\boldsymbol{x}$ mittels CG-Verfahren (Alg. 4.67)

1. Es werden gesetzt

$$\boldsymbol{x}_0 = \begin{pmatrix} 1.0000 \\ 1.0000 \end{pmatrix}, \qquad \boldsymbol{d}_0 = \begin{pmatrix} -1.0000 \\ -6.0000 \end{pmatrix}, \qquad \boldsymbol{g}_0 = \begin{pmatrix} 1.0000 \\ 6.0000 \end{pmatrix}$$

	Für $k = 0$ wird berechnet
Schritt	Rechnung
2.1	$\alpha_0 = 0.108504$
2.2	$\boldsymbol{x}_1 = \boldsymbol{x}_0 + \alpha_0\, \boldsymbol{d}_0 = \begin{pmatrix} 0.8915 \\ 0.3490 \end{pmatrix}$
2.3	$\boldsymbol{g}_1 = \boldsymbol{g}_0 + \alpha_0\, \boldsymbol{A}\, \boldsymbol{d}_0 = \begin{pmatrix} 1.7595 \\ -0.2933 \end{pmatrix}$
2.4	$\beta_0 = 0.085999$
2.5	$\boldsymbol{d}_1 = -\boldsymbol{g}_0 + \beta_0\, \boldsymbol{d}_0 = \begin{pmatrix} -1.8455 \\ -0.2227 \end{pmatrix}$

	Für $k = 1$ wird berechnet
Schritt	Rechnung
2.1	$\alpha_1 = 0.200353$
2.2	$\boldsymbol{x}_2 = \boldsymbol{x}_1 + \alpha_1\, \boldsymbol{d}_1 = \begin{pmatrix} 0.5217 \\ 0.3043 \end{pmatrix}$
2.3	$\boldsymbol{g}_2 = \boldsymbol{g}_1 + \alpha_1\, \boldsymbol{A}\, \boldsymbol{d}_1 = \begin{pmatrix} -0.0000 \\ 0.0000 \end{pmatrix}$

Da die Norm von $\boldsymbol{g}_2$ verschwindet, wird $\boldsymbol{x}_2$ als Lösung verwendet.

Probe: $\boldsymbol{A} \cdot \begin{pmatrix} 0.5217 \\ 0.3043 \end{pmatrix} = \begin{pmatrix} 2.0000 \\ 2.0000 \end{pmatrix} \stackrel{!}{=} \boldsymbol{a}$ □

Beispiel 4.69.

Gegeben: Das lineare Gleichungssystem $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ mit

$$\boldsymbol{A} = \begin{pmatrix} 7 & 2 & -1 & 2 & -1 \\ 2 & 7 & 2 & -1 & 2 \\ -1 & 2 & 7 & 2 & -1 \\ 2 & -1 & 2 & 7 & 2 \\ -1 & 2 & -1 & 2 & 7 \end{pmatrix}, \qquad \boldsymbol{b} = \begin{pmatrix} -13 \\ 8 \\ 3 \\ -8 \\ 19 \end{pmatrix}$$

Gesucht: Die Lösung $\boldsymbol{x}$ unter Anwendung des CG-Verfahrens (Alg. 4.67).

Lösung: Da $\boldsymbol{A}$ symmetrisch ist und alle Hauptabschnittsdeterminanten (7, 45, 272, 1280 und 3072) positiv sind, $\boldsymbol{A}$ also auch positiv definit ist, ist das CG-Verfahren anwendbar.

Der Algorithmus führt folgende Schritte durch:

$$\boldsymbol{x}_0 = \begin{pmatrix} 1.0000 \\ 1.0000 \\ 1.0000 \\ 1.0000 \\ 1.0000 \end{pmatrix}, \qquad \boldsymbol{g}_0 = \begin{pmatrix} 22.0000 \\ 4.0000 \\ 6.0000 \\ 20.0000 \\ -10.0000 \end{pmatrix}, \qquad \boldsymbol{d}_0 = \begin{pmatrix} -22.0000 \\ -4.0000 \\ -6.0000 \\ -20.0000 \\ 10.0000 \end{pmatrix}$$

$\alpha_0 = 0.113646\,,$ $\qquad \beta_0 = 0.029001$

$$\boldsymbol{x}_1 = \begin{pmatrix} -1.5002 \\ 0.5454 \\ 0.3181 \\ -1.2729 \\ 2.1365 \end{pmatrix}, \quad \boldsymbol{g}_1 = \begin{pmatrix} -1.4111 \\ -1.0004 \\ -2.8644 \\ 0.4528 \\ -4.3177 \end{pmatrix} \qquad \boldsymbol{d}_1 = \begin{pmatrix} 0.7731 \\ 0.8844 \\ 2.6904 \\ -1.0328 \\ 4.6077 \end{pmatrix}$$

$\alpha_1 = 0.169390\,,$ $\qquad \beta_1 = 0.528165$

$$\boldsymbol{x}_2 = \begin{pmatrix} -1.3693 \\ 0.6952 \\ 0.7739 \\ -1.4479 \\ 2.9170 \end{pmatrix}, \quad \boldsymbol{g}_2 = \begin{pmatrix} -1.7809 \\ 2.9576 \\ -0.6360 \\ 1.8127 \\ 0.5088 \end{pmatrix} \qquad \boldsymbol{d}_2 = \begin{pmatrix} 2.1893 \\ -2.4905 \\ 2.0570 \\ -2.3582 \\ 1.9248 \end{pmatrix}$$

$\alpha_2 = 1.082218$

$$\boldsymbol{x}_3 = \begin{pmatrix} 1.0000 \\ -2.0000 \\ 3.0000 \\ -4.0000 \\ 5.0000 \end{pmatrix}, \quad \boldsymbol{g}_3 = \begin{pmatrix} -0.0000 \\ 0.0000 \\ -0.0000 \\ -0.0000 \\ 0.0000 \end{pmatrix}$$

Norm von $\boldsymbol{g}_3$ verschwindet $\quad\Rightarrow\quad$ $\boldsymbol{x}_3$ ist Lösung.

Damit erhält man als Lösung $\boldsymbol{x}$ des Gleichungssystems $\boldsymbol{x}_3 = (1, -2, 3, -4, 5)^{\mathsf{T}}$. □

4.8 Das Gauß-Jordan-Verfahren

Das Gauß-Jordan-Verfahren ist eine Modifikation des Gaußschen Algorithmus, welche die rekursive Berechnung der Lösungen x_i gemäß (4.8) erspart. Der erste Schritt des Verfahrens ist identisch mit dem ersten Eliminationsschritt des Gaußschen Algorithmus; man erhält somit (4.6). Die Gleichungen 2 bis n sind so umgeordnet, dass $a_{22}^{(1)}$ das betragsgrößte Element der $a_{i2}^{(1)}$ für $i = 2(1)n$ ist. Jetzt wird die 2. Gleichung von (4.6) nacheinander für $i = 1$ mit $-a_{12}^{(0)}/a_{22}^{(1)}$ und für $i = 3(1)n$ mit $-a_{i2}^{(1)}/a_{22}^{(1)}$ multipliziert und jeweils zur i-ten Gleichung addiert. Man erhält nach diesem ersten Jordan-Schritt ein Gleichungssystem der Form

$$\left\{\begin{array}{lllllll} a_{11}^{(1)}x_1+ & & a_{13}^{(1)}x_3+ & a_{14}^{(1)}x_4 & +\ldots+ & a_{1n}^{(1)}x_n = & a_1^{(1)}, \\ & a_{22}^{(1)}x_2+ & a_{23}^{(1)}x_3+ & a_{24}^{(1)}x_4 & +\ldots+ & a_{2n}^{(1)}x_n = & a_2^{(1)}, \\ & & a_{33}^{(1)}x_3+ & a_{34}^{(1)}x_4 & +\ldots+ & a_{3n}^{(1)}x_n = & a_3^{(1)}, \\ & & \vdots & & & & \\ & & a_{n3}^{(1)}x_3+ & a_{n4}^{(1)}x_4 & +\ldots+ & a_{nn}^{(1)}x_n = & a_n^{(1)}. \end{array}\right. \tag{4.11}$$

Dabei ist $a_{11}^{(1)} = a_{11}^{(0)}$ und $a_{22}^{(1)} = a_{22}^{(1)}$; für diese unveränderten und für die neu gewonnenen Elemente soll die Bezeichnung mit dem oberen Index verwendet werden. Die Gleichungen 3 bis n von (4.11) sind bereits so umgeordnet, dass $a_{33}^{(1)}$ das betragsgrößte Element der $a_{i3}^{(1)}$ für $i = 3(1)n$ ist. In einem zweiten Jordan-Schritt multipliziert man die dritte Gleichung von (4.11) mit $-a_{i3}^{(1)}/a_{33}^{(1)}$ für $i = 1(1)n$ und $i \neq 3$ und addiert sie zur i-ten Gleichung. So fortfahrend erhält man nach $n-1$ Jordan-Schritten schließlich n Gleichungen der Form $a_{ii}^{(n-1)} x_i = a_i^{(n-1)}$, $i = 1(1)n$, aus denen sich unmittelbar die x_i berechnen lassen. Da die Anzahl der Punktoperationen $n^3/2 + O(n^2)$ ist und beim Gauß-Algorithmus nur $n^3/3 + O(n^2)$, ist der Gauß-Algorithmus vorzuziehen !

4.9 Gleichungssysteme mit tridiagonaler Matrix

4.9.1 Systeme mit tridiagonaler Matrix

Eine Matrix $\boldsymbol{A} = (a_{ik})$ heißt *tridiagonal*, falls gilt $a_{ik} = 0$ für $|i-k| > 1$, $i, k = 1(1)n$, $n \geq 3$. Ein Gleichungssystem (4.1) bzw. (4.2) mit tridiagonaler Matrix hat die Gestalt

$$\begin{pmatrix} a_{11} & a_{12} & & & & \\ a_{21} & a_{22} & a_{23} & & & \\ & a_{32} & a_{33} & a_{34} & & \\ & & \ddots & \ddots & \ddots & \\ & & a_{n-1,n-2} & a_{n-1,n-1} & a_{n-1,n} & \\ & & & a_{n,n-1} & a_{nn} & \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_{n-1} \\ a_n \end{pmatrix}. \quad (4.12)$$

Prinzip des Verfahrens

Das System $\boldsymbol{Ax} = \boldsymbol{a}$ lässt sich mit der Zerlegung $\boldsymbol{A} = \boldsymbol{LR}$, wo $\boldsymbol{L}$ eine bidiagonale untere Dreiecksmatrix und $\boldsymbol{R}$ eine normierte bidiagonale obere Dreiecksmatrix ist, in ein äquivalentes System $\boldsymbol{Rx} = \boldsymbol{r}$ überführen. Voraussetzung für die Zerlegbarkeit ohne Zeilenvertauschung ist strenge Regularität von $\boldsymbol{A}$, d. h. es muss gelten (siehe Satz 4.9)

$$\det(\boldsymbol{A}_k) \neq 0 \quad \text{für} \quad k = 1(1)n-1\,.$$

Ist diese Voraussetzung verletzt, muss mit Spaltenpivotsuche gearbeitet werden, wodurch sich jedoch im Allgemeinen die Bandbreite erhöht (vgl. Abschnitt 4.12). Die Überführung von $\boldsymbol{Ax} = \boldsymbol{a}$ in $\boldsymbol{Rx} = \boldsymbol{r}$ erfolgt in den Schritten:

1. (Faktorisierung) $\quad \boldsymbol{A} = \boldsymbol{LR} \quad \Rightarrow \boldsymbol{L}, \boldsymbol{R}$,
2. (Vorwärtselimination) $\quad \boldsymbol{a} = \boldsymbol{Lr} \quad \Rightarrow \boldsymbol{r}$,
3. (Rückwärtselimination) $\quad \boldsymbol{Rx} = \boldsymbol{r} \quad \Rightarrow \boldsymbol{x}$.

Durchführung des Verfahrens

Die Elemente der Matrizen $\boldsymbol{A}, \boldsymbol{L}, \boldsymbol{R}$ werden wegen vektorieller Speicherung der Diagonalen wie folgt bezeichnet

$$\left\{\begin{array}{l} \boldsymbol{A} = \begin{pmatrix} d_1 & c_1 & & & \\ b_2 & d_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & b_{n-1} & d_{n-1} & c_{n-1} \\ & & & b_n & d_n \end{pmatrix}, \ \boldsymbol{R} = \begin{pmatrix} 1 & \gamma_1 & & & \\ & 1 & \gamma_2 & & \\ & & \ddots & \ddots & \\ & & & 1 & \gamma_{n-1} \\ & & & & 1 \end{pmatrix}, \\ \boldsymbol{L} = \begin{pmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \beta_n & \alpha_n \end{pmatrix}, \qquad \boldsymbol{r} = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{pmatrix}. \end{array}\right. \tag{4.13}$$

1. $\boldsymbol{A} = \boldsymbol{LR} \Longrightarrow$

$$\boldsymbol{A} = \begin{pmatrix} d_1 & c_1 & & & \\ b_2 & d_2 & c_2 & & \\ & b_3 & d_3 & c_3 & \\ & & \ddots & \ddots & \ddots \\ & & & b_{n-1} & d_{n-1} & c_{n-1} \\ & & & & b_n & d_n \end{pmatrix} = \begin{pmatrix} \alpha_1 & & & & \\ \beta_2 & \alpha_2 & & & \\ & \beta_3 & \alpha_3 & & \\ & & \ddots & \ddots & \\ & & & \beta_{n-1} & \alpha_{n-1} \\ & & & & \beta_n & \alpha_n \end{pmatrix} \begin{pmatrix} 1 & \gamma_1 & & & \\ & 1 & \gamma_2 & & \\ & & 1 & \gamma_3 & \\ & & & \ddots & \ddots \\ & & & & 1 & \gamma_{n-1} \\ & & & & & 1 \end{pmatrix}$$

$$= \begin{pmatrix} \alpha_1 & \alpha_1\gamma_1 & & & & \\ \beta_2 & \beta_2\gamma_1 + \alpha_2 & \alpha_2\gamma_2 & & & \\ & \beta_3 & \beta_3\gamma_2 + \alpha_3 & \alpha_3\gamma_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & \beta_{n-1} & \beta_{n-1}\gamma_{n-2} + \alpha_{n-1} & \alpha_{n-1}\gamma_{n-1} \\ & & & & \beta_n & \beta_n\gamma_{n-1} + \alpha_n \end{pmatrix}$$

Koeffizientenvergleich liefert:

α_1	$=$	d_1	$\Rightarrow$	$\boxed{\alpha_1 = d_1}$	(a)	
c_1	$=$	$\alpha_1\,\gamma_1$	$\Rightarrow$	$\boxed{\gamma_1 = \frac{c_1}{\alpha_1}}$	(b)	
				$\boxed{\beta_i = b_i}$	(c)	$i = 2(1)n$
$b_i\,\gamma_{i-1} + \alpha_i$	$=$	d_i	$\Rightarrow$	$\boxed{\alpha_i = d_i - b_i\,\gamma_{i-1}}$	(d)	$i = 2(1)n$
$\alpha_i\,\gamma_i$	$=$	c_i	$\Rightarrow$	$\boxed{\gamma_i = \frac{c_i}{\alpha_i}}$	(e)	$i = 2(1)n{-}1$

immer zeilenweise, also zuerst (a), dann (b), dann (d) und (e) für $i = 2(1)n{-}1$ und anschließend (d) für $i = n$; (c) ist nur Umbenennung.

2. Vorwärtselimination $\boldsymbol{a} = \boldsymbol{L}\,\boldsymbol{r}$

$$\begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_{n-1} \\ a_n \end{pmatrix} = \begin{pmatrix} \alpha_1 & & & & & \\ b_2 & \alpha_2 & & & & \\ & b_3 & \alpha_3 & & & \\ & & \ddots & \ddots & & \\ & & & b_{n-1} & \alpha_{n-1} & \\ & & & & b_n & \alpha_n \end{pmatrix} \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ \vdots \\ r_{n-1} \\ r_n \end{pmatrix}$$

Beginnend mit der 1. Zeile erhält man

$$a_1 = \alpha_1\, r_1 \qquad \Longrightarrow \qquad r_1 = \frac{a_1}{\alpha_1}$$

und für $i = 2(1)n$

$$r_i = (a_i - b_i\, r_{i-1}) \;/\; \alpha_i \,.$$

3. Rückwärtselimination $\boldsymbol{R}\,\boldsymbol{x} = \boldsymbol{r}$

$$\begin{pmatrix} 1 & \gamma_1 & & & & \\ & 1 & \gamma_2 & & & \\ & & 1 & \gamma_3 & & \\ & & & \ddots & \ddots & \\ & & & & 1 & \gamma_{n-1} \\ & & & & & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ \vdots \\ r_{n-1} \\ r_n \end{pmatrix}$$

Beginnend mit der letzten Zeile erhält man

$$x_n = r_n$$

und für $i = n{-}1, n{-}2, \ldots, 2, 1$

$$x_i = r_i - \gamma_i\, x_{i+1} \,.$$

Algorithmus 4.70.

Gegeben: $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}$ mit tridiagonaler Matrix $\boldsymbol{A}$, $\det(\boldsymbol{A}_k) \neq 0$ für $k = 1(1)n{-}1$, $n \geq 3$.
Gesucht: $\boldsymbol{x} = (x_i)$, $i = 1(1)n$.

Dann sind nacheinander folgende Schritte auszuführen:

1. (Zerlegung $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{R}$)
   - 1.1 $\alpha_1 = d_1$
   - 1.2 $\gamma_1 = c_1/\alpha_1$
   - 1.3 Für jedes $i = 2(1)n{-}1$ sind zu berechnen:
     - 1.3.1 $\alpha_i = d_i - b_i\gamma_{i-1}$
     - 1.3.2 $\gamma_i = c_i/\alpha_i$
   - 1.4 $\alpha_n = d_n - b_n\gamma_{n-1}$

2. (Vorwärtselimination $\boldsymbol{a} = \boldsymbol{Lr}$)
   2.1 $r_1 = a_1/d_1$
   2.2 Für jedes $i = 2(1)n$ sind zu berechnen:
   $r_i = (a_i - b_i r_{i-1})/\alpha_i$
3. (Rückwärtselimination $\boldsymbol{Rx} = \boldsymbol{r}$)
   3.1 $x_n = r_n$
   3.2 Für jedes $i = n{-}1(-1)1$ sind zu berechnen:
   $x_i = r_i - \gamma_i x_{i+1}$

Anzahl der Punktoperationen:

zu 1. $1 + (n{-}1) + (n{-}2) = 2(n{-}1)$; zu 2. $1 + 2(n{-}1)$; zu 3. $n{-}1$.

Insgesamt sind dies $2(n-1) + 1 + 2(n-1) + n - 1 = 5n - 4$ Punktoperationen. Im Vergleich dazu benötigt der Gauß-Algorithmus für vollbesetzte Matrizen

$$\frac{n^3}{3} + n^2 - \frac{n}{3}$$

Punktoperationen, d. h. zum Beispiel bei $n = 1000$ benötigt der Gauß-Algorithmus $\sim 10^9/3$ Punktoperationen, während das Verfahren speziell für tridiagonale Matrizen nur knapp 5 000 Punktoperationen erfordert.

Beispiel 4.71.

Gegeben: Das Gleichungssystem $\boldsymbol{Ax} = \boldsymbol{a}$ mit

$$\begin{pmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} -5 \\ 1 \\ 4 \\ -1 \end{pmatrix}$$

Gesucht: Die Lösung $\boldsymbol{x}$ mit dem Algorithmus 4.70.

Lösung:

1. Zerlegung $\boldsymbol{A} = \boldsymbol{LR}$

$$\begin{pmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{pmatrix} = \begin{pmatrix} 2 & 0 & 0 & 0 \\ -1 & \frac{3}{2} & 0 & 0 \\ 0 & -1 & \frac{4}{3} & 0 \\ 0 & 0 & -1 & \frac{5}{4} \end{pmatrix} \begin{pmatrix} 1 & \frac{-1}{2} & 0 & 0 \\ 0 & 1 & \frac{-2}{3} & 0 \\ 0 & 0 & 1 & \frac{-3}{4} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

2. Vorwärtselimination $\boldsymbol{Lr} = \boldsymbol{a}$

$$\begin{pmatrix} 2 & 0 & 0 & 0 \\ -1 & \frac{3}{2} & 0 & 0 \\ 0 & -1 & \frac{4}{3} & 0 \\ 0 & 0 & -1 & \frac{5}{4} \end{pmatrix} \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{pmatrix} = \begin{pmatrix} -5 \\ 1 \\ 4 \\ -1 \end{pmatrix} \quad \Longrightarrow \quad \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{pmatrix} = \begin{pmatrix} \frac{-5}{2} \\ -1 \\ \frac{9}{4} \\ 1 \end{pmatrix}$$

3. Rückwärtselimination $\boldsymbol{Rx} = \boldsymbol{r}$

$$\begin{pmatrix} 1 & \frac{-1}{2} & 0 & 0 \\ 0 & 1 & \frac{-2}{3} & 0 \\ 0 & 0 & 1 & \frac{-3}{4} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} \frac{-5}{2} \\ -1 \\ \frac{9}{4} \\ 1 \end{pmatrix} \quad \Longrightarrow \quad \boldsymbol{x} = \begin{pmatrix} -2 \\ 1 \\ 3 \\ 1 \end{pmatrix}.$$

□

Die tridiagonale Matrix $\boldsymbol{A}$ (4.13) ist regulär, d. h. $\det \boldsymbol{A} \neq 0$, wenn gilt:

$$|d_1| > |c_1| > 0 \qquad |d_i| \geq |b_i| + |c_i| \quad \text{für} \quad i = 2(1)n{-}1\,.$$
$$b_i c_i \neq 0 \quad \text{für} \quad i = 2(1)n{-}1 \qquad |d_n| > |b_n| > 0\,.$$

Es liegt dann eine tridiagonale, diagonaldominante Matrix vor. ([CONT1987], S. 184). Für die Determinante einer tridiagonalen Matrix $\boldsymbol{A}$ gilt mit $\boldsymbol{A} = \boldsymbol{LR}$, (4.12) und (4.13) wegen $\det \boldsymbol{R} = 1$

$$\det \boldsymbol{A} = \det \boldsymbol{L} \det \boldsymbol{R} = \det \boldsymbol{L} = \alpha_1 \alpha_2 \ldots \alpha_n.$$

Bei Gleichungssystemen mit symmetrischen, tridiagonalen bzw. zyklisch tridiagonalen, diagonaldominanten und anderen positiv definiten Matrizen ist der Gaußsche Algorithmus auch ohne Pivotsuche numerisch stabil; Konditionsverbesserung und Nachiteration tragen nicht zur Verbesserung der Lösung bei (s. [FORS1971], 8, 10, 11; [SPAT1986], S.15, [WILK1961]). In allen anderen Fällen ist Pivotsuche erforderlich. Dadurch kann sich jedoch die Bandbreite (s. Abschnitt 4.12) erhöhen, sie kann sich aber höchstenfalls verdoppeln.

4.9.2 Systeme mit symmetrischer, tridiagonaler, positiv definiter Matrix

Ist $\boldsymbol{A}$ eine symmetrische, tridiagonale, positiv definite (n, n)-Matrix, so kann bei der Lösung des Systems $\boldsymbol{Ax} = \boldsymbol{a}$ ein zum Cholesky-Verfahren äquivalentes Verfahren angewandt werden; es kann dabei Speicherplatz eingespart werden. Die Überführung von $\boldsymbol{Ax} = \boldsymbol{a}$ in ein äquivalentes System $\boldsymbol{Rx} = \boldsymbol{r}$ geschieht gemäß Algorithmus 4.64 in den Schritten:

1. Faktorisierung:	$\boldsymbol{A} = \boldsymbol{R}^{\mathsf{T}} \boldsymbol{D} \boldsymbol{R}$	$\Rightarrow \boldsymbol{R}$ und $\boldsymbol{D}$,
2. Vorwärtselimination:	$\boldsymbol{R}^{\mathsf{T}} \boldsymbol{z} = \boldsymbol{a}$	$\Rightarrow \boldsymbol{z}$,
	$\boldsymbol{D}\boldsymbol{r} = \boldsymbol{z}$	$\Rightarrow \boldsymbol{r}$,
3. Rückwärtselimination:	$\boldsymbol{Rx} = \boldsymbol{r}$	$\Rightarrow \boldsymbol{x}$.

Durchführung des Verfahrens

Die Elemente von $\boldsymbol{A}, \boldsymbol{R}, \boldsymbol{D}, \boldsymbol{r}, \boldsymbol{z}$ und $\boldsymbol{x}$ werden wie folgt bezeichnet:

$$\boldsymbol{A} = \begin{pmatrix} d_1 & c_1 & & & \\ c_1 & d_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & c_{n-2} & d_{n-1} & c_{n-1} \\ & & & c_{n-1} & d_n \end{pmatrix}, \quad \boldsymbol{R} = \begin{pmatrix} 1 & \gamma_1 & & & \\ & 1 & \gamma_2 & & \\ & & \ddots & \ddots & \\ & & & 1 & \gamma_{n-1} \\ & & & & 1 \end{pmatrix},$$

$$\boldsymbol{D} = \begin{pmatrix} \alpha_1 & & & \\ & \alpha_2 & & \\ & & \ddots & \\ & & & \alpha_n \end{pmatrix}, \quad \boldsymbol{r} = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{pmatrix}, \quad \boldsymbol{z} = \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{pmatrix}, \quad \boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \quad \boldsymbol{a} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}$$

Algorithmus 4.72.

Gegeben: $\boldsymbol{Ax} = \boldsymbol{a}$, $\boldsymbol{A}$ symmetrisch, tridiagonal, positiv definit.

Gesucht: $\boldsymbol{x} = (x_i)$, $i = 1(1)n$.

Dann sind nacheinander folgende Schritte auszuführen:

1. (Zerlegung $\boldsymbol{A} = \boldsymbol{R}^{\mathsf{T}}\boldsymbol{DR}$)
   - 1.1 $\alpha_1 = d_1$
   - 1.2 $\gamma_1 = c_1/\alpha_1$
   - 1.3 Für jedes $i = 2(1)n{-}1$ sind durchzuführen:
     - 1.3.1 $\alpha_i = d_i - c_{i-1}\gamma_{i-1}$
     - 1.3.2 $\gamma_i = c_i/\alpha_i$
   - 1.4 $\alpha_n = d_n - c_{n-1}\gamma_{n-1}$
2. (Vorwärtselimination $\boldsymbol{R}^{\mathsf{T}}\boldsymbol{z} = \boldsymbol{a}$, $\boldsymbol{Dr} = \boldsymbol{z}$)
   - 2.1 $z_1 = a_1$
   - 2.2 Für jedes $i = 2(1)n$ ist zu berechnen:
     $z_i = a_i - \gamma_{i-1}z_{i-1}$
   - 2.3 Für jedes $i = 1(1)n$ ist zu berechnen:
     $r_i = z_i/\alpha_i$
3. (Rückwärtselimination $\boldsymbol{Rx} = \boldsymbol{r}$)
   - 3.1 $x_n = r_n$
   - 3.2 Für jedes $i = n{-}1(-1)1$ ist zu berechnen:
     $x_i = r_i - \gamma_i x_{i+1}$

Für die Determinante von $\boldsymbol{A}$ gilt

$$\det \boldsymbol{A} = \det(\boldsymbol{R}^{\mathsf{T}}\boldsymbol{DR}) = \det(\boldsymbol{R}^{\mathsf{T}})\ \det \boldsymbol{D}\ \det \boldsymbol{R} = \det \boldsymbol{D} = \alpha_1\alpha_2 \ldots \alpha_n.$$

Beispiel 4.73.

Gegeben: Das Gleichungssystem

$$\begin{pmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} -5 \\ 1 \\ 4 \\ -1 \end{pmatrix}$$

Gesucht: Die Lösung $\boldsymbol{x}$ mit dem Algorithmus 4.72.

Lösung:

1. Zerlegung $\boldsymbol{A} = \boldsymbol{R}^{\mathsf{T}} \boldsymbol{D} \boldsymbol{R}$

$$\begin{pmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 & 0 \\ 0 & -\frac{2}{3} & 1 & 0 \\ 0 & 0 & -\frac{3}{4} & 1 \end{pmatrix} \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & \frac{3}{2} & 0 & 0 \\ 0 & 0 & \frac{4}{3} & 0 \\ 0 & 0 & 0 & \frac{5}{4} \end{pmatrix} \begin{pmatrix} 1 & -\frac{1}{2} & 0 & 0 \\ 0 & 1 & -\frac{2}{3} & 0 \\ 0 & 0 & 1 & -\frac{3}{4} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

2. Vorwärtselimination $\boldsymbol{R}^{\mathsf{T}} \boldsymbol{z} = \boldsymbol{a}, \boldsymbol{D} \boldsymbol{r} = \boldsymbol{z}$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 & 0 \\ 0 & -\frac{2}{3} & 1 & 0 \\ 0 & 0 & -\frac{3}{4} & 1 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{pmatrix} = \begin{pmatrix} -5 \\ 1 \\ 4 \\ -1 \end{pmatrix} \Rightarrow \boldsymbol{z} = \begin{pmatrix} -5 \\ -\frac{3}{2} \\ 3 \\ \frac{5}{4} \end{pmatrix}$$

$$\begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & \frac{3}{2} & 0 & 0 \\ 0 & 0 & \frac{4}{3} & 0 \\ 0 & 0 & 0 & \frac{5}{4} \end{pmatrix} \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{pmatrix} = \begin{pmatrix} -5 \\ -\frac{3}{2} \\ 3 \\ \frac{5}{4} \end{pmatrix} \Rightarrow \boldsymbol{r} = \begin{pmatrix} -\frac{5}{2} \\ -1 \\ \frac{9}{4} \\ 1 \end{pmatrix}$$

3. Rückwärtselimination $\boldsymbol{R}\boldsymbol{x} = \boldsymbol{r}$

$$\begin{pmatrix} 1 & -\frac{1}{2} & 0 & 0 \\ 0 & 1 & -\frac{2}{3} & 0 \\ 0 & 0 & 1 & -\frac{3}{4} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} -\frac{5}{2} \\ -1 \\ \frac{9}{4} \\ 1 \end{pmatrix} \Rightarrow \boldsymbol{x} = \begin{pmatrix} -2 \\ 1 \\ 3 \\ 1 \end{pmatrix}.$$

□

4.10 Gleichungssysteme mit zyklisch tridiagonaler Matrix

4.10.1 Systeme mit zyklisch tridiagonaler Matrix

Eine Matrix $\boldsymbol{A} = (a_{ik})$, $i,k = 1(1)n$, $n \geq 4$, heißt *zyklisch tridiagonal*, falls gilt $a_{ik} = 0$ für $1 < |i-k| < n-1$, $i,k = 1(1)n$.

Es sei $\boldsymbol{Ax} = \boldsymbol{a}$ ein System mit zyklisch tridiagonaler Matrix $\boldsymbol{A}$.

Prinzip des Verfahrens

Das System kann mit der Zerlegung $\boldsymbol{A} = \boldsymbol{LR}$ in ein äquivalentes System $\boldsymbol{Rx} = \boldsymbol{r}$ überführt werden, sofern $\det(\boldsymbol{A}_k) \neq 0$ für $k = 1(1)n-1$ gilt. Die Lösung erfolgt gemäß Algorithmus 4.42 in den Schritten:

1. (Faktorisierung) $\boldsymbol{A} = \boldsymbol{LR} \Rightarrow \boldsymbol{L}$ und $\boldsymbol{R}$,
2. (Vorwärtselimination) $\boldsymbol{Lr} = \boldsymbol{a} \Rightarrow \boldsymbol{r}$,
3. (Rückwärtselimination) $\boldsymbol{Rx} = \boldsymbol{r} \Rightarrow \boldsymbol{x}$.

Durchführung des Verfahrens

Die Elemente von $\boldsymbol{A}$, $\boldsymbol{L}$, $\boldsymbol{R}$, $\boldsymbol{r}$, $\boldsymbol{x}$, $\boldsymbol{a}$ werden wie folgt bezeichnet

$$\boldsymbol{A} = \begin{pmatrix} d_1 & c_1 & & & e_1 \\ b_2 & d_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & b_{n-1} & d_{n-1} & c_{n-1} \\ c_n & & & b_n & d_n \end{pmatrix}, \quad \boldsymbol{R} = \begin{pmatrix} 1 & \gamma_1 & & & \delta_1 \\ & 1 & \gamma_2 & & \delta_2 \\ & & \ddots & \ddots & \vdots \\ & & & \gamma_{n-2} & \delta_{n-2} \\ & & & 1 & \gamma_{n-1} \\ & & & & 1 \end{pmatrix},$$

$$\boldsymbol{L} = \begin{pmatrix} \alpha_1 & & & & \\ \beta_2 & \alpha_2 & & & \\ & \ddots & \ddots & & \\ & & \beta_{n-1} & \alpha_{n-1} & \\ \varepsilon_3 & \varepsilon_4 \cdots & \varepsilon_n & \beta_n & \alpha_n \end{pmatrix}, \quad \boldsymbol{r} = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{pmatrix}, \quad \boldsymbol{a} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}, \boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}.$$

Analog zur Vorgehensweise bei den tridiagonalen Systemen (Algorithmus 4.70) ergeben sich auch hier durch Koeffizientenvergleich die Elemente von $\boldsymbol{L}$, $\boldsymbol{R}$, $\boldsymbol{r}$ und schließlich von der Lösung $\boldsymbol{x}$.

Algorithmus 4.74.

Gegeben: $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}$ mit zyklisch tridiagonaler Matrix $\boldsymbol{A}$ und $\det(\boldsymbol{A}_k) \neq 0$ für $k = 1(1)n{-}1$, $n \geq 4$.

Gesucht: $\boldsymbol{x} = (x_i)$, $i = 1(1)n$.

Dann sind nacheinander folgende Schritte auszuführen:

1. (Faktorisierung $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{R}$)
   - 1.1 $\alpha_1 = d_1$
   - 1.2 $\gamma_1 = c_1/\alpha_1$
   - 1.3 $\delta_1 = e_1/\alpha_1$
   - 1.4 Für jedes $i = 2(1)n{-}2$ sind durchzuführen:
     - 1.4.1 $\alpha_i = d_i - b_i\gamma_{i-1}$
     - 1.4.2 $\gamma_i = c_i/\alpha_i$
     - 1.4.3 $\beta_i = b_i$
     - 1.4.4 $\delta_i = -\beta_i\delta_{i-1}/\alpha_i$
   - 1.5 $\alpha_{n-1} = d_{n-1} - b_{n-1}\gamma_{n-2}$
   - 1.6 $\beta_{n-1} = b_{n-1}$
   - 1.7 $\varepsilon_3 = c_n$
   - 1.8 Für jedes $i = 4(1)n$ ist zu berechnen: $\varepsilon_i = -\varepsilon_{i-1}\gamma_{i-3}$
   - 1.9 $\gamma_{n-1} = (c_{n-1} - \beta_{n-1}\delta_{n-2})/\alpha_{n-1}$
   - 1.10 $\beta_n = b_n - \varepsilon_n\gamma_{n-2}$
   - 1.11 $\alpha_n = d_n - \sum_{i=3}^{n} \varepsilon_i\delta_{i-2} - \beta_n\gamma_{n-1}$
2. (Vorwärtselimination $\boldsymbol{L}\boldsymbol{r} = \boldsymbol{a}$)
   - 2.1 $r_1 = a_1/\alpha_1$
   - 2.2 Für jedes $i = 2(1)n{-}1$ ist zu berechnen: $r_i = (a_i - r_{i-1}\beta_i)/\alpha_i$
   - 2.3 $r_n = (a_n - \sum_{i=3}^{n} \varepsilon_i r_{i-2} - \beta_n r_{n-1})/\alpha_n$
3. (Rückwärtselimination $\boldsymbol{R}\boldsymbol{x} = \boldsymbol{r}$)
   - 3.1 $x_n = r_n$
   - 3.2 $x_{n-1} = r_{n-1} - \gamma_{n-1}x_n$
   - 3.3 Für jedes $i = n{-}2(-1)1$ ist zu berechnen: $x_i = r_i - \gamma_i x_{i+1} - \delta_i x_n$

Für die Determinante von $\boldsymbol{A}$ gilt:

$$\det \boldsymbol{A} = \det \boldsymbol{L} \ \det \boldsymbol{R} = \det \boldsymbol{L} = \alpha_1\alpha_2 \ldots \alpha_n.$$

Beispiel 4.75.

Gegeben: Das Gleichungssystem $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}$ mit zyklisch tridiagonaler Matrix $\boldsymbol{A}$ und $\det(\boldsymbol{A}_k) \neq 0$ für $k = 1(1)4$

$$\begin{pmatrix} 2 & -1 & 0 & 0 & 1 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ -1 & 0 & 0 & -1 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 5 \\ -8 \\ 9 \\ -6 \\ 2 \end{pmatrix}.$$

Gesucht: $\boldsymbol{x} = (x_i)$, $i = 1(1)5$ mit dem Algorithmus 4.74

Lösung:

1. Faktorisierung $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{R}$:

$$\begin{pmatrix} 2 & -1 & 0 & 0 & 1 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ -1 & 0 & 0 & -1 & 2 \end{pmatrix} = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 \\ -1 & \frac{3}{2} & 0 & 0 & 0 \\ 0 & -1 & \frac{4}{3} & 0 & 0 \\ 0 & 0 & -1 & \frac{5}{4} & 0 \\ -1 & -\frac{1}{2} & -\frac{1}{3} & -\frac{5}{4} & 2 \end{pmatrix} \begin{pmatrix} 1 & -\frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & 1 & -\frac{2}{3} & 0 & \frac{1}{3} \\ 0 & 0 & 1 & -\frac{3}{4} & \frac{1}{4} \\ 0 & 0 & 0 & 1 & -\frac{3}{5} \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

2. Vorwärtselimination $\boldsymbol{L}\boldsymbol{r} = \boldsymbol{a}$

$$\begin{pmatrix} 2 & 0 & 0 & 0 & 0 \\ -1 & \frac{3}{2} & 0 & 0 & 0 \\ 0 & -1 & \frac{4}{3} & 0 & 0 \\ 0 & 0 & -1 & \frac{5}{4} & 0 \\ -1 & -\frac{1}{2} & -\frac{1}{3} & -\frac{5}{4} & 2 \end{pmatrix} \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \end{pmatrix} = \begin{pmatrix} 5 \\ -8 \\ 9 \\ -6 \\ 4 \end{pmatrix} \quad \Rightarrow \quad \boldsymbol{r} = \begin{pmatrix} \frac{5}{2} \\ -\frac{11}{3} \\ 4 \\ -\frac{8}{5} \\ 2 \end{pmatrix}$$

3. Rückwartselimination $\boldsymbol{R}\boldsymbol{x} = \boldsymbol{r}$

$$\begin{pmatrix} 1 & -\frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & 1 & -\frac{2}{3} & 0 & \frac{1}{3} \\ 0 & 0 & 1 & -\frac{3}{4} & \frac{1}{4} \\ 0 & 0 & 0 & 1 & -\frac{3}{5} \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} \frac{5}{2} \\ -\frac{11}{3} \\ 4 \\ -\frac{8}{5} \\ 1 \end{pmatrix} \quad \Rightarrow \quad \boldsymbol{x} = \begin{pmatrix} 1 \\ -2 \\ 3 \\ -1 \\ 1 \end{pmatrix}$$

□

4.10.2 Systeme mit symmetrischer, zyklisch tridiagonaler Matrix

Die Matrix $\boldsymbol{A} = (a_{ik})$, $i,k = 1(1)n$, sei zyklisch tridiagonal, symmetrisch und positiv definit.

Prinzip des Verfahrens

Zur Lösung eines Systems $\boldsymbol{Ax} = \boldsymbol{a}$ mit einer zyklisch tridiagonalen, symmetrischen, positiv definiten Matrix $\boldsymbol{A}$ kann ein zum Cholesky-Verfahren äquivalentes Verfahren angewandt werden. Es kann gegenüber 4.11.1 Speicherplatz eingespart werden. Mit der Zerlegung $\boldsymbol{A} = \boldsymbol{R}^{\mathsf{T}}\boldsymbol{DR}$ wird $\boldsymbol{Ax} = \boldsymbol{a}$ gemäß Algorithmus 4.64 in ein äquivalentes System $\boldsymbol{Rx} = \boldsymbol{r}$ überführt in den Schritten:

1. (Faktorisierung)	$\boldsymbol{A} = \boldsymbol{R}^{\mathsf{T}}\boldsymbol{DR}$	$\Rightarrow \boldsymbol{R}$ und $\boldsymbol{D}$,
2. (Vorwärtselimination)	$\boldsymbol{R}^{\mathsf{T}}\boldsymbol{z} = \boldsymbol{a}$	$\Rightarrow \boldsymbol{z}$,
	$\boldsymbol{Dr} = \boldsymbol{z}$	$\Rightarrow \boldsymbol{r}$,
3. (Rückwärtselimination)	$\boldsymbol{Rx} = \boldsymbol{r}$	$\Rightarrow \boldsymbol{x}$.

Durchführung des Verfahrens

Die Elemente von $\boldsymbol{A}, \boldsymbol{R}, \boldsymbol{D}, \boldsymbol{r}, \boldsymbol{z}, \boldsymbol{x}, \boldsymbol{a}$ werden wie folgt bezeichnet:

$$\boldsymbol{A} = \begin{pmatrix} d_1 & c_1 & & & c_n \\ c_1 & d_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & c_{n-2} & d_{n-1} & c_{n-1} \\ c_n & & & c_{n-1} & d_n \end{pmatrix}, \boldsymbol{R} = \begin{pmatrix} 1 & \gamma_1 & & & \delta_1 \\ & 1 & \gamma_2 & & \delta_2 \\ & & \ddots & \ddots & \vdots \\ & & & \gamma_{n-2} & \delta_{n-2} \\ & & & 1 & \gamma_{n-1} \\ & & & & 1 \end{pmatrix},$$

$$\boldsymbol{D} = \begin{pmatrix} \alpha_1 & & & \\ & \alpha_2 & & \\ & & \ddots & \\ & & & \alpha_n \end{pmatrix}, \boldsymbol{r} = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{pmatrix}, \boldsymbol{a} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}, \boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \boldsymbol{z} = \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{pmatrix}.$$

Durch Koeffizientenvergleich in den drei Schritten ergibt sich der folgende Algorithmus.

Algorithmus 4.76.

Gegeben: $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}$ mit symmetrischer, zyklisch tridiagonaler, positiv definiter Matrix $\boldsymbol{A}$.

Gesucht: $\boldsymbol{x} = (x_i), i = 1(1)n$, $n \geq 4$.

Dann sind nacheinander folgende Schritte auszuführen:

1. (Faktorisierung $\boldsymbol{A} = \boldsymbol{R}^{\mathsf{T}}\boldsymbol{D}\boldsymbol{R}$)
   - 1.1 $\alpha_1 = d_1$
   - 1.2 $\gamma_1 = c_1/\alpha_1$
   - 1.3 $\delta_1 = c_n/\alpha_1$
   - 1.4 Für jedes $i = 2(1)n{-}2$ sind zu berechnen:
     - 1.4.1 $\alpha_i = d_i - c_{i-1}\gamma_{i-1}$
     - 1.4.2 $\gamma_i = c_i/\alpha_i$
     - 1.4.3 $\delta_i = -\delta_{i-1}c_{i-1}/\alpha_i$
   - 1.5 $\alpha_{n-1} = d_{n-1} - c_{n-2}\gamma_{n-2}$
   - 1.6 $\gamma_{n-1} = (c_{n-1} - c_{n-2}\delta_{n-2})/\alpha_{n-1}$
   - 1.7 $\alpha_n = d_n - \sum_{i=1}^{n-2} \alpha_i\delta_i^2 - c_{n-1}\gamma_{n-1}$
2. (Vorwärtselimination $\boldsymbol{R}^{\mathsf{T}}\boldsymbol{z} = \boldsymbol{a}$, $\boldsymbol{D}\boldsymbol{r} = \boldsymbol{z}$)
   - 2.1 $z_1 = a_1$
   - 2.2 Für jedes $i = 2(1)n{-}1$ ist zu berechnen: $z_i = a_i - z_{i-1}\gamma_{i-1}$
   - 2.3 $z_n = a_n - \sum_{i=1}^{n-2} \delta_i z_i - \gamma_{n-1}z_{n-1}$
   - 2.4 Für jedes $i = 1(1)n$ ist zu berechnen: $r_i = z_i/\alpha_i$
3. (Rückwärtselimination $\boldsymbol{R}\boldsymbol{x} = \boldsymbol{r}$)
   - 3.1 $x_n = r_n$
   - 3.2 $x_{n-1} = r_{n-1} - \gamma_{n-1}x_n$
   - 3.3 Für jedes $i = n{-}2(-1)1$ ist zu berechnen: $x_i = r_i - \gamma_i x_{i+1} - \delta_i x_n$

Für die Determinante von $\boldsymbol{A}$ gilt:

$$\det \boldsymbol{A} = \det(\boldsymbol{R}^{\mathsf{T}}\boldsymbol{D}\boldsymbol{R}) = \det(\boldsymbol{R}^{\mathsf{T}})\ \det \boldsymbol{D}\ \det \boldsymbol{R} = \det \boldsymbol{D} = \alpha_1\alpha_2 \ldots \alpha_n.$$

4.11 Gleichungssysteme mit fünfdiagonaler Matrix

4.11.1 Systeme mit fünfdiagonaler Matrix

Eine Matrix $\boldsymbol{A} = (a_{ik})$, $i,k = 1(1)n$, heißt fünfdiagonal, falls gilt $a_{ik} = 0$ für $|i-k| \geq 2$, $n \geq 4$, $i,k = 1(1)n$.

Prinzip des Lösungsverfahrens

Mit der Zerlegung $\boldsymbol{A} = \boldsymbol{LR}$, wobei $\boldsymbol{L}$ eine tridiagonale Subdiagonalmatrix und $\boldsymbol{R}$ eine normierte tridiagonale Superdiagonalmatrix ist, lässt sich ein System $\boldsymbol{Ax} = \boldsymbol{a}$ mit fünfdiagonaler Matrix $\boldsymbol{A}$ ohne Zeilenvertauschung gemäß Algorithmus 4.42 in ein äquivalentes System $\boldsymbol{Rx} = \boldsymbol{r}$ überführen, sofern $\det(\boldsymbol{A}_k) \neq 0$ für $k = 1(1)n-1$:

1. (Faktorisierung) $\boldsymbol{A} = \boldsymbol{LR} \quad \Rightarrow \boldsymbol{L}$ und $\boldsymbol{R}$,
2. (Vorwärtselimination) $\boldsymbol{a} = \boldsymbol{Lr} \quad \Rightarrow \boldsymbol{r}$,
3. (Rückwärtselimination) $\boldsymbol{Rx} = \boldsymbol{r} \quad \Rightarrow \boldsymbol{x}$.

Durchführung des Verfahrens

Wegen der Speicherung der Diagonalen als Vektoren werden die Elemente der beteiligten Vektoren und Matrizen wie folgt bezeichnet und anschließend durch Koeffizientenvergleich in den Schritten 1., 2., 3. ermittelt:

$$\boldsymbol{A} = \begin{pmatrix} d_1 & e_1 & f_1 & & & & \\ c_2 & d_2 & e_2 & f_2 & & & \\ g_3 & c_3 & d_3 & e_3 & f_3 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & g_{n-2} & c_{n-2} & d_{n-2} & e_{n-2} & f_{n-2} \\ & & & g_{n-1} & c_{n-1} & d_{n-1} & e_{n-1} \\ & & & & g_n & c_n & d_n \end{pmatrix}, \quad \boldsymbol{a} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_{n-1} \\ a_n \end{pmatrix}, \quad \boldsymbol{r} = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_{n-1} \\ r_n \end{pmatrix},$$

$$\boldsymbol{L} = \begin{pmatrix} \alpha_1 & & & & & \\ \beta_2 & \alpha_2 & & & & \\ \varepsilon_3 & \beta_3 & \alpha_3 & & & \\ & \ddots & \ddots & \ddots & & \\ & & \varepsilon_{n-1} & \beta_{n-1} & \alpha_{n-1} & \\ & & & \varepsilon_n & \beta_n & \alpha_n \end{pmatrix}, \quad \boldsymbol{R} = \begin{pmatrix} 1 & \gamma_1 & \delta_1 & & & & \\ & 1 & \gamma_2 & \delta_2 & & & \\ & & 1 & \gamma_3 & \delta_3 & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & 1 & \gamma_{n-2} & \delta_{n-2} \\ & & & & & 1 & \gamma_{n-1} \\ & & & & & & 1 \end{pmatrix}.$$

Algorithmus 4.77.

Gegeben: $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}$ mit fünfdiagonaler Matrix $\boldsymbol{A}$, $\det(\boldsymbol{A}_k) \neq 0$ für $k = 1(1)n-1$.
Gesucht: $\boldsymbol{x} = (x_i)$, $i = 1(1)n$, $n \geq 5$.

Dann sind nacheinander folgende Schritte auszuführen:

1. (Faktorisierung $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{R}$)
   - 1.1 $\alpha_1 = d_1$
   - 1.2 $\gamma_1 = e_1/\alpha_1$
   - 1.3 $\delta_1 = f_1/\alpha_1$
   - 1.4 $\beta_2 = c_2$
   - 1.5 $\alpha_2 = d_2 - \beta_2\gamma_1$
   - 1.6 $\gamma_2 = (e_2 - \beta_2\delta_1)/\alpha_2$
   - 1.7 $\delta_2 = f_2/\alpha_2$
   - 1.8 Für jedes $i = 3(1)n-2$ sind zu berechnen:
     - 1.8.1 $\beta_i = c_i - g_i\gamma_{i-2}$
     - 1.8.2 $\alpha_i = d_i - g_i\delta_{i-2} - \beta_i\gamma_{i-1}$
     - 1.8.3 $\gamma_i = (e_i - \beta_i\delta_{i-1})/\alpha_i$
     - 1.8.4 $\delta_i = f_i/\alpha_i$
   - 1.9 $\beta_{n-1} = c_{n-1} - g_{n-1}\gamma_{n-3}$
   - 1.10 $\alpha_{n-1} = d_{n-1} - g_{n-1}\delta_{n-3} - \beta_{n-1}\gamma_{n-2}$
   - 1.11 $\gamma_{n-1} = (e_{n-1} - \beta_{n-1}\delta_{n-2})/\alpha_{n-1}$
   - 1.12 $\beta_n = c_n - g_n\gamma_{n-2}$
   - 1.13 $\alpha_n = d_n - g_n\delta_{n-2} - \beta_n\gamma_{n-1}$
   - 1.14 Für jedes $i = 3(1)n$
     $\varepsilon_i = g_i$
2. (Vorwärtselimination $\boldsymbol{a} = \boldsymbol{L}\boldsymbol{r}$)
   - 2.1 $r_1 = a_1/\alpha_1$
   - 2.2 $r_2 = (a_2 - \beta_2 r_1)/\alpha_2$
   - 2.3 Für jedes $i = 3(1)n$ sind zu berechnen:
     $r_i = (a_i - \varepsilon_i r_{i-2} - \beta_i r_{i-1})/\alpha_i$
3. (Rückwärtselimination $\boldsymbol{R}\boldsymbol{x} = \boldsymbol{r}$)
   - 3.1 $x_n = r_n$
   - 3.2 $x_{n-1} = r_{n-1} - \gamma_{n-1}x_n$
   - 3.3 Für jedes $i = n-2(-1)1$ ist zu berechnen:
     $x_i = r_i - \gamma_i x_{i+1} - \delta_i x_{i+2}$

Für die Determinante von $\boldsymbol{A}$ gilt:

$$\det \boldsymbol{A} = \det \boldsymbol{L} \ \det \boldsymbol{R} = \det \boldsymbol{L} = \alpha_1\alpha_2 \ldots \alpha_n.$$

Beispiel 4.78.

Gegeben: $\boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{a}$ mit

$$\boldsymbol{A} = \begin{pmatrix} 2 & -2 & -2 & 0 & 0 & 0 \\ -2 & 5 & -4 & -3 & 0 & 0 \\ -1 & -2 & 11 & -1 & -4 & 0 \\ 0 & -1 & 1 & 7 & -4 & -10 \\ 0 & 0 & -1 & -1 & 9 & -8 \\ 0 & 0 & 0 & -1 & 0 & 5 \end{pmatrix}, \quad \boldsymbol{a} = \begin{pmatrix} -2 \\ -4 \\ 3 \\ -7 \\ -1 \\ 4 \end{pmatrix}, \quad \boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix}$$

Gesucht: $\boldsymbol{x}$, $\det(\boldsymbol{A})$ mit Algorithmus 4.77

Lösung:

1. Dreieckszerlegung $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{R}$ mit normierter oberer Dreiecksmatrix $\boldsymbol{R}$ und unterer Dreiecksmatrix $\boldsymbol{L}$

$$\boldsymbol{A} = \begin{pmatrix} 2 & -2 & -2 & 0 & 0 & 0 \\ -2 & 5 & -4 & -3 & 0 & 0 \\ -1 & -2 & 11 & -1 & -4 & 0 \\ 0 & -1 & 1 & 7 & -4 & -10 \\ 0 & 0 & -1 & -1 & 9 & -8 \\ 0 & 0 & 0 & -1 & 0 & 5 \end{pmatrix} = \boldsymbol{L}\boldsymbol{R}$$

$$\boldsymbol{L}\boldsymbol{R} = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ -2 & 3 & 0 & 0 & 0 & 0 \\ -1 & -3 & 4 & 0 & 0 & 0 \\ 0 & -1 & -1 & 5 & 0 & 0 \\ 0 & 0 & -1 & -2 & 6 & 0 \\ 0 & 0 & 0 & -1 & -1 & 1 \end{pmatrix} \begin{pmatrix} 1 & -1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -2 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 & -2 \\ 0 & 0 & 0 & 0 & 1 & -2 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

2. Vorwärtselimination $\boldsymbol{a} = \boldsymbol{L}\,\boldsymbol{r}$

$$\begin{pmatrix} -2 \\ -4 \\ 3 \\ -7 \\ -1 \\ 4 \end{pmatrix} = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ -2 & 3 & 0 & 0 & 0 & 0 \\ -1 & -3 & 4 & 0 & 0 & 0 \\ 0 & -1 & -1 & 5 & 0 & 0 \\ 0 & 0 & -1 & -2 & 6 & 0 \\ 0 & 0 & 0 & -1 & -1 & 1 \end{pmatrix} \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \\ r_6 \end{pmatrix} \quad \Longrightarrow \quad \boldsymbol{r} = \begin{pmatrix} -1 \\ -2 \\ -1 \\ -2 \\ -1 \\ 1 \end{pmatrix}$$

3. Rückwärtselimination $\boldsymbol{R}\,\boldsymbol{x} = \boldsymbol{r}$

$$\begin{pmatrix} 1 & -1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -2 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 & -2 \\ 0 & 0 & 0 & 0 & 1 & -2 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} = \begin{pmatrix} -1 \\ -2 \\ -1 \\ -2 \\ -1 \\ 1 \end{pmatrix} \quad \Longrightarrow \quad \boldsymbol{x} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

$\det(\boldsymbol{A}) = \det(\boldsymbol{L}) \cdot \det(\boldsymbol{R}) = \det(\boldsymbol{L}) = 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 = 720$ □

4.11.2 Systeme mit symmetrischer, fünfdiagonaler, positiv definiter Matrix

Ist $\boldsymbol{A}$ symmetrisch, fünfdiagonal und positiv definit, so kann zur Lösung von $\boldsymbol{Ax} = \boldsymbol{a}$ ein zum Cholesky-Verfahren äquivalentes Verfahren angewandt werden. Wegen der positiven Definitheit von $\boldsymbol{A}$ ist die Voraussetzung $\det(\boldsymbol{A}_k) \neq 0$, $k = 1(1)n$, für die Zerlegbarkeit ohne Zeilenvertauschungen immer erfüllt, d. h. $\boldsymbol{A}$ ist streng regulär. Es kann etwa die Hälfte Speicherplatz eingespart werden.

Prinzip des Verfahrens

Mit der Zerlegung $\boldsymbol{A} = \boldsymbol{R}^{\mathsf{T}}\boldsymbol{DR}$, wobei $\boldsymbol{R}$ eine normierte tridiagonale Superdiagonalmatrix ist und $\boldsymbol{D}$ eine Diagonalmatrix, lässt sich das System $\boldsymbol{Ax} = \boldsymbol{a}$ in ein äquivalentes System $\boldsymbol{Rx} = \boldsymbol{r}$ überführen gemäß Algorithmus 4.64:

1. (Faktorisierung)	$\boldsymbol{A} = \boldsymbol{R}^{\mathsf{T}}\boldsymbol{DR}$	$\Rightarrow \boldsymbol{R}$ und $\boldsymbol{D}$,
2. (Vorwärtselimination)	$\boldsymbol{R}^{\mathsf{T}}\boldsymbol{z} = \boldsymbol{a}$	$\Rightarrow \boldsymbol{z}$,
	$\boldsymbol{Dr} = \boldsymbol{z}$	$\Rightarrow \boldsymbol{r}$,
3. (Rückwärtselimination)	$\boldsymbol{Rx} = \boldsymbol{r}$	$\Rightarrow \boldsymbol{x}$.

Durchführung des Verfahrens

Wegen der vektoriellen Abspeicherung werden folgende Bezeichnungen eingeführt:

$$\boldsymbol{A} = \begin{pmatrix} d_1 & c_1 & e_1 & & & & \\ c_1 & d_2 & c_2 & e_2 & & & \\ e_1 & c_2 & d_3 & c_3 & e_3 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & e_{n-4} & c_{n-3} & d_{n-2} & c_{n-2} & e_{n-2} \\ & & & e_{n-3} & c_{n-2} & d_{n-1} & c_{n-1} \\ & & & & e_{n-2} & c_{n-1} & d_n \end{pmatrix}, \quad \boldsymbol{a} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_{n-1} \\ a_n \end{pmatrix},$$

$$\boldsymbol{D} = \begin{pmatrix} \alpha_1 & & & & \\ & \alpha_2 & & & \\ & & \alpha_3 & & \\ & & & \ddots & \\ & & & & \alpha_n \end{pmatrix}, \quad \boldsymbol{R} = \begin{pmatrix} 1 & \gamma_1 & \delta_1 & & & & \\ & 1 & \gamma_2 & \delta_2 & & & \\ & & 1 & \gamma_3 & \delta_3 & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & 1 & \gamma_{n-2} & \delta_{n-2} \\ & & & & & 1 & \gamma_{n-1} \\ & & & & & & 1 \end{pmatrix},$$

$$\boldsymbol{r} = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{pmatrix}, \quad \boldsymbol{z} = \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{pmatrix}.$$

Mit Koeffizientenvergleich ergeben sich aus den Schritten 1., 2., 3. die Formeln des folgenden Algorithmus:

Algorithmus 4.79.

Gegeben: $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}$ mit symmetrischer, fünfdiagonaler, positiv definiter Matrix $\boldsymbol{A}$.
Gesucht: $\boldsymbol{x} = (x_i)$, $i = 1(1)n$.

Dann sind nacheinander folgende Schritte auszuführen:

1. (Zerlegung $\boldsymbol{A} = \boldsymbol{R}^\mathsf{T}\boldsymbol{D}\boldsymbol{R}$)
   - 1.1 $\alpha_1 = d_1$
   - 1.2 $\gamma_1 = c_1/\alpha_1$
   - 1.3 $\delta_1 = e_1/\alpha_1$
   - 1.4 $\alpha_2 = d_2 - c_1\gamma_1$
   - 1.5 $\gamma_2 = (c_2 - e_1\gamma_1)/\alpha_2$
   - 1.6 $\delta_2 = e_2/\alpha_2$
   - 1.7 Für jedes $i = 3(1)n{-}2$ sind zu berechnen:
     - 1.7.1 $\alpha_i = d_i - e_{i-2}\delta_{i-2} - \alpha_{i-1}\gamma_{i-1}^2$
     - 1.7.2 $\gamma_i = (c_i - e_{i-1}\gamma_{i-1})/\alpha_i$
     - 1.7.3 $\delta_i = e_i/\alpha_i$
   - 1.8 $\alpha_{n-1} = d_{n-1} - e_{n-3}\delta_{n-3} - \alpha_{n-2}\gamma_{n-2}^2$
   - 1.9 $\gamma_{n-1} = (c_{n-1} - e_{n-2}\gamma_{n-2})/\alpha_{n-1}$
   - 1.10 $\alpha_n = d_n - e_{n-2}\delta_{n-2} - \alpha_{n-1}\gamma_{n-1}^2$
2. (Vorwärtselimination $\boldsymbol{R}^\mathsf{T}\boldsymbol{z} = \boldsymbol{a}$, $\boldsymbol{D}\boldsymbol{r} = \boldsymbol{z}$)
   - 2.1 $z_1 = a_1$
   - 2.2 $z_2 = a_2 - \gamma_1 z_1$
   - 2.3 Für jedes $i = 3(1)n$ ist zu berechnen:
     $z_i = a_i - \gamma_{i-1}z_{i-1} - \delta_{i-2}z_{i-2}$
   - 2.4 Für jedes $i = 1(1)n$ ist zu berechnen:
     $r_i = z_i/\alpha_i$
3. (Rückwärtselimination $\boldsymbol{R}\boldsymbol{x} = \boldsymbol{r}$)
   - 3.1 $x_n = r_n$
   - 3.2 $x_{n-1} = r_{n-1} - \gamma_{n-1}x_n$
   - 3.3 Für jedes $i = n{-}2(-1)1$ ist zu berechnen:
     $x_i = r_i - \gamma_i x_{i+1} - \delta_i x_{i+2}$

Für die Determinante von $\boldsymbol{A}$ gilt:

$$\det \boldsymbol{A} = \det(\boldsymbol{R}^\mathsf{T})\ \det \boldsymbol{D}\ \det \boldsymbol{R} = \det \boldsymbol{D} = \alpha_1\alpha_2 \ldots \alpha_n.$$

Beispiel 4.80.

Gegeben: $\boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{a}$ mit

$$\boldsymbol{A} = \begin{pmatrix} 2 & -2 & -2 & 0 & 0 & 0 \\ -2 & 5 & -1 & -3 & 0 & 0 \\ -2 & -1 & 9 & -1 & -4 & 0 \\ 0 & -3 & -1 & 12 & -1 & -5 \\ 0 & 0 & -4 & -1 & 15 & -1 \\ 0 & 0 & 0 & -5 & -1 & 12 \end{pmatrix}, \quad \boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix}, \quad \boldsymbol{a} = \begin{pmatrix} -2 \\ -1 \\ 1 \\ 2 \\ 9 \\ 6 \end{pmatrix},$$

$\boldsymbol{A}$ positiv definit

Gesucht: $\boldsymbol{x}$, $\det(\boldsymbol{A})$

Lösung:

1. Schritt: Faktorisierung $\boldsymbol{A} = \boldsymbol{R}^{\mathsf{T}}\boldsymbol{D}\boldsymbol{R}$ mit
normierter oberer Dreiecksmatrix $\boldsymbol{R}$ und Diagonalmatrix $\boldsymbol{D}$.

$$\boldsymbol{A} = \underset{\boldsymbol{R}^{\mathsf{T}}}{\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ -1 & -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & -1 & 1 \end{pmatrix}} \underset{\boldsymbol{D}}{\begin{pmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}} \underset{\boldsymbol{R}}{\begin{pmatrix} 1 & -1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}}$$

2. Schritt: $\boldsymbol{R}^{\mathsf{T}}\boldsymbol{z} = \boldsymbol{a}\,, \quad \boldsymbol{D}\,\boldsymbol{r} = \boldsymbol{z}$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ -1 & -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & -1 & 1 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \end{pmatrix} = \begin{pmatrix} -2 \\ -1 \\ 1 \\ 2 \\ 9 \\ 6 \end{pmatrix} \quad \Rightarrow \quad \boldsymbol{z} = \begin{pmatrix} -2 \\ -3 \\ -4 \\ -5 \\ 0 \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \\ r_6 \end{pmatrix} = \begin{pmatrix} -2 \\ -3 \\ -4 \\ -5 \\ 0 \\ 1 \end{pmatrix} \quad \Rightarrow \quad \boldsymbol{r} = \begin{pmatrix} -1 \\ -1 \\ -1 \\ -1 \\ 0 \\ 1 \end{pmatrix}$$

3. Schritt: $\boldsymbol{R}\boldsymbol{x} = \boldsymbol{r}$

$$\begin{pmatrix} 1 & -1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} = \begin{pmatrix} -1 \\ -1 \\ -1 \\ -1 \\ 0 \\ 1 \end{pmatrix} \quad \Rightarrow \quad \boldsymbol{z} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Es gilt: $\det(\boldsymbol{A}_i) > 0$ für $i = 1(1)6$, d. h.:

sämtliche Hauptabschnittsdeterminanten größer Null $\quad\Longrightarrow\quad$ $\boldsymbol{A}$ positiv definit

$\det(\boldsymbol{A}) = \det(\boldsymbol{R}) \cdot \det(\boldsymbol{D}) \cdot \det(\boldsymbol{R}) = \det(\boldsymbol{D}) = 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 = 720$ □

4.12 Gleichungssysteme mit Bandmatrix

Eine Matrix $\boldsymbol{A} = (a_{ik})$, $i, k = 1(1)n$, deren Elemente außerhalb eines Bandes längs der Hauptdiagonale verschwinden, heißt *Bandmatrix* oder *bandstrukturierte Matrix* (vgl. Definition 4.22).

Bei der Zerlegung $\boldsymbol{A} = \boldsymbol{LR}$ werden die Dreiecksmatrizen $\boldsymbol{R}$ und $\boldsymbol{L}$ ebenfalls bandförmig, wodurch sich der Rechenaufwand bei (gegenüber n) kleinen Zahlen m_ℓ (Anzahl der unteren Nebendiagonalen), m_r (Anzahl der oberen Nebendiagonalen) bedeutend verringert. Es sind somit die Algorithmen 4.42 und 4.44 anwendbar unter der Berücksichtigung, dass die Zerlegungsmatrizen bandförmig sind. Im Folgenden werden drei Algorithmen angegeben, von denen der erste eine Transformation von $\boldsymbol{A}$ auf obere Dreiecksform beinhaltet, d. h. bei der LR-Zerlegung ist $\boldsymbol{L}$ normierte untere Dreiecksmatrix und $\boldsymbol{R}$ obere Dreiecksmatrix. Der zweite Algorithmus gibt eine Transformation auf untere Dreiecksform an, d. h. bei der LR-Zerlegung ist $\boldsymbol{L}$ normierte obere Dreiecksmatrix und $\boldsymbol{R}$ untere Dreiecksmatrix.

Der dritte Algorithmus arbeitet mit einer gepackten Matrix $\boldsymbol{A}$ und transformiert je nach Rechenaufwand auf obere bzw. untere Dreiecksform. Das Packen von $\boldsymbol{A}$ wird so vorgenommen, dass die Diagonalen von $\boldsymbol{A}$ zu Spalten der gepackten Matrix werden; dieser Algorithmus wurde von Elmar Pohl entwickelt. Hier wird auf eine Darstellung mit der Permutationsmatrix $\boldsymbol{P}$ (vgl. Algorithmus 4.44) verzichtet. Stattdessen werden die Zeilenvertauschungen mit dem Vektor $\boldsymbol{p}$ verwaltet.

Es bedeutet: $p_i = k$: Zeile i wurde mit der Zeile $i + k$ vertauscht.

Falls $p_i = 0$ wurde keine Vertauschung vorgenommen. Der Parameter SIG liefert das Vorzeichen der Determinante von $\boldsymbol{A}$, es gilt nach der Zerlegung bzw. Transformation auf obere oder untere Dreiecksform

$$\det \boldsymbol{A} = \text{ SIG } r_{11} r_{22} \dots r_{nn}$$

mit SIG $= (-1)^k$, $k =$ Anzahl der Zeilenvertauschungen.

In den folgenden Algorithmen wird $\boldsymbol{A}$ mit den Zerlegungsmatrizen überspeichert, so dass sich für die Determinante ergibt

$$\det \boldsymbol{A} = \text{ SIG } a_{11} a_{22} \dots a_{nn}.$$

Algorithmus 4.81. (*Transformation auf obere Dreiecksform*)

Gegeben: $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}$ mit der Bandmatrix $\boldsymbol{A} = (a_{ik})$, $i,k = 1(1)n$ (m_ℓ = Anzahl der unteren Nebendiagonalen, m_r = Anzahl der oberen Nebendiagonalen) und der rechten Seite $\boldsymbol{a} = (a_i)$, $i = 1(1)n$.

Gesucht: $\boldsymbol{x}$ mit $\boldsymbol{x} = (x_i)$, $i = 1(1)n$.

1. Faktorisierung $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{R}$ mit einer normierten unteren Dreiecksmatrix $\boldsymbol{L}$ und oberen Dreiecksmatrix $\boldsymbol{R}$ mit Pivotisierung, d. h. die Matrix $\boldsymbol{A}$ wird auf obere Dreiecksform transformiert.

   1.1 SIG := 1

   1.2 Für jedes $i = 1(1)n-1$

   1.2.1 $p_i := \nu$ für
   $|a_{i+\nu,i}| = \max\{|a_{i+k,i}|, k = 0(1)\min\{m_\ell, n-i\}\}$

   1.2.2 Wenn $\nu \neq 0$, dann

   1.2.2.1 SIG := −SIG

   1.2.2.2 Für jedes $k = 0(1)\min\{m_\ell + m_r, n-i\}$
   Vertausche $a_{i,i+k}$ mit $a_{i+\nu,i+k}$

   1.2.3 Für $k = 1(1)\min\{m_\ell, n-i\}$

   1.2.3.1 Ersetze $a_{i+k,i}$ durch $a_{i+k,i}/a_{ii}$

   1.2.3.2 Für $j = 1(1)\min\{m_\ell + m_r, n-i\}$
   Ersetze $a_{i+k,i+j}$ durch $a_{i+k,i+j} - a_{i+k,i}a_{i,i+j}$

   1.3 $p_n := 0$ (letzte Zeile wird nicht vertauscht)

2. Vorwärtselimination $\boldsymbol{a} = \boldsymbol{L}\boldsymbol{r}$, $\boldsymbol{a}$ wird mit $\boldsymbol{r}$ überspeichert, die Zeilenvertauschungen der Zerlegung werden berücksichtigt.

   2.1 Für $i = 1(1)n-1$ ist durchzuführen

   2.1.1 Wenn $p_i \neq 0$, dann vertausche a_i mit a_{i+p_i}

   2.1.2 Für $k = 1(1)\min\{m_\ell, n-i\}$
   Ersetze a_{i+k} durch $a_{i+k} - a_{i+k,i}a_i$

3. Rückwärtselimination $\boldsymbol{R}\boldsymbol{x} = \boldsymbol{r}$, die Lösung $\boldsymbol{x}$ wird in $\boldsymbol{a}$ gespeichert.

   3.1 Ersetze a_n durch a_n/a_{nn}

   3.2 Für jedes $i = n-1(-1)1$

   3.2.1 Für $k = 1(1)\min\{n-i, m_\ell + m_r\}$
   Ersetze a_i durch $a_i - a_{i,i+k}a_{i+k}$

   3.2.2 Ersetze a_i durch a_i/a_{ii}

Algorithmus 4.82. (*Transformation auf untere Dreiecksform*)

Gegeben: $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}$ mit $\boldsymbol{A} = (a_{ik})$, $i,k = 1(1)n$, bandförmig,
$\boldsymbol{a} = (a_i)$, $i = 1(1)n$, rechte Seite

Gesucht: $\boldsymbol{x} = (x_i)$, $i = 1(1)n$

1. Faktorisierung $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{R}$ mit einer normierten oberen Dreiecksmatrix $\boldsymbol{L}$ und unteren Dreiecksmatrix $\boldsymbol{R}$, d. h. Transformation von $\boldsymbol{A}$ auf untere Dreiecksform.

   1.1 SIG : = 1

   1.2 Für jedes $i = n(-1)2$

   1.2.1 $p_i := \nu$ für
   $|a_{i+\nu,i}| = \max\{|a_{i+k,i}|,\ k = 0(-1)\max\{1-i, -m_r\}\}$

   1.2.2 Wenn $\nu \neq 0$, dann

   1.2.2.1 $\bar{\text{S}}\text{IG} := -\text{SIG}$

   1.2.2.2 Für jedes $k = 0(-1)\max\{1-i, -m_r - m_\ell\}$
   Vertausche $a_{i,i+k}$ mit $a_{i+\nu,i+k}$

   1.2.3 Für $k = -1(-1)\max\{1-i, -m_r\}$

   1.2.3.1 Ersetze $a_{i+k,i}$ durch $a_{i+k,i}/a_{ii}$

   1.2.3.2 Für $j = -1(-1)\max\{1-i, m_r - m_\ell\}$ setze
   $a_{i+k,i+j} := a_{i+k,i+j} - a_{i+k,i}a_{i,i+j}$

   1.3 $p_1 := 0$ (Zeile 1 wird nicht vertauscht)

2. Rückwärtselimination ($\boldsymbol{a} = \boldsymbol{L}\boldsymbol{r}$, $\boldsymbol{r}$ wird auf $\boldsymbol{a}$ gespeichert)

   2.1 Für $i = n(-1)2$ ist durchzuführen:

   2.1.1 Wenn $p_i \neq 0$, dann vertausche a_i mit a_{i+p_i}

   2.1.2 Für $k = -1(-1)\max\{1-i, -m_r\}$ setze
   $a_{i+k} := a_{i+k} - a_{i+k,i}\ a_i$

3. Vorwärtselimination ($\boldsymbol{R}\boldsymbol{x} = \boldsymbol{r}$, $\boldsymbol{x}$ wird auf $\boldsymbol{a}$ gespeichert)

   3.1 $a_1 := a_1/a_{11}$

   3.2 Für jedes $i = 2(1)n$

   3.2.1 Für $k = -1(-1)\max\{1-i, -m_\ell - m_r\}$
   Ersetze a_i durch $a_i - a_{i,i+k}\ a_{i+k}$

   3.2.2 Ersetze a_i durch a_i/a_{ii}

Algorithmus 4.83. (*Gepackte Matrix, Transformation auf obere bzw. untere Dreiecksform*)

Gegeben: $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}$, $\boldsymbol{A} = (a_{ik})$, $i,k = 1(1)n$, bandförmig,
$\boldsymbol{a} = (a_i)$, $i = 1(1)n$, rechte Seite

Gesucht: $\boldsymbol{x} = (x_i)$, $i = 1(1)n$

1. Packung der Matrix $\boldsymbol{A}$ zu einer Matrix $\boldsymbol{A}^*$ mit
$a^*_{\alpha,\beta} := a_{\alpha,\alpha+\beta-m_\ell-1}$ bzw. $a_{ik} = a^*_{i,m_\ell+1+k-i}$
für $i = 1(1)n$, $k = \max\{1, i-m_\ell\}(1)\min\{n, i+m_r\}$

   Die (n,n)-Matrix $\boldsymbol{A}$ wird zu einer Matrix $\boldsymbol{A}^*$ mit n Zeilen und $m = m_\ell + m_r + 1$ Spalten.

Bemerkung. (zur Speicherung der zusätzlichen Nebendiagonalen)
Bei der Transformation auf untere Dreiecksform werden zusätzliche Nebendiagonalen unterhalb des Diagonalbandes erzeugt. Nach der in 1. beschriebenen Packung wären das hier Spalten links von Spalte 1 in $\boldsymbol{A}^*$. Da zu Beginn die Anzahl dieser Spalten unbekannt ist, werden die zusätzlichen Nebendiagonalen rechts von $\boldsymbol{A}^*$ gespeichert, so dass die Gesamtspaltenzahl von $\boldsymbol{A}^*$ ist:

$$m^* := m + \min\{m_\ell, m_r\}.$$

2. Faktorisierung

   2.1 SIG: $= 1$
   $m^* = m_\ell + m_r + 1 + \min\{m_\ell, m_r\}$

   2.2 Wenn $m_\ell \le m_r$, dann wird gesetzt
   $i_a := 1,\ i_e := n-1,\ i_s := 1,\ k_a := 1,\ j_a := 1,$
   andernfalls
   $i_a := n,\ i_e := 2,\ i_s := -1,\ k_a := -1,\ j_a := -1$

   2.3 Für jedes $i = i_a(i_s)i_e$ wird durchgeführt:

      2.3.1 Wenn $m_\ell \le m_r$, dann
      $k_e := \min\{m_\ell, n-i\}$
      andernfalls
      $k_e := \max\{1-i, -m_r\}$

      2.3.2 $p_i := \mu$ für
      $|a^*_{i+\mu,m_\ell+1-\mu}| = \max\{|a^*_{i+k,m_\ell+1-k}|,\ k = 0(i_s)k_e\}$

      2.3.3 Wenn $m_\ell \le m_r$, dann
      $j_e := \min\{\mu + m_r, n-i\}$
      andernfalls
      $j_e := \max\{1-i, \mu - m_\ell\}$

      2.3.4 Wenn $\mu \ne 0$, dann

         2.3.4.1 SIG: $= -$SIG

         2.3.4.2 Für $k = 0(i_s)j_e$

            2.3.4.2.1 $k_m := k + m_\ell + 1$

2.3.4.2.2 Wenn $k_m \le 0$, dann $k_m := k_m + m^*$

2.3.4.2.3 Vertausche a^*_{i,k_m} mit $a^*_{i+\mu,k_m-\mu}$

2.3.5 Für $k = k_a(i_s)k_e$

$a^*_{i+k,m_\ell+1-k} := a^*_{i+k,m_\ell+1-k}/a^*_{i,m_\ell+1}$

Für $j = j_a(i_s)j_e$

$j_k := j + m_\ell + 1 - k$

$j_m := j + m_\ell + 1$

Wenn $j_k \le 0$, dann $j_k := j_k + m^*$

Wenn $j_m \le 0$, dann $j_m := j_m + m^*$

Setze $a^*_{i+k,j_k} := a^*_{i+k,j_k} - a^*_{i+k,m_\ell+1-k}\, a^*_{i,j_m}$

2.4 $p_{i_e+i_s} := 0$

3. Rückwärts- oder Vorwärtselimination

3.1 $m^* := m_\ell + m_r + 1 + \min\{m_\ell, m_r\}$

Wenn $m_\ell \le m_r$, dann

$i_a := 1,\ i_e := n-1,\ i_s := 1,\ k_a := 1$

andernfalls

$i_a := n,\ i_e := 2,\ i_s := -1,\ k_a := -1$

3.2 Für $i = i_a(i_s)i_e$

3.2.1 Wenn $p_i \ne 0$, vertausche a_i mit a_{i+p_i}

3.2.2 Wenn $m_\ell \le m_r$, dann

$k_e := \min\{m_\ell, n-i\}$

andernfalls

$k_e := \max\{1-i, -m_r\}$

3.2.3 Für $k = k_a(i_s)k_e$

$a_{i+k} = a_{i+k} - a^*_{i+k,m_\ell+1-k} a_i$

3.3 $a_{i_e+i_s} = a_{i_e+i_s}/a^*_{i_e+i_s,m_e+1}$

3.4 Für $i = i_e(-i_s)i_a$

3.4.1 Wenn $m_\ell \le m_r$, dann

$k_e := \min\{n-i, m_\ell + m_r\}$

andernfalls

$k_e := \max\{1-i, -m_\ell - m_r\}$

3.4.2 Für $k = k_a(i_s)k_e$

3.4.2.1 $k_m := k + m_\ell + 1$

3.4.2.2 Wenn $k_m \le 0$, dann $k_m := k_m + m^*$

3.4.2.3 $a_i := a_i - a^*_{i,k_m}\, a_{i+k}$

3.4.3 $a_i := a_i/a^*_{i,m_\ell+1}$

Bemerkung. (zur Wahl der Transformation auf untere bzw. obere Dreiecksform)
Im Falle vollbesetzter Matrizen ist es gleichgültig, ob eine Transformation auf obere oder untere Dreiecksform erfolgt. Bei Bandmatrizen mit Zeilenvertauschungen ergibt sich jedoch ein Unterschied in der Rechenzeit und der Speichereffizienz im Falle $m_\ell \neq m_r$. Bei der Transformation auf obere Dreiecksform entstehen durch die Zeilenvertauschungen m_ℓ zusätzliche obere Nebendiagonalen, bei Transformation auf untere Dreiecksform m_r zusätzliche untere Nebendiagonalen. Bei der oberen Dreiecksform sind pro Eliminationsschritt m_ℓ Zeilen zu behandeln, bei der unteren Dreiecksform m_r Zeilen, so dass der Aufwand von der entsprechenden Wahl der Transformationsart abhängt.

Beispiel 4.84.

Problem: Zur Dimensionierung der Stäbe des skizzierten Kragarms sollen die durch die äußeren Kräfte F_1 und F_2 hervorgerufenen Stabkräfte ermittelt werden.

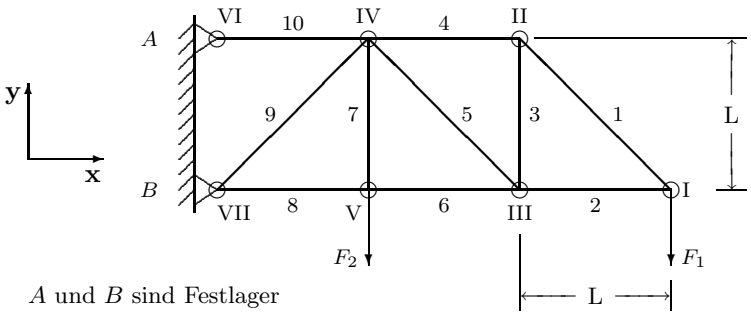


Abb. 4.8.

Die Formulierung der Gleichgewichtsbedingungen an den herausgeschnittenen Knotenpunkten liefert ein Gleichungssystem $\boldsymbol{A} \cdot \boldsymbol{x} = \boldsymbol{y}$.

Gegeben: $F_1 = 10\,\text{kN}$ und $F_2 = 20\,\text{kN}$.

Anleitung:
1. Man skizziere die herausgeschnittenen Knotenpunkte mit den an den durchgeschnittenen Stäben angreifenden Stabkräften und stelle das Gleichungssystem zur Bestimmung der Stabkräfte und der Auflagerkräfte A_x, B_x, A_y, B_y unter Vernachlässigung des Eigengewichts der Stäbe auf.
2. Man löse das System mit Hilfe einer geeigneten Variante des Gauß-Algorithmus.

Lösung: Man setzt zweckmässig alle Stabkräfte als Zugkräfte an (also die Pfeile von den Knotenpunkten weggerichtet), zerlegt sie in Komponenten bezüglich des x, y-Koordinatensystems und setzt die Gleichgewichtsbedingungen (Knotenschnittbedingungen $\sum F_{ix} = 0$, $\sum F_{iy} = 0$) für alle Knoten an.

Knoten I:

$$\sum F_{ix} = 0: \quad -S_1 \cdot \cos 45^o - S_2 = 0 \tag{1}$$

$$\sum F_{iy} = 0: \quad S_1 \cdot \sin 45^o - F_1 = 0 \tag{2}$$

Knoten II:

$$\sum F_{ix} = 0: \quad S_1 \cdot \cos 45^o - S_4 = 0 \tag{3}$$

$$\sum F_{iy} = 0: \quad -S_1 \cdot \sin 45^o - S_3 = 0 \tag{4}$$

Knoten III:

$$\sum F_{ix} = 0: \quad S_2 - S_5 \cdot \cos 45^o - S_6 = 0 \tag{5}$$

$$\sum F_{iy} = 0: \quad S_3 + S_5 \cdot \sin 45^o = 0 \tag{6}$$

Knoten IV:

$$\sum F_{ix} = 0: \quad S_4 + S_5 \cdot \cos 45^o - S_9 \cdot \cos 45^o - S_{10} = 0 \tag{7}$$

$$\sum F_{iy} = 0: \quad -S_5 \cdot \sin 45^o - S_7 - S_9 \cdot \sin 45^o = 0 \tag{8}$$

Knoten V:

$$\sum F_{ix} = 0: \quad S_6 - S_8 = 0 \tag{9}$$

$$\sum F_{iy} = 0: \quad S_7 - F_2 = 0 \tag{10}$$

Knoten VI:

$$\sum F_{ix} = 0: \quad S_{10} + A_x = 0 \tag{11}$$

$$\sum F_{iy} = 0: \quad A_y = 0 \tag{12}$$

Knoten VII:

$$\sum F_{ix} = 0: \quad S_8 + S_9 \cdot \cos 45^o + B_x = 0 \tag{13}$$

$$\sum F_{iy} = 0: \quad S_9 \cdot \sin 45^o + B_y = 0 \tag{14}$$

Mit $w := \frac{1}{2}\sqrt{2}$ ergibt sich folgende Gleichung:

$$
\begin{array}{l}
(1)\\(2)\\(3)\\(4)\\(5)\\(6)\\(7)\\(8)\\(9)\\(10)\\(11)\\(12)\\(13)\\(14)
\end{array}
\begin{pmatrix}
-w & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
w & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
w & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
-w & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 1 & 0 & 0 & -w & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 1 & 0 & w & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 1 & w & 0 & 0 & 0 & -w & -1 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & -w & 0 & -1 & 0 & -w & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & w & 0 & 0 & 0 & 1 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & w & 0 & 0 & 0 & 0 & 1
\end{pmatrix}
\begin{pmatrix} S_1\\S_2\\S_3\\S_4\\S_5\\S_6\\S_7\\S_8\\S_9\\S_{10}\\A_x\\A_y\\B_x\\B_y \end{pmatrix}
=
\begin{pmatrix} 0\\F_1\\0\\0\\0\\0\\0\\0\\0\\F_2\\0\\0\\0\\0 \end{pmatrix}.
$$

Durch Zeilenvertauschungen entsteht eine Bandmatrix (Breite 7):

$$
\begin{array}{l}
(2)\\(1)\\(4)\\(3)\\(5)\\(6)\\(7)\\(8)\\(9)\\(10)\\(13)\\(14)\\(11)\\(12)
\end{array}
\begin{pmatrix}
\boldsymbol{w} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
\boldsymbol{-w} & \boldsymbol{-1} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
\boldsymbol{-w} & \boldsymbol{0} & \boldsymbol{-1} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
\boldsymbol{w} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{-1} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & \boldsymbol{1} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{-w} & \boldsymbol{-1} & \boldsymbol{0} & \boldsymbol{0} & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & \boldsymbol{1} & \boldsymbol{0} & \boldsymbol{w} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & \boldsymbol{1} & \boldsymbol{w} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{-w} & \boldsymbol{-1} & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & \boldsymbol{-w} & \boldsymbol{0} & \boldsymbol{-1} & \boldsymbol{0} & \boldsymbol{-w} & \boldsymbol{0} & \boldsymbol{0} & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & \boldsymbol{1} & \boldsymbol{0} & \boldsymbol{-1} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & \boldsymbol{1} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \boldsymbol{1} & \boldsymbol{w} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{1} & \boldsymbol{0}\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \boldsymbol{w} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{1}\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \boldsymbol{1} & \boldsymbol{1} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0}\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \boldsymbol{0} & \boldsymbol{1} & \boldsymbol{0} & \boldsymbol{0}
\end{pmatrix}
\begin{pmatrix} S_1\\S_2\\S_3\\S_4\\S_5\\S_6\\S_7\\S_8\\S_9\\S_{10}\\A_x\\A_y\\B_x\\B_y \end{pmatrix}
=
\begin{pmatrix} F_1\\0\\0\\0\\0\\0\\0\\0\\0\\F_2\\0\\0\\0\\0 \end{pmatrix}.
$$

Mit dem Gauß-Algorithmus für Bandmatrizen erhält man folgende Lösung (jeweils in kN):

Stabkräfte

i	1	2	3	4	5	6	7	8	9	10
S_i	$10 \cdot \sqrt{2}$	-10	-10	10	$10 \cdot \sqrt{2}$	-20	20	-20	$-30 \cdot \sqrt{2}$	50

Auflagerkräfte

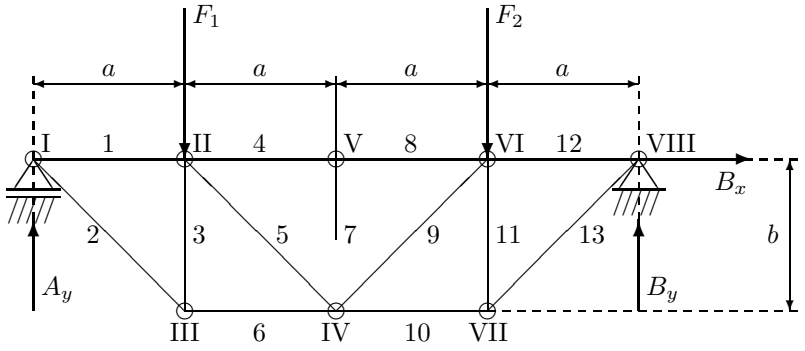
A_x	A_y	B_x	B_y
-50	0	50	30

□

Beispiel 4.85.
Gegeben ist ein Fachwerkträger mit 8 Knoten, 13 Stäben (S_1 bis S_{13}), den Auflagerkräften A_y, B_x und B_y und den äußeren Kräften F_1 und F_2.

Um die unbekannten Stabkräfte und Auflagerkräfte zu bestimmen, soll ein Gleichungssystem $\boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{y}$ formuliert und gelöst werden. Hierzu sind die Gleichgewichtsbedingungen an den herausgeschnittenen Knotenpunkten aufzustellen und in Matrizenform $\boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{y}$ zu übertragen.

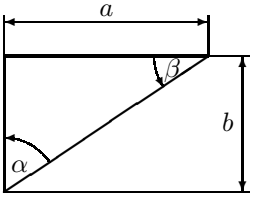
Skizze des Fachwerkträgers:



$F_1 = 30\,\text{kN}$, $F_2 = 50\,\text{kN}$, $a = 0.3\,\text{m}$ und $b = 0.2\,\text{m}$

Abb. 4.9.

Festlegen der Winkel α und β:

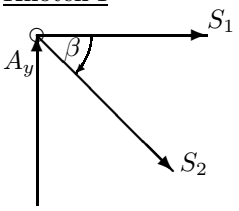


$$\tan\alpha = \frac{a}{b} = \frac{0.3}{0.2} \quad \Longrightarrow \quad \alpha = 56.31°$$

$$\tan\beta = \frac{b}{a} = \frac{0.2}{0.3} \quad \Longrightarrow \quad \beta = 33.69°$$

Gleichgewichtsbedingungen an herausgeschnittenen Knoten:

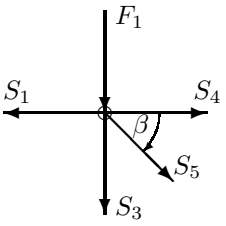
Knoten I



$$\sum F_x = 0 = S_1 + S_2 \cos\beta$$

$$\sum F_y = 0 = A_y - S_2 \sin\beta$$

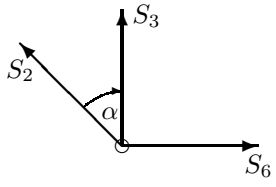
Knoten II



$$\sum F_x = 0 = -S_1 + S_4 + S_5 \cos\beta$$

$$\sum F_y = 0 = -F_1 - S_3 - S_5 \sin\beta$$

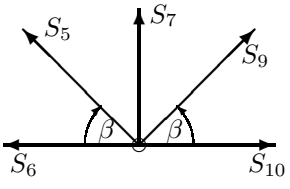
Knoten III



$$\sum F_x = 0 = -S_2 \sin\alpha + S_6$$

$$\sum F_y = 0 = S_3 + S_2 \cos\alpha$$

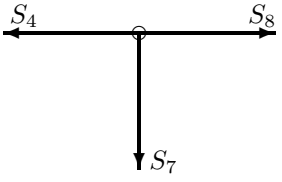
Knoten IV



$$\sum F_x = 0 = -S_5 \cos\beta - S_6 + S_9 \cos\beta + S_{10}$$

$$\sum F_y = 0 = S_5 \sin\beta + S_7 + S_9 \sin\beta$$

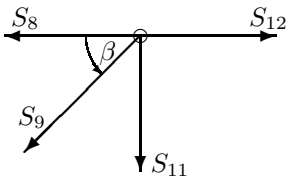
Knoten V



$$\sum F_x = 0 = -S_4 + S_8$$

$$\sum F_y = 0 = -S_7$$

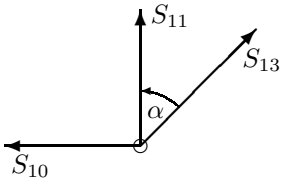
Knoten VI



$$\sum F_x = 0 = -S_8 - S_9 \cos\beta + S_{12}$$

$$\sum F_y = 0 = -S_{11} - S_9 \sin\beta$$

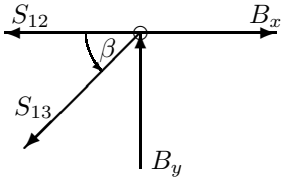
Knoten VII



$$\sum F_x = 0 = -S_{10} + S_{13} \sin\alpha$$

$$\sum F_y = 0 = S_{11} + S_{13} \cos\alpha$$

Knoten VIII



$$\sum F_x = 0 = -S_{12} - S_{13} \cos\beta + B_x$$

$$\sum F_y = 0 = B_y - S_{13} \sin\beta$$

Gleichungssystem:

$$\begin{pmatrix}
1 & \cos\beta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -\sin\beta & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 1 & \cos\beta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & -\sin\beta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -\cos\beta & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \sin\beta & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -\cos\beta & -1 & 0 & 0 & \cos\beta & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \sin\beta & 0 & 1 & 0 & \sin\beta & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -\cos\beta & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\sin\beta & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & \cos\beta & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \sin\beta & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -\cos\beta & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\sin\beta & 0 & 1
\end{pmatrix}
\begin{pmatrix} S_1 \\ S_2 \\ A_y \\ S_3 \\ S_4 \\ S_5 \\ S_6 \\ S_7 \\ S_8 \\ S_9 \\ S_{10} \\ S_{11} \\ S_{12} \\ S_{13} \\ B_x \\ B_y \end{pmatrix}
=
\begin{pmatrix} 0 \\ 0 \\ 0 \\ F_1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ F_2 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

mit $\sin\beta = 0.554700196$ und $\cos\beta = 0.832050294$.

Die Lösung dieses Gleichungssystems sind folgende Kräfte:

S_1	=	-52.5	kN	S_8	=	$-60.$	kN
S_2	=	63.097	kN	S_9	=	-9.014	kN
A_y	=	$35.$	kN	S_{10}	=	67.5	kN
S_3	=	$-35.$	kN	S_{11}	=	$-45.$	kN
S_4	=	$-60.$	kN	S_{12}	=	-67.5	kN
S_5	=	9.014	kN	S_{13}	=	81.125	kN
S_6	=	52.5	kN	B_x	=	$0.$	kN
S_7	=	$0.$	kN	B_y	=	$45.$	kN

□

4.13 Lösung überbestimmter linearer Gleichungssysteme mit Householdertransformation

Die Lösung überbestimmter linearer Systeme mit Hilfe der Householdertransformation ist von besonderer Bedeutung, weil bei der Dreieckszerlegung von $\boldsymbol{A}$ in das Produkt einer orthogonalen Matrix $\boldsymbol{Q}$ und einer oberen Dreiecksmatrix $\boldsymbol{R}$ die Kondition von $\boldsymbol{A}$ nicht verschlechtert wird (vgl. [STOE1989]).

Algorithmus 4.86. (*für ein überbestimmtes lineares Gleichungssystem*)

Gegeben: $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}$ mit (m,n)-Matrix $\boldsymbol{A}$, $m > n$, Rang $(\boldsymbol{A}) = n$ und $\boldsymbol{a} \in \mathbb{R}^m$.

Gesucht: $\boldsymbol{x} \in \mathbb{R}^n$.

1. Schritt: Zerlegung $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{R}$ mit einer orthogonalen (m,m)-Matrix $\boldsymbol{Q}$ und einer oberen (m,n)-Dreiecksmatrix $\boldsymbol{R}$. Es gilt wegen der Orthogonalität $\boldsymbol{Q}^{\mathsf{T}} = \boldsymbol{Q}^{-1}$.
2. Schritt: Berechnung von $\boldsymbol{b} \in \mathbb{R}^m$ aus $\boldsymbol{b} = \boldsymbol{Q}^{\mathsf{T}}\boldsymbol{a}$.
3. Schritt: Rückwärtselimination $\boldsymbol{R}\boldsymbol{x} = \boldsymbol{b}$ mit dem Ergebnis $\boldsymbol{x} \in \mathbb{R}^n$.

Householder-Transformation

Ziel der Householder-Transformation ist die Überführung der (m,n)-Matrix $\boldsymbol{A}$ vom Rang n in eine obere (m,n)-Dreiecksmatrix $\boldsymbol{R} = (r_{ik})$, $i = 1(1)m$, $k = 1(1)n$, $r_{ik} = 0$ für $i > k$, durch sukzessive Linksmultiplikation von $\boldsymbol{A}$ mit symmetrischen, orthogonalen (m,m)-Matrizen $\boldsymbol{H}_i$ (Householder-Matrizen) mit $\boldsymbol{H}_i\boldsymbol{H}_i^{\mathsf{T}} = \boldsymbol{E}$, $\boldsymbol{H}_i^{-1} = \boldsymbol{H}_i^{\mathsf{T}}$

$$\boldsymbol{A}_{q+1} = \underbrace{\boldsymbol{H}_q\boldsymbol{H}_{q-1}\ldots\boldsymbol{H}_2\boldsymbol{H}_1}_{=:\boldsymbol{H}}\boldsymbol{A}_1 =: \boldsymbol{H}\boldsymbol{A} = \boldsymbol{R}$$

mit $\boldsymbol{A}_{i+1} = \boldsymbol{H}_i\boldsymbol{A}_i$ für $i = 1(1)q$, $q = \min(m-1, n)$, $\boldsymbol{A}_1 := \boldsymbol{A}$.

Aus $\boldsymbol{R} = \boldsymbol{HA}$ ergibt sich eine QR-Zerlegung

$$\boldsymbol{A} = \boldsymbol{H}^{\mathsf{T}}\boldsymbol{R} =: \boldsymbol{QR}, \quad \boldsymbol{Q} := \boldsymbol{H}^{\mathsf{T}}$$

mit der orthogonalen Matrix $\boldsymbol{Q}(\boldsymbol{Q}^{\mathsf{T}} = \boldsymbol{Q}^{-1})$.

Durchführung der Householder-Transformation

Es wird gesetzt

$$\boldsymbol{A}_1 = (a_{ik}^{(1)}) := \boldsymbol{A} = (a_{ik}) \quad \text{für} \quad i = 1(1)m,\ k = 1(1)n.$$

Dann gilt

$$\boldsymbol{A}_1 = \begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \dots & a_{1n}^{(1)} \\ a_{21}^{(1)} & a_{22}^{(1)} & \dots & a_{2n}^{(1)} \\ \vdots & & & \\ a_{m1}^{(1)} & a_{m2}^{(1)} & \dots & a_{mn}^{(1)} \end{pmatrix} = (\boldsymbol{a}_1^{(1)}, \boldsymbol{a}_2^{(1)}, \dots, \boldsymbol{a}_n^{(1)}),$$

wobei $\boldsymbol{a}_j^{(1)}$ die j-te Spalte von $\boldsymbol{A}_1$ ist.

Mit der (m, m)-Householder-Matrix $\boldsymbol{H}_1$, die gemäß Satz 4.20 konstruiert wird,

$$\boldsymbol{H}_1 := \boldsymbol{E} - \frac{2}{\|\boldsymbol{v}_1\|^2}\boldsymbol{v}_1\boldsymbol{v}_1^{\mathsf{T}}$$

mit

$$\boldsymbol{v}_1 = \begin{pmatrix} a_{11}^{(1)} + \operatorname{sign}(a_{11}^{(1)})\|\boldsymbol{a}_1^{(1)}\| \\ a_{21}^{(1)} \\ \vdots \\ a_{m1}^{(1)} \end{pmatrix}$$

ergibt sich

$$\boldsymbol{A}_2 = \boldsymbol{H}_1\boldsymbol{A}_1 = \begin{pmatrix} a_{11}^{(2)} & a_{12}^{(2)} & \dots & a_{1n}^{(2)} \\ 0 & a_{22}^{(2)} & \dots & a_{2n}^{(2)} \\ \vdots & & & \\ 0 & a_{m2}^{(2)} & \dots & a_{mn}^{(2)} \end{pmatrix} = \begin{pmatrix} a_{11}^{(2)} & a_{12}^{(2)} & \dots & a_{1n}^{(2)} \\ 0 & & & \\ \vdots & & \widetilde{\boldsymbol{A}}_2 & \\ 0 & & & \end{pmatrix}$$

Analog fortfahrend erhält man

$$\boldsymbol{A}_{i+1} = \boldsymbol{H}_i\boldsymbol{A}_i \quad \text{für} \quad i = 1(1)p$$

mit der (m, n)-Matrix

$$\boldsymbol{A}_i = \begin{pmatrix} a_{11}^{(2)} & a_{12}^{(2)} & & & \dots & a_{1n}^{(2)} \\ & a_{22}^{(3)} & & & \dots & a_{2n}^{(3)} \\ & \ddots & & & & \\ & & a_{i-1,i-1}^{(i)} & \dots & & a_{i-1,n}^{(i)} \\ & & 0 & & & \\ & & \vdots & & \widetilde{\boldsymbol{A}}_i & \\ & & 0 & & & \end{pmatrix}, \qquad \widetilde{\boldsymbol{A}}_i = \begin{pmatrix} a_{ii}^{(i)} & \dots & a_{in}^{(i)} \\ \vdots & & \vdots \\ a_{mi}^{(i)} & \dots & a_{mn}^{(i)} \end{pmatrix}$$

und der (m,m)-Matrix

$$\boldsymbol{H}_i = \begin{pmatrix} \boldsymbol{E}_{i-1} & O \\ O & \widetilde{\boldsymbol{H}}_i \end{pmatrix} \begin{matrix} \} \\ \} \end{matrix} \begin{matrix} i{-}1 \text{ Zeilen} \\ m{-}i{+}1 \text{ Zeilen} \end{matrix}$$

wobei $\widetilde{\boldsymbol{H}}_i$ eine $(m{-}i{+}1,m{-}i{+}1)$-Matrix ist, die gemäß Satz 4.20 berechnet werden muss

$$\widetilde{\boldsymbol{H}}_i = \boldsymbol{E} - \frac{2}{\|\boldsymbol{v}_i\|^2} \boldsymbol{v}_i \boldsymbol{v}_i^{\mathsf{T}}.$$

Der Vektor $\boldsymbol{v}_i \in \mathbb{R}^{m-i+1}$ ergibt sich aus

$$\boldsymbol{v}_i = \begin{pmatrix} a_{ii}^{(i)} + \operatorname{sign}(a_{ii}^{(i)}) \|\boldsymbol{a}_i^{(i)}\| \\ a_{i+1,i}^{(i)} \\ \vdots \\ a_{mi}^{(i)} \end{pmatrix} \quad \text{mit} \quad \boldsymbol{a}_i = \begin{pmatrix} a_{ii}^{(i)} \\ a_{i+1,i}^{(i)} \\ \vdots \\ a_{mi}^{(i)} \end{pmatrix}.$$

Die (m,n)-Matrix $\boldsymbol{A}_{i+1}$ hat dann die Form

$$\boldsymbol{A}_{i+1} = \begin{pmatrix} a_{11}^{(2)} & a_{12}^{(2)} & & \dots & a_{1n}^{(2)} \\ & a_{22}^{(3)} & & \dots & a_{2n}^{(3)} \\ & \ddots & & & \\ & & a_{ii}^{(i+1)} & \dots & a_{in}^{(i+1)} \\ & & 0 & & \\ & & \vdots & \widetilde{\boldsymbol{A}}_{i+1} & \\ & & 0 & & \end{pmatrix} \quad \text{mit} \quad \widetilde{\boldsymbol{A}}_{i+1} = \begin{pmatrix} a_{i+1,i+1}^{(i+1)} & \dots & a_{i+1,n}^{(i+1)} \\ \vdots & & \vdots \\ a_{m,i+1}^{(i+1)} & \dots & a_{m,n}^{(i+1)} \end{pmatrix}.$$

Verwendet man die Householder-Transformation für die Erzeugung der QR-Zerlegung in Algorithmus 4.86 und berücksichtigt dabei die Beziehungen

$$\boldsymbol{R} = \boldsymbol{H}_q \boldsymbol{H}_{q-1} \cdots \boldsymbol{H}_1 \boldsymbol{A} =: \boldsymbol{H}\boldsymbol{A}$$

und

$$\boldsymbol{R}\boldsymbol{x} = \boldsymbol{H}\boldsymbol{A}\boldsymbol{x} = \boldsymbol{H}\,\boldsymbol{a} =: \boldsymbol{r},$$

so können die beiden ersten Schritte in Algorithmus 4.86 ersetzt werden durch die gleichzeitige Erzeugung von $\boldsymbol{R}$ und $\boldsymbol{r}$ durch sukzessive Multiplikation von $\boldsymbol{A}$ und $\boldsymbol{a}$ mit den Transformationsmatrizen $\boldsymbol{H}_i$; d. h. man umgeht die explizite Herstellung von $\boldsymbol{Q}$ und $\boldsymbol{Q}^{\mathsf{T}}$ und erhält den

Algorithmus 4.87. (*Householder-Transformation*)

Gegeben: $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}$, $\boldsymbol{A} = (a_{ik})$, $i = 1(1)m$, $k = 1(1)n$, $\boldsymbol{a} \in \mathbb{R}^m$, $m > n$, $\mathrm{Rg}(\boldsymbol{A}) = n$.

Gesucht: $\boldsymbol{x} \in \mathbb{R}^n$ mit Hilfe der Householder-Transformation.

1. (Householder-Transformation zur Erzeugung von $\boldsymbol{R} = \boldsymbol{H}\boldsymbol{A}$, $\boldsymbol{r} = \boldsymbol{H}\,\boldsymbol{a}$ mit $\boldsymbol{H} = \boldsymbol{H}_q\boldsymbol{H}_{q-1}\cdots\boldsymbol{H}_1$)
   Für jedes $i = 1(1)n$

   1.1 Berechnung der folgenden Größen in der angegebenen Reihenfolge
   $r := \sum_{k=i}^{m} a_{ki}^2$
   $\alpha := \sqrt{r}\ \mathrm{sign}(a_{ii})$
   $ak := 1/(r + \alpha \cdot a_{ii})$
   $a_{ii} := a_{ii} + \alpha$

   1.2 Multiplikation der Matrix $\boldsymbol{A}$ und der rechten Seite $\boldsymbol{a}$ (als $(n+1)$-te Spalte von $\boldsymbol{A}$) von links mit der neuen Transformationsmatrix $d_i := -\alpha$
   Für jedes $k = i{+}1(1)n{+}1$ sind durchzuführen
   $f := 0$
   Für jedes $j = i(1)m$
   $f := f + a_{jk}a_{ji}$
   $f := f \cdot ak$
   Für jedes $j = i(1)m$
   $a_{jk} := a_{jk} - f \cdot a_{ji}$

2. (Rückwärtselimination zur Bestimmung der Lösung $\boldsymbol{x}$ aus $\boldsymbol{R}\boldsymbol{x} = \boldsymbol{r}$)
   Für jedes $i = n(-1)1$ sind durchzuführen:
   $x_i := a_{i,n+1}$
   Für jedes $k = i + 1(1)n$
   $x_i := x_i - a_{ik}x_k$
   $x_i := x_i/d_i$

Beispiel 4.88.

Gegeben: Die Matrix

$$\boldsymbol{A} = \begin{pmatrix} 2 & 2 \\ 1 & 6 \end{pmatrix}$$

Gesucht: Überführung der Matrix $\boldsymbol{A}$ in eine obere Dreiecksmatrix $\boldsymbol{R}$ mit Hilfe der Householder-Transformation (Algorithmus 4.87, 1.).

Lösung:

$$\textcircled{1.}\quad i = 1 \quad r = \sum_{k=1}^{2} a_{k1}^2 = 4 + 1 = 5$$
$$\alpha = \sqrt{5} \cdot 1; \quad ak = \tfrac{1}{5+\sqrt{5}\cdot 2}; \quad a_{11} = a_{11} + \alpha = 2 + \sqrt{5}$$

$$
\begin{array}{lll}
\textcircled{2.} & d_1 = -\alpha = -\sqrt{5} \\
k=2; j=1(1)2 & f = a_{12}\cdot a_{11} + a_{22}\cdot a_{21} \\
& = 2\cdot(2+\sqrt{5}) + 6\cdot 1 \\
& = 4 + 2\sqrt{5} + 6 = 10 + 2\sqrt{5} \\
& f = f \cdot ak = \frac{10+2\sqrt{5}}{5+2\sqrt{5}} \\
j=1(1)2 & a_{12} = 2 - \frac{10+2\sqrt{5}}{5+2\sqrt{5}}\cdot(2+\sqrt{5}) \\
& a_{22} = 6 - \frac{10+2\sqrt{5}}{5+2\sqrt{5}}\cdot 1 \\
& \boldsymbol{v} = \begin{pmatrix} 2 \\ 1 \end{pmatrix} + \sqrt{5}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 2+\sqrt{5} \\ 1 \end{pmatrix} \\
& \boldsymbol{H}_1 = E - \frac{2}{||v||^2}\boldsymbol{v}\boldsymbol{v}^{\mathsf{T}}
\end{array}
$$

$$
\boldsymbol{v}^{\mathsf{T}}\boldsymbol{v} = (2+\sqrt{5},1)\begin{pmatrix} 2+\sqrt{5} \\ 1 \end{pmatrix} = (2+\sqrt{5})^2 + 1 = 10 + 4\sqrt{5}
$$

$$
\begin{aligned}
\boldsymbol{v}\boldsymbol{v}^{\mathsf{T}} &= \begin{pmatrix} 2+\sqrt{5} \\ 1 \end{pmatrix}(2+\sqrt{5},1) \\
&= \begin{pmatrix} (2+\sqrt{5})^2 & 2+\sqrt{5} \\ 2+\sqrt{5} & 1 \end{pmatrix} = \begin{pmatrix} (9+4\sqrt{5})^2 & 2+\sqrt{5} \\ 2+\sqrt{5} & 1 \end{pmatrix}
\end{aligned}
$$

$$
\boldsymbol{H}_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \frac{2}{10+4\sqrt{5}}\begin{pmatrix} 9+4\sqrt{5} & 2+\sqrt{5} \\ 2+\sqrt{5} & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \frac{1}{5+2\sqrt{5}}\begin{pmatrix} 9+4\sqrt{5} & 2+\sqrt{5} \\ 2+\sqrt{5} & 1 \end{pmatrix}
$$

$$
= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} -\frac{9+4\sqrt{5}}{5+2\sqrt{5}} & -\frac{2+\sqrt{5}}{5+2\sqrt{5}} \\ -\frac{2+\sqrt{5}}{5+2\sqrt{5}} & -\frac{1}{5+2\sqrt{5}} \end{pmatrix} = \begin{pmatrix} \frac{5+2\sqrt{5}-9-4\sqrt{5}}{5+2\sqrt{5}} & -\frac{2+\sqrt{5}}{5+2\sqrt{5}} \\ -\frac{2+\sqrt{5}}{5+2\sqrt{5}} & \frac{5+2\sqrt{5}-1}{5+2\sqrt{5}} \end{pmatrix}
$$

Mit $\boldsymbol{A}_1 = \boldsymbol{A}$ folgt

$$
\boldsymbol{A}_2 = \boldsymbol{H}_1\boldsymbol{A}_1 = \begin{pmatrix} \frac{-4-2\sqrt{5}}{5+2\sqrt{5}} & -\frac{2+\sqrt{5}}{5+2\sqrt{5}} \\ -\frac{2+\sqrt{5}}{5+2\sqrt{5}} & \frac{4+2\sqrt{5}}{5+2\sqrt{5}} \end{pmatrix}\begin{pmatrix} 2 & 2 \\ 1 & 6 \end{pmatrix}
$$

$$
= \begin{pmatrix} -\frac{-8-4\sqrt{5}-2-\sqrt{5}}{5+2\sqrt{5}} & -\frac{-8-4\sqrt{5}-12-6\sqrt{5}}{5+2\sqrt{5}} \\ 0 & \frac{-4-2\sqrt{5}+24+12\sqrt{5}}{5+2\sqrt{5}} \end{pmatrix}
$$

$$
= \frac{1}{5+2\sqrt{5}}\begin{pmatrix} -10-5\sqrt{5} & -20-10\sqrt{5} \\ 0 & 20+10\sqrt{5} \end{pmatrix} = \begin{pmatrix} -\sqrt{5} & -2\sqrt{5} \\ 0 & 2\sqrt{5} \end{pmatrix} = \boldsymbol{R}
$$

□

4.14 Fehler, Kondition und Nachiteration

4.14.1 Fehler und Kondition

Die mit Hilfe direkter Methoden ermittelte Lösung eines linearen Gleichungssystems ist meist nicht die exakte Lösung, da

1. im Verlaufe der Rechnung Rundungsfehler auftreten, deren Akkumulation zur Verfälschung der Ergebnisse führen kann.
2. Ungenauigkeiten in den Ausgangsgrößen bestehen können, die Ungenauigkeiten in den Lösungen hervorrufen.

Wenn kleine Änderungen in den Ausgangsdaten große Änderungen in der Lösung hervorrufen, heißt die Lösung *instabil*; man spricht von einem *schlecht konditionierten* System.

Beispiel 4.89. (zu Punkt 2.)

Gegeben: Das Gleichungssystem

$$\left\{\begin{array}{rcl} 3.3x_1 + 1.2x_2 & = & 1.1 \\ 6.9x_1 + 2.5x_2 & = & 2.7 \end{array}\right. \tag{4.14}$$

dessen exakte Lösung $x_1 = 16.\overline{3},\ x_2 = -44$ ist.

Der Koeffizient $a_{11} = 3.3$ wird geringfügig abgeändert, so dass man zu dem neuen System

$$\left\{\begin{array}{rcl} 3.31x_1 + 1.2x_2 & = & 1.1 \\ 6.9\ x_1 + 2.5x_2 & = & 2.7 \end{array}\right. \tag{4.15}$$

gelangt. Dieses System hat die exakte Lösung $x_1 = 98$, $x_2 = -269.4$; d. h. die Änderung des Koeffizienten a_{11} um $1 \cdot 10^{-2}$ ruft also eine sehr große Änderung in der Lösung hervor. Das System (4.14) ist demnach schlecht konditioniert und seine Lösung instabil. □

Woher kommt diese Instabilität? Man betrachte das System $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}$ und stelle sich die Aufgabe, $\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{a}$ zu berechnen. Sind nun die a_{ik} nur näherungsweise gegeben, so hat möglicherweise die Frage, ob $\det \boldsymbol{A} = 0$ ist oder nicht, gar keinen Sinn; denn es könnte der Fall eintreten, dass $\det \boldsymbol{A} \neq 0$ ist, wenn man die a_{ik} als exakt gegeben ansieht, dass jedoch bei einer Änderung der a_{ik} in ihren Genauigkeitsgrenzen $\det \boldsymbol{A} = 0$ wird. Es ist völlig klar, dass sich ein solches System nicht befriedigend lösen lässt.

Es ist erforderlich, ein Maß für die Güte einer Näherungslösung $\boldsymbol{x}^{(0)}$ für $\boldsymbol{x}$ zu finden. Das Einsetzen der Näherungslösung $\boldsymbol{x}^{(0)}$ in das System $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}$ liefert den Fehlervektor

$$\boldsymbol{r}^{(0)} = \boldsymbol{a} - \boldsymbol{A}\boldsymbol{x}^{(0)}; \tag{4.16}$$

man bezeichnet $\boldsymbol{r}^{(0)}$ auch als das *Residuum*. Ist $\boldsymbol{x}^{(0)}$ eine gute Approximation der exakten Lösung $\boldsymbol{x}$, so werden notwendig die Komponenten von $\boldsymbol{r}^{(0)}$ sehr klein sein, so dass gilt $|\boldsymbol{r}^{(0)}| < \varepsilon$. Umgekehrt ist $|\boldsymbol{r}^{(0)}| < \varepsilon$ nicht hinreichend dafür, dass $\boldsymbol{x}^{(0)}$ eine gute Approximation für $\boldsymbol{x}$ darstellt; das gilt nur für die Lösungen gut konditionierter Systeme. Das Residuum ist also als Maß für die Güte einer Näherungslösung nicht geeignet.

Ebensowenig reicht als Kennzeichen für schlechte Kondition die Kleinheit des Betrages der Determinante aus.

Beispiel 4.90. (Fortsetzung von Beispiel 4.89)

Angenommen, man hätte die exakte Lösung des Systems (4.14) als Näherungslösung $\boldsymbol{x}^{(0)}$ für das System (4.15) erhalten. Dann ergibt sich für das Residuum nach (4.15)

$$\boldsymbol{r}^{(0)} = \begin{pmatrix} 1.1 \\ 2.7 \end{pmatrix} - \begin{pmatrix} 3.31 & 1.2 \\ 6.9 & 2.5 \end{pmatrix} \begin{pmatrix} 16.\overline{3} \\ -44 \end{pmatrix} = \begin{pmatrix} -0.16\overline{3} \\ 0 \end{pmatrix},$$

also $|\boldsymbol{r}^{(0)}| = 0.163 < 0.2$. Mit $\varepsilon = 2 \cdot 10^{-1}$ sieht das Residuum vernünftig aus, obwohl die Lösung, wie man weiß, völlig falsch ist. Da das System schlecht konditioniert ist, kann man aus $|\boldsymbol{r}^{(0)}| < \varepsilon$ nicht auf die Güte der Lösung schließen. □

Nun entsteht die Frage, wie man die Kondition definieren kann. Sicher ist die Kleinheit des Betrages der Determinante ein Kennzeichen für schlechte Kondition (die Determinante des Systems (4.14) ist $3 \cdot 10^{-2}$, die von (4.15) ist $5 \cdot 10^{-3}$). Multipliziert man jedoch eine Zeile eines Systems, dessen Determinante „klein“ ist, mit einem genügend großen Faktor, so wird auch die Determinante „größer“, ohne dass damit an der Kondition eines Systems etwas verbessert wäre.

Dividiert man jedoch die Elemente jeder Zeile einer Matrix $\boldsymbol{A} = (a_{ik})$ durch die Quadratwurzel aus der Summe der Quadrate ihrer Elemente (also durch den Betrag des Zeilenvektors), so erhält man dadurch eine Matrix

$$\boldsymbol{A}^* = \begin{pmatrix} \frac{a_{11}}{\alpha_1} & \frac{a_{12}}{\alpha_1} & \dots & \frac{a_{1n}}{\alpha_1} \\ \frac{a_{21}}{\alpha_2} & \frac{a_{22}}{\alpha_2} & \dots & \frac{a_{2n}}{\alpha_2} \\ \vdots & \vdots & & \vdots \\ \frac{a_{n1}}{\alpha_n} & \frac{a_{n2}}{\alpha_n} & \dots & \frac{a_{nn}}{\alpha_n} \end{pmatrix}$$

$$\text{mit} \qquad \alpha_i = \sqrt{a_{i1}^2 + a_{i2}^2 + \ldots + a_{in}^2} = \sqrt{\sum_{k=1}^{n} a_{ik}^2}, \quad i = 1(1)n,$$

deren Determinante $\det \boldsymbol{A}^*$ invariant gegenüber Multiplikationen der Zeilen von $\boldsymbol{A}$ mit beliebigen von Null verschiedenen Faktoren ist. Ebenso ist $|\det \boldsymbol{A}^*|$ invariant gegenüber Zeilenvertauschungen in $\boldsymbol{A}$. Damit ist eine Möglichkeit zur Einführung eines Konditionsmaßes gefunden.

Erstes Konditionsmaß

Die Zahl

$$K_H(\boldsymbol{A}) = \frac{|\det \boldsymbol{A}|}{\alpha_1 \alpha_2 \ldots \alpha_n} \quad \text{mit} \quad \alpha_i = \sqrt{a_{i1}^2 + a_{i2}^2 + \ldots + a_{in}^2},\ i = 1(1)n, \tag{4.17}$$

heißt *Hadamardsches Konditionsmaß* der Matrix $\boldsymbol{A}$. Eine Matrix $\boldsymbol{A}$ heißt schlecht konditioniert, wenn gilt $K_H(\boldsymbol{A}) \ll 1$.

Erfahrungswerte:	$K_H(\boldsymbol{A}) < 0.01$	schlechte Kondition,
	$K_H(\boldsymbol{A}) > 0.1$	gute Kondition,
	$0.01 \leq K_H(\boldsymbol{A}) \leq 0.1$	keine genaue Aussage.

Für Gleichungssysteme, bei denen $K_H = O(10^{-k})$ ist, kann – muss aber nicht – eine Änderung in der k-ten oder früheren sicheren Stelle eines Koeffizienten von $\boldsymbol{A}$ zu Änderungen der Ordung $O(10^k)$ in der Lösung führen (s. dazu [CONT1987], 4.6; [WILK1969], S.116ff., S.133ff., S.143ff.).

Beispiel 4.91. (Fortsetzung von Beispiel 4.89)
Nachträglich wird das Konditionsmaß $K_H(\boldsymbol{A})$ für das System (4.15) bestimmt:

$$K_H(\boldsymbol{A}) = \frac{|\det \boldsymbol{A}|}{\alpha_1 \cdot \alpha_2} = \frac{5 \cdot 10^{-3}}{\sqrt{3.31^2 + 1.2^2}\sqrt{6.9^2 + 2.5^2}} = 0.194 \cdot 10^{-3} < 0.01,$$

das System ist schlecht konditioniert. □

Beispiel 4.92.
Gegeben: Das Gleichungssystem $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}$ mit

$$\boldsymbol{A} = \begin{pmatrix} 1.1 & 1.7 & 1.3 \\ 1.3 & 1.9 & 2.3 \\ 2.1 & 3.1 & 2.9 \end{pmatrix}, \qquad \boldsymbol{a} = \begin{pmatrix} 2.0 \\ 1.6 \\ 2.2 \end{pmatrix}, \qquad \boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}. \tag{4.18}$$

Es besitzt die exakte Lösung $x_1 = -21$, $x_2 = 14$, $x_3 = 1$.

Mit $\det \boldsymbol{A} = 0.72 \cdot 10^{-1}$ folgt für das Konditionsmaß

$$K_H(\boldsymbol{A}) = \frac{|\det \boldsymbol{A}|}{\alpha_1\alpha_2\alpha_3} = \frac{0.72 \cdot 10^{-1}}{\sqrt{5.79 \cdot 10.59 \cdot 22.43}} \approx 0.19 \cdot 10^{-2} < 0.01.$$

Daraus kann man auf eine schlechte Kondition schließen. Es soll nachgeprüft werden, wie sich diese schlechte Kondition auswirkt, wenn man ein Element von $\boldsymbol{A}$ in (4.18) um $1 \cdot 10^{-1}$ abändert; es sei $a_{11} = 1.0$. Dann erhält man die Lösung $\boldsymbol{x}^{\mathsf{T}} = (-6.4615; 4.4872; 0.6410)$. Diese Lösung weicht sehr stark von der Lösung des Systems (4.18) ab. Das ist eine Folge der schlechten Kondition. (Bei stetiger Abänderung des Elementes a_{11} von (4.18) bis zu dem angegebenen Wert ändert sich jedoch auch die Lösung stetig bis zur Lösung des geänderten Systems.) □

Zweites Konditionsmaß

$$\operatorname{cond}(\boldsymbol{A}) = \|\boldsymbol{A}\| \, \|\boldsymbol{A}^{-1}\|.$$

(vgl. Abschnitt 5.2 zur Definition der Matrixnorm)

Drittes Konditionsmaß

$$\mu(\boldsymbol{A}) = \frac{\max\limits_i |\lambda_i|}{\min\limits_i |\lambda_i|},$$

wo λ_i, $i = 1(1)n$, die Eigenwerte der Matrix $\boldsymbol{A}$ sind (s. Kapitel 7).

Im Falle der Konditionsmaße $\text{cond}(\boldsymbol{A})$, $\mu(\boldsymbol{A})$ zeigen große Werte für $\text{cond}(\boldsymbol{A})$ bzw. $\mu(\boldsymbol{A})$ schlechte Kondition an.
Keine der drei genannten Konditionszahlen gibt eine erschöpfende Auskunft über die Kondition einer Matrix.

Eine Reihe anderer Möglichkeiten zur Einführung eines Konditionsmaßes sind in [BERE1971] Bd.2, S.270ff.; [COLL1968], S.81/82; [FADD1979] S.149-159; [ISAA1973], S.39/40; [SPEL1985], S.39/40; [MAES1985], [NIEM1987], 6.1, 6.2 angegeben. Testmatrizen beliebiger Ordnung mit freien Parametern findet man in [ZIEL1974], [ZIEL1975], [ZIEL1986].

Auf schlechte Kondition eines linearen Gleichungssystems kann man auch im Verlaufe seiner Lösung mit Hilfe des Gaußschen Algorithmus schließen, wenn die Elemente $a_{jj}^{(j-1)}$ des gestaffelten Systems (4.7) nacheinander einen Verlust von einer oder mehreren sicheren Stellen erleiden, der z. B. bei der Subtraktion fast gleich großer Zahlen entsteht.

Beispiel 4.93. (Fortsetzung von Beispiel 4.92)

Gegeben: Das lineare Gleichungssystem $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}$ aus Beispiel 4.92.

Gesucht: Seine Lösung mit Hilfe des Gaußschen Algorithmus. Lässt sich eine Aussage über die Kondition des Systems aus dem Rechnungsverlauf machen?

Lösung: Man erhält bei Rechnung mit 3-stelliger Mantisse nach Rechenschema 4.46:

Bezeichnung der Zeilen	$a_{ik}^{(j)}, \widetilde{a}_{ik}^{(j)}$			$a_i^{(j)}, \widetilde{a}_i^{(j)}$	erfolgte Operationen
$\widetilde{1}^{(0)}$	1.1	1.7	1.3	2.0	
$\widetilde{2}^{(0)}$	1.3	1.9	2.3	1.6	—
$\widetilde{3}^{(0)}$	2.1	3.1	2.9	2.2	
$1^{(0)}$	2.1	3.1	2.9	2.2	
$2^{(0)}$	1.3	1.9	2.3	1.6	—
$3^{(0)}$	1.1	1.7	1.3	2.0	
$\widetilde{2}^{(1)}$	0	−0.0200	0.500	0.240	$-\frac{1.3}{2.1} \cdot 1^{(0)} + 2^{(0)}$
$\widetilde{3}^{(1)}$	0	0.0800	−0.220	0.850	$-\frac{1.1}{2.1} \cdot 1^{(0)} + 3^{(0)}$
$2^{(1)}$	0	0.0800	−0.220	0.850	
$3^{(1)}$	0	−0.0200	0.500	0.240	—
$3^{(2)}$	0	0	0.445	0.453	$-\frac{0.0200}{0.0800} \cdot 2^{(1)} + 3^{(1)}$

Man erhält also das gestaffelte System $\boldsymbol{R}\boldsymbol{x} = \boldsymbol{r}$, aus dem die Näherungslösung $\boldsymbol{x}^{(0)}$ für $\boldsymbol{x}$ folgt

$$\boldsymbol{R} = \begin{pmatrix} 2.10 & 3.10 & 2.900 \\ 0.00 & 0.0800 & -0.220 \\ 0.00 & 0.00 & 0.445 \end{pmatrix}, \ \boldsymbol{r} = \begin{pmatrix} 2.20 \\ 0.850 \\ 0.453 \end{pmatrix} \quad \Rightarrow \quad \boldsymbol{x}^{(0)} = \begin{pmatrix} -20.1 \\ 13.4 \\ 1.02 \end{pmatrix}.$$

Man erkennt aus dem Rechnungsverlauf, dass die gegebene Matrix schlecht konditioniert ist; denn die Diagonalelemente des gestaffelten Systems haben zwei bzw. eine sichere Ziffer verloren. □

Zusammenfassend lässt sich sagen, dass ein System $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}$ mit $\boldsymbol{A} = (a_{ik})$ schlecht konditioniert ist, wenn eine der folgenden Aussagen für das System zutrifft:

1. $K_H(\boldsymbol{A}) < 0.01$;
2. $\mathrm{cond}(\boldsymbol{A}) \gg 1$;
3. $\mu(\boldsymbol{A}) \gg 1$;
4. kleine Änderungen der Koeffizienten a_{ik} bewirken große Änderungen der Lösung;
5. die Koeffizienten $a_{jj}^{(j-1)}$ des nach dem Gaußschen Algorithmus erhaltenen gestaffelten Systems verlieren nacheinander eine oder mehrere sichere Stellen;
6. die Elemente der Inversen $\boldsymbol{A}^{-1}$ von $\boldsymbol{A}$ sind groß im Vergleich zu den Elementen von $\boldsymbol{A}$ selbst;
7. langsame Konvergenz der Nachiteration (vgl. Abschnitt 4.14.4).

4.14.2 Konditionsschätzung

Von den vorgenannten Konditionsmaßen ist nur das Hadamardsche Konditionsmaß $K_H(\boldsymbol{A})$ mit vertretbarem Aufwand zu berechnen, sofern eine LR-Zerlegung von $\boldsymbol{A}$ vorliegt. Mit der LR-Zerlegung lässt sich die Determinante von $\boldsymbol{A}$ leicht berechnen, ebenso ist der Aufwand zur Berechnung der α_i relativ gering. Insgesamt ist die Anzahl der erforderlichen Punktoperationen $O(n^2)$.

Der Aufwand zur Berechnung der Konditionszahl $\mathrm{cond}(\boldsymbol{A}) = \|\boldsymbol{A}\| \, \|\boldsymbol{A}^{-1}\|$ ist wegen der erforderlichen Berechnung von $\boldsymbol{A}^{-1}$ nicht gerechtfertigt; deshalb gibt man sich hier mit Schätzungen zufrieden.

Konditionsschätzung nach Forsythe/Moler

In [FORS1971], 13. ist eine Möglichkeit zur Berechnung der heuristischen Schätzung der Konditionszahl $\mathrm{cond}(\boldsymbol{A}) = \|\boldsymbol{A}\| \, \|\boldsymbol{A}^{-1}\|$ angegeben, die recht brauchbare Ergebnisse liefert, wie eine Reihe von Tests ergeben haben. Es gilt

$$\mathrm{cond}(\boldsymbol{A}) \approx \frac{\|\boldsymbol{z}^{(1)}\|}{\|\boldsymbol{x}^{(0)}\|} \cdot \frac{1}{\mathrm{EPS}}$$

wobei $\boldsymbol{z}^{(1)}$ der Korrekturvektor aus der ersten Nachiteration (vgl. Abschnitt 4.14.4) ist, $\boldsymbol{x}^{(0)}$ die Lösung von $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}$ mit dem Gauß-Algorithmus und EPS die Maschinengenauigkeit (Def. 1.13).

Algorithmus 4.94. (*Konditionsschätzung nach Forsythe/Moler*)

Gegeben: (i) $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}$, (n,n)-Matrix $\boldsymbol{A}$, $\det(\boldsymbol{A}) \neq 0$
(ii) $\boldsymbol{x}^{(0)}$ = Lösung des Systems (i) mit dem Gauß-Algorithmus
(iii) die Maschinengenauigkeit EPS

Gesucht: Schätzung der Konditionszahl $\text{cond}(\boldsymbol{A}) = \|\boldsymbol{A}\|\,\|\boldsymbol{A}^{-1}\|$

Es sind folgende Schritte durchzuführen:

1. Berechnung des Fehlervektors $\boldsymbol{r}^{(0)}$ mit doppelter Genauigkeit
$$\boldsymbol{r}^{(0)} := \boldsymbol{a} - \boldsymbol{A}\boldsymbol{x}^{(0)}$$
2. Lösung des Gleichungssystems $\boldsymbol{A}\boldsymbol{z}^{(1)} = \boldsymbol{r}^{(0)}$ unter Verwendung der Dreieckszerlegung von $\boldsymbol{A}$ mit dem Gauß-Algorithmus
3. Berechnung des Schätzwertes
$$\text{cond}(\boldsymbol{A}) \approx \frac{\|\boldsymbol{z}^{(1)}\|}{\|\boldsymbol{x}^{(0)}\|} \cdot \frac{1}{\text{EPS}}$$

Konditionsschätzung nach Cline (und anderen) [CLIN1979]

Der folgende Algorithmus geht auf eine Arbeit von Cline, Moler, Stewart und Wilkinson [CLIN1979] zurück, wurde in modifizierter Form in LINPACK eingesetzt und ist in [KIEL1988], 5.4 sehr gut beschrieben und bewiesen.

Der Algorithmus beruht auf der für $\boldsymbol{A}\boldsymbol{z} = \boldsymbol{x}$ mit $\boldsymbol{x} \neq \boldsymbol{0}$ folgenden Ungleichung

(1) $$\|\boldsymbol{A}^{-1}\|_\infty \geq \frac{\|\boldsymbol{z}\|_\infty}{\|\boldsymbol{x}\|_\infty}$$

bzw.

(2) $$\|\boldsymbol{A}^{-1}\|_\infty = \|(\boldsymbol{A}^{-1})^\mathsf{T}\|_1 = \|(\boldsymbol{L}\boldsymbol{R}^{-1})^\mathsf{T}\|_1 \geq \frac{\|\boldsymbol{z}\|_1}{\|\boldsymbol{x}\|_1}$$

für $(\boldsymbol{L}\boldsymbol{R})^\mathsf{T}\boldsymbol{z} = \boldsymbol{x}$ bei $\boldsymbol{x} \neq \boldsymbol{0}$. In beiden Darstellungen muss $\|\boldsymbol{z}\|$ so vergrößert werden, dass $\|\boldsymbol{z}\|/\|\boldsymbol{x}\|$ eine gute Näherung für $\|\boldsymbol{A}^{-1}\|$ ist. Im zweiten Fall sind wegen $(\boldsymbol{L}\boldsymbol{R})^\mathsf{T}\boldsymbol{z} = \boldsymbol{R}^\mathsf{T}\boldsymbol{L}^\mathsf{T}\boldsymbol{z} = \boldsymbol{x}$ die beiden Dreiecksysteme

$$\begin{aligned} \boldsymbol{R}^\mathsf{T}\boldsymbol{y} &= \boldsymbol{x} \\ \boldsymbol{L}^\mathsf{T}\boldsymbol{z} &= \boldsymbol{y} \end{aligned}$$

zu lösen. Beim Einsatz von Spaltenpivotsuche ist es sinnvoll, $\boldsymbol{x}$ und $\boldsymbol{y}$ so zu ermitteln, dass $\|\boldsymbol{y}\|_1/\|\boldsymbol{x}\|_1$ maximiert wird und anschließend $\boldsymbol{L}^\mathsf{T}\boldsymbol{z} = \boldsymbol{y}$ gelöst und damit $\|\boldsymbol{z}\|_1/\|\boldsymbol{x}\|_1$ berechnet; dieser Fall wird hier beschrieben. Der Beweis sowie weitere Fallunterscheidungen sind in [KIEL1988], 5.4 zu finden.

Algorithmus 4.95. (*Konditionsschätzung nach Cline u.a.*)

Gegeben: LR-Zerlegung von $\boldsymbol{A}$ bzw. $\boldsymbol{PA}$ mit normierter unterer Dreiecksmatrix $\boldsymbol{L} = (l_{ik})$ und regulärer oberer Dreiecksmatrix $\boldsymbol{R} = (r_{ik})$, $i, k = 1(1)n$.

Gesucht: Schätzwert für $\text{cond}_\infty(\boldsymbol{A}) = \|\boldsymbol{A}\|_\infty \|\boldsymbol{A}^{-1}\|_\infty$

1. Für $\boldsymbol{R}^\mathsf{T}$ werden $\boldsymbol{x} = (x_i)$ mit $x_i = \pm 1$ und $\boldsymbol{y} = (y_i) = (\boldsymbol{R}^\mathsf{T})^{-1}\boldsymbol{x}$, $i = 1(1)n$, so bestimmt, dass $\|\boldsymbol{y}\|_1 = \sum_{i=1}^{n} |y_i|$ möglichst groß wird. Dazu werden folgende Schritte durchgeführt:

   1.1 Setze $x_1 := 1$, $y_1 := 1/r_{11}$ und $y_i = -r_{1i} \cdot y_1 / r_{ii}$ für $i = 2(1)n$

   1.2 Für jedes $k = 2(1)n$

   1.2.1 $v := \frac{1}{r_{kk}}$

   1.2.2 $x_k := y_k - v, \quad y_k := y_k + v$

   1.2.3 $\text{SMI} := |x_k|, \quad \text{SPL} := |y_k|$

   1.2.4 Für jedes $i = k{+}1(1)n$

   $v := r_{ki}/r_{ii}$

   $x_i := y_i - v \cdot x_k, \quad y_i := y_i - v \cdot y_k$

   $\text{SMI} := \text{SMI} + |x_i|, \quad \text{SPL} := \text{SPL} + |y_i|$

   1.2.5 Falls SMI > SPL ist, wird gesetzt

   $y_i := x_i$ für $i = k(1)n$

   $x_k := -1$

   andernfalls

   $x_k := 1$

2. Durch Rückwärtselimination wird die Lösung $\boldsymbol{z}$ des Dreieckssystems $\boldsymbol{L}^\mathsf{T}\boldsymbol{z} = \boldsymbol{y}$ berechnet.

3. Die Zahl $K_\infty := \|\boldsymbol{z}\|_1 / \|\boldsymbol{x}\|_1$ wird in [KIEL1988] und [NIEM1987] als Näherungswert für $\|\boldsymbol{A}^{-1}\|_\infty$ verwendet. (Wegen (1) und (2) lässt sich auch $K_\infty := \|\boldsymbol{z}\|_\infty / \|\boldsymbol{x}\|_\infty$ verwenden; damit erhält man einen besseren Schätzwert für $\text{cond}_\infty(\boldsymbol{A})$, wie umfangreiche Tests gezeigt haben.)

4. Der Schätzwert
$$\text{cond}_\infty(\boldsymbol{A}) \approx \|\boldsymbol{A}\|_\infty \cdot K_\infty$$
wird berechnet.

Beispiel 4.96.

Gegeben: Das sehr schlecht konditionierte Gleichungssystem

$$\begin{pmatrix} 1.985 & -1.358 \\ 0.953 & -0.652 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 2.212 \\ 1.062 \end{pmatrix}$$

hat auf 5 Stellen gerundet die exakte Lösung

$$\boldsymbol{x}_{ex} = \begin{pmatrix} 0.60870 \\ -0.73913 \end{pmatrix}$$

Gesucht: Die Konditionszahl, nachdem für das voranstehende Gleichungssystem die Lösung mit dem Gauß-Algorithmus berechnet wurde.

Lösung: Aussagen zur Konditionszahl der Matrix $\boldsymbol{A}$ nach den Verfahren:

(1) Ermittlung des Hadamardschen Konditionsmaßes
(2) Konditionsschätzung nach Forsythe/Moler
(3) Konditionsschätzung nach Cline (und anderen)

(1) *Ermittlung des Hadamardschen Konditionsmaßes* (gemäß (4.17)).

$$\boldsymbol{A} = \begin{pmatrix} 1.985 & -1.358 \\ 0.953 & -0.652 \end{pmatrix} \text{ mit } \det \boldsymbol{A} = 1.985 \cdot (-0.652) - 0.953 \cdot (-1.358) = -4.6 \cdot 10^{-5}$$

$$\begin{aligned} K_H(\boldsymbol{A}) = \frac{|\det A|}{\alpha_1 \alpha_2} &= \frac{|-4.6 \cdot 10^{-5}|}{\sqrt{1.985^2 + (-1.358)^2} \cdot \sqrt{0.953^2 + (-0.652^2)}} \\ &= \frac{4.6 \cdot 10^{-5}}{\sqrt{5.784} \cdot \sqrt{1.333}} = \frac{4.6 \cdot 10^{-5}}{2.777} = 1.656 \cdot 10^{-5} \quad < \quad 0.01 \end{aligned}$$

$\Rightarrow$ schlechte Kondition

(2) *Konditionsschätzung nach Forsythe/Moler.*

$$\text{cond}(\boldsymbol{A}) \approx \frac{||\boldsymbol{z}^{(1)}||}{||\boldsymbol{x}^{(0)}||} \cdot \frac{1}{\varepsilon}$$

$$\begin{array}{cc} \boldsymbol{A} & \boldsymbol{x} = \boldsymbol{a} \\ \begin{pmatrix} 1.985 & -1.358 \\ 0.953 & -0.652 \end{pmatrix} & \boldsymbol{x} = \begin{pmatrix} 2.212 \\ 1.062 \end{pmatrix} \end{array}$$

$$\boldsymbol{x}_{ex} = \begin{pmatrix} 0.60870 \\ -0.73913 \end{pmatrix}; \qquad \boldsymbol{x}^{(0)} = \begin{pmatrix} 0.60868842 \\ -0.73913682 \end{pmatrix}$$

$$\begin{aligned} \boldsymbol{r}^{(0)} &= \boldsymbol{a} - \boldsymbol{A}\boldsymbol{x}^{(0)} \\ &= \begin{pmatrix} 2.212 \\ 1.062 \end{pmatrix} - \begin{pmatrix} 2.211994315 \\ 1.061997271 \end{pmatrix} = \begin{pmatrix} 0.00000568474 \\ 0.0000027291 \end{pmatrix} \end{aligned}$$

$$\boldsymbol{A}\boldsymbol{z}^{(1)} = \boldsymbol{r}^{(0)} \quad \Rightarrow \quad \boldsymbol{z}^{(1)} = \begin{pmatrix} 0.000013689 \\ 0.000016086 \end{pmatrix}$$

$$\text{cond}(\boldsymbol{A}) = \frac{0.000021239}{0.957509703} \cdot \frac{1}{\varepsilon} = \frac{2.218 \cdot 10^{-5}}{\varepsilon} > 1 \quad \Leftrightarrow \quad \varepsilon < 2.218 \cdot 10^{-5}$$

Da Maschinengenauigkeit ε im Allgemeinen $\ll 2 \cdot 10^{-5}$ $\Rightarrow$ schlechte Kondition

(3) *Konditionsschätzung nach Cline (und anderen).*

Bei dieser Schätzung von cond($\boldsymbol{A}$) wird vorausgesetzt, dass eine ohne oder mit Spaltenpivotsuche erzeugte LR-Zerlegung von $\boldsymbol{A}$ bzw. $\boldsymbol{PA}$ vorliegt mit normierter unterer Dreiecksmatrix $\boldsymbol{L}$ und regulärer oberer Dreiecksmatrix $\boldsymbol{R}$. Aus den Faktoren $\boldsymbol{L}$ und $\boldsymbol{R}$ wird ein Schätzwert für $\|(\boldsymbol{LR})^{-1}\|_\infty$ berechnet und damit wegen $\|\boldsymbol{A}^{-1}\| = \|(\boldsymbol{LR})^{-1}\|$ (Invarianz der Matrixnormen 1,2 und ∞ gegenüber Zeilen- und Spaltenvertauschungen) einen Schätzwert für cond($\boldsymbol{A}$).

$$\begin{array}{ccccc} \boldsymbol{A} & = & \boldsymbol{L} & & \boldsymbol{R} \\ \begin{pmatrix} 1.985 & -1.358 \\ 0.953 & -0.652 \end{pmatrix} & = & \begin{pmatrix} 1 & 0 \\ 0.4801 & 1 \end{pmatrix} & \begin{pmatrix} 1.985 & -1.358 \\ 0 & -2.3173 \cdot 10^{-5} \end{pmatrix} \end{array}$$

Mit Algorithmus 4.95 erhält man:

1.1 $x_1 := 1\,;$ $\qquad y_1 := \dfrac{1}{1.985} = 0.5038$

$$y_2 := 1.358 \cdot \frac{1}{1.985 \cdot (-2.317 \cdot 10^{-5})} = -29527$$

1.2.1 $$v := \frac{1}{-2.317 \cdot 10^{-5}} = -43159$$

1.2.2 $x_2 := y_2 - v = 13632\,;$ $\qquad y_2 := y_2 + v = -72686$

1.2.3 $\text{SMI} := |x_2| = 13632\,;$ $\qquad \text{SPL} := |y_2| = 72686$

1.2.4 $\text{SMI} < \text{SPL}$ $\qquad x_2 := 1$

$$\begin{array}{ccccc} \boldsymbol{L}^{\mathsf{T}} & & \boldsymbol{z} & = & \boldsymbol{y} \\ \begin{pmatrix} 1 & 0.4801 \\ 0 & 1 \end{pmatrix} & \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} & = & \begin{pmatrix} 0.5038 \\ -72686 \end{pmatrix} \end{array}$$

$\Rightarrow \quad z_2 = -72686; \quad z_1 = 34890$

$$K_\infty := \frac{||\boldsymbol{z}||_1}{||\boldsymbol{x}||_1} = \frac{\sum\limits_{i=1}^{n} |z_i|}{\sum\limits_{i=1}^{n} |x_i|} = \frac{107576}{2} = 53788$$

$$||\boldsymbol{A}||_\infty := \text{Zeilensummennorm} = \max\left\{ \sum_{k=1}^{n} |a_{ik}| \,\middle|\, i = 1, \ldots, n \right\} = \max\left\{3.343, 1.605\right\}$$

$$\text{cond}_\infty(\boldsymbol{A}) \approx ||\boldsymbol{A}||_\infty \cdot K_\infty \quad = \quad 3.343 \cdot 53788 \quad \gg \quad 1$$

$\Rightarrow$ schlechte Kondition $\square$

4.14.3 Möglichkeiten zur Konditionsverbesserung

(a) *Äquilibrierung* (s. [WERN1993], S.160): Man multipliziert die Zeilen von $\boldsymbol{A}$ mit einem konstanten Faktor, d. h. man geht vom gegebenen System $\boldsymbol{Ax} = \boldsymbol{a}$ zu

$$\boldsymbol{D}_1\boldsymbol{Ax} = \boldsymbol{D}_1\boldsymbol{a},$$

über, wobei $\boldsymbol{D}_1$ eine nicht singuläre Diagonalmatrix darstellt. Nach Ergebnissen von Wilkinson erhält man im Allgemeinen dann optimale Konditionszahlen, wenn man so multipliziert, dass alle Zeilenvektoren der Matrix $\boldsymbol{A}$ gleiche Norm haben.

(b) *Skalierung* (s. [BJOR1979], 5; [FORS1971], 11): Man multipliziert die k-te Spalte von $\boldsymbol{A}$ mit einem konstanten Faktor. Physikalisch bedeutet dies die Änderung des Maßstabs für die Unbekannte x_k. Das gleiche kann man für die rechte Seite machen. Auf alle Spalten bezogen ergibt sich statt $\boldsymbol{Ax} = \boldsymbol{a}$ das System

$$\boldsymbol{AD}_2\boldsymbol{x} = \boldsymbol{D}_2\boldsymbol{a}.$$

(c) Auch Linearkombination von Gleichungen kann zur Konditionsverbesserung führen. Die Kondition kann dadurch allerdings auch verschlechtert werden (s. Beispiel dazu in [POLO1964], S.345/346).

4.14.4 Nachiteration

Wenn die Koeffizienten a_{ik} eines linearen Gleichungssystems $\boldsymbol{Ax} = \boldsymbol{a}$ mit $\boldsymbol{A} = (a_{ik})$ *exakt* gegeben sind, das System aber schlecht konditioniert ist, kann eine mit Rundungsfehlern behaftete Näherungslösung, die mittels einer direkten Methode bestimmt wurde, iterativ verbessert werden. Sei $\boldsymbol{x}^{(0)}$ die mit Hilfe des Gaußschen Algorithmus gewonnene Näherungslösung des Systems $\boldsymbol{Ax} = \boldsymbol{a}$, dann ist durch (4.16) das Residuum (Fehlervektor) $\boldsymbol{r}^{(0)}$ definiert (vgl. Abschnitt 4.14.1). Mit einem Korrekturvektor $\boldsymbol{z}^{(1)}$ macht man den Ansatz $\boldsymbol{x} = \boldsymbol{x}^{(0)} + \boldsymbol{z}^{(1)}$ und erhält

$$\boldsymbol{Az}^{(1)} = \boldsymbol{a} - \boldsymbol{Ax}^{(0)} = \boldsymbol{r}^{(0)}. \tag{4.19}$$

Da $\boldsymbol{a}$, $\boldsymbol{A}$ und $\boldsymbol{x}^{(0)}$ bekannt sind, lässt sich das Residuum $\boldsymbol{r}^{(0)}$ berechnen. Zur Berechnung von $\boldsymbol{x}^{(0)}$ ist das System $\boldsymbol{Ax} = \boldsymbol{a}$ mit Hilfe des Gaußschen Algorithmus bereits auf obere Halbdiagonalform gebracht worden, so dass sich $\boldsymbol{z}^{(1)}$ aus (4.18) rasch bestimmen lässt; man muss nur noch die rechte Seite transformieren. Statt der exakten Lösung $\boldsymbol{z}^{(1)}$ von (4.19) erhält man nur eine Näherungslösung, so dass sich aus $\boldsymbol{x}^{(0)} + \boldsymbol{z}^{(1)}$ statt des exakten $\boldsymbol{x}$ ein Näherungswert

$$\boldsymbol{x}^{(1)} = \boldsymbol{x}^{(0)} + \boldsymbol{z}^{(1)}$$

errechnet. Für $\boldsymbol{x}^{(1)}$ ergibt sich dann der Fehlervektor $\boldsymbol{r}^{(1)} = \boldsymbol{a} - \boldsymbol{Ax}^{(1)}$, so dass sich der soeben beschriebene Prozess wiederholen lässt.

Die allgemeine Vorschrift zur Berechnung eines $(\nu+1)$-ten Korrekturvektors $\boldsymbol{z}^{(\nu+1)}$ lautet

$$\boldsymbol{Az}^{(\nu+1)} = \boldsymbol{a} - \boldsymbol{Ax}^{(\nu)} = \boldsymbol{r}^{(\nu)}, \quad \nu = 0, 1, 2, \ldots$$

Es wird so lange gerechnet, bis sich für ein ν die Komponenten aufeinander folgender Korrekturvektoren in der gewünschten Stellenzahl nicht mehr ändern bzw. zu vorgegebenem $\varepsilon > 0$ die *relative Verbesserung* $\|\boldsymbol{z}^{(\nu+1)}\|_\infty \;/\; \|\boldsymbol{x}^{(\nu)}\|_\infty < \varepsilon$ ist. Dann gilt für die gesuchte Lösung

$$\boldsymbol{x} \approx \boldsymbol{x}^{(\nu+1)} = \boldsymbol{x}^{(\nu)} + \boldsymbol{z}^{(\nu+1)}.$$

Es empfiehlt sich, die Residuen mit doppelter Stellenzahl zu berechnen und jeweils erst das Ergebnis auf die einfache Stellenzahl zu runden.

Algorithmus 4.97. (*Nachiteration*)

Gegeben: $\boldsymbol{A}$, $\boldsymbol{a}$, Näherungslösung $\boldsymbol{x}^{(0)}$ von $\boldsymbol{Ax} = \boldsymbol{a}$.

Gesucht: Verbesserte Lösung $\boldsymbol{x}^{(\nu+1)} \approx \boldsymbol{x}$.

Dann sind für jedes $\nu = 0, 1, 2, \ldots$ nacheinander folgende Schritte auszuführen:

1. $\boldsymbol{r}^{(\nu)} = \boldsymbol{a} - \boldsymbol{A}\boldsymbol{x}^{(\nu)}$,
2. $\boldsymbol{A}\boldsymbol{z}^{(\nu+1)} = \boldsymbol{r}^{(\nu)}$,
3. $\boldsymbol{x}^{(\nu+1)} = \boldsymbol{x}^{(\nu)} + \boldsymbol{z}^{(\nu+1)}$.

Da eine Dreieckszerlegung $\boldsymbol{A} = \boldsymbol{LR}$ (bzw. $\boldsymbol{PA} = \boldsymbol{LR}$ bei Spaltenpivotsuche) bereits mit dem direkten Verfahren durchgeführt wurde, ist bei 2. nur noch auszuführen:

2.1 (Vorwärtselimination) $\boldsymbol{r}^{(\nu)} = \boldsymbol{Lr} \Rightarrow \boldsymbol{r}$
(bzw. $\boldsymbol{Pr}^{(\nu)} = \boldsymbol{Lr}$ bei Spaltenpivotsuche)

2.2 (Rückwärtselimination) $\boldsymbol{Rz}^{(\nu+1)} = \boldsymbol{r} \Rightarrow \boldsymbol{z}^{(\nu+1)}$

Die Rechnung wird abgebrochen, wenn die relative Verbesserung kleiner als eine vorzugebende Schranke ε ist: $\frac{||\boldsymbol{z}^{(\nu+1)}||_\infty}{||\boldsymbol{z}^{(\nu)}||_\infty} < \varepsilon$.

Eine hinreichende Konvergenzbedingung für die Nachiteration ist zwar bekannt ([WILK1969], S.155), jedoch für die Praxis zu aufwändig. Die Konvergenz ist umso schlechter, je schlechter die Kondition des Systems ist (vgl. auch [MCCA1967], 5.8; [STIE1976], S.24/25; [ZURM1965], S.163).

Beispiel 4.98. (Fortsetzung von Beispiel 4.92)

Gegeben: Das Gleichungssystem $\boldsymbol{Ax} = \boldsymbol{a}$ mit (4.18) und seine mit Hilfe des Gaußschen Algorithmus gewonnene Näherungslösung $\boldsymbol{x}^{(0)}$.

Gesucht: Eine Verbesserung von $\boldsymbol{x}^{(0)}$ durch einen Nachiterationsschritt.

Lösung:

Mit $\boldsymbol{x}^{(0)\mathsf{T}} = (-20.1, 13.4, 1.02)$ folgt bei Rechnung mit 6-stelliger und Rundung auf 3-stellige Mantisse für das Residuum $\boldsymbol{r}^{(0)}$ gemäß (4.15)

$$\boldsymbol{r}^{(0)} = \boldsymbol{a} - \boldsymbol{A}\boldsymbol{x}^{(0)} = \begin{pmatrix} 0.00400 \\ -0.0760 \\ -0.0880 \end{pmatrix}, \qquad |\boldsymbol{r}^{(0)}| = 0.0010.$$

Aus $\boldsymbol{A}\boldsymbol{z}^{(1)} = \boldsymbol{r}^{(0)}$ folgt mit Hilfe des Gaußschen Algorithmus für den Korrekturvektor

$$\boldsymbol{z}^{(1)} = \begin{pmatrix} -0.857 \\ 0.571 \\ -0.0202 \end{pmatrix}.$$

Daraus ergibt sich eine gegenüber $\boldsymbol{x}^{(0)}$ verbesserte Näherung $\boldsymbol{x}^{(1)}$ für $\boldsymbol{x}$ mit

$$\boldsymbol{x}^{(1)} = \boldsymbol{x}^{(0)} + \boldsymbol{z}^{(1)} = \begin{pmatrix} -20.957 \\ 13.971 \\ 0.9998 \end{pmatrix};$$

d. h. nach Rundung auf die 3-stellige Mantisse erhält man für $\boldsymbol{x}^{(1)}$ die exakte Lösung $(-21, 14, 1)^{\mathsf{T}}$. □

4.15 Gleichungssysteme mit Blockmatrix

4.15.1 Vorbemerkungen

Es liege ein Gleichungssystem von n Gleichungen mit n Unbekannten der Form von (4.2) vor:

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}$$

vor. Eine Zerlegung der (n, n)-Matrix $\boldsymbol{A} = (a_{ik})$ in *Blöcke* (*Untermatrizen*) geschieht durch horizontale und vertikale Trennungslinien, die die ganze Matrix durchschneiden. Man erhält eine sogenannte *Blockmatrix*, die aus Untermatrizen $\boldsymbol{A}_{ik}$ kleinerer Ordnung aufgebaut ist: $\boldsymbol{A} = (\boldsymbol{A}_{ik})$.

Zerlegt man nun die quadratische Matrix $\boldsymbol{A}$ so, dass die *Diagonalblöcke* $\boldsymbol{A}_{ii}$ quadratische (n_i, n_i)-Matrizen sind und die Blöcke $\boldsymbol{A}_{ik}$ Matrizen mit n_i Zeilen und n_k Spalten, so erhält man bei entsprechender Zerlegung der Vektoren $\boldsymbol{x}$ und $\boldsymbol{a}$ das System $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}$ in der Form

$$\begin{pmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} & \cdots & \boldsymbol{A}_{1N} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} & \cdots & \boldsymbol{A}_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{A}_{N1} & \boldsymbol{A}_{N2} & \cdots & \boldsymbol{A}_{NN} \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_N \end{pmatrix} = \begin{pmatrix} \boldsymbol{a}_1 \\ \boldsymbol{a}_2 \\ \vdots \\ \boldsymbol{a}_N \end{pmatrix}. \tag{4.20}$$

Es gilt $\sum_{i=1}^{N} n_i = n$, $\sum_{k=1}^{N} \boldsymbol{A}_{ik}\boldsymbol{x}_k = \boldsymbol{a}_i$, $i = 1(1)N$.

Es werden nur solche Zerlegungen betrachtet, deren Diagonalblöcke quadratisch sind, weil man mit ihnen so operieren kann, als wären die Blöcke Zahlen. Man kann deshalb zur Lösung von Gleichungssystemen (4.20) mit Blockmatrix im Wesentlichen die bisher behandelten Methoden verwenden, nur rechnet man jetzt mit Matrizen und Vektoren statt mit Zahlen. Divisionen durch Matrixelemente sind jetzt durch Multiplikationen mit deren Inversen zu ersetzen. Die Pivotsuche kann nicht angewendet werden.

Beispiel 4.99.

$$\overbrace{\underbrace{\left(\begin{array}{ccc|cc} 1 & 6 & -1 & -7 & 5 \\ 2 & 7 & -2 & -10 & 6 \\ 3 & 8 & -3 & 2 & 7 \\ \hline 4 & 9 & -4 & 3 & 8 \\ 5 & 10 & -5 & 4 & 9 \end{array}\right)}_{\boldsymbol{A}_{21}} \underbrace{\left(\begin{array}{cc} -7 & 5 \\ -10 & 6 \\ 2 & 7 \\ \hline 3 & 8 \\ 4 & 9 \end{array}\right)}_{\boldsymbol{A}_{22}}}^{\boldsymbol{A}_{11}} \overbrace{\underbrace{\left(\begin{array}{c} x_1 \\ x_2 \\ x_3 \\ \hline x_4 \\ x_5 \end{array}\right)}_{\boldsymbol{x}_2}}^{\boldsymbol{x}_1} = \left(\begin{array}{c} 1 \\ 1 \\ 1 \\ \hline 1 \\ 1 \end{array}\right) \begin{array}{l} \Big\} \boldsymbol{a}_1 \\ \Big\} \boldsymbol{a}_2 \end{array}$$

$$\Rightarrow \begin{pmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{pmatrix} \cdot \begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{pmatrix} = \begin{pmatrix} \boldsymbol{a}_1 \\ \boldsymbol{a}_2 \end{pmatrix}.$$

□

Hier wird zur Veranschaulichung der Vorgehensweise bei Blockmethoden der Gaußsche Algorithmus für vollbesetzte Blocksysteme und für blockweise tridiagonale Systeme angegeben. Anschließend werden Methoden zur Behandlung spezieller Blocksysteme mit entsprechenden Literaturangaben genannt.

4.15.2 Gauß-Algorithmus für Blocksysteme

1. Eliminationsschritt

Formal verläuft die Elimination analog zu Abschnitt 4.5 ohne Pivotisierung. Die Division durch das Diagonalelement wird hier ersetzt durch die Multiplikation mit der Inversen $\left(\boldsymbol{A}_{jj}^{(j-1)}\right)^{-1}$. Multiplikation von

$$1^{(0)}: \qquad \boldsymbol{A}_{11}^{(0)}\boldsymbol{x}_1 + \boldsymbol{A}_{12}^{(0)}\boldsymbol{x}_2 + \ldots + \boldsymbol{A}_{1N}^{(0)}\boldsymbol{x}_N = \boldsymbol{a}_1^{(0)}$$

mit $-\boldsymbol{A}_{i1}^{(0)}(\boldsymbol{A}_{11}^{(0)})^{-1}$ von links und Addition zur i-ten Zeile (nacheinander für $i = 2, 3, \ldots, N$) liefert das System

$$\begin{array}{lllll}
1^{(0)}: & \boldsymbol{A}_{11}^{(0)}\boldsymbol{x}_1+ & \boldsymbol{A}_{12}^{(0)}\boldsymbol{x}_2 & +\ldots+ & \boldsymbol{A}_{1N}^{(0)}\boldsymbol{x}_N = \quad \boldsymbol{a}_1^{(0)},\\
2^{(1)}: & & \boldsymbol{A}_{22}^{(1)}\boldsymbol{x}_2 & +\ldots+ & \boldsymbol{A}_{2N}^{(1)}\boldsymbol{x}_N = \quad \boldsymbol{a}_2^{(1)},\\
3^{(1)}: & & \boldsymbol{A}_{32}^{(1)}\boldsymbol{x}_2 & +\ldots+ & \boldsymbol{A}_{3N}^{(1)}\boldsymbol{x}_N = \quad \boldsymbol{a}_3^{(1)},\\
\vdots & & \vdots & & \vdots \qquad\quad \vdots\\
N^{(1)}: & & \boldsymbol{A}_{N2}^{(1)}\boldsymbol{x}_2 & +\ldots+ & \boldsymbol{A}_{NN}^{(1)}\boldsymbol{x}_N = \quad \boldsymbol{a}_N^{(1)}.
\end{array}$$

2. Eliminationsschritt

Multiplikation von $2^{(1)}$ mit $-\boldsymbol{A}_{i2}^{(1)}\left(\boldsymbol{A}_{22}^{(1)}\right)^{-1}$ von links und Addition zur i-ten Zeile nacheinander für $i = 3, 4, \ldots, N$ liefert das System

$$\begin{array}{llllll}
1^{(0)}: & \boldsymbol{A}_{11}^{(0)}\boldsymbol{x}_1+ & \boldsymbol{A}_{12}^{(0)}\boldsymbol{x}_2+ & \boldsymbol{A}_{13}^{(0)}\boldsymbol{x}_3 & +\ldots+ & \boldsymbol{A}_{1N}^{(0)}\boldsymbol{x}_N = \quad \boldsymbol{a}_1^{(0)},\\
2^{(1)}: & & \boldsymbol{A}_{22}^{(1)}\boldsymbol{x}_3+ & \boldsymbol{A}_{23}^{(1)}\boldsymbol{x}_3 & +\ldots+ & \boldsymbol{A}_{2N}^{(1)}\boldsymbol{x}_N = \quad \boldsymbol{a}_2^{(1)},\\
3^{(2)}: & & & \boldsymbol{A}_{33}^{(2)}\boldsymbol{x}_3 & +\ldots+ & \boldsymbol{A}_{3N}^{(2)}\boldsymbol{x}_N = \quad \boldsymbol{a}_3^{(2)},\\
\vdots & & & \vdots & & \vdots \qquad\quad \vdots\\
N^{(2)}: & & & \boldsymbol{A}_{N3}^{(2)}\boldsymbol{x}_3 & +\ldots+ & \boldsymbol{A}_{NN}^{(2)}\boldsymbol{x}_N = \quad \boldsymbol{a}_N^{(2)}.
\end{array}$$

Nach $N-1$ analogen Eliminationsschritten erhält man das blockweise gestaffelte System $\boldsymbol{B}\boldsymbol{x} = \boldsymbol{b}$, wobei $\boldsymbol{B}$ eine Block-Superdiagonalmatrix ist, der Form

$$\begin{array}{llllll}
1^{(0)}: & \boldsymbol{A}_{11}^{(0)}\boldsymbol{x}_1+ & \boldsymbol{A}_{12}^{(0)}\boldsymbol{x}_2+ & \ldots+ & \boldsymbol{A}_{1N}^{(0)}\boldsymbol{x}_N = \quad \boldsymbol{a}_1^{(0)},\\
2^{(1)}: & & \boldsymbol{A}_{22}^{(1)}\boldsymbol{x}_2+ & \ldots+ & \boldsymbol{A}_{2N}^{(1)}\boldsymbol{x}_N = \quad \boldsymbol{a}_2^{(1)},\\
\vdots & & & \ddots & \vdots \qquad\quad \vdots\\
N^{(N-1)}: & & & & \boldsymbol{A}_{NN}^{(N-1)}\boldsymbol{x}_N = \quad \boldsymbol{a}_N^{(N-1)}.
\end{array}$$

Durch Rückrechnung ergeben sich daraus die $\boldsymbol{x}_i$, $i = 1(1)N$, man erhält N Gleichungssysteme

$$\begin{aligned}
\boldsymbol{A}_{NN}^{(N-1)}\boldsymbol{x}_N &= \boldsymbol{a}_N^{(N-1)},\\
\boldsymbol{A}_{jj}^{(j-1)}\boldsymbol{x}_j &= \boldsymbol{a}_j^{(j-1)} - \sum_{k=j+1}^{N} \boldsymbol{A}_{jk}^{(j-1)}\boldsymbol{x}_k,
\end{aligned}$$

wobei die $\boldsymbol{A}_{jj}^{(j-1)}$ quadratisch sind. Diese Systeme lassen sich jetzt mit dem Gaußschen Algorithmus (mit Pivotisierung) gemäß Abschnitt 4.5 behandeln.

Bei der numerischen Lösung partieller Differentialgleichungen und Integralgleichungen treten häufig Gleichungssysteme mit blockweise tridiagonalen Matrizen auf. Im Folgenden wird ein Algorithmus für diesen Fall angegeben.

4.15.3 Gauß-Algorithmus für tridiagonale Blocksysteme

Gegeben sei das lineare Gleichungssystem (4.20) mit tridiagonaler Blockmatrix der Form

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{D}_1 & \boldsymbol{C}_1 & & & \\ \boldsymbol{B}_2 & \boldsymbol{D}_2 & \boldsymbol{C}_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \boldsymbol{B}_{N-1} & \boldsymbol{D}_{N-1} & \boldsymbol{C}_{N-1} \\ & & & \boldsymbol{B}_N & \boldsymbol{D}_N \end{pmatrix}$$

Dann ergeben sich die Lösungen des Systems, indem man die Matrix $\boldsymbol{A}$ analog zu Abschnitt 4.9 in das Produkt $\boldsymbol{CB}$ zweier bidiagonaler Blockmatrizen zerlegt und damit das System $\boldsymbol{Ax} = \boldsymbol{a}$ in ein äquivalentes System $\boldsymbol{Bx} = \boldsymbol{b}$ mit $\boldsymbol{b} = \boldsymbol{C}^{-1}\boldsymbol{a}$ überführt, welches rekursiv gelöst werden muss. Mit

$$\boldsymbol{C} = \begin{pmatrix} \boldsymbol{A}_1 & & & \\ \boldsymbol{B}_2 & \boldsymbol{A}_2 & & \\ & \ddots & \ddots & \\ & & \boldsymbol{B}_N & \boldsymbol{A}_N \end{pmatrix}, \; \boldsymbol{B} = \begin{pmatrix} 1 & \boldsymbol{G}_1 & & & \\ & 1 & \boldsymbol{G}_2 & & \\ & & \ddots & \ddots & \\ & & & 1 & \boldsymbol{G}_{N-1} \\ & & & & 1 \end{pmatrix}, \; \boldsymbol{b} = \begin{pmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \vdots \\ \boldsymbol{b}_{N-1} \\ \boldsymbol{b}_N \end{pmatrix}$$

berechnet man aus

$$\boldsymbol{A} = \boldsymbol{CB} \qquad \text{und} \qquad \boldsymbol{a} = \boldsymbol{C}\,\boldsymbol{b}$$

die Elemente der Matrizen $\boldsymbol{C}$, $\boldsymbol{B}$ und des Vektors $\boldsymbol{b}$ und die Lösung $\boldsymbol{x}$ durch Rückrechnung aus $\boldsymbol{Bx} = \boldsymbol{b}$. Dies wird an einem Blocksystem mit einer tridiagonalen (3,3)-Blockmatrix demonstriert:

$$\begin{pmatrix} \boldsymbol{D}_1 & \boldsymbol{C}_1 & \boldsymbol{0} \\ \boldsymbol{B}_2 & \boldsymbol{D}_2 & \boldsymbol{C}_2 \\ \boldsymbol{0} & \boldsymbol{B}_3 & \boldsymbol{D}_3 \end{pmatrix} \cdot \begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{x}_3 \end{pmatrix} = \begin{pmatrix} \boldsymbol{a}_1 \\ \boldsymbol{a}_2 \\ \boldsymbol{a}_3 \end{pmatrix}.$$

Aus der Zerlegung $\boldsymbol{A} = \boldsymbol{CB}$ ergeben sich mit den Einheitsmatrizen $\boldsymbol{Z}$ wegen

$$\begin{aligned} \begin{pmatrix} \boldsymbol{D}_1 & \boldsymbol{C}_1 & \boldsymbol{0} \\ \boldsymbol{B}_2 & \boldsymbol{D}_2 & \boldsymbol{C}_2 \\ \boldsymbol{0} & \boldsymbol{B}_3 & \boldsymbol{D}_3 \end{pmatrix} &= \begin{pmatrix} \boldsymbol{A}_1 & \boldsymbol{0} & \boldsymbol{0} \\ \widetilde{\boldsymbol{B}}_2 & \boldsymbol{A}_2 & \boldsymbol{0} \\ \boldsymbol{0} & \widetilde{\boldsymbol{B}}_3 & \boldsymbol{A}_3 \end{pmatrix} \cdot \begin{pmatrix} \boldsymbol{Z} & \boldsymbol{G}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Z} & \boldsymbol{G}_2 \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{Z} \end{pmatrix} \\ &= \begin{pmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_1\boldsymbol{G}_1 & \boldsymbol{0} \\ \widetilde{\boldsymbol{B}}_2 & \widetilde{\boldsymbol{B}}_2\boldsymbol{G}_1 + \boldsymbol{A}_2 & \boldsymbol{A}_2\boldsymbol{G}_2 \\ \boldsymbol{0} & \widetilde{\boldsymbol{B}}_3 & \widetilde{\boldsymbol{B}}_3\boldsymbol{G}_2 + \boldsymbol{A}_3 \end{pmatrix} \end{aligned}$$

die Beziehungen

$$\begin{aligned} \boldsymbol{D}_1 &= \boldsymbol{A}_1 & \qquad \boldsymbol{C}_2 &= \boldsymbol{A}_2\boldsymbol{G}_2 \\ \boldsymbol{C}_1 &= \boldsymbol{A}_1\boldsymbol{G}_1 & \qquad \boldsymbol{B}_3 &= \widetilde{\boldsymbol{B}}_3 \\ \boldsymbol{B}_2 &= \widetilde{\boldsymbol{B}}_2 & \qquad \boldsymbol{D}_3 &= \widetilde{\boldsymbol{B}}_3\boldsymbol{G}_2 + \boldsymbol{A}_3, \\ \boldsymbol{D}_2 &= \widetilde{\boldsymbol{B}}_2\boldsymbol{G}_1 + \boldsymbol{A}_2 \end{aligned}$$

also allgemein

$$\begin{array}{lcll} \boldsymbol{A}_1 & = & \boldsymbol{D}_1 & \\ \boldsymbol{G}_i & = & \boldsymbol{A}_i^{-1}\boldsymbol{C}_i & i=1,2 \\ \boldsymbol{A}_i & = & \boldsymbol{D}_i - \boldsymbol{B}_i\boldsymbol{G}_{i-1} & i=2,3. \end{array}$$

Analog erhält man aus $\boldsymbol{a} = \boldsymbol{C}\boldsymbol{b}$ mit der bereits bestehenden Matrix $\boldsymbol{C}$ die Vektoren $\boldsymbol{b}_i$

$$\begin{array}{lcll} \boldsymbol{b}_1 & = & \boldsymbol{A}_1^{-1}\boldsymbol{a}_1 & \\ \boldsymbol{b}_i & = & \boldsymbol{A}_i^{-1}(\boldsymbol{a}_i - \boldsymbol{B}_i\boldsymbol{b}_{i-1}), & i=2,3. \end{array}$$

Schließlich wird die Rückrechnung $\boldsymbol{B}\boldsymbol{x} = \boldsymbol{b}$ durchgeführt, um die Lösungen $\boldsymbol{x}_i$ zu erhalten:

$$\begin{pmatrix} \boldsymbol{Z} & \boldsymbol{G}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Z} & \boldsymbol{G}_2 \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{Z} \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{x}_3 \end{pmatrix} = \begin{pmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \boldsymbol{b}_3 \end{pmatrix}.$$

$$\Rightarrow \boldsymbol{Z}\boldsymbol{x}_3 = \boldsymbol{x}_3 = \boldsymbol{b}_3 \qquad \boldsymbol{x}_i = \boldsymbol{b}_i - \boldsymbol{G}_i\boldsymbol{x}_{i+1}, \qquad i=2,1.$$

Zusammengefasst und auf (N,N)-Systeme erweitert, ergeben sich die Lösungen eines tridiagonalen Blocksystems, wenn man die folgenden Berechnungen in der angegebenen Reihenfolge ausführt (siehe auch [ISAA1973], S.61-64):

(1) $\boldsymbol{A}_1 = \boldsymbol{D}_1,\ \boldsymbol{G}_1 = \boldsymbol{A}_1^{-1}\boldsymbol{C}_1$

(2) $\boldsymbol{A}_i = \boldsymbol{D}_i - \boldsymbol{B}_i\boldsymbol{G}_{i-1}$, $i = 2(1)N$
$\boldsymbol{G}_i = \boldsymbol{A}_i^{-1}\boldsymbol{C}_i$, $i = 2(1)N-1$

(3) $\boldsymbol{b}_1 = \boldsymbol{A}_i^{-1}\boldsymbol{a}_1$
$\boldsymbol{b}_i = \boldsymbol{A}_i^{-1}(\boldsymbol{a}_i - \boldsymbol{B}_i\boldsymbol{b}_{i-1})$, $i = 2(1)N$

(4) $\boldsymbol{x}_N = \boldsymbol{b}_N$
$\boldsymbol{x}_i = \boldsymbol{b}_i - \boldsymbol{G}_i\boldsymbol{x}_{i+1}$, $i = N-1, N-2, \ldots, 1.$

4.15.4 Weitere Block-Verfahren

(1) Ist $\boldsymbol{A} = (\boldsymbol{A}_{ik})$ positiv definit und besitzen alle Diagonalblöcke $\boldsymbol{A}_{ii}$ eine und dieselbe Ordnung $n_1 = n_2 = \ldots = n_N = n/N$, so lässt sich die in [BERE1971], Bd.2, S.49-51 beschriebene Quadratwurzelmethode (Analogon zum Verfahren von Cholesky) anwenden.

(2) Sind alle Blöcke $\boldsymbol{A}_{ik}$ quadratische Matrizen der gleichen Ordnung, so lässt sich eine Blockmethode anwenden, die eine Modifikation des Verfahrens von Gauß-Jordan darstellt, (s. [BERE1971], Bd.2, S.51-54).

(3) Ein Beispiel zu Systemen mit tridiagonalen Blockmatrizen ist in [SCHW1972], S.210 zu finden. Dort liegt speziell eine diagonal blockweise tridiagonale Matrix vor, d. h. eine blockweise tridiagonale Matrix, deren Diagonalblöcke Diagonalmatrizen sind.

(4) Zur Blockiteration und Blockrelaxation s. [ISAA1973], S.63ff.; [SCHW1972], S.216ff.

4.16 Algorithmus von Cuthill-McKee für dünn besetzte, symmetrische Matrizen

Etwa bei der Lösung von Randwertproblemen für gewöhnliche und partielle Differentialgleichungen sowie beim Einsatz von Finite-Element-Methoden treten häufig dünn besetzte Matrizen auf.

Hier ist folgende Vorgehensweise zu empfehlen:

1. Die Anwendung des Algorithmus von Cuthill-McKee überführt die dünnbesetzte Matrix (z. B. Steifigkeitsmatrix) in eine Bandmatrix mit fast optimaler Bandbreite, aber eben im Allgemeinen noch nicht mit der möglichen minimalen Bandbreite.
2. Anschließend wird mit den Nummerierungen aus Cuthill-McKee als Startnummerierung der Algorithmus von Rosen angewandt, der im Allgemeinen die Bandbreite weiter verringert. Es gibt aber auch Fälle, bei denen damit keine weitere Verminderung der Bandbreite erzielt werden kann.

Hier wird der Algorithmus von Cuthill-McKee in exakt der Form aus [ENGE1996], Programmteil, verwendet. Die Theorie zu dem Algorithmus ist z. B. in [WEIS1990], 1.5.2 zu finden.

Algorithmus 4.100. (*Cuthill-McKee-Algorithmus*)

Gegeben: Die Nichtnull-Elemente a_{ik} einer symmetrischen, dünn besetzten (n,n)-Matrix $\boldsymbol{A}$ und ihre Zeilen- und Spaltenindizes i,k.

Gesucht: Die Cuthill-McKee-Nummerierung der Matrix.

1. Die Matrix $\boldsymbol{A}$ ist in Listenform zu speichern:
   - 1.1 Setze $NV := 0$
   - 1.2 Für $i = 1(1)n$:
     - 1.2.1 Setze $IR_i := NV + 1$
     - 1.2.2 Für alle Nichtnull-Elemente in Zeile i:
       - 1.2.2.1 Bestimme das nächste Nichtnull-Element a_{ik}
       - 1.2.2.2 Setze $NV := NV + 1; V_{NV} := a_{ik}; IC_{NV} := k$
   - 1.3 Setze $IR_{n+1} := NV + 1$
2. Konstruktion des Besetzungsgraphen:
   - 2.1 Setze $\mu := 0$
   - 2.2 Für $i = 1(1)n$:
     - 2.2.1 Setze $INB_i := \mu + 1$
     - 2.2.2 Für $k = IR_i(1)IR_{i+1} - 1$:

2.2.2.1 Wenn $IC_k \neq i$, dann
$\mu := \mu + 1$;
$NEIGHB_\mu := IC_k$

2.2.3 $IDEG_i := \mu + 1 - INB_i$

2.3 $INB_{n+1} := \mu + 1$

3. Berechnung der Cuthill-McKee-Nummerierung:

3.1 Felder initialisieren:
Für $i = 1(1)n$:

3.1.1 Setze Marke für Knoten i auf „falsch“. Setze $ICM_i := 0$.

3.2 Setze $NFOUND := 0$

3.3 Für $i = 1(1)n$:

3.3.1 Wenn Knoten i noch nicht markiert ist:

3.3.1.1 Suche zu der von Knoten i induzierten Komponente des Graphen einen Startknoten $IROOT$, der eine möglichst lange Stufenstruktur ergibt (Unter-Algorithmus 4.101).

3.3.1.2 Berechne die Cuthill-McKee-Nummerierung dieser Komponente mit dem Startknoten $IROOT$ und der Anfangsnummer
$ISTART := NFOUND + 1$.
Markiere alle Knoten dieser Komponente. (Unter-Algorithmus 4.102).

3.3.1.3 Setze $NFOUND := NFOUND +$ Anzahl der Knoten der neuen Komponente.

3.4 Das Feld ICM enthält jetzt die CM-Nummerierung des ganzen Graphen: Für $i = 1(1)n$ ist ICM_i die ursprüngliche Nummer des Knotens mit der CM-Nummer i. Berechne die Umkehrpermutation $ICMREV$:
Für $i = 1(1)n$:
Setze $ICMREV_{ICM_i} := i$.

Durch die CM-Permutation geht ein lineares Gleichungssystem $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ über in ein äquivalentes Gleichungssystem $\boldsymbol{A}^*\boldsymbol{x}^* = \boldsymbol{b}^*$, wobei

$$\begin{aligned} a^*_{ik} &= a_{ICM_i, ICM_k}, \\ b^*_i &= b_{ICM_i}, \\ x^*_i &= x_{ICM_i}, \qquad i = 1(1)n,\ k = 1(1)n \end{aligned}$$

bzw.

$$\begin{aligned} a_{ik} &= a^*_{ICMREV_i, ICMREV_k}, \\ b_i &= b^*_{ICMREV_i}, \\ x_i &= x^*_{ICMREV_i}, \qquad i = 1(1)n,\ k = 1(1)n. \end{aligned}$$

Algorithmus 4.101. (*Unter-Algorithmus zu Algorithmus 4.100*)

Gegeben: (a) Der Besetzungsgraph einer (n,n)-Matrix $\boldsymbol{A}$ in den Feldern $NEIGHB$ und INB aus Algorithmus 4.100 Schritt 2.

(b) Feld $IDEG$ mit Knotengraden aus Algorithmus 4.100 Schritt 2.2.3.

(c) Gewisse Knoten in $NEIGHB$ können von früheren Ausführungen des Schrittes 3.3.1 im Algorithmus 4.100 her markiert sein.

(d) Die Nummer $IROOT$ eines nicht markierten Knotens.

Gesucht: Ein Startknoten für eine möglichst lange Stufenstruktur der von $IROOT$ erzeugten Komponente des Graphen.

1. Setze $NLVOLD := 0$.
   $NLVOLD$ ist die bisher gefundene maximale Stufenstrukturlänge.
2. Bilde die Stufenstruktur zu $IROOT$ in den Feldern $LEVEL$ und ILV:
   - 2.1 Setze $NLV := 0; LEVEL_1 := IROOT; \mu := 1; LEVEND := 0$.
     Markiere Knoten $IROOT$.
   - 2.2 Setze $NLV := NLV + 1; LEVBEG := LEVEND + 1;$
     $LEVEND := \mu; ILV_{NLV} := LEVBEG$.
     Hiernach ist NLV die Anzahl der bisher gefundenen Stufen (die erste Stufe besteht aus $IROOT$). $LEVBEG$ zeigt auf den Anfang, $LEVEND$ auf das Ende der zuletzt gefundenen Stufe (NLV) im Feld $LEVEL$.
   - 2.3 Bestimme die Knoten der nächsten Stufe $NLV+1$. Dazu finde alle noch nicht markierten Nachbarn von Knoten der Stufe NLV und trage sie in $LEVEL$ ein:
     Für $i = LEVBEG(1)LEVEND$:
     - 2.3.1 Für $j = INB_{LEVEL_i}(1)INB_{LEVEL_{i+1}} - 1$:
       - 2.3.1.1 Wenn Knoten $NEIGHB_j$ noch nicht markiert ist, dann setze $\mu := \mu + 1; LEVEL_\mu := NEIGHB_j$ und markiere Knoten $NEIGHB_j$.
   - 2.4 Wenn $\mu > LEVEND$, d. h. wenn in 2.3 neue Knoten gefunden wurden, gehe zu 2.2. Ansonsten ist die Stufenstruktur der von $IROOT$ erzeugten Komponente fertig.
   - 2.5 Setze $LVNODES := LEVEND$. Dies ist die Anzahl der Knoten in dieser Komponente. Setze $ILV_{NLV+1} := LVNODES + 1$.
   - 2.6 Setze alle in 2.1 bis 2.3 gesetzten Knotenmarkierungen zurück.
3. Wenn $NLV \leq NLVOLD$, dann hat sich die Stufenstruktur gegenüber der letzten Ausführung von Schritt 2 nicht verlängert. In diesem Fall beende Algorithmus 4.101. Andernfalls:
4. Setze $NLVOLD := NLV$.
5. Suche einen Knoten minimalen Grades auf der letzten Stufe, d. h. suche i_0 mit $IDEG_{i_0} = \min\{IDEG_{LEVEL_i} | i = ILV_{NLV}(1)ILV_{NLV+1} - 1\}$
   Ersetze $IROOT$ durch $LEVEL_{i_0}$.
6. Gehe zu 2.

Algorithmus 4.102. (*Unter-Algorithmus zu Algorithmus 4.100*)

Gegeben: (a) Der Besetzungsgraph einer (n, n)-Matrix $\boldsymbol{A}$ in den Feldern $NEIGHB$ und INB aus Algorithmus 4.100 Schritt 2.

(b) Feld $IDEG$ mit Knotengraden aus Algorithmus 4.100 Schritt 2.2.3. von Algorithmus 4.101 bestimmt.

(d) Anfangsnummer $ISTART$ für die CM-Nummerierung dieser Komponente.

(e) In früheren Ausführungen von Algorithmus 4.100 Schritt 3.3.1 wurden ggf. schon Komponenten des Graphen erfasst. Die CM-Nummerierung dieser Komponenten steht im Feld ICM in den Elementen ICM_i, $i = 1(1)ISTART - 1$. Die Knoten dieser Komponente sind markiert, alle anderen sind nicht markiert.

Gesucht: Die CM-Nummerierung der von $IROOT$ erzeugten Komponente.

1. Setze $ICM_{ISTART} := IROOT$, $NEWEND := ISTART$; $LEVEND := ISTART - 1$. Markiere Knoten $IROOT$.
2. Bilde die Stufenstruktur zu $IROOT$ im Feld ICM:

   2.1 Setze $LEVBEG := LEVEND + 1; LEVEND := NEWEND$. Hiernach zeigt $LEVBEG$ auf den Anfang, $LEVEND$ auf das Ende der zuletzt gefundenen Stufe im Feld ICM. (Die erste Stufe besteht nur aus $IROOT$.)

   2.2 Bestimme die Knoten der nächsten Stufe: dazu finde alle noch nicht markierten Nachbarn von Knoten der letzten Stufe und trage sie in ICM ein: Für $i = LEVBEG(1)LEVEND$:

      2.2.1 Setze $NEWBEG := NEWEND + 1$. $NEWBEG$ zeigt auf den Anfang der jetzt neu zu bildenden Stufe in ICM. $NEWEND$ zeigt immer auf den zuletzt gefundenen Knoten in ICM.

      2.2.2 Für $j = INB_{ICM_i}(1)INB_{ICM_i+1} - 1$:

         2.2.2.1 Wenn Knoten $NEIGHB_j$ noch nicht markiert ist, dann setze $NEWEND := NEWEND + 1$; $ICM_{NEWEND} := NEIGHB_j$ und markiere Knoten $NEIGHB_j$.

      2.2.3 Sortiere die Elemente ICM_i, $i = NEWBEG(1)NEWEND$, nach steigendem Grad.

   2.3 Wenn $NEWEND > LEVEND$, d. h. wenn in Schritt 2.2 neue Knoten gefunden werden, gehe zu 2.1, andernfalls ist die Durchführung des Algorithmus 4.102 beendet.

Programme dazu sind in der Programmsammlung auf der CD-ROM enthalten. Ein Nachvollziehen „per Hand“ ist hier nicht sinnvoll.

4.17 Entscheidungshilfen für die Auswahl des Verfahrens

Trotz der Vielzahl numerischer Verfahren, die zur Lösung linearer Gleichungssysteme zur Verfügung stehen, ist die praktische Bestimmung der Lösungen für große Werte von n eine problematische numerische Aufgabe. Die Gründe hierfür sind

(1) der Arbeitsaufwand (die Rechenzeit),

(2) der Speicherplatzbedarf,

(3) die Verfälschung der Ergebnisse durch Rundungsfehler oder mathematische Instabilität des Problems.

Zu (1): Der Arbeitsaufwand lässt sich über die Anzahl erforderlicher Punktoperationen (Multiplikationen, Divisionen) abschätzen.
Die folgende Tabelle liefert die Anzahl der Punktoperationen, die erforderlich sind, um ein lineares Gleichungssystem aus n Gleichungen mit n Unbekannten nach den angegebenen Verfahren zu lösen. Die Anzahl erforderlicher Additionen und Subtraktionen bleibt in diesem Vergleich unberücksichtigt.

Tabelle. (*Anzahl der Punktoperationen bei n Gleichungen mit n Unbekannten*)

Verfahren	Anzahl der Punktoperationen
Gauß-Algorithmus	$\frac{n^3}{3} + n^2 - \frac{n}{3}$
Cholesky-Verfahren	$\frac{n^3}{6} + O(n^2)$
Gauß-Jordan-Verfahren	$\frac{n^3}{2} + n^2 + \frac{n}{2}$
Austauschverfahren	$n^3 + n^2$
Verf. für tridiagonale Matrizen	$5n - 4$
Verf. für zyklisch tridiagonale Matrizen	$11n - 16$
Verf. für fünfdiagonale Matrizen	$11n - 16$
Iterationsverf. (pro Iterationsschritt)	$2n^2 - 2n$

Zu (2): Vom Computer her gesehen ergeben sich bezüglich des Speicherplatzes zwei kritische Größen für *sehr* große n:

(a) der für die Speicherung der a_{ik} verfügbare Platz im Arbeitsspeicher (Hauptspeicher),

(b) der dafür verfügbare Platz in den Hintergrundspeichern.

Der Speicherplatzbedarf verringert sich, wenn $\boldsymbol{A}$ spezielle Eigenschaften, z. B. Bandstruktur, besitzt, dünn besetzt ist oder symmetrisch ist. Es entsteht praktisch kein Speicherplatzbedarf, wenn sich die a_{ik} aufgrund einer im Einzelfall gegebenen Vorschrift jeweils im Computer berechnen lassen („generated Matrix"), siehe auch die folgende Bemerkung.

Zu (3): Durch geeignete Gestaltung des Ablaufs der Rechnung kann die Akkumulation von Rundungsfehlern unter Kontrolle gehalten werden, sofern die Ursache nicht in mathematischer Instabilität des Problems liegt. Deshalb sollte grundsätzlich mit skalierter teilweiser Pivotisierung gearbeitet werden, es sei denn, die spezielle Struktur des Systems garantiert numerische Stabilität. Mit relativ geringem Aufwand lassen sich die Ergebnisse jeweils durch Nachiteration verbessern.

Im Allgemeinen lassen sich weder die Kondition des Systems noch die Frage, ob die Bedingungen für die eindeutige Lösbarkeit erfüllt sind, vor Beginn der numerischen Rechnung prüfen. Daher sollten die Programme so gestaltet sein, dass sie den Benutzern im Verlaufe der Rechnung darüber Auskunft geben.

Bemerkungen zu großen Systemen und dünnbesetzten Matrizen:

Bei sehr großen Systemen, bei denen die Elemente von $\boldsymbol{A}$ und $\boldsymbol{a}$ nicht vollständig im Arbeitsspeicher unterzubringen sind (selbst nicht in gepackter Form), können sogenannte Blockmethoden angewandt werden, s. dazu Abschnitt 4.15. Solche Systeme treten vorwiegend im Zusammenhang mit der numerischen Lösung partieller Differentialgleichungen auf. Sind die Matrizen dünn besetzt, wie es häufig bei der Lösung von Randwertproblemen für gewöhnliche und partielle Differentialgleichungen durch Differenzenverfahren oder die Finite-Elemente-Methode auftritt, sollten entsprechende Lösungsalgorithmen verwendet werden, siehe dazu z. B. [MAES1985] und [WEIS1990], 1.

1. Die Anwendung des Algorithmus von Cuthill-McKee [CUTH1969] überführt die dünnbesetzte Matrix (z. B. Steifigkeitsmatrix) in eine Bandmatrix mit fast optimaler Bandbreite, aber eben im Allgemeinen noch nicht mit der möglichen minimalen Bandbreite.

2. Anschließend wird mit den Nummerierungen aus Cuthill-McKee als Startnummerierung der Algorithmus von Rosen angewandt, der im Allgemeinen die Bandbreite weiter verringert. Es gibt aber auch Fälle, wo damit keine weitere Verminderung der Bandbreite erzielt werden kann.

Weitere geeignete Verfahren, insbesondere auch Iterationsverfahren, sind in [WEIS1990] zu finden.

Ergänzende Literatur zu Kapitel 4

[DEUF2002] Bd.1, Kap.1; [GRAM2000], Kap.5; [HAMM1994], Kap.2; [KNOR2003], Kap.3; [KOCK1990]; [OPFE2002], Kap.6, [PLAT2000], Kap.4; [PREU2001], Kap.3; [QUAR2002] I, Kap.3; [STOE1999], Kap.4; [TORN1990] Bd.1; [WEIS1984], 6.3, 6.5; [WERN1993] III 1, 2, S.149/150; [YOUN2003], 14, 18; [ZURM1997], 6.1, 6.6, 6.7, 8.

Kapitel 5

Iterationsverfahren zur Lösung linearer Gleichungssysteme

5.1 Vorbemerkungen

Bei den direkten Methoden besteht aufgrund der großen Anzahl von Punktoperationen proportional zu n^3 die Gefahr der Akkumulation von Rundungsfehlern, so dass bei schlecht konditioniertem System die Lösung völlig unbrauchbar werden kann. Dagegen sind die iterativen Methoden gegenüber Rundungsfehlern weitgehend unempfindlich, da jede Näherungslösung als Ausgangsnäherung für die folgende Iterationsstufe angesehen werden kann. Die Iterationsverfahren konvergieren jedoch nicht für alle lösbaren Systeme.

Die hier angegebenen Verfahren in Gesamt- und Einzelschritten konvergieren nur linear und außerdem (wegen eines für wachsendes n ungünstiger werdenden Wertes der Lipschitzkonstante) bei den meisten in der Praxis vorkommenden Problemen auch noch sehr langsam. Deshalb sind die iterativen Methoden den direkten nur in sehr speziellen Fällen überlegen, nämlich dann, wenn $\boldsymbol{A}$ schwach besetzt, sehr groß und so strukturiert ist, dass bei Anwendung eines der direkten Verfahren die zu verarbeitenden Matrizen nicht mehr in den oder die verfügbaren Speicher passen. Die Konvergenz kann im Allgemeinen mit einem auf dem Gesamt- bzw. Einzelschrittverfahren aufbauenden Relaxationsverfahren beschleunigt werden. Dies erfordert jedoch zusätzlich eine möglichst genaue Bestimmung des betragsgrößten und des betragskleinsten Eigenwertes der Iterationsmatrix bei Anwendung des Gesamtschrittverfahrens bzw. des betragsgrößten Eigenwertes bei Anwendung des Einzelschrittverfahrens.

5.2 Vektor- und Matrizennormen

$\mathbf{R}^n$ sei ein n-dimensionaler Vektorraum und $\boldsymbol{x}$ ein Element von $\mathbf{R}^n$. Unter der Norm von $\boldsymbol{x}$ versteht man eine diesem Vektor zugeordnete reelle Zahl $\|\boldsymbol{x}\|$, die die folgenden ***Vektor-Norm-Axiome*** erfüllt:

1. $\|\boldsymbol{x}\| \geq 0$ für alle $\boldsymbol{x} \in \mathbb{R}^n$, $\|\boldsymbol{x}\| = 0$ genau dann, wenn $\boldsymbol{x} = \boldsymbol{0}$ ist,
2. $\|\alpha\boldsymbol{x}\| = |\alpha| \, \|\boldsymbol{x}\|$ für alle $\boldsymbol{x} \in \mathbb{R}^n$ und beliebige Zahlen $\alpha \in \mathbb{R}$.
3. $\|\boldsymbol{x} + \boldsymbol{y}\| \leq \|\boldsymbol{x}\| + \|\boldsymbol{y}\|$ für alle $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ (Dreiecksungleichung).

Vektor-Normen sind z. B.:

$$\|\boldsymbol{x}\|_\infty := \max_{1 \leq i \leq n} |x_i| \qquad \text{(sup-Norm oder Maximumnorm)},$$

$$\|\boldsymbol{x}\|_1 := \sum_{i=1}^{n} |x_i| \qquad \text{(Norm der Komponenten-Betragssumme)},$$

$$\|\boldsymbol{x}\|_2 := \sqrt{\sum_{i=1}^{n} |x_i|^2} \qquad \text{(Euklidische Norm)}.$$

Beispiel 5.1.

$$\begin{aligned} \boldsymbol{x} &= (2, -4, 8)^\mathsf{T} \\ \|\boldsymbol{x}\|_\infty &= \max\{2, 4, 8\} = 8 \\ \|\boldsymbol{x}\|_1 &= |2| + |-4| + |8| = 14 \\ \|\boldsymbol{x}\|_2 &= \sqrt{2^2 + 4^2 + 8^2} = \sqrt{84} = 9.17 \end{aligned}$$

□

Ist $\boldsymbol{A}$ eine (n, n)-Matrix mit $\boldsymbol{A} = (a_{ik})$, $a_{ik} \in \mathbb{R}$, so heißt eine reelle Zahl $\|\boldsymbol{A}\|$ Norm der (n, n)-Matrix $\boldsymbol{A}$, wenn sie den ***Matrix-Norm-Axiomen*** genügt:

1. $\|\boldsymbol{A}\| \geq 0$ für alle $\boldsymbol{A}$,
   $\|\boldsymbol{A}\| = 0$ genau dann, wenn $\boldsymbol{A} = \boldsymbol{0}$ (Nullmatrix, d. h. alle $a_{ik} = 0$) ist,
2. $\|\alpha\boldsymbol{A}\| = |\alpha| \, \|\boldsymbol{A}\|$ für alle $\boldsymbol{A}$ und beliebige Zahlen $\alpha \in \mathbb{R}$,
3. $\|\boldsymbol{A} + \boldsymbol{B}\| \leq \|\boldsymbol{A}\| + \|\boldsymbol{B}\|$ für alle $\boldsymbol{A}, \boldsymbol{B}$.

Außerdem gilt für die hier betrachteten Matrixnormen:

$$\|\boldsymbol{A}\boldsymbol{B}\| \leq \|\boldsymbol{A}\| \, \|\boldsymbol{B}\|.$$

Matrix-Normen sind z. B.:

$$\|\boldsymbol{A}\|_\infty := \max_{1 \leq i \leq n} \sum_{k=1}^{n} |a_{ik}| \qquad \text{(Zeilensummennorm)},$$

$$\|\boldsymbol{A}\|_1 := \max_{1 \leq k \leq n} \sum_{i=1}^{n} |a_{ik}| \qquad \text{(Spaltensummennorm)},$$

$$\|\boldsymbol{A}\|_2 := \sqrt{\sum_{i,k=1}^{n} |a_{ik}|^2} \qquad \text{(Euklidische Norm)}.$$

Beispiel 5.2.

$$\begin{aligned}
\boldsymbol{A} &= \begin{pmatrix} 1 & -2 & 3 \\ 4 & 5 & -6 \\ -7 & 8 & 9 \end{pmatrix} \\
\|\boldsymbol{A}\|_\infty &= \max\{6,\, 15,\, 24\} = 24 \\
\|\boldsymbol{A}\|_1 &= \max\{12,\, 15,\, 18\} = 18 \\
\|\boldsymbol{A}\|_2 &= \sqrt{1+4+9+16+25+36+49+64+81} = \sqrt{285} = 16.88
\end{aligned}$$

□

Die eingeführten Matrix-Normen müssen mit den Vektor-Normen verträglich sein.

Definition 5.3.
Eine Matrix-Norm heißt mit einer Vektor-Norm verträglich, wenn für jede Matrix $\boldsymbol{A}$ und jeden Vektor $\boldsymbol{x}$ die Ungleichung

$$\|\boldsymbol{A}\boldsymbol{x}\| \le \|\boldsymbol{A}\|\,\|\boldsymbol{x}\|$$

erfüllt ist. Die Bedingung heißt *Verträglichkeitsbedingung.*

Die Matrix-Normen $\|\boldsymbol{A}\|_j$ sind mit den Vektor-Normen $\|\boldsymbol{x}\|_j$ verträglich für $j = 1, 2, \infty$.

5.3 Das Iterationsverfahren in Gesamtschritten

Gegeben sei das lineare Gleichungssystem $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}$ mit $\det \boldsymbol{A} \neq 0$, das ausgeschrieben die Form

$$\left\{\begin{array}{lllllllll}
a_{11}x_1 & + & a_{12}x_2 & + & \ldots & + & a_{1n}x_n & = & a_1\,, \\
a_{21}x_1 & + & a_{22}x_2 & + & \ldots & + & a_{2n}x_n & = & a_2\,, \\
\vdots & & & & & & & & \vdots \\
a_{n1}x_1 & + & a_{n2}x_2 & + & \ldots & + & a_{nn}x_n & = & a_n\,,
\end{array}\right. \tag{5.1}$$

besitzt. Um einen Näherungsvektor für $\boldsymbol{x}$ zu finden, konstruiert man eine Folge $\{\boldsymbol{x}^{(\nu)}\}$, $\nu = 1, 2, \ldots$, für die unter gewissen Voraussetzungen $\lim\limits_{\nu\to\infty} \boldsymbol{x}^{(\nu)} = \boldsymbol{x}$ gilt.

Es sei o. B. d. A. vorausgesetzt, dass keines der Diagonalelemente a_{jj} von $\boldsymbol{A}$ verschwindet, andernfalls werden die Zeilen entsprechend vertauscht. Indem man jeweils die i-te Gleichung von (5.1) nach x_i auflöst, bringt man das System auf die äquivalente Form:

$$x_i = -\sum_{\substack{k=1 \\ k\neq i}}^{n} \frac{a_{ik}}{a_{ii}} x_k + \frac{a_i}{a_{ii}}, \quad i = 1(1)n, \tag{5.2}$$

die mit den Abkürzungen

$$c_i = \frac{a_i}{a_{ii}}, \quad b_{ik} = \begin{cases} -\frac{a_{ik}}{a_{ii}} & \text{für} \quad i \neq k \\ 0 & \text{für} \quad i = k \end{cases} \tag{5.3}$$

in Matrizenschreibweise lautet

$$\boldsymbol{x} = \boldsymbol{B}\boldsymbol{x} + \boldsymbol{c} \quad \text{mit} \quad \boldsymbol{B} = (b_{ik}), \quad \boldsymbol{c} = \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix}. \tag{5.4}$$

Man definiert eine vektorielle Schrittfunktion durch

$$\boldsymbol{\varphi}(\boldsymbol{x}) := \boldsymbol{B}\boldsymbol{x} + \boldsymbol{c}$$

und konstruiert mit einem Startvektor $\boldsymbol{x}^{(0)}$ und der Vorschrift

$$\begin{cases} \boldsymbol{x}^{(\nu+1)} = \boldsymbol{\varphi}(\boldsymbol{x}^{(\nu)}) = \boldsymbol{B}\boldsymbol{x}^{(\nu)} + \boldsymbol{c} \\ \\ \text{mit} \quad \boldsymbol{x}^{(\nu)} = \begin{pmatrix} x_1^{(\nu)} \\ x_2^{(\nu)} \\ \vdots \\ x_n^{(\nu)} \end{pmatrix}, \quad \nu = 0, 1, 2, \ldots, \end{cases} \tag{5.5}$$

eine Folge $\{\boldsymbol{x}^{(\nu)}\}$; komponentenweise lautet die *Iterationsvorschrift* (s. auch Kapitel 2, Abschnitt 2.3)

$$x_i^{(\nu+1)} = c_i + \sum_{k=1}^{n} b_{ik} x_k^{(\nu)} = \frac{a_i}{a_{ii}} - \sum_{\substack{k=1 \\ k \neq 1}}^{n} \frac{a_{ik}}{a_{ii}} x_k^{(\nu)}, \qquad \begin{matrix} i = 1(1)n, \\ \nu = 0, 1, 2, \ldots \end{matrix} \tag{5.6}$$

Die Matrix $\boldsymbol{B}$ heißt *Iterationsmatrix*. Die Rechnung wird zweckmäßig in einem Schema der folgenden Form durchgeführt:

Rechenschema 5.4. (*Iteration in Gesamtschritten für n = 3*)

c_i	b_{ik}			$x_i^{(0)}$	$x_i^{(1)}$	$\cdots$	
$\frac{a_1}{a_{11}}$	0	$-\frac{a_{12}}{a_{11}}$	$-\frac{a_{13}}{a_{11}}$	0			
$\frac{a_2}{a_{22}}$	$-\frac{a_{21}}{a_{22}}$	0	$-\frac{a_{23}}{a_{22}}$	0			
$\frac{a_3}{a_{33}}$	$-\frac{a_{31}}{a_{33}}$	$-\frac{a_{32}}{a_{33}}$	0	0			

Beispiel 5.5.

Gegeben: Das lineare Gleichungssystem

$$\left\{\begin{array}{rcrcrcl} 10x_1 & + & 2x_2 & + & x_3 & = & 13, \\ x_1 & + & 10x_2 & + & 2x_3 & = & 13, \\ 2x_1 & + & x_2 & + & 10x_3 & = & 13; \end{array}\right. \tag{5.7}$$

es besitzt die exakte Lösung $\boldsymbol{x}^{\mathsf{T}} = (1,1,1)$. Durch Auflösen jeder Zeile von (5.7) nach dem Diagonalelement erhält man die äquivalente Form (5.2):

$$\left\{\begin{array}{rcrcrcl} x_1 & = & 1.3 & - & 0.2x_2 & - & 0.1x_3, \\ x_2 & = & 1.3 & - & 0.1x_1 & - & 0.2x_3, \\ x_3 & = & 1.3 & - & 0.2x_1 & - & 0.1x_2 \end{array}\right.$$

bzw. $\boldsymbol{x} = \boldsymbol{B}\boldsymbol{x} + \boldsymbol{c}$ mit

$$\boldsymbol{B} = \begin{pmatrix} 0 & -0.2 & -0.1 \\ -0.1 & 0 & -0.2 \\ -0.2 & -0.1 & 0 \end{pmatrix}, \quad \boldsymbol{c} = \begin{pmatrix} 1.3 \\ 1.3 \\ 1.3 \end{pmatrix}. \tag{5.8}$$

Die zugehörige Iterationsvorschrift (5.5) lautet ausgeschrieben

$$\begin{pmatrix} x_1^{(\nu+1)} \\ x_2^{(\nu+1)} \\ x_3^{(\nu+1)} \end{pmatrix} = \begin{pmatrix} 0 & -0.2 & -0.1 \\ -0.1 & 0 & -0.2 \\ -0.2 & -0.1 & 0 \end{pmatrix} \begin{pmatrix} x_1^{(\nu)} \\ x_2^{(\nu)} \\ x_3^{(\nu)} \end{pmatrix} + \begin{pmatrix} 1.3 \\ 1.3 \\ 1.3 \end{pmatrix}$$

Rechenschema

$\boldsymbol{c}$	$\boldsymbol{B}$			$\boldsymbol{x}^{(0)}$	$\boldsymbol{x}^{(1)}$	$\boldsymbol{x}^{(2)}$	$\boldsymbol{x}^{(3)}$	...
1.3	0	−0.2	−0.1	0	1.3	0.91	1.027	
1.3	−0.1	0	−0.2	0	1.3	0.91	1.027	
1.3	−0.2	−0.1	0	0	1.3	0.91	1.027	

Man erkennt schon aus den ersten drei Iterationsschritten, dass die Folge $\{\boldsymbol{x}^{(\nu)}\}$ wahrscheinlich konvergiert.

Benutzt man allerdings die Gleichungen (5.7) in der Reihenfolge

$$\left\{\begin{array}{rcrcrcl} x_1 & + & 10x_2 & + & 2x_3 & = & 13, \\ 2x_1 & + & x_2 & + & 10x_3 & = & 13, \\ 10x_1 & + & 2x_2 & + & x_3 & = & 13, \end{array}\right.$$

so erhält man die Iterationsvorschrift

$$\begin{pmatrix} x_1^{(\nu+1)} \\ x_2^{(\nu+1)} \\ x_3^{(\nu+1)} \end{pmatrix} = \begin{pmatrix} 0 & -10 & -2 \\ -2 & 0 & -10 \\ -10 & -2 & 0 \end{pmatrix} \begin{pmatrix} x_1^{(\nu)} \\ x_2^{(\nu)} \\ x_3^{(\nu)} \end{pmatrix} + \begin{pmatrix} 13 \\ 13 \\ 13 \end{pmatrix}$$

und damit das Rechenschema

$\boldsymbol{c}$	$\boldsymbol{B}$			$\boldsymbol{x}^{(0)}$	$\boldsymbol{x}^{(1)}$	$\boldsymbol{x}^{(2)}$	$\boldsymbol{x}^{(3)}$	...
13	0	−10	−2	0	13	−143	1729	
13	−2	0	−10	0	13	−143	1729	
13	−10	−2	0	0	13	−143	1729	

Die so konstruierte Folge $\{\boldsymbol{x}^{(\nu)}\}$ divergiert offensichtlich. □

Aus diesem Beispiel erkennt man bereits, dass in dem zu lösenden System die Diagonalelemente betragsmäßig überwiegen sollten.

Mit den Begriffen *Vektor-Norm* und *Matrix-Norm* kann nun die Frage beantwortet werden, unter welchen Bedingungen die Folge $\{\boldsymbol{x}^{(\nu)}\}$ konvergiert.

Satz 5.6.
Es sei $\boldsymbol{x} \in \mathbb{R}^n$ eine Lösung der Gleichung $\boldsymbol{x} = \boldsymbol{\varphi}(\boldsymbol{x})$; $\boldsymbol{\varphi}(\boldsymbol{x})$ erfülle die Lipschitzbedingung bezüglich einer Vektornorm

$$\|\boldsymbol{\varphi}(\boldsymbol{x}) - \boldsymbol{\varphi}(\boldsymbol{x}')\| \leq L\|\boldsymbol{x} - \boldsymbol{x}'\| \quad \text{mit} \quad 0 \leq L < 1$$

für alle $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^n$.
Dann gilt für die durch $\boldsymbol{x}^{(\nu+1)} = \boldsymbol{\varphi}(\boldsymbol{x}^{(\nu)})$ mit dem beliebigen Startvektor $\boldsymbol{x}^{(0)} \in \mathbb{R}^n$ definierte Iterationsfolge $\{\boldsymbol{x}^{(\nu)}\}$:

1) $\lim\limits_{\nu\to\infty} \boldsymbol{x}^{(\nu)} = \boldsymbol{x}$;

2) $\boldsymbol{x}$ ist eindeutig bestimmt;

3) $\|\boldsymbol{x}^{(\nu)} - \boldsymbol{x}\| \leq \frac{L}{1-L}\|\boldsymbol{x}^{(\nu)} - \boldsymbol{x}^{(\nu-1)}\|$ (a posteriori-Fehlerabschätzung)
$\leq \frac{L^\nu}{1-L}\|\boldsymbol{x}^{(1)} - \boldsymbol{x}^{(0)}\|$ (a priori-Fehlerabschätzung).

Es interessieren nun die Bedingungen, die an die Elemente a_{ik} der gegebenen Matrix $\boldsymbol{A}$ eines Systems der Form (5.1) zu stellen sind, damit die Lipschitzbedingung aus Satz 5.6 bezüglich einer Vektor-Norm $\|.\|$ erfüllt ist. Siehe dazu auch Abschnitt 2.3.4.

Satz 5.7.
Ist für die Koeffizienten a_{ik} des linearen Gleichungssystems $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}$ mit $\boldsymbol{A} = (a_{ik})$, $i, k = 1(1)n$, das

a) *Zeilensummenkriterium*

$$\max_{1\le i\le n} \sum_{k=1}^{n} |b_{ik}| = \max_{1\le i\le n} \sum_{\substack{k=1\\k\neq i}}^{n} \left|\frac{a_{ik}}{a_{ii}}\right| \le L_\infty < 1, \tag{5.9}$$

b) *Spaltensummenkriterium*

$$\max_{1\le k\le n} \sum_{i=1}^{n} |b_{ik}| = \max_{1\le k\le n} \sum_{\substack{i=1\\i\neq k}}^{n} \left|\frac{a_{ik}}{a_{ii}}\right| \le L_1 < 1, \tag{5.10}$$

c) *Kriterium von Schmidt - v. Mises*

$$\sqrt{\sum_{i=1}^{n}\sum_{k=1}^{n} |b_{ik}|^2} = \sqrt{\sum_{i=1}^{n}\sum_{\substack{k=1\\k\neq i}}^{n} \left|\frac{a_{ik}}{a_{ii}}\right|^2} \le L_2 < 1 \tag{5.11}$$

erfüllt, dann konvergiert die durch (5.5) bzw. (5.6) definierte Iterationsfolge mit (5.3) und (5.4) für jeden Startvektor $\boldsymbol{x}^{(0)} \in \mathbb{R}^n$ gegen die eindeutig bestimmte Lösung $\boldsymbol{x}$, und es gilt die Fehlerabschätzung

$$\begin{aligned} f_\infty^{(\nu)} &:= \|\boldsymbol{x}^{(\nu)} - \boldsymbol{x}\|_\infty = \max_{1\le i\le n} |x_i^{(\nu)} - x_i| \\ &\le \tfrac{L_\infty}{1-L_\infty} \max_{1\le i\le n} |x_i^{(\nu)} - x_i^{(\nu-1)}| \qquad \text{(a posteriori)} \\ &\le \tfrac{L_\infty^\nu}{1-L_\infty} \max_{1\le i\le n} |x_i^{(1)} - x_i^{(0)}| \qquad \text{(a priori)} \end{aligned} \tag{5.12}$$

bzw.

$$\begin{aligned} f_1^{(\nu)} &:= \|\boldsymbol{x}^{(\nu)} - \boldsymbol{x}\|_1 = \sum_{i=1}^{n} |x_i^{(\nu)} - x_i| \\ &\le \tfrac{L_1}{1-L_1} \sum_{i=1}^{n} |x_i^{(\nu)} - x_i^{(\nu-1)}| \qquad \text{(a posteriori)} \\ &\le \tfrac{L_1^\nu}{1-L_1} \sum_{i=1}^{n} |x_i^{(1)} - x_i^{(0)}| \qquad \text{(a priori)} \end{aligned} \tag{5.13}$$

bzw.

$$\begin{aligned} f_2^{(\nu)} &:= \|\boldsymbol{x}^{(\nu)} - \boldsymbol{x}\|_2 = \sqrt{\sum_{i=1}^{n} |x_i^{(\nu)} - x_i|^2} \\ &\le \tfrac{L_2}{1-L_2} \sqrt{\sum_{i=1}^{n} |x_i^{(\nu)} - x_i^{(\nu-1)}|^2} \qquad \text{(a posteriori)} \\ &\le \tfrac{L_2^\nu}{1-L_2} \sqrt{\sum_{i=1}^{n} |x_i^{(1)} - x_i^{(0)}|^2} \qquad \text{(a priori)} \end{aligned} \tag{5.14}$$

Beweis.
Mit $\boldsymbol{\varphi}(\boldsymbol{x}) := \boldsymbol{B}\boldsymbol{x} + \boldsymbol{c}$ und der Verträglichkeitsbedingung (Definition 5.3) lautet die Lipschitzbedingung aus Satz 5.6:

$$\|\boldsymbol{\varphi}(\boldsymbol{x}) - \boldsymbol{\varphi}(\widetilde{\boldsymbol{x}})\| = \|\boldsymbol{B}(\boldsymbol{x} - \widetilde{\boldsymbol{x}})\| \leq \|\boldsymbol{B}\|\,\|\boldsymbol{x} - \widetilde{\boldsymbol{x}}\| \leq L\|\boldsymbol{x} - \widetilde{\boldsymbol{x}}\|\,;$$

daraus folgt

$$0 \leq \|\boldsymbol{B}\| \leq L < 1.$$

Hiermit und mit (5.3) erhält man unter Verwendung der Zeilensummennorm als hinreichende Konvergenzbedingung das sogenannte *Zeilensummenkriterium:*

$$\|\boldsymbol{B}\|_\infty = \max_{1\leq i\leq n} \sum_{k=1}^{n} |b_{ik}| = \max_{1\leq i\leq n} \sum_{\substack{k=1\\k\neq i}}^{n} \left|\frac{a_{ik}}{a_{ii}}\right| \leq L_\infty < 1.$$

Unter Verwendung der sup-Norm ergeben sich die Fehlerabschätzungen für die ν-te Näherung gemäß Satz 5.6:

$$f_\infty^{(\nu)} = \|\boldsymbol{x}^{(\nu)} - \boldsymbol{x}\|_\infty \leq \frac{L_\infty}{1 - L_\infty}\|\boldsymbol{x}^{(\nu)} - \boldsymbol{x}^{(\nu-1)}\|_\infty \leq \frac{L_\infty^\nu}{1 - L_\infty}\|\boldsymbol{x}^{(1)} - \boldsymbol{x}^{(0)}\|_\infty,$$

d. h. ausgeschrieben

$$\begin{aligned} f_\infty^{(\nu)} = \max_{1\leq i\leq n} |x_i^{(\nu)} - x_i| \quad &\leq \tfrac{L_\infty}{1-L_\infty} \max_{1\leq i\leq n} |x_i^{(\nu)} - x_i^{(\nu-1)}| \qquad \text{a posteriori} \\ &\leq \tfrac{L_\infty^\nu}{1-L_\infty} \max_{1\leq i\leq n} |x_i^{(1)} - x_i^{(0)}| \qquad \text{a priori.} \end{aligned}$$

Unter Verwendung der Spaltensummennorm erhält man das sogenannte *Spaltensummenkriterium:*

$$\|\boldsymbol{B}\|_1 = \max_{1\leq k\leq n} \sum_{\substack{i=1\\i\neq k}}^{n} \left|\frac{a_{ik}}{a_{ii}}\right| \leq L_1 < 1.$$

Mit der Norm der Komponenten-Betragssumme ergeben sich hier als Fehlerabschätzungen für die ν-te Näherung gemäß Satz 5.6:

$$f_1^{(\nu)} = \|\boldsymbol{x}^{(\nu)} - \boldsymbol{x}\|_1 \leq \frac{L_1}{1 - L_1}\|\boldsymbol{x}^{(\nu)} - \boldsymbol{x}^{(\nu-1)}\|_1 \leq \frac{L_1^{(\nu)}}{1 - L_1}\|\boldsymbol{x}^{(1)} - \boldsymbol{x}^{(0)}\|_1\,,$$

bzw.

$$\begin{aligned} f_1^{(\nu)} = \sum_{i=1}^{n} |x_i^{(\nu)} - x_i| \quad &\leq \tfrac{L_1}{1-L_1} \sum_{i=1}^{n} |x_i^{(\nu)} - x_i^{(\nu-1)}| \qquad \text{a posteriori} \\ &\leq \tfrac{L_1^\nu}{1-L_1} \sum_{i=1}^{n} |x_i^{(1)} - x_i^{(0)}| \qquad \text{a priori.} \end{aligned}$$

Dabei muss die Lipschitzkonstante L_1 der Bedingung

$$\max_{1\leq k\leq n} \sum_{\substack{i=1\\i\neq k}}^{n} \left|\frac{a_{ik}}{a_{ii}}\right| \leq L_1 < 1$$

genügen.

Entsprechend erhält man mit der Euklidischen Norm als hinreichende Konvergenzbedingung das *Kriterium von Schmidt-v. Mises:*

$$\sqrt{\sum_{i=1}^{n} \sum_{\substack{k=1 \\ i \neq k}}^{n} \left| \frac{a_{ik}}{a_{ii}} \right|^2} \leq L_2 < 1.$$

Mit der zugeordneten Euklidischen Vektor-Norm erhält man hier als Fehlerabschätzung für die ν-te Näherung:

$$\begin{aligned} f_2^{(\nu)} = \sqrt{\sum_{i=1}^{n} |x_i^{(\nu)} - x_i|^2} \quad &\leq \tfrac{L_2}{1-L_2} \sqrt{\sum_{i=1}^{n} |x_i^{(\nu)} - x_i^{(\nu-1)}|^2} \quad \text{a posteriori} \\ &\leq \tfrac{L_2^{\nu}}{1-L_2} \sqrt{\sum_{i=1}^{n} |x_i^{(1)} - x_i^{(0)}|^2} \quad \text{a priori.} \end{aligned}$$

Für die $f_i^{(\nu)}$, $i = \infty, 1, 2$, gilt die Ungleichung

$$f_\infty^{(\nu)} \leq f_1^{(\nu)} \leq f_2^{(\nu)}.$$

Bemerkung. Im Fall (5.9) konvergiert die Iterationsfolge sogar komponentenweise. □

Algorithmus 5.8. (*Iteration in Gesamtschritten*)

Gegeben: Das lineare Gleichungssystem $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}$ mit

$$\boldsymbol{A} = (a_{ik}), \quad \boldsymbol{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, \quad \boldsymbol{a} = \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix}, \quad i, k = 1(1)n,$$

Gesucht: Seine Lösung $\boldsymbol{x}$ mittels Iteration in Gesamtschritten.

1. Schritt: Das gegebene System wird auf die äquivalente Form (5.4) gebracht mit den Größen (5.3).

2. Schritt: Man prüfe, ob eines der in Satz 5.7 angegebenen hinreichenden Konvergenzkriterien erfüllt ist. Falls nicht, versuche man, durch geeignete Linearkombinationen von Gleichungen ein System mit überwiegenden Diagonalelementen herzustellen, welches einem der Konvergenzkriterien genügt. Ist dies nicht möglich, so berechne man die Lösung mit einer direkten Methode.

3. Schritt: Falls eines der Konvergenzkriterien erfüllt ist, wähle man einen beliebigen Startvektor $\boldsymbol{x}^{(0)}$; o. B. d. A. kann man $\boldsymbol{x}^{(0)} = \boldsymbol{0}$ wählen.

4. Schritt: Man erzeuge eine Iterationsfolge $\{\boldsymbol{x}^{(\nu)}\}$ nach der Vorschrift (5.5) bzw. (5.6). Dazu verwende man zweckmäßig das Rechenschema 5.4. Es wird so lange iteriert, bis eine der drei folgenden Abfragen erfüllt ist:

a) *Abfrage auf den absoluten Fehler:*

$$\max_{1\le i\le n} |x_i^{(\nu+1)} - x_i^{(\nu)}| < \delta, \quad \delta > 0 \quad \text{vorgegeben.}$$

b) *Abfrage auf den relativen Fehler:*

$$\max_{1\le i\le n} |x_i^{(\nu+1)} - x_i^{(\nu)}| \le \max_{1\le i\le n} |x_i^{(\nu+1)}|\varepsilon \text{ zu vorgegebenem } \varepsilon > 0.$$

c) $\nu > \nu_0$, ν_0 vorgegebene Zahl, die etwa aus einer a priori-Fehlerabschätzung ermittelt wurde.

5. Schritt: (Fehlerabschätzung). Falls (5.9) erfüllt ist, wird die Fehlerabschätzung (5.12) verwendet. Ist (5.9) nicht erfüllt, sondern (5.10), so wird die Fehlerabschätzung (5.13) verwendet. Ist nur (5.11) erfüllt, so kann nur die gröbste Fehlerabschätzung (5.14) benutzt werden.

Die Abfrage a) im 4. Schritt des Algorithmus 5.8 ist praktisch einem Konvergenznachweis gleichzusetzen; denn für $0 \le L_\infty < 1$ und hinreichend großes ν kann 4a) immer erfüllt werden wegen

$$\begin{aligned}
\max_{1\le i\le n} |x_i^{(\nu+1)} - x_i^{(\nu)}| &= \|\boldsymbol{x}^{(\nu+1)} - \boldsymbol{x}^{(\nu)}\|_\infty \\
&= \|\boldsymbol{x}^{(\nu+1)} - \boldsymbol{x} + \boldsymbol{x} - \boldsymbol{x}^{(\nu)}\|_\infty \le \|\boldsymbol{x}^{(\nu+1)} - \boldsymbol{x}\|_\infty + \|\boldsymbol{x}^{(\nu)} - \boldsymbol{x}\|_\infty \\
&\le L_\infty^{\nu+1}\|\boldsymbol{x}^{(0)} - \boldsymbol{x}\|_\infty + L_\infty^{\nu}\|\boldsymbol{x}^{(0)} - \boldsymbol{x}\|_\infty \\
&= L_\infty^{\nu}(1 + L_\infty)\|\boldsymbol{x}^{(0)} - \boldsymbol{x}\|_\infty < 2L_\infty^{\nu}\|\boldsymbol{x}^{(0)} - \boldsymbol{x}\|_\infty < \varepsilon
\end{aligned}$$

unter Verwendung der sup-Norm.

Um für die Abfrage c) im 4. Schritt des Algorithmus 5.8 den Index ν_0 zu erhalten, gibt man für $\|\boldsymbol{x}^{(\nu_0)} - \boldsymbol{x}\|$ eine Schranke α vor und bestimmt ν_0 mittels der a priori-Fehlerabschätzung des Satzes 5.6

$$\|\boldsymbol{x}^{(\nu_0)} - \boldsymbol{x}\| \le \frac{L^{\nu_0}}{1-L}\|\boldsymbol{x}^{(1)} - \boldsymbol{x}^{(0)}\| \le \alpha\,.$$

Beispiel 5.9. (Fortsetzung von Beispiel 5.5)

Gegeben: Das lineare Gleichungssystem (5.7):

$$\left\{\begin{array}{rcrcrcl} 10x_1 & + & 2x_2 & + & x_3 & = & 13, \\ x_1 & + & 10x_2 & + & 2x_3 & = & 13, \\ 2x_1 & + & x_2 & + & 10x_3 & = & 13. \end{array}\right.$$

Gesucht: Eine Näherungslösung für $\boldsymbol{x}$ mit Hilfe des Iterationsverfahrens in Gesamtschritten.

Lösung: Die Vorgehensweise erfolgt nach Algorithmus 5.8.

1. Schritt: Das System (5.7) liegt bereits in der äquivalenten Form (5.8) vor.

2. Schritt: Mit dem Zeilensummenkriterium erhält man

$$\max_{1\le i\le 3}\sum_{k=1}^{n}\left|\frac{a_{ik}}{a_{ii}}\right| = \max_{1\le i\le 3}\sum_{\substack{k=1\\ i\neq k}}^{3}|b_{ik}| = 0.3 \le L_\infty < 1\,;$$

es wird $L_\infty = 0.3$ gesetzt.

3. Schritt: Wahl des Startvektors $\boldsymbol{x}^{(0)^\mathsf{T}} = (0,0,0)$.

4. Schritt: Erzeugung der Iterationsfolge $\{\boldsymbol{x}^{(\nu)}\}$. Bei 6-stelliger Mantisse und Vorgabe von $\varepsilon = 1\cdot 10^{-5}$ erhält man

c_i	b_{ik}			$x_i^{(0)}$	$x_i^{(1)}$	$x_i^{(2)}$	$x_i^{(3)}$	$x_i^{(4)}$
1.3	0	−0.2	−0.1	0	1.3	0.91	1.027	0.9919
1.3	−0.1	0	−0.2	0	1.3	0.91	1.027	0.9919
1.3	−0.2	−0.1	0	0	1.3	0.91	1.027	0.9919

$x_i^{(5)}$	$x_i^{(6)}$	$x_i^{(7)}$	$x_i^{(8)}$	$x_i^{(9)}$	$x_i^{(10)}$	$x_i^{(11)}$
1.00243	0.999271	1.00022	0.999934	1.00002	0.99994	1.00000
1.00243	0.999271	1.00022	0.999934	1.00002	0.99994	1.00000
1.00243	0.999271	1.00022	0.999934	1.00002	0.99994	1.00000

5. Schritt: Fehlerabschätzung

$$\begin{aligned}\|\boldsymbol{x}^{(11)} - \boldsymbol{x}\|_\infty &= \max_{1\le i\le 3}|x_i^{(11)} - x_i| \le \tfrac{L_\infty}{1-L_\infty}\max_{1\le i\le 3}|x_i^{(11)} - x_i^{(10)}|\\ &= \tfrac{0.3}{1-0.3}\cdot 6\cdot 10^{-5} \le 2.6\cdot 10^{-5}.\end{aligned}$$

Für die a priori-Fehlerabschätzung folgt bei Vorgabe von $\alpha = 10^{-5}$

$$f_\infty^{(\nu_0)} \le \frac{L_\infty^{\nu_0}}{1-L_\infty}\max_{1\le i\le 3}|x_i^{(1)} - x_i^{(0)}| < \alpha, \quad \text{d. h.} \frac{(0.3)^{\nu_0}}{1-0.3}\cdot 1.3 < 10^{-5}$$

Daraus ergibt sich $\nu_0 \ge 11$. □

5.4 Das Gauß-Seidelsche Iterationsverfahren, Iteration in Einzelschritten

Das Gauß-Seidelsche Iterationsverfahren unterscheidet sich vom Iterationsverfahren in Gesamtschritten nur dadurch, dass zur Berechnung der $(\nu+1)$-ten Näherung von x_i die bereits berechneten $(\nu+1)$-ten Näherungen von $x_1,\ x_2,\ ...,\ x_{i-1}$ berücksichtigt werden. Hat man das gegebene Gleichungssystem (5.1) auf die äquivalente Form (5.4) mit (5.3) gebracht, so lautet hier die Iterationsvorschrift

$$\left\{\begin{array}{l} \boldsymbol{x}^{(\nu+1)} = \boldsymbol{B}_r\boldsymbol{x}^{(\nu)} + \boldsymbol{B}_\ell\boldsymbol{x}^{(\nu+1)} + \boldsymbol{c} \quad \text{mit} \\ \\ \boldsymbol{B}_r = \begin{pmatrix} 0 & b_{12} & b_{13} & \cdots & b_{1n} \\ 0 & 0 & \cdot & & \vdots \\ \vdots & \vdots & \cdot & \cdot & \cdot \\ \cdot & \cdot & & \cdot & b_{n-1,n} \\ 0 & 0 & \cdot & \cdots & 0 \end{pmatrix}, \quad \boldsymbol{B}_\ell = \begin{pmatrix} 0 & \cdot & \cdot & \cdot & 0 \\ b_{21} & \cdot & & & \vdots \\ \vdots & \cdot & \cdot & & \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ b_{n1} & b_{n2} & \cdots & b_{n,n-1} & 0 \end{pmatrix} \end{array}\right. \tag{5.15}$$

bzw. in Komponenten geschrieben für $i = 1(1)n$, $\nu = 0,1,2,\ldots$

$$\begin{aligned} x_i^{(\nu+1)} &= c_i + \sum_{k=i+1}^{n} b_{ik}x_k^{(\nu)} + \sum_{k=1}^{i-1} b_{ik}x_k^{(\nu+1)} \\ &= \frac{a_i}{a_{ii}} - \sum_{k=i+1}^{n} \frac{a_{ik}}{a_{ii}}x_k^{(\nu)} - \sum_{k=1}^{i-1} \frac{a_{ik}}{a_{ii}}x_k^{(\nu+1)}. \end{aligned} \tag{5.16}$$

Hinreichende Konvergenzkriterien für das Iterationsverfahren in Einzelschritten sind:

1. das Zeilensummenkriterium (5.9);
2. das Spaltensummenkriterium (5.10);
3. ist $\boldsymbol{A}$ symmetrisch ($a_{ik} = a_{ki}$) und positiv definit ($\boldsymbol{x}^\mathsf{T}\boldsymbol{A}\boldsymbol{x} > 0$ für $\boldsymbol{x} \neq \boldsymbol{0}$), so konvergiert das Verfahren.

Jedes System $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}$ lässt sich mit Hilfe der Gaußschen Transformation so umformen, dass das 3. Konvergenzkriterium erfüllt ist. Dazu wird $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}$ von links mit $\boldsymbol{A}^\mathsf{T}$ multipliziert, so dass sich ergibt: $\boldsymbol{A}^\mathsf{T}\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^\mathsf{T}\boldsymbol{a}$.
Falls das Iterationsverfahren jedoch bereits für das ursprüngliche System konvergiert hätte und dies lediglich nicht nachprüfbar war, wird durch die Gaußsche Transformation die Konvergenz verschlechtert. Die Zahl der erforderlichen Punktoperationen bei Anwendung der Gaußschen Transformation und anschließenden Iteration ist größer als die Zahl der Punktoperationen beim Gaußschen Algorithmus. Es lohnt sich also nicht, diese Transformation durchzuführen.

Die Rechnung wird zweckmäßig in einem Rechenschema der folgenden Form durchgeführt:

Rechenschema 5.10. (*Iterationsverfahren in Einzelschritten für $n = 3$*)

c_i	b_{ik} für $k \geq i$			b_{ik} für $k < i$			$x_i^{(0)}$	$x_i^{(1)}$	$\cdots$	
$\frac{a_1}{a_{11}}$	0	$-\frac{a_{12}}{a_{11}}$	$-\frac{a_{13}}{a_{11}}$	·	·	·	0			
$\frac{a_2}{a_{22}}$	·	0	$-\frac{a_{23}}{a_{22}}$	$-\frac{a_{21}}{a_{22}}$	·	·	0			
$\frac{a_3}{a_{33}}$	·	·	0	$-\frac{a_{31}}{a_{33}}$	$-\frac{a_{32}}{a_{33}}$	·	0			

Hier wird kein eigener Algorithmus formuliert, weil der Wortlaut völlig mit dem des Algorithmus 5.8 übereinstimmen würde; hier kommen lediglich für den 2. Schritt andere Konvergenzkriterien in Frage, und im 4. Schritt lautet die Iterationsvorschrift (5.15) bzw. (5.16).

Beispiel 5.11. (Fortsetzung von Beispiel 5.9)

Gegeben: Das Gleichungssystem (5.7).

Gesucht: Eine Näherungslösung mit Hilfe des Gauß-Seidelschen Iterationsverfahrens (Rundung auf 6-stellige Mantisse).

Lösung: Die Iterationsvorschrift lautet für $\nu = 0,1,2,\ldots$

$$\left\{\begin{array}{lcl} x_1^{(\nu+1)} & = & 1.3 - 0.2x_2^{(\nu)} - 0.1x_3^{(\nu)}\,, \\ x_2^{(\nu+1)} & = & 1.3 - 0.1x_1^{(\nu+1)} - 0.2x_3^{(\nu)}\,, \\ x_3^{(\nu+1)} & = & 1.3 - 0.2x_1^{(\nu+1)} - 0.1x_2^{(\nu+1)}\,, \end{array}\right.$$

c_i	b_{ik} für $k \geq i$			b_{ik} für $k < i$			$x_i^{(0)}$	$x_i^{(1)}$	$x_i^{(2)}$	$x_i^{(3)}$	$x_i^{(4)}$	$x_i^{(5)}$	$x_i^{(6)}=x_i^{(7)}$
1.3	0	−0.2	−0.1	·	·	·	0	1.3	0.973700	0.996048	0.999977	1.00003	1.00000
1.3	·	0	−0.2	−0.1	·	·	0	1.17	1.01803	0.999704	0.999838	0.999993	1.00000
1.3	·	·	0	−0.2	−0.1	·	0	0.923	1.00346	1.00082	1.00002	0.999995	1.00000

Fehlerabschätzung:

$$\begin{aligned} \|\boldsymbol{x}^{(6)} - \boldsymbol{x}\|_\infty &= \max_{1\leq i\leq 3} |x_i^{(6)} - x_i| \leq \tfrac{L_\infty}{1-L_\infty}\|x^{(6)} - x^{(5)}\|_\infty \\ &= \tfrac{L_\infty}{1-L_\infty} \max_{1\leq i\leq 3} |x_i^{(6)} - x_i^{(5)}| \leq \tfrac{0.3}{0.7} \cdot 3 \cdot 10^{-5} \leq 0.13 \cdot 10^{-4}. \end{aligned}$$

□

5.5 Relaxation beim Gesamtschrittverfahren

Beim Gesamtschrittverfahren erfolgt die Iteration nach der Vorschrift

$$\boldsymbol{x}^{(\nu+1)} = \boldsymbol{c} + \boldsymbol{B}\boldsymbol{x}^{(\nu)}, \qquad \nu = 0, 1, 2, \ldots \tag{5.17}$$

mit der Iterationsmatrix $\boldsymbol{B}$ bzw. umgeformt nach der Vorschrift

$$\boldsymbol{x}^{(\nu+1)} = \boldsymbol{x}^{(\nu)} + \boldsymbol{z}^{(\nu)} \tag{5.18}$$

mit

$$\left\{ \begin{array}{lcl} \boldsymbol{z}^{(\nu)} & = & \boldsymbol{c} - \boldsymbol{B}^* \boldsymbol{x}^{(\nu)}, \\ \boldsymbol{B}^* & = & \boldsymbol{E} - \boldsymbol{B}; \end{array} \right. \tag{5.19}$$

$\boldsymbol{z}^{(\nu)}$ heißt *Korrekturvektor*. Man versucht nun, den Wert $\boldsymbol{x}^{(\nu)}$ durch $\omega \boldsymbol{z}^{(\nu)}$ statt durch $\boldsymbol{z}^{(\nu)}$ zu verbessern; ω heißt *Relaxationskoeffizient*. Das Iterationsverfahren (5.18) erhält so die Form

$$\boldsymbol{x}^{(\nu+1)} = \boldsymbol{x}^{(\nu)} + \omega(\boldsymbol{c} - \boldsymbol{B}^* \boldsymbol{x}^{(\nu)}), \tag{5.20}$$

ω ist so zu wählen, dass die Konvergenzgeschwindigkeit gegenüber der des Gesamtschrittverfahrens erhöht wird.

Besitzt nun die Iterationsmatrix $\boldsymbol{B}$ des Gesamtschrittverfahrens (5.17) die reellen Eigenwerte

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \quad \text{mit} \quad \lambda_1 \neq -\lambda_n,$$

so ist mit dem Relaxationskoeffizienten

$$\omega = \frac{2}{2 - \lambda_1 - \lambda_n}$$

die Konvergenz des Relaxationsverfahrens (5.20) mit (5.19) besser als die des Gesamtschrittverfahrens (Beweis s. [WERN1993], S.188ff.).

Im Falle $\omega < 1$ spricht man von *Unterrelaxation*; für $\omega > 1$ von *Überrelaxation*. Zur Durchführung der Relaxation benötigt man scharfe Schranken für die Eigenwerte von $\boldsymbol{B}$, die das Vorzeichen berücksichtigen. Verfahren zur näherungsweisen Bestimmung der Eigenwerte sind in Kapitel 7 angegeben.

5.6 Relaxation beim Einzelschrittverfahren – SOR-Verfahren

Die Iterationsvorschrift für das Einzelschrittverfahren lautet

$$\boldsymbol{x}^{(\nu+1)} = \boldsymbol{c} + \boldsymbol{B}_r \boldsymbol{x}^{(\nu)} + \boldsymbol{B}_\ell \boldsymbol{x}^{(\nu+1)}, \quad \nu = 0, 1, 2, \ldots$$

bzw. umgeformt

$$\left\{\begin{array}{lcl} \boldsymbol{x}^{(\nu+1)} & = & \boldsymbol{x}^{(\nu)} + \boldsymbol{z}^{(\nu)} \quad \text{mit} \\ \boldsymbol{z}^{(\nu)} & = & \boldsymbol{c} + \boldsymbol{B}_\ell \boldsymbol{x}^{(\nu+1)} - (\boldsymbol{E} - \boldsymbol{B}_r)\boldsymbol{x}^{(\nu)}. \end{array}\right. \tag{5.21}$$

Ersetzt man nun in (5.21) analog zu Abschnitt 5.5 den Korrekturvektor $\boldsymbol{z}^{(\nu)}$ durch $\omega \boldsymbol{z}^{(\nu)}$ mit dem Relaxationskoeffizienten ω, so erhält man als Iterationsvorschrift für das Verfahren der sukzessiven Relaxation

$$\boldsymbol{x}^{(\nu+1)} = \boldsymbol{x}^{(\nu)} + \omega(\boldsymbol{c} + \boldsymbol{B}_\ell \boldsymbol{x}^{(\nu+1)} - (\boldsymbol{E} - \boldsymbol{B}_r)\boldsymbol{x}^{(\nu)}). \tag{5.22}$$

Die Berechnung des optimalen Wertes für ω ist schwierig. Es lässt sich zeigen, dass Relaxationsverfahren (5.22) überhaupt nur konvergent sein können für $0 < \omega < 2$ (s. [STOE1989], S.236).

Für ein Gleichungssystem mit symmetrischer, positiv definiter, tridiagonaler bzw. diagonal blockweise tridiagonaler Matrix (d. h. einer tridiagonalen Blockmatrix, deren Diagonalblöcke Diagonalmatrizen sind, siehe Abschnitt 4.15) ist der optimale Überrelaxationsfaktor für das *Verfahren der sukzessiven Überrelaxation (kurz SOR)*

$$\omega_{opt} = \frac{2}{1 + \sqrt{1 - \lambda_1^2}} \ ;$$

λ_1 ist der größte Eigenwert der Matrix $\boldsymbol{B} = \boldsymbol{B}_\ell + \boldsymbol{B}_r$ (s. [SCHW1972], S.60, S.208/210, S.214). Solche Matrizen treten bei der Diskretisierung von Randwertaufgaben vom elliptischen Typ auf. SOR mit ω_{opt} konvergiert hier erheblich rascher als die Relaxation beim Gesamtschrittverfahren.

Für Gleichungssysteme mit symmetrischer, aber nicht diagonal blockweise tridiagonaler Matrix sowie mit schiefsymmetrischer Matrix wird in [NIET1970] eine günstige Näherung für ω angegeben.

Im Falle dünn besetzter Matrizen wird auf [WEIS1990] hingewiesen.

5.6.1 Schätzung des Relaxationskoeffizienten – Adaptives SOR-Verfahren

In [BUNS1995] ist ein adaptives Verfahren zur Berechnung des optimalen Relaxationskoeffizienten w_{opt} angegeben, welches begleitend zur Anwendung des Gauß-Seidel-Verfahrens mit Relaxation den Relaxationskoeffizienten ermittelt. Zur Beschleunigung des Verfahrens werden vorher einige Schritte (die Anzahl j ist frei wählbar, $j \geq 1$) jeweils neu mit dem Einzelschrittverfahren mit festem Schätzwert für den Relaxationskoeffizienten w durchgeführt, dann wird die Schätzung des Relaxationskoeffizienten neu angepasst (vgl. [BUNS1995]).

Algorithmus 5.12.

Gegeben: $\boldsymbol{Ax} = \boldsymbol{a}$, $\boldsymbol{A}$ erfülle die Voraussetzungen für die Anwendung des Gauß-Seidel-Verfahrens; alle Eigenwerte der Iterationsmatrix $\boldsymbol{B}$ seien reell.

Gesucht: Näherungslösung für $\boldsymbol{x}$.

Setze: $w := 1, q := 1, v := 0$

Wähle: Genauigkeitsschranke $\varepsilon \in \mathbb{R}$, $\varepsilon > 0$, Häufigkeit j der Anpassung des Relaxationsparameters $j \geq 1, j \in \mathbb{N}$, Startvektor $\boldsymbol{x}^{(0)}$

Für jedes $v = 0, 1, 2, \ldots$ wird wie folgt verfahren:

1. Berechnung von $\boldsymbol{x}^{(\nu+1)}$ nach der Vorschrift (5.22). Falls v ein ganzzahliges Vielfaches von j ist, wird mit dem 2. Schritt fortgesetzt, andernfalls mit dem 3. Schritt.
2. Zur Anpassung der Schätzung des Relaxationskoeffizienten wird berechnet

$$q := \max_k \frac{|x_k^{(\nu+1)} - x_k^{(\nu)}|}{|x_k^{(\nu)} - x_k^{(\nu-1)}|}$$

   Falls $q > 1$ ist, wird v um 1 erhöht und mit 1. fortgesetzt, andernfalls wird mit

$$q := \max(q, w - 1)$$

   eine neue Anpassung für den Relaxationskoeffizienten berechnet:

$$w := \frac{2}{1 + \sqrt{1 - \frac{1}{q}\left(\frac{q+w-1}{w}\right)^2}}$$

3. Falls gilt

$$\|\boldsymbol{x}^{(\nu+1)} - \boldsymbol{x}^{(\nu)}\|_\infty \leq \varepsilon \|\boldsymbol{x}^{(\nu+1)}\|_\infty ,$$

   wird mit dem Ergebnis $\boldsymbol{x} \approx \boldsymbol{x}^{(\nu+1)}$ die Rechnung abgebrochen, andernfalls wird mit $v := v + 1$ mit dem 1. Schritt fortgesetzt.

Beispiel 5.13.

Gegeben: Das Gleichungssystem

$$\begin{pmatrix} 4 & -1 & -1 & 0 \\ -1 & 4 & 0 & -1 \\ -1 & 0 & 4 & -1 \\ 0 & -1 & -1 & 4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1000 \\ 1000 \end{pmatrix}$$

Gesucht: Anzahl der Iterationsschritte bei den Verfahren: adaptives SOR-Verfahren, SOR-Verfahren mit festen Relaxationskoeffizienten und Gauß-Seidel-Verfahren mit den vier verschiedenen Konvergenzkriterien: kein Konvergenzkriterium,

Zeilensummenkriterium, Spaltensummenkriterium, Kriterum von Schmidt-v. Mises. Die Genauigkeit soll sein: $\varepsilon < 10^{-8}$, außerdem soll der Relaxationskoeffizient erst wieder nach vier Iterationsschritten angepasst werden. Als Startvektor wird der Nullvektor gewählt.

Lösung:

Anzahl der Iterationsschritte

	kein Konvergenz-kriterium	Zeilen-summen-kriterium	Spalten-summen-kriterium	Kriterium von Schmidt-v. Mises
adaptives SOR-Verfahren	13	13	13	13
SOR-Verfahren mit festen Relaxations-koeffizienten	10	10	10	10
Gauß-Seidel-Verfahren	17	17	17	17

□

Ergänzende Literatur zu Kapitel 5

[GOLU1996], 10.; [GRAM2000], Kap.5; [HAMM1994], 2.; [MAES1985], 2.7; [NIEM1987], 6.5; [PREU2001], Kap.3; [QUAR2001] Bd.1, 4; [RALS1979], III.; [RICE1993], 6.3; [SCHE1989]; [STOE1990], 8.3; [TORN1990] Bd.1, 6; [WERN1993], III §5; [YOUN2003], 6-8.

Kapitel 6

Systeme nichtlinearer Gleichungen

6.1 Aufgabenstellung und Motivation

Nichtlineare Gleichungssysteme sind einerseits eine Verallgemeinerung der linearen Gleichungssysteme (siehe Kapitel 4), andererseits der nichtlinearen Gleichungen (Kapitel 2). Es handelt sich hier um n nichtlineare Gleichungen, die zu einem System zusammengefügt sind, sich aber nicht in der Form $\boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{a}$ linearer Systeme formulieren lassen. Im Gegensatz zu den linearen Systemen, die eine, unendlich viele oder keine Lösung haben, besitzen nichtlineare Systeme keine, eine, mehrere oder unendlich viele Lösungen.

Ein Beispiel für ein nichtlineares System ist

$$\begin{aligned} f_1(x_1, x_2) &= x_1 - 2\,\ln x_2 = 0 \\ f_2(x_1, x_2) &= -x_1^2 + 2.5x_1 - 5x_2 + 20 = 0 \end{aligned}$$

Verallgemeinert man diese Schreibweise, so erhält man ein System aus n nichtlinearen Gleichungen ($n \in \mathbb{N}, n \geq 2$)

$$\left\{ \begin{array}{l} f_1(x_1, x_2, \ldots, x_n) = 0, \\ f_2(x_1, x_2, \ldots, x_n) = 0, \\ \vdots \\ f_n(x_1, x_2, \ldots, x_n) = 0. \end{array} \right. \tag{6.1}$$

D_f sei ein endlicher, abgeschlossener Bereich des $\mathbb{R}^n$, auf dem die Funktionen $f_i(x_1, x_2, ..., x_n)$, $i = 1(1)n$, definiert sind, die f_i seien stetig und reellwertig.

Mit

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \qquad \text{und} \qquad \boldsymbol{f} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{pmatrix}$$

lässt sich (6.1) ausdrücken durch

$$\boldsymbol{f} : D_f \subset \mathbb{R}^n \to \mathbb{R}^n, \quad \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}. \tag{6.2}$$

In dieser Schreibweise lässt sich obiges Beispiel so darstellen

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{pmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{pmatrix} = \begin{pmatrix} x_1 - 2\ln x_2 \\ -x_1^2 + 2.5x_1 - 5x_2 + 20 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \boldsymbol{0}.$$

Gesucht sind nun Lösungen $\overline{\boldsymbol{x}} \in D_f$ von (6.1) mit $\boldsymbol{f}(\overline{\boldsymbol{x}}) = \boldsymbol{0}$.

Beispiel 6.1.

Gegeben: Zwei nichtlineare Gleichungen

$$\begin{cases} f_1(x_1, x_2) & = x_1^2 + x_2 - 11 & = 0, \\ f_2(x_1, x_2) & = x_1 + x_2^2 - 7 & = 0. \end{cases} \tag{6.3}$$

Gesucht: Die Lösungen des Gleichungssystems.

Lösung: In der Form (6.2) lautet (6.3)

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{pmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{pmatrix} = \begin{pmatrix} x_1^2 & + & x_2 & - & 11 \\ x_1 & + & x_2^2 & - & 7 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \boldsymbol{0}$$

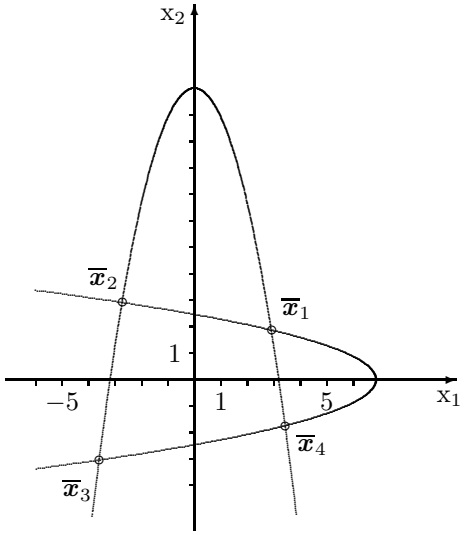


Abb. 6.1.

Abbildung 6.1 zeigt die durch $f_1(x_1, x_2) = 0$ und $f_2(x_1, x_2) = 0$ implizit dargestellten Kurven. Es zeigt sich, dass das System (6.3) vier reelle Lösungen $\overline{\boldsymbol{x}}_k$, $k = 1(1)4$, besitzt. Ausgangsnäherungen $\boldsymbol{x}_i^{(0)}$, $i = 1(1)4$, für diese Lösungen lassen sich aus Abb. 6.1 ablesen:

$$\boldsymbol{x}_1^{(0)} = \begin{pmatrix} 3 \\ 2 \end{pmatrix}, \quad \boldsymbol{x}_2^{(0)} = \begin{pmatrix} -2.8 \\ 3.2 \end{pmatrix},$$

$$\boldsymbol{x}_3^{(0)} = \begin{pmatrix} -3.8 \\ -3.3 \end{pmatrix}, \quad \boldsymbol{x}_4^{(0)} = \begin{pmatrix} 3.4 \\ -1.7 \end{pmatrix}.$$

□

Beispiel 6.2.

Gegeben: Ein *unsymmetrischer Stabzweischlag.*

Gesucht: Für das in Abbildung 6.2 dargestellte System soll die statische Gleichgewichtslage nach Aufbringen einer Kraft F ermittelt werden. Zu berechnen sind die Koordinaten x_g, y_g des Gelenkpunktes für das angegebene Koordinatensystem.

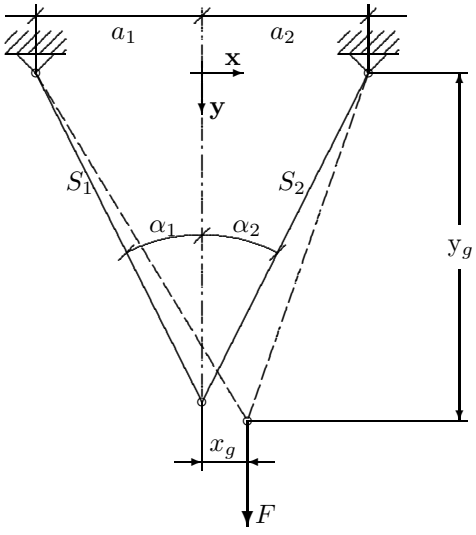


Abb. 6.2. Unsymmetrischer Stabzweischlag

Lösung:

Eingabeparameter:

Stab S_1 :	E_1	=	100 GPa ,	$d_1 = 10$ mm ,	$l_1 = 2$ m ,	$\alpha_1 = 30°$
Stab S_2 :	E_2	=	200 GPa ,	$d_2 = 10$ mm ,	$l_2 = 2$ m ,	$\alpha_2 = 30°$
Kraft :	F	=	10 k N			(1 GPa = 10^9 N/m^2)

Ausgabeparameter: x_g, y_g

Zur Ermittlung der Koordinaten x_g, y_g werden die Gleichgewichtsbedingungen für den Gelenkpunkt

$$\sum F_{ix} = 0, \qquad \sum F_{iy} = 0$$

unter Berücksichtigung der Beziehungen

$$\Delta l_i = \frac{F_i l_i}{E_i A_i} = l_{b_i} - l_i, \quad i = 1, 2, \tag{6.4}$$

aufgestellt, wobei l_{b_i} die Länge des belasteten Stabes S_i und A_i die Querschnittsfläche darstellt.

Man erhält

$$\left\{\begin{array}{lrcl} \sum F_{ix} = - & F_1 \cos\overline{\alpha}_1 + & F_2 \cos\overline{\alpha}_2 & = 0, \\ \sum F_{iy} = & F_1 \sin\overline{\alpha}_1 + & F_2 \sin\overline{\alpha}_2 - F & = 0. \end{array}\right. \tag{6.5}$$

Mit den geometrischen Beziehungen

$$\begin{array}{rcl} a_1 = a_2 = l_{1/2} \cdot \sin 30^\circ & = & 2 \cdot \dfrac{1}{2} = 1, \\ l_{b_1} = \sqrt{y_g^2 + (a_1 + x_g)^2} \quad , & & l_{b_2} = \sqrt{y_g^2 + (a_2 - x_g)^2}, \\ \cos\overline{\alpha}_1 = (a_1 + x_g)/l_{b_1} \quad , & & \sin\overline{\alpha}_1 = y_g/l_{b_1}, \\ \cos\overline{\alpha}_2 = (a_2 - x_g)/l_{b_2} \quad , & & \sin\overline{\alpha}_2 = y_g/l_{b_2} \end{array}$$

und den aus (6.4) folgenden Gleichungen

$$\begin{array}{rcl} F_1 & = & (l_{b_1} - l_1) E_1 A_1 / l_1, \\ F_2 & = & (l_{b_2} - l_2) E_2 A_2 / l_2 \end{array}$$

erhält man schließlich nach Einsetzen in (6.5) das nichtlineare Gleichungssystem für x_g, y_g:

$$\left\{\begin{array}{llcl} E_1 A_1 (1 + x_g) \cdot z_1 - & E_2 A_2 (1 - x_g) \cdot z_2 & = & 0, \\ E_1 A_1 y_g \cdot z_1 + & E_2 A_2 y_g \cdot z_2 - F & = & 0, \\ \text{mit} \quad z_1 & := \frac{1}{2} - \left[(1 + x_g)^2 + y_g^2\right]^{-1/2}, & & \\ z_2 & := \frac{1}{2} - \left[(1 - x_g)^2 + y_g^2\right]^{-1/2}. & & \end{array}\right.$$

Mit dem Startvektor $\boldsymbol{x}^{(0)} = (0.0, 1.0)^\mathsf{T}$ und wegen der Querschnittsflächen $A_1 = A_2 = \frac{d^2}{4}\pi = \frac{1}{4} \cdot 10^{-4} \cdot \pi[\mathrm{m}^2]$ erhält man schließlich nach 8 Iterationen und Rundung auf 5 Dezimalstellen mit dem Verfahren von Brown (s. Abschnitt 6.3.4) das Ergebnis:

$$\begin{array}{rcr} x_g & = & 0.73416 \ \text{mm}, \\ y_g & = & 1733.32335 \ \text{mm}. \end{array}$$

□

6.2 Allgemeines Iterationsverfahren für Systeme

Zu dem nichtlinearen System $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}$ wird ein äquivalentes System

$$\boldsymbol{\varphi} : D_\varphi \subseteq D_f \subset \mathbb{R}^n \to \mathbb{R}^n, \quad \boldsymbol{x} = \boldsymbol{\varphi}(\boldsymbol{x}); \quad \boldsymbol{\varphi}(\boldsymbol{x}) = \big(\varphi_1(x), \varphi_2(x), \ldots, \varphi_n(x)\big)^\mathsf{T}$$

erzeugt. $\overline{\boldsymbol{x}}$ heißt *Fixpunkt* von $\boldsymbol{\varphi}$ bzw. Lösung von $\boldsymbol{x} = \boldsymbol{\varphi}(\boldsymbol{x})$, falls gilt $\overline{\boldsymbol{x}} = \boldsymbol{\varphi}(\overline{\boldsymbol{x}})$. Mit Hilfe eines Startvektors $\boldsymbol{x}^{(0)} \in D_\varphi$ wird eine Iterationsfolge $\{\boldsymbol{x}^{(\nu)}\}$ konstruiert nach der Vorschrift

$$\boldsymbol{x}^{(\nu+1)} = \boldsymbol{\varphi}(\boldsymbol{x}^{(\nu)}), \qquad \nu = 0, 1, 2, \ldots \tag{6.6}$$

$\boldsymbol{\varphi}$ heißt vektorielle *Schrittfunktion*, (6.6) *Iterationsvorschrift*.

Satz 6.3. (*Fixpunktsatz für Systeme*)
Es sei $D \subseteq D_\varphi$ ein endlicher, abgeschlossener Bereich und es gelte

(i) $\boldsymbol{\varphi}(\boldsymbol{x}) \in D$ für alle $\boldsymbol{x} \in D$, d. h. $\boldsymbol{\varphi}$ ist eine Abbildung von D in sich.

(ii) Es gibt eine Konstante L mit $0 \leq L < 1$ und eine Norm $\|\cdot\|$, so dass für alle $\boldsymbol{x}, \boldsymbol{x}' \in D$ die Lipschitzbedingung erfüllt ist

$$\|\boldsymbol{\varphi}(\boldsymbol{x}) - \boldsymbol{\varphi}(\boldsymbol{x}')\| \leq L\|\boldsymbol{x} - \boldsymbol{x}'\|. \tag{6.7}$$

Dann gilt:

(a) Es gibt genau einen Fixpunkt $\overline{\boldsymbol{x}}$ in D .

(b) Die Iteration (6.6) konvergiert für jeden Startwert $\boldsymbol{x}^{(0)} \in D$ gegen $\overline{\boldsymbol{x}}$.

(c) Es gelten die Fehlerabschätzungen

$$\begin{aligned} \|\boldsymbol{x}^{(\nu)} - \overline{\boldsymbol{x}}\| &\leq \tfrac{L^\nu}{1-L}\|\boldsymbol{x}^{(1)} - \boldsymbol{x}^{(0)}\| && \text{(a priori-Fehlerabschätzung),} \\ \|\boldsymbol{x}^{(\nu)} - \overline{\boldsymbol{x}}\| &\leq \tfrac{L}{1-L}\|\boldsymbol{x}^{(\nu)} - \boldsymbol{x}^{(\nu-1)}\| && \text{(a posteriori-Fehlerabschätzung).} \end{aligned}$$

Zum Beweis siehe [HENR1972] Bd. 1, S.131 ff.

Ein Analogon zur Bedingung $|\varphi'| \leq L < 1$ im eindimensionalen Fall lässt sich mit Hilfe der Funktionalmatrix formulieren. Besitzen die φ_i in D stetige partielle Ableitungen nach den x_k, so kann mit der Funktionalmatrix (Jacobi-Matrix)

$$\boldsymbol{J}_\varphi := \left(\frac{\partial \varphi_i}{\partial x_k}\right)_{\substack{i=1(1)n \\ k=1(1)n}} = \begin{pmatrix} \frac{\partial \varphi_1}{\partial x_1} & \frac{\partial \varphi_1}{\partial x_2} & \cdots & \frac{\partial \varphi_1}{\partial x_n} \\ \frac{\partial \varphi_2}{\partial x_1} & \frac{\partial \varphi_2}{\partial x_2} & \cdots & \frac{\partial \varphi_2}{\partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial \varphi_n}{\partial x_1} & \frac{\partial \varphi_n}{\partial x_2} & \cdots & \frac{\partial \varphi_n}{\partial x_n} \end{pmatrix}$$

die Lipschitzbedingung (6.7) in Satz 6.3 ersetzt werden durch

$$\|\boldsymbol{J}_\varphi\| \leq L < 1, \tag{6.8}$$

sofern D konvex und abgeschlossen ist.

Unter Verwendung der verschiedenen Matrixnormen aus Abschnitt 5.2 ergeben sich für (6.8) die folgenden Konvergenzkriterien:

Zeilensummenkriterium:

$$\|\boldsymbol{J}_\varphi\|_\infty = \max_{\substack{i=1(1)n \\ \boldsymbol{x}\in D}} \sum_{k=1}^{n} \left|\frac{\partial \varphi_i}{\partial x_k}\right| \leq L_\infty < 1, \tag{6.9}$$

Spaltensummenkriterium:

$$\|\boldsymbol{J}_\varphi\|_1 = \max_{\substack{k=1(1)n \\ \boldsymbol{x}\in D}} \sum_{i=1}^{n} \left|\frac{\partial \varphi_i}{\partial x_k}\right| \leq L_1 < 1, \tag{6.10}$$

Kriterium von E. Schmidt und R. v. Mises:

$$\|\boldsymbol{J}_\varphi\|_2 = \max_{\boldsymbol{x}\in D} \left(\sum_{i=1}^{n}\sum_{k=1}^{n} \left(\frac{\partial \varphi_i}{\partial x_k}\right)^2\right)^{1/2} \leq L_2 < 1. \tag{6.11}$$

Ist das Kriterium (6.9) erfüllt, so lassen sich unter Verwendung der Maximumnorm $\|\cdot\|_\infty$ gemäß Satz 6.3 die zugehörigen Fehlerabschätzungen angeben; entsprechendes gilt für die Kriterien (6.10) und (6.11).

Definition 6.4. (*Konvergenzordnung*)
Die Iterationsfolge $\{\boldsymbol{x}^{(\nu)}\}$ konvergiert von mindestens p-ter Ordnung gegen $\overline{\boldsymbol{x}}$, wenn eine Konstante $0 \leq M < \infty$ existiert, so dass gilt

$$\lim_{\nu\to\infty} \frac{\|\boldsymbol{x}^{(\nu+1)} - \overline{\boldsymbol{x}}\|}{\|\boldsymbol{x}^{(\nu)} - \overline{\boldsymbol{x}}\|^p} = M < \infty.$$

Das Iterationsverfahren $\boldsymbol{x}^{(\nu+1)} = \boldsymbol{\varphi}(\boldsymbol{x}^{(\nu)})$ heißt dann ein Verfahren von mindestens p-ter Konvergenzordnung; es besitzt genau die Konvergenzordnung p für $M \neq 0$.

Praktikable Fehlerabschätzung ohne Verwendung der Lipschitzkonstante
(s. [KIOU1978]; [KIOU1979]; [MOOR1980])

Die Methode beruht auf einem Satz von Carlo Miranda:

$$\text{Sei} \qquad K := \left\{\boldsymbol{x} \in \mathbb{R}^n | \max_{1\leq i\leq n} |x_i - \widetilde{x}_i| \leq p\right\}$$

ein n-dimensionaler Kubus und seien E_i^+ und E_i^- mit

$$E_i^\pm := \{\boldsymbol{x} \in K | x_i = \widetilde{x}_i \pm p\}, \qquad i = 1(1)n$$

jeweils die Paare von senkrecht zur x_i-Achse liegenden Oberflächenebenen von K. Ist dann $\boldsymbol{f} : K \to \mathbf{R}^n$ stetig und gilt

$$\begin{aligned} & f_i(E_i^+) > 0, && f_i(E_i^-) < 0, && i = 1(1)n \\ \text{bzw.} \quad & f_i(\boldsymbol{x}) > 0, \quad \boldsymbol{x} \in E_i^+; && f_i(\boldsymbol{y}) < 0, \quad \boldsymbol{y} \in E_i^-, && i = 1(1)n, \end{aligned}$$

so gibt es in K eine Nullstelle $\overline{\boldsymbol{x}}$ von $\boldsymbol{f}$.

Bei beliebigen stetigen Abbildungen $\boldsymbol{f}$ hat der Satz wenig praktischen Nutzen. Ist $\boldsymbol{f}$ jedoch differenzierbar, so lässt sich auf der Grundlage des Satzes eine praktikable Fehlerabschätzung ableiten, sofern man mit irgend einem Verfahren zur Lösung nichtlinearer Gleichungssysteme bereits Näherungen $\widetilde{\boldsymbol{x}}$ für die Lösung $\overline{\boldsymbol{x}}$ mit $\boldsymbol{f}(\widetilde{\boldsymbol{x}}) \approx \boldsymbol{0}$ berechnet hat.

Approximiert man dann nämlich $\boldsymbol{f}$ in einer Umgebung von $\widetilde{\boldsymbol{x}}$ mit Hilfe der Taylorentwicklung bis zum linearen Glied, so gilt

$$\boldsymbol{f}(\boldsymbol{x}) \approx \boldsymbol{f}(\widetilde{\boldsymbol{x}}) + \boldsymbol{J}_f(\widetilde{\boldsymbol{x}})(\boldsymbol{x} - \widetilde{\boldsymbol{x}}) \approx \boldsymbol{J}_f(\widetilde{\boldsymbol{x}})(\boldsymbol{x} - \widetilde{\boldsymbol{x}}), \tag{6.12}$$

wobei $\boldsymbol{J}_f(\widetilde{\boldsymbol{x}}) := \left(\frac{\partial f_i(\widetilde{\boldsymbol{x}})}{\partial x_k}\right)_{\substack{i=1(1)n \\ k=1(1)n}}$ die Funktionalmatrix von $\boldsymbol{f}$ im Punkt $\widetilde{\boldsymbol{x}}$ darstellt.

Aus (6.12) folgt

$$\boldsymbol{J}_f^{-1}(\widetilde{\boldsymbol{x}})\, \boldsymbol{f}(\boldsymbol{x}) \approx \boldsymbol{x} - \widetilde{\boldsymbol{x}} \qquad \text{für} \qquad \widetilde{\boldsymbol{x}} \approx \overline{\boldsymbol{x}}. \tag{6.13}$$

Die Abbildung $\boldsymbol{h}(\boldsymbol{x}) := \boldsymbol{x} - \widetilde{\boldsymbol{x}}$ hat aber genau die gewünschte Eigenschaft

$$h_i(E_i^+) = p > 0, \quad h_i(E_i^-) = -p < 0, \qquad i = 1(1)n.$$

Wegen (6.13) ist also zu erwarten, dass jede Abbildung der Form

$$\boldsymbol{g}(\boldsymbol{x}) := \left(\frac{\partial f_i(\widetilde{\boldsymbol{x}})}{\partial x_k}\right)^{-1} \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{J}_f^{-1}(\widetilde{\boldsymbol{x}})\, \boldsymbol{f}(\boldsymbol{x}) \tag{6.14}$$

auch diese Eigenschaft besitzt und sogar

$$g_i(E_i^+) \approx p, \quad g_i(E_i^-) \approx -p, \qquad i = 1(1)n,$$

erfüllt, wenn die Approximation $\boldsymbol{g}(\boldsymbol{x}) \approx \boldsymbol{x} - \widetilde{\boldsymbol{x}}$ genügend gut ist, der betrachtete Kubus K also klein genug. Da $\boldsymbol{g}(\boldsymbol{x})$ dieselben Nullstellen wie $\boldsymbol{f}$ besitzt, besteht nun ein Fehlerabschätzungsverfahren für die Näherungslösung $\widetilde{\boldsymbol{x}}$ von $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}$ in folgenden Einzelschritten:

Algorithmus 6.5. (*Fehlerschätzung ohne Lipschitzkonstante*)

1. Schritt: Näherungsweise Berechnung der Funktionalmatrix $\boldsymbol{J}_f(\widetilde{\boldsymbol{x}})$ im Punkte $\widetilde{\boldsymbol{x}}$.

2. Schritt: Berechnung der Inversen $\boldsymbol{J}_f^{-1}(\widetilde{\boldsymbol{x}})$ zur Funktionalmatrix. Die praktische Erfahrung zeigt, dass die Näherungen des 1. und 2. Schrittes nicht allzu gut sein müssen.

3. Schritt: Konstruktion von $\boldsymbol{g}(\boldsymbol{x})$ gemäß (6.14) und Überprüfung der Bedingungen

$$g_i(\widetilde{\boldsymbol{x}} + p\boldsymbol{u}_i) > 0, \quad g_i(\widetilde{\boldsymbol{x}} - p\boldsymbol{u}_i) < 0, \qquad i = 1(1)n, \tag{6.15}$$

nacheinander für $p := p_k = 0.5 \cdot 10^{-k}$, $k = k_0, k_0 + 1, \ldots$, $k_0 \in \mathbb{N}$, wobei $\boldsymbol{u}_i$ der Einheitsvektor in Richtung x_i ist.

4. Schritt: Ist (6.15) für $p = p_N$, aber nicht mehr für $p = p_{N+1}$ erfüllt, so müsste mit Hilfe von Methoden der Intervallanalysis die Gültigkeit der Bedingungen

$$g_i(\boldsymbol{x}) > 0, \quad \boldsymbol{x} \in E_i^+, \quad g_i(\boldsymbol{y}) < 0, \quad \boldsymbol{y} \in E_i^-, \quad i = 1(1)n, \tag{6.16}$$

für alle Punkte $\boldsymbol{x}, \boldsymbol{y}$ der Ebenen E_i^+, E_i^- geprüft werden (vgl. dazu [MOOR1980]).

Ist (6.16) erfüllt, so gilt $\max\limits_{1 \le i \le n} \|\widetilde{x}_i - \overline{x}_i\| < p_N$. Da das Nachprüfen von (6.16) jedoch recht mühsam ist, kann man sich meistens auch mit dem Schritt 4* zufriedengeben.

4*. Schritt: Es ist zu prüfen, ob die Bedingungen

$$g_i(\widetilde{\boldsymbol{x}} + p\boldsymbol{u}_i) \approx +p, \quad g_i(\widetilde{\boldsymbol{x}} - p\boldsymbol{u}_i) \approx -p, \qquad i = 1(1)n$$

erfüllt sind. Dies ist ein starkes Indiz (aber keine Garantie) für die Richtigkeit der Behauptung, dass für die Lösung $\overline{\boldsymbol{x}}$ gilt:

$$\max_{1 \le i \le n} |\widetilde{x}_i - \overline{x}_i| < p_N.$$

Beispiel 6.6. (aus [MOOR1980])

Gegeben: Ein Gleichungssystem und eine Näherungslösung

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{pmatrix} x_1^2 + x_2^2 - 1 \\ x_1 - x_2^2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \widetilde{\boldsymbol{x}} = \begin{pmatrix} \widetilde{x}_1 \\ \widetilde{x}_2 \end{pmatrix} = \begin{pmatrix} 0.618034 \\ 0.786151 \end{pmatrix}.$$

Gesucht: p_N mit $\max\limits_{1 \le i \le 2} |\widetilde{x}_i - \overline{x}_i| < p_N$, $\overline{\boldsymbol{x}}^{\mathsf{T}} = (\overline{x}_1, \overline{x}_2)$ ist die exakte Lösung.

Lösung: Gesucht wird der Schnittpunkt des Einheitskreises mit der Parabel im ersten Quadranten.

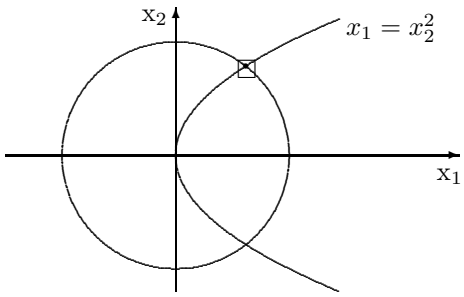


Abb. 6.3. Geometrische Interpretation

1. Schritt:

$$\boldsymbol{J}_f(\widetilde{\boldsymbol{x}}) = \begin{pmatrix} 2\widetilde{x}_1 & 2\widetilde{x}_2 \\ 1 & -2\widetilde{x}_2 \end{pmatrix} = \begin{pmatrix} 1.236068 & 1.572302 \\ 1 & -1.572302 \end{pmatrix}.$$

2. Schritt:

$$\boldsymbol{J}_f^{-1} = \left(\frac{\partial f_i(\widetilde{\boldsymbol{x}})}{\partial x_k}\right)^{-1} = \begin{pmatrix} 0.447214 & 0.447214 \\ 0.284432 & -0.351578 \end{pmatrix}$$

3. Schritt:

$$\begin{aligned} \boldsymbol{J}_f^{-1} \cdot f(\boldsymbol{x}) &= \begin{pmatrix} 0.447214 & 0.447214 \\ 0.284432 & -0.351578 \end{pmatrix} \begin{pmatrix} x_1^2 + x_2^2 - 1 \\ x_1 - x_2^2 \end{pmatrix} \\ &= \begin{pmatrix} 0.447214(x_1^2 + x_2^2 - 1) & + & 0.447214(x_1 - x_2^2) \\ 0.284432(x_1^2 + x_2^2 - 1) & - & 0.351578(x_1 - x_2^2) \end{pmatrix} = \begin{pmatrix} g_1(\boldsymbol{x}) \\ g_2(\boldsymbol{x}) \end{pmatrix}. \end{aligned}$$

Annahme: $p = 10^{-6}$: Dann sind wegen

$$\begin{array}{llllllll} g_1(\widetilde{\boldsymbol{x}} + p\boldsymbol{u}_1) & = & g_1(\widetilde{x}_1 + p, \widetilde{x}_2) & = & 1.011 & \cdot & 10^{-6} & > \; 0 \\ g_1(\widetilde{\boldsymbol{x}} - p\boldsymbol{u}_1) & = & g_1(\widetilde{x}_1 - p, \widetilde{x}_2) & = & -0.988 & \cdot & 10^{-6} & < \; 0 \\ g_2(\widetilde{\boldsymbol{x}} + p\boldsymbol{u}_2) & = & g_2(\widetilde{x}_1, \widetilde{x}_2 + p) & = & 0.622 & \cdot & 10^{-6} & > \; 0 \\ g_2(\widetilde{\boldsymbol{x}} - p\boldsymbol{u}_2) & = & g_2(\widetilde{x}_1, \widetilde{x}_2 - p) & = & -1.337 & \cdot & 10^{-6} & < \; 0 \end{array}$$

die Bedingungen der a priori-Fehlerabschätzung (Satz 6.3) erfüllt.

4*. Schritt: Man sieht, dass die im 3. Schritt berechneten $g_i(\widetilde{\boldsymbol{x}} + p\boldsymbol{u}_i)$ und $g_i(\widetilde{\boldsymbol{x}} - p\boldsymbol{u}_i)$ die Größenordnung von p bzw. $-p$ besitzen, und kann daraus ohne exakten Beweis folgern

$$\max_{1 \le i \le n} |\widetilde{x}_i - \overline{x}_i| < 10^{-6};$$

eigentlich hätte man (um ganz sicher zu sein) nach dem 4. Schritt verfahren müssen.

4. Schritt:

$$\begin{array}{lll} g_1(\widetilde{\boldsymbol{x}} + p\boldsymbol{u}_1) & \in & [\;\; 0.95{\cdot}10^{-6}, \;\; 1.45{\cdot}10^{-6}\;], \\ g_1(\widetilde{\boldsymbol{x}} - p\boldsymbol{u}_1) & \in & [\,-0.62{\cdot}10^{-6}, \;\; -0.56{\cdot}10^{-6}\;], \\ g_2(\widetilde{\boldsymbol{x}} + p\boldsymbol{u}_2) & \in & [\;\; 0.62{\cdot}10^{-6}, \;\; 0.84{\cdot}10^{-6}\;], \\ g_2(\widetilde{\boldsymbol{x}} - p\boldsymbol{u}_2) & \in & [\,-1.85{\cdot}10^{-6}, \;\; -1.37{\cdot}10^{-6}\;], \end{array}$$

so dass die Bedingungen (6.16) erfüllt sind. Für die absoluten Fehler der einzelnen Komponenten gilt dann die Abschätzung

$$\max_{1 \le i \le n} |\widetilde{x}_i - \overline{x}_i| < 10^{-6}.$$

□

6.3 Spezielle Iterationsverfahren

6.3.1 Newtonsche Verfahren für nichtlineare Systeme

6.3.1.1 Das quadratisch konvergente Newton-Verfahren

Es liege ein System (6.1) vor mit einer Lösung $\overline{\boldsymbol{x}}$ im Inneren von D_f. Die f_i sollen in D_f stetige zweite partielle Ableitungen besitzen, und für die Funktionalmatrix (Jakobimatrix)

$$\boldsymbol{J}_f := \left(\frac{\partial f_i}{\partial x_k}\right)_{\substack{i=1(1)n\\k=1(1)n}} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{pmatrix} =: \begin{pmatrix} f_{11} & f_{12} & \cdots & f_{1n} \\ f_{21} & f_{22} & \cdots & f_{2n} \\ \vdots & \vdots & & \vdots \\ f_{n1} & f_{n2} & \cdots & f_{nn} \end{pmatrix}$$

gelte $\det(\boldsymbol{J}_f) \neq 0$. Dann existiert immer eine Umgebung $D \subset D_f$ von $\overline{\boldsymbol{x}}$ so, dass die Voraussetzungen des Satzes 6.3 für die Schrittfunktion des Newton-Verfahrens

$$\boldsymbol{\varphi}(\boldsymbol{x}) := \boldsymbol{x} - \boldsymbol{J}_f^{-1}(\boldsymbol{x})\boldsymbol{f}(\boldsymbol{x})$$

erfüllt sind. Die Iterationsvorschrift lautet

$$\boldsymbol{x}^{(\nu+1)} = \boldsymbol{x}^{(\nu)} - \boldsymbol{J}_f^{-1}(\boldsymbol{x}^{(\nu)})\boldsymbol{f}(\boldsymbol{x}^{(\nu)}) \qquad \text{bzw.} \tag{6.17}$$

$$\boldsymbol{x}^{(\nu+1)} = \boldsymbol{x}^{(\nu)} + \Delta\boldsymbol{x}^{(\nu+1)}, \qquad \nu = 0,1,2,\ldots, \qquad \text{mit} \tag{6.18}$$

$$\Delta\boldsymbol{x}^{(\nu+1)} = -\boldsymbol{J}_f^{-1}(\boldsymbol{x}^{(\nu)})\boldsymbol{f}(\boldsymbol{x}^{(\nu)})\,.$$

Für $n = 2$ ist die erforderliche Berechung der inversen Jakobimatrix unproblematisch (siehe Beispiel 6.8).

Im Allgemeinen wird stattdessen das lineare Gleichungssystem gelöst:

$$\boldsymbol{J}_f(\boldsymbol{x}^{(\nu)})\Delta\boldsymbol{x}^{(\nu+1)} = -\boldsymbol{f}(\boldsymbol{x}^{(\nu)}). \tag{6.19}$$

Algorithmus 6.7.
Für jedes ν = 0,1,2,... sind nacheinander folgende Schritte auszuführen:

(i) Lösung des linearen Gleichungssystems (6.19) zur Berechnung von $\Delta\boldsymbol{x}^{(\nu+1)}$.

(ii) Berechnung von $\boldsymbol{x}^{(\nu+1)}$ gemäß (6.18).
Mögliche Abbruchbedingungen:

(a) $\nu \geq \nu_{\max}, \quad \nu_{\max} \in \mathbb{N}$.
(b) $\|\boldsymbol{x}^{(\nu+1)} - \boldsymbol{x}^{(\nu)}\| \leq \|\boldsymbol{x}^{(\nu+1)}\|\varepsilon_1\,, \quad \varepsilon_1 > 0, \quad \varepsilon_1 \in \mathbb{R}$.
(c) $\|\boldsymbol{x}^{(\nu+1)} - \boldsymbol{x}^{(\nu)}\| \leq \varepsilon_2, \quad \varepsilon_2 > 0, \quad \varepsilon_2 \in \mathbb{R}$.
(d) $\|\boldsymbol{f}(\boldsymbol{x}^{(\nu+1)})\| \leq \varepsilon_3, \quad \varepsilon_3 > 0, \quad \varepsilon_3 \in \mathbb{R}$.

Mit Algorithmus 6.7 wird die Berechnung der Inversen in (6.17) durch die Lösung eines linearen Gleichungssystems ersetzt. Die Konvergenz ist immer gewährleistet, wenn die Iteration nahe genug an der Lösung $\overline{\boldsymbol{x}}$ beginnt.

Beispiel 6.8. (Fortsetzung von Beispiel 6.1)

Gegeben: Das System

$$\boldsymbol{f}(x_1, x_2) = \begin{pmatrix} x_1^2 + x_2 - 11 \\ x_1 + x_2^2 - 7 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Für die Funktionalmatrix $\boldsymbol{J}_f(\boldsymbol{x})$ gilt

$$\boldsymbol{J}_f = \begin{pmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \end{pmatrix} = \begin{pmatrix} 2x_1 & 1 \\ 1 & 2x_2 \end{pmatrix}$$

mit $\det(\boldsymbol{J}_f) = 4x_1x_2 - 1 \neq 0$ für $x_2 \neq \frac{1}{4x_1}$.

I. Näherungsweise Berechnung der Lösung $\overline{\boldsymbol{x}}_4$ im 4. Quadranten (s. Abb. 6.1) mit Hilfe der direkt berechneten inversen Jakobimatrix

$$\boldsymbol{J}_f^{-1} = \frac{1}{f_{11}f_{22} - f_{12}f_{21}} \begin{pmatrix} f_{22} & -f_{12} \\ -f_{21} & f_{11} \end{pmatrix}.$$

Damit lautet die Iterationsvorschrift (6.17) des Newton-Verfahrens für $n = 2$:

$$\begin{aligned} x_1^{(\nu+1)} &= x_1^{(\nu)} - \frac{f_1 f_{22} - f_2 f_{12}}{f_{11}f_{22} - f_{12}f_{21}} \Bigg|_{x_1 = x_1^{(\nu)}, x_2 = x_2^{(\nu)}} \\ x_2^{(\nu+1)} &= x_2^{(\nu)} - \frac{f_2 f_{11} - f_1 f_{21}}{f_{11}f_{22} - f_{12}f_{21}} \Bigg|_{x_1 = x_1^{(\nu)}, x_2 = x_2^{(\nu)}} \end{aligned}$$

$$x_1^{(\nu+1)} = x_1^{(\nu)} - \frac{2(x_1^{(\nu)})^2 x_2^{(\nu)} + (x_2^{(\nu)})^2 - 22x_2^{(\nu)} - x_1^{(\nu)} + 7}{4x_1^{(\nu)}x_2^{(\nu)} - 1},$$

$$x_2^{(\nu+1)} = x_2^{(\nu)} - \frac{2x_1^{(\nu)}(x_2^{(\nu)})^2 + (x_1^{(\nu)})^2 - 14x_1^{(\nu)} - x_2^{(\nu)} + 11}{4x_1^{(\nu)}x_2^{(\nu)} - 1}.$$

Als Startvektor wird $\boldsymbol{x}^{(0)} = (3.4, -1.7)^\mathsf{T}$ verwendet. Man erhält bei Rundung auf eine 5-stellige Mantisse

ν	$x_1^{(\nu)}$	$x_2^{(\nu)}$
0	3.4000	−1.7000
1	3.5901	−1.8529
2	3.5844	−1.8481
3	3.5844	−1.8481

Die Iteration bleibt ab $\nu = 2$ stehen, so dass sich als Näherung $\overline{\boldsymbol{x}}_4 \approx (3.5844, -1.8481)^\mathsf{T}$ ergibt.

II. Näherungsweise Berechnung von $\overline{\boldsymbol{x}}_1$ im ersten Quadranten nach Algorithmus 6.7: Es gilt exakt $\overline{\boldsymbol{x}}_1 = (3,2)^\mathsf{T}$. Mit dem sehr groben Startvektor $\boldsymbol{x}^{(0)} = (4,1)^\mathsf{T}$ werden zwei Iterationsschritte ausgeführt (Rechnung mit Rundung auf 6-stellige Mantisse).

Man erhält für den 1. Iterationsschritt

$$\boldsymbol{\nu = 0:} \qquad \boldsymbol{J}_f(\boldsymbol{x}^{(0)}) = \begin{pmatrix} 8 & 1 \\ 1 & 2 \end{pmatrix}, \qquad \boldsymbol{f}(\boldsymbol{x}^{(0)}) = \begin{pmatrix} 6 \\ -2 \end{pmatrix}.$$

(i) Berechnung von $\Delta\boldsymbol{x}^{(1)}$ aus $\boldsymbol{J}_f(\boldsymbol{x}^{(0)})\Delta\boldsymbol{x}^{(1)} = -\boldsymbol{f}(\boldsymbol{x}^{(0)})$

$$\begin{pmatrix} 8 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} \Delta x_1^{(1)} \\ \Delta x_2^{(1)} \end{pmatrix} = \begin{pmatrix} -6 \\ 2 \end{pmatrix} \Rightarrow \Delta\boldsymbol{x}^{(1)} = \begin{pmatrix} -0.93333 \\ 1.46667 \end{pmatrix}.$$

(ii) Berechnung von $\boldsymbol{x}^{(1)} = \boldsymbol{x}^{(0)} + \Delta\boldsymbol{x}^{(1)}$

$$\boldsymbol{x}^{(1)} = \begin{pmatrix} 4 \\ 1 \end{pmatrix} + \begin{pmatrix} -0.93333 \\ 1.46667 \end{pmatrix} = \begin{pmatrix} 3.06667 \\ 2.46667 \end{pmatrix}.$$

Für den 2. Iterationsschritt ergibt sich mit

$$\boldsymbol{\nu = 1:} \qquad \boldsymbol{J}_f(\boldsymbol{x}^{(1)}) = \begin{pmatrix} 6.13333 & 1 \\ 1 & 4.93333 \end{pmatrix}, \quad \boldsymbol{f}(\boldsymbol{x}^{(1)}) = \begin{pmatrix} 0.87111 \\ 2.15111 \end{pmatrix}.$$

(i) Berechnung von $\Delta\boldsymbol{x}^{(2)}$ aus $\boldsymbol{J}_f(\boldsymbol{x}^{(1)})\Delta\boldsymbol{x}^{(2)} = -\boldsymbol{f}(\boldsymbol{x}^{(1)})$

$$\begin{pmatrix} 6.13333 & 1 \\ 1 & 4.93333 \end{pmatrix} \begin{pmatrix} \Delta x_1^{(2)} \\ \Delta x_1^{(2)} \end{pmatrix} = \begin{pmatrix} -0.87111 \\ -2.15111 \end{pmatrix}$$

$$\Rightarrow \Delta\boldsymbol{x}^{(2)} = \begin{pmatrix} -0.07336 \\ -0.42117 \end{pmatrix}.$$

(ii) Berechnung von $\boldsymbol{x}^{(2)} = \boldsymbol{x}^{(1)} + \Delta\boldsymbol{x}^{(2)}$

$$\boldsymbol{x}^{(2)} = \begin{pmatrix} 3.06667 \\ 2.46667 \end{pmatrix} + \begin{pmatrix} -0.07336 \\ -0.42117 \end{pmatrix} = \begin{pmatrix} 2.99331 \\ 2.04550 \end{pmatrix} \approx \overline{\boldsymbol{x}}_1,$$

$$\|\boldsymbol{f}(\boldsymbol{x}^{(2)})\|_2 = 0.17746\ldots$$

□

Primitivform des Newton-Verfahrens für Systeme

Um sich die Lösung eines linearen Gleichungssystems (6.19) in jedem Iterationsschritt zu ersparen, kann statt (6.19) das Gleichungssystem

$$\boldsymbol{J}_f(\boldsymbol{x}^{(0)})\Delta\boldsymbol{x}^{(\nu+1)} = -\boldsymbol{f}(\boldsymbol{x}^{(\nu)}) \tag{6.20}$$

mit fester Matrix $\boldsymbol{J}_f(\boldsymbol{x}^{(0)})$ für alle Iterationsschritte verwendet werden. Dann ist die Dreieckszerlegung (**LR**-Zerlegung, siehe Abschnitt 4.2) der Matrix $\boldsymbol{J}_f$ nur einmal auszuführen, Vorwärts- und Rückwärtselimination sind für jede neue rechte Seite notwendig, ebenso die Berechnung von $\boldsymbol{x}^{(\nu+1)}$ nach (6.18).

Man kann auch so verfahren, dass man etwa ν_0 Schritte mit fester Matrix $\boldsymbol{J}_f(\boldsymbol{x}^{(0)})$ gemäß (6.20) iteriert, dann die nächsten ν_1 Schritte mit fester Matrix $\boldsymbol{J}_f(\boldsymbol{x}^{(\nu_0)})$ usw.; auf diese Weise wird die Konvergenzgeschwindigkeit etwas erhöht.

6.3.1.2 Gedämpftes Newton-Verfahren für Systeme

Das gedämpfte Newton-Verfahren ist eine Variante des quadratisch konvergenten Newton-Verfahrens, (vgl. [CONT1987], 5.2).
Eine Newton-Iterierte $\boldsymbol{x}^{(\nu+1)}$ wird erst akzeptiert, wenn in der Euklidischen Norm gilt

$$\|\boldsymbol{f}(\boldsymbol{x}^{(\nu+1)})\|_2 < \|\boldsymbol{f}(\boldsymbol{x}^{(\nu)})\|_2.$$

Mit den gleichen Voraussetzungen wie für das Newton-Verfahren in Abschnitt 6.3.1.1 gilt der

Algorithmus 6.9. (*Gedämpftes Newton-Verfahren für Systeme*)
Für $\nu = 0,1,2,...$ sind nacheinander folgende Schritte auszuführen:

(i) Berechnung von $\Delta\boldsymbol{x}^{(\nu+1)}$ aus (6.19).

(ii) Berechnung eines j so, dass gilt

$$j := \min\left\{ i \;\middle|\; i \geq 0 \quad \left\| \boldsymbol{f}\left(\boldsymbol{x}^{(\nu)} + \frac{\Delta\boldsymbol{x}^{(\nu+1)}}{2^i}\right) \right\|_2 < \left\| \boldsymbol{f}(\boldsymbol{x}^{(\nu)}) \right\|_2 \right\}$$

(iii) $\boldsymbol{x}^{(\nu+1)} := \boldsymbol{x}^{(\nu)} + \frac{\Delta\boldsymbol{x}^{(\nu+1)}}{2^j}$.

Den Schritt (ii) führt man zu vorgegebenem $i_{\max}$ nur für $0 \leq i \leq i_{\max}$ durch. Sollte die Bedingung dann noch immer nicht erfüllt sein, rechnet man mit $j = 0$ weiter. Das Verfahren ist quadratisch konvergent.

In der Praxis verwendet man das gedämpfte Newton-Verfahren in zwei Varianten. Die eine arbeitet mit Vorgabe der Jakobi-Matrix, die zweite Variante schätzt die Jakobi-Matrix mit dem vorderen Differenzenquotienten. Beim zweiten Verfahren ist es möglich anzugeben, wieviele Iterationsschritte IUPD mit fester geschätzter Jakobi-Matrix durchgeführt werden sollen. Falls diese Anzahl > 1 ist, entspricht das Verfahren der gedämpften Primitivform des Newtonschen Iterationsverfahrens.

Umfangreiche Tests haben ergeben, dass das gedämpfte Newton-Verfahren bzw. die gedämpfte Primitivform im Allgemeinen weit besser sind als das normale Newton-Verfahren, das Verfahren von Brown oder das Gradientenverfahren. Es hat sich auch gezeigt, dass die Dämpfungsgröße $i_{\max}$ stark vom Problem abhängig ist und das Verfahren bei gleichem Startvektor bei verschiedener Vorgabe von $i_{\max}$ einmal konvergiert, ein anderes Mal nicht; es kann insbesondere bei verschiedenen $i_{\max}$ gegen verschiedene Nullstellen konvergieren. Bei völliger Offenheit der Situation sollte deshalb zunächst mit $i_{\max} = 4$, IUPD $= 1$ und maximal 1000 Iterationen gearbeitet werden.

Beispiel 6.10.

Gegeben: Ein System aus zwei nichtlinearen Gleichungen

$$\begin{aligned} f_1 = x_1^2 - x_2 - 1 &= 0 \\ f_2 = (x_1 - 2)^2 + (x_2 - 0.5)^2 - 1 &= 0\,. \end{aligned}$$

Gesucht: Die Anzahl der Iterationsschritte mit den verschiedenen Verfahren, wobei $\varepsilon_1 = \varepsilon_2 = 1.0 \cdot 10^{-14}$, $\varepsilon_3 = 1.28 \cdot 10^{-12}$ (vgl. Algorithmus 6.7).

Lösung:

Verfahren	Anzahl Iterationsschritte
Newton-Verfahren	7
gedämpftes Newton-Verfahren	7
Primitivform des Newton-Verfahrens	10
Primitivform des gedämpften Newton-Verfahrens	10
Brown-Verfahren	8

□

6.3.2 Sekantenverfahren für nichtlineare Systeme

Gegeben sei das nichtlineare Gleichungssystem (6.1). Man bildet damit die Vektoren

$$\begin{aligned} \delta \boldsymbol{f}(x_j, \widetilde{x}_j) &= \tfrac{1}{x_j - \widetilde{x}_j}\big(\boldsymbol{f}(x_1, \ldots, x_j, \ldots, x_n) - \boldsymbol{f}(x_1, \ldots, \widetilde{x}_j, \ldots, x_n)\big) \\ &=: \widetilde{\boldsymbol{f}}(x_j, \widetilde{x_j})\,, \qquad j = 1(1)n. \end{aligned} \tag{6.21}$$

Mit den Vektoren (6.21) wird die folgende Matrix gebildet

$$\boldsymbol{Df}(\boldsymbol{x}, \widetilde{\boldsymbol{x}}) = \big(\widetilde{\boldsymbol{f}}(x_1, \widetilde{x}_1), \widetilde{\boldsymbol{f}}(x_2, \widetilde{x}_2), \ldots, \widetilde{\boldsymbol{f}}(x_n, \widetilde{x}_n)\big)\,.$$

Sie entspricht der Funktionalmatrix beim Newton-Verfahren, wenn dort die Ableitungen durch Differenzenquotienten ersetzt werden; d. h. $\boldsymbol{D}$ ist die geschätzte Jakobimatrix $\boldsymbol{J}$.

Ist $\overline{\boldsymbol{x}} \in B$ eine Lösung von (6.1) und sind $\boldsymbol{x}^{(\nu-1)}$, $\boldsymbol{x}^{(\nu)} \in B$ Näherungen für $\overline{\boldsymbol{x}}$, so errechnet sich für jedes ν = 1,2,3,... eine weitere Näherung $\boldsymbol{x}^{(\nu+1)}$ nach der *Iterationsvorschrift des Sekantenverfahrens*

$$\boldsymbol{x}^{(\nu+1)} = \boldsymbol{x}^{(\nu)} - \big(\boldsymbol{Df}^{-1}(\boldsymbol{x}^{(\nu)}, \boldsymbol{x}^{(\nu-1)})\big)\boldsymbol{f}(\boldsymbol{x}^{(\nu)}) = \boldsymbol{\varphi}(\boldsymbol{x}^{(\nu)}, \boldsymbol{x}^{(\nu-1)}); \tag{6.22}$$

es sind also stets zwei Startvektoren $\boldsymbol{x}^{(0)}$, $\boldsymbol{x}^{(1)}$ erforderlich. Die Berechnung der Inversen $\boldsymbol{Df}^{-1}$ kann analog zum Newton-Verfahren vermieden werden, wenn gesetzt wird:

$$\begin{aligned} \boldsymbol{x}^{(\nu+1)} &= \boldsymbol{x}^{(\nu)} + \Delta\boldsymbol{x}^{(\nu+1)} \qquad \text{mit} \\ \Delta\boldsymbol{x}^{(\nu+1)} &= -\big(\boldsymbol{Df}^{-1}(\boldsymbol{x}^{(\nu)}, \boldsymbol{x}^{(\nu-1)})\big)\boldsymbol{f}(\boldsymbol{x}^{(\nu)}), \end{aligned} \tag{6.23}$$

so dass $\Delta \boldsymbol{x}^{(\nu+1)}$ als Lösung des linearen Gleichungssystems

$$\boldsymbol{Df}(\boldsymbol{x}^{(\nu)}, \boldsymbol{x}^{(\nu+1)})\Delta \boldsymbol{x}^{(\nu+1)} = -\boldsymbol{f}(\boldsymbol{x}^{(\nu)})$$

gewonnen und in (6.23) eingesetzt wird. Hinreichende Bedingungen für die Konvergenz sind in [SCHM1963] angegeben; die Bedingungen sind für die praktische Durchführung unbrauchbar. Ist jedoch $\det \boldsymbol{J}(\overline{\boldsymbol{x}}) \neq 0$, so konvergiert das Verfahren sicher, wenn die Startvektoren nahe genug bei bei $\overline{\boldsymbol{x}}$ liegen; die Konvergenzordnung ist dann $p = (1+\sqrt{5})/2$.

Fehlerabschätzungen (vgl. [SCHM1963], S.3):

$$\|\overline{\boldsymbol{x}} - \boldsymbol{x}^{(\nu)}\| \leq \prod_{k=1}^{\nu-1} \left(\frac{s_k}{1-s_k}\right) \frac{1-2s_1}{1-3s_1} \|\boldsymbol{x}^{(2)} - \boldsymbol{x}^{(1)}\|, \qquad \nu = 2, 3, \ldots$$

$$\text{mit} \qquad s_1 \leq \frac{2}{7}, \quad s_2 = \frac{s_1}{1-s_1}, \quad s_k = \frac{s_{k-1}}{1-s_{k-1}} \frac{s_{k-2}}{1-s_{k-2}}, \qquad k \geq 3.$$

In [SCHM1963], S.99 ist eine Variante dieses Verfahrens zu finden.

6.3.3 Das Verfahren des stärksten Abstiegs (Gradientenverfahren) für nichtlineare Systeme

Gegeben sei ein nichtlineares Gleichungssystem (6.1). Es besitze in B eine Lösung $\overline{\boldsymbol{x}}$. Bildet man die Funktion

$$Q(\boldsymbol{x}) := \sum_{i=1}^{n} f_i^2(\boldsymbol{x}), \quad Q(\boldsymbol{x}) = Q(x_1, x_2, \ldots, x_n), \tag{6.24}$$

so ist genau dann, wenn $f_i(\boldsymbol{x}) = 0$ für $i = 1(1)n$ gilt, auch $Q(\boldsymbol{x}) = 0$. Die Aufgabe, Lösungen $\overline{\boldsymbol{x}}$ zu suchen, für die $Q(\boldsymbol{x}) = 0$ ist, ist also äquivalent zu der Aufgabe, das System (6.1) aufzulösen.

Mit Hilfe von (6.24) und

$$\nabla Q(\boldsymbol{x}) = \operatorname{grad} Q(\boldsymbol{x}) = \begin{pmatrix} Q_{x_1} \\ Q_{x_2} \\ \vdots \\ Q_{x_n} \end{pmatrix}, \; Q_{x_i} := \frac{\partial Q}{\partial x_i}, \qquad i = 1(1)n,$$

ergibt sich ein Iterationsverfahren zur näherungsweisen Bestimmung von $\overline{\boldsymbol{x}}$ mit der *Iterationsvorschrift*

$$\boldsymbol{x}^{(\nu+1)} = \boldsymbol{x}^{(\nu)} - \frac{Q(\boldsymbol{x}^{(\nu)})}{\left(\nabla Q(\boldsymbol{x}^{(\nu)})\right)^2} \nabla Q(\boldsymbol{x}^{(\nu)}) := \boldsymbol{\varphi}(\boldsymbol{x}^{(\nu)}), \tag{6.25}$$

$$\text{mit} \qquad \left(\nabla Q\left(\boldsymbol{x}^{(\nu)}\right)\right)^2 = \sum_{i=1}^{n} Q_{x_i}^2(\boldsymbol{x}^{(\nu)}).$$

Die Schrittfunktion lautet somit

$$\boldsymbol{\varphi}(\boldsymbol{x}) = \boldsymbol{x} - \frac{Q(\boldsymbol{x})}{\big(\nabla Q(\boldsymbol{x})\big)^2}\nabla Q(\boldsymbol{x}).$$

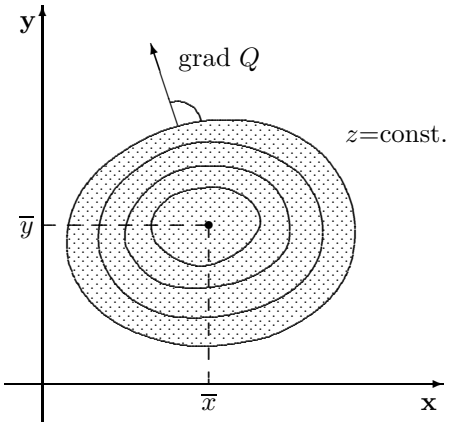


Abb. 6.4. Zum Gradientenverfahren

Zur Konvergenz gelten die entsprechenden Aussagen wie beim Newtonschen Verfahren. Die Konvergenz der Folge $\{\boldsymbol{x}^{(\nu)}\}$ mit den nach der Vorschrift (6.25) gebildeten Vektoren gegen $\overline{\boldsymbol{x}}$ ist wie dort gewährleistet, wenn der Startvektor $\boldsymbol{x}^{(0)}$ nahe genug bei $\overline{\boldsymbol{x}}$ liegt. Im Allgemeinen kann man jedoch beim Gradientenverfahren mit gröberen Ausgangsnäherungen (Startvektoren) $\boldsymbol{x}^{(0)}$ arbeiten als beim Newtonschen Verfahren; das Gradientenverfahren konvergiert allerdings nur linear. Über eine Methode zur Konvergenzverbesserung s. [BERE1971] Bd.2, S.150/151.

Die Anwendung des Gradientenverfahrens wird allerdings erschwert, wenn in der Umgebung der gesuchten Lösung $\boldsymbol{x}$ auch Nichtnull-Minima der Funktion $Q(\boldsymbol{x})$ existieren. Dann kann es vorkommen, dass die Iterationsfolge gegen eines dieser Nichtnull-Minima konvergiert (vgl. dazu [BERE1971] Bd.2, S.152).

Über allgemeine Gradientenverfahren und die zugehörigen Konvergenzbedingungen s. [STOE1989], 5.4.1; [STUM1982], 9.2.2.

Gradientenverfahren für n = 2:

Mit $x_1 = x,\ x_2 = y,\ f_1 = f,\ f_2 = g$ und $Q = f^2(x,y) + g^2(x,y)$ lautet (6.25)

$$\begin{aligned} x^{(\nu+1)} &= x^{(\nu)} - \frac{Q(x^{(\nu)},y^{(\nu)})Q_x(x^{(\nu)},y^{(\nu)})}{Q_x^2(x^{(\nu)},y^{(\nu)}) + Q_y^2(x^{(\nu)},y^{(\nu)})} \\ y^{(\nu+1)} &= y^{(\nu)} - \frac{Q(x^{(\nu)},y^{(\nu)})Q_y(x^{(\nu)},y^{(\nu)})}{Q_x^2(x^{(\nu)},y^{(\nu)}) + Q_y^2(x^{(\nu)},y^{(\nu)})}\,. \end{aligned}$$

Einen geeigneten Startvektor $(x^{(0)}, y^{(0)})$ beschafft man sich hier durch grobes Aufzeichnen von $f = 0$ und $g = 0$.

Beispiel 6.11. (Fortsetzung von Beispiel 6.8)

Gegeben: Das nichtlineare Gleichungssystem (6.3).

Lösung: Es sind

$$\begin{aligned} Q(x,y) &= f^2(x,y) + g^2(x,y) \\ &= (x^2+y-11)^2 + (x+y^2-7)^2, \\ \nabla Q &= \begin{pmatrix} 4x(x^2+y-11) & + & 2(x+y^2-7) \\ 2(x^2+y-11) & + & 4y(x+y^2-7) \end{pmatrix}, \end{aligned}$$

als Startwert wird wieder $x^{(0)} = 3.4, y^{(0)} = -1.7$ gesetzt. Die iterierten Werte $x^{(\nu)}, y^{(\nu)}$ für $\nu = 1,2,3$ und die zu ihrer Berechnung benötigten Werte $Q^{(\nu)}, Q_x^{(\nu)}, Q_y^{(\nu)}$ für $\nu = 0,1,2$ sind in der folgenden Tabelle zusammengestellt:

ν	$x^{(\nu)}$	$y^{(\nu)}$	$Q^{(\nu)}$	$Q_x^{(\nu)}$	$Q_y^{(\nu)}$
0	3.4000	−1.7000	1.8037	−16.9240	2.5480
1	3.5042	−1.7157	0.4952	−7.2196	2.9169
2	3.5632	−1.7395	0.1707	−1.4362	2.7731
3	3.5883	−1.7880	0.05385		

□

6.3.4 Das Verfahren von Brown für Systeme

Das Verfahren von Brown [BROW1971] zur Lösung eines Systems (6.1) von n nichtlinearen Gleichungen mit n Unbekannten ist ein (lokal) quadratisch konvergentes, Newton-ähnliches Iterationsverfahren, das ohne vorherige Kenntnis der partiellen Ableitungen arbeitet. Die Approximation des nichtlinearen Systems in der Umgebung der Lösung geschieht hier durch ein lineares System nacheinander komponentenweise. Bei der Berechnung einer neuen Komponente kann deshalb die letzte Information über die vorherbestimmten Komponenten bereits verwendet werden. Pro Iterationsschritt benötigt das Verfahren nur etwa halb so viele Funktionsauswertungen wie das Newton-Verfahren.

Beispiel 6.12.

Gegeben: Die Gleichungen

$$\begin{aligned} f_1 = x_1 - \mathrm{e}^{\sin(x_2)} &= 0 \\ f_2 = \pi \cdot \mathrm{e}^{\sin(\pi x_1)} + x_2 &= 0\,. \end{aligned}$$

Gesucht: Die Anzahl der Lösungsschritte mit den Verfahren

1. Newton-Verfahren
2. gedämpftes Newton-Verfahren
3. Brown-Verfahren für Systeme

Lösung: Die fünf Lösungen der Gleichungen lauten

$$\begin{array}{lll} (1) & x_1 = 0.733877 & x_2 = -6.597761 \\ (2) & x_1 = 0.367895 & x_2 = -7.844702 \\ (3) & x_1 = 1.0 & x_2 = -3.141592 \\ (4) & x_1 = 2.074316 & x_2 = -3.959375 \\ (5) & x_1 = 2.181405 & x_2 = -5.388568 \end{array}$$

Verfahren	Anzahl Iterationsschritte	Nullstellen
Newton-Verfahren	100	(–)
Gedämpftes Newton-Verfahren ($i_{\max} = 2$)	51	(1)
Gedämpftes Newton-Verfahren ($i_{\max} = 3$)	13	(4)
Gedämpftes Newton-Verfahren ($i_{\max} = 9$)	18	(4)
Gedämpftes Newton-Verfahren ($i_{\max} = 19$)	68	(5)
Brown-Verfahren für Systeme	100	(–)

Es wurde mit der Numerik-Bibliothek auf der CD gerechnet. Die Abbruchschranke lag bei $\varepsilon = 0.5 \cdot 10^{-8}$. Bei 100 Iterationsschritten wurde die Berechnung beendet. □

6.4 Entscheidungshilfen für die Auswahl der Methode

Von den angegebenen Verfahren sind das gedämpfte Newton-Verfahren bzw. die gedämpfte Primitivform des Newton-Verfahrens den übrigen Verfahren vorzuziehen. Für den praktischen Einsatz ist besonders die Modifikation mit geschätzter Jakobimatrix bei beiden Verfahren zu empfehlen, um die Erstellung der partiellen Ableitungen für die Jakobimatrix umgehen zu können, siehe dazu auch die Bemerkungen am Ende des Abschnittes 6.3.1.2.

Ergänzende Literatur zu Kapitel 6

[CARN1990], 5.9; [DENN1996]; [DEUF2002] Bd.1, Kap.4; [HAMM1994], 8. §2; [IGAR1985]; [OPFE2002], Kap.10; [PREU2001], Kap.4; [QUAR2002], Kap.7; [RICE1993], S.239; [SCHW1997], 5; [SCHWE1979]; [TORN1990] Bd.1, 7.1 f.

Kapitel 7

Eigenwerte und Eigenvektoren von Matrizen

7.1 Definitionen und Aufgabenstellungen

Gegeben ist eine (n,n)-Matrix $\boldsymbol{A} = (a_{ik})$, $i,k = 1(1)n$, und gesucht sind Vektoren $\boldsymbol{x}$ derart, dass der Vektor $\boldsymbol{Ax}$ dem Vektor $\boldsymbol{x}$ proportional ist mit einem zunächst noch unbestimmten Parameter λ

$$\boldsymbol{Ax} = \lambda \boldsymbol{x}. \tag{7.1}$$

Mit der (n,n)-Einheitsmatrix $\boldsymbol{E}$ lässt sich (7.1) in der Form

$$\boldsymbol{Ax} - \lambda \boldsymbol{x} = (\boldsymbol{A} - \lambda \boldsymbol{E})\boldsymbol{x} = \boldsymbol{0} \tag{7.2}$$

schreiben. (7.2) ist ein homogenes lineares Gleichungssystem, das genau dann nichttriviale Lösungen $\boldsymbol{x} \neq \boldsymbol{0}$ besitzt, wenn

$$P(\lambda) := \det(\boldsymbol{A} - \lambda \boldsymbol{E}) = 0 \tag{7.3}$$

ist, ausführlich geschrieben

$$P(\lambda) = \begin{vmatrix} a_{11}-\lambda & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22}-\lambda & a_{23} & \cdots & a_{2n} \\ \vdots & & \ddots & & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn}-\lambda \end{vmatrix} = 0. \tag{7.4}$$

(7.3) bzw. (7.4) heißt *charakteristische Gleichung* der Matrix $\boldsymbol{A}$; $P(\lambda)$ ist ein Polynom in λ vom Grade n und heißt entsprechend *charakteristisches Polynom* der Matrix $\boldsymbol{A}$. Die Nullstellen λ_i , $i = 1(1)n$, von $P(\lambda)$ heißen *charakteristische Zahlen* oder *Eigenwerte* (EWe) von $\boldsymbol{A}$. Nur für die EWe λ_i besitzt (7.2) nichttriviale Lösungen. Ein zu einem EW λ_i gehöriger Lösungsvektor $\boldsymbol{x}_i$ heißt *Eigenvektor* (EV) der Matrix $\boldsymbol{A}$ zum EW λ_i. Es gilt

$$\boldsymbol{Ax}_i = \lambda_i \boldsymbol{x}_i \quad \text{bzw.} \quad (\boldsymbol{A} - \lambda_i \boldsymbol{E})\boldsymbol{x}_i = \boldsymbol{0}. \tag{7.5}$$

Die Aufgabe, die EWe und EVen einer Matrix $\boldsymbol{A}$ zu bestimmen, heißt *Eigenwertaufgabe* (EWA).

Es wird zwischen der *vollständigen* und der *teilweisen* EWA unterschieden. Die vollständige EWA verlangt die Bestimmung sämtlicher EWe und EVen, die teilweise EWA verlangt nur die Bestimmung eines (oder mehrerer) EWes (EWe) ohne oder mit dem (den) zugehörigen EV (EVen).

Man unterscheidet zwei Klassen von *Lösungsmethoden*:

1. *Iterative Methoden*: Sie umgehen die Aufstellung des charakteristischen Polynoms $P(\lambda)$ und versuchen, die EWe und EVen schrittweise anzunähern.
2. *Direkte Methoden*: Sie erfordern die Aufstellung des charakteristischen Polynoms $P(\lambda)$, die Bestimmung der EWe λ_i als Nullstellen von $P(\lambda)$ und die anschließende Berechnung der EVen $\boldsymbol{x}_i$ als Lösungen der homogenen Gleichungssysteme (7.5). Sie sind zur Lösung der vollständigen EWA geeignet; unter ihnen gibt es auch solche, die das Ausrechnen umfangreicher Determinanten vermeiden, z. B. das Verfahren von Krylov.

Die Berechnung der Eigenwerte und Eigenvektoren einer komplexen (n,n)- Matrix kann auf die entsprechende Aufgabe für eine reelle $(2n,2n)$-Matrix zurückgeführt werden. Es sei

$$\begin{array}{lcll} \boldsymbol{A} & = & \boldsymbol{B} + \mathrm{i}\boldsymbol{C}, & \boldsymbol{B}, \boldsymbol{C} \quad \text{reelle Matrizen,} \\ \boldsymbol{x} & = & \boldsymbol{u} + \mathrm{i}\boldsymbol{v}, & \boldsymbol{u}, \boldsymbol{v} \quad \text{reelle Vektoren.} \end{array}$$

Dann erhält man durch Einsetzen in (7.1) zwei reelle lineare homogene Gleichungssysteme

$$\begin{aligned} \boldsymbol{B}\boldsymbol{u} - \boldsymbol{C}\boldsymbol{v} &= \lambda \boldsymbol{u} \\ \boldsymbol{C}\boldsymbol{u} + \boldsymbol{B}\boldsymbol{v} &= \lambda \boldsymbol{v}. \end{aligned}$$

Diese lassen sich mit der $(2n,2n)$-Matrix $\widetilde{\boldsymbol{A}}$ und dem Vektor $\widetilde{\boldsymbol{x}}$

$$\widetilde{\boldsymbol{A}} = \begin{pmatrix} \boldsymbol{B} & -\boldsymbol{C} \\ \boldsymbol{C} & \boldsymbol{B} \end{pmatrix} \qquad \widetilde{\boldsymbol{x}} = \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{v} \end{pmatrix}$$

zu der reellen Ersatzaufgabe $\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{x}} = \lambda\widetilde{\boldsymbol{x}}$ zusammenfassen.

7.2 Diagonalähnliche Matrizen

Eine (n,n)-Matrix $\boldsymbol{A}$, die zu einem k_j-fachen EW stets k_j linear unabhängige EVen und wegen $\Sigma k_j = n$ genau n linear unabhängige EVen zu der Gesamtheit ihrer EWe besitzt, heißt *diagonalähnlich*. Die n linear unabhängigen EVen spannen den n-dimensionalen Vektorraum $\mathbf{R}^n$ auf.

Die EVen sind bis auf einen willkürlichen Faktor bestimmt. Der EV $\boldsymbol{x}_i$ wird so normiert, dass gilt

$$\|\boldsymbol{x}_i\|_2 = |\boldsymbol{x}_i| = \sqrt{\boldsymbol{x}_i^{\mathsf{T}}\boldsymbol{x}_i} = \sqrt{\sum_{k=1}^{n} x_{i,k}^2} = 1 \quad \text{mit} \quad \boldsymbol{x}_i = \begin{pmatrix} x_{i,1} \\ x_{i,2} \\ \vdots \\ x_{i,n} \end{pmatrix}, \tag{7.6}$$

d. h. die Euklidische Norm von $\boldsymbol{x}_i$ nimmt den Wert 1 an.

Bezeichnet man mit $\boldsymbol{X}$ die nicht singuläre Eigenvektormatrix (Modalmatrix)

$$\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n), \tag{7.7}$$

so gilt mit der Diagonalmatrix $\boldsymbol{D}$ (Spektralmatrix) der Eigenwerte

$$\boldsymbol{D} = \begin{pmatrix} \lambda_1 & 0 & . & . & . & 0 \\ 0 & \lambda_2 & 0 & . & . & 0 \\ . & & . & & & . \\ . & & & . & & . \\ . & & & & . & . \\ 0 & . & . & . & 0 & \lambda_n \end{pmatrix}$$

und wegen $\det \boldsymbol{X} \neq 0$

$$\boldsymbol{D} = \boldsymbol{X}^{-1} \boldsymbol{A} \boldsymbol{X}.$$

Jede Matrix mit n linear unabhängigen EVen $\boldsymbol{x}_i$ lässt sich also auf Hauptdiagonalform transformieren. Es gilt der folgende

Satz 7.1. (*Entwicklungssatz*)
Ist $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n$ ein System von n linear unabhängigen Eigenvektoren, so lässt sich jeder beliebige Vektor $\boldsymbol{z} \neq \boldsymbol{0}$ des n-dimensionalen Vektorraumes $\mathbb{R}^n$ als Linearkombination

$$\boldsymbol{z} = c_1 \boldsymbol{x}_1 + c_2 \boldsymbol{x}_2 + \ldots + c_n \boldsymbol{x}_n, \quad c_i = \text{const.},$$

darstellen, wobei für mindestens einen Index i gilt $c_i \neq 0$.

Als Sonderfall enthalten die diagonalähnlichen Matrizen die hermiteschen Matrizen $\boldsymbol{H} = (h_{ik})$ mit $\boldsymbol{H} = \overline{\boldsymbol{H}}^{\mathsf{T}}$ bzw. $h_{ik} = \overline{h}_{ki}$ ($\overline{h}_{ki}$ sind die zu h_{ki} konjugiert komplexen Elemente) und diese wiederum die symmetrischen Matrizen $\boldsymbol{S} = (s_{ik})$ mit reellen Elementen $s_{ik} = s_{ki}$, d. h. $\boldsymbol{S} = \boldsymbol{S}^{\mathsf{T}}$.

Hermitesche (und damit auch symmetrische) Matrizen besitzen die folgenden *Eigenschaften*:

1. Sämtliche EWe sind reell; bei symmetrischen Matrizen sind auch die EVen reell.
2. Die zu verschiedenen EWen gehörenden EVen sind unitär (konjugiert orthogonal): $\overline{\boldsymbol{x}_i}^{\mathsf{T}} \boldsymbol{x}_k = 0$ für $i \neq k$; für analog zu (7.6) normierte EVen gilt

$$\|x_i\|_2 = \sqrt{\sum_{k=1}^{n} \overline{x_{i,k}}\, x_{i,k}} = 1$$

$$\overline{\boldsymbol{x}_i}^{\mathsf{T}} \boldsymbol{x}_k = \delta_{ik} = \begin{cases} 1 & \text{für} \quad i = k, \\ 0 & \text{für} \quad i \neq k. \end{cases}$$

3. Die Eigenvektormatrix (7.7) ist unitär ($\overline{\boldsymbol{X}}^{\mathsf{T}} = \boldsymbol{X}^{-1}$) .

Bei symmetrischen Matrizen ist in 2. und 3. unitär durch orthogonal zu ersetzen.

7.3 Das Iterationsverfahren nach v. Mises

7.3.1 Bestimmung des betragsgrößten Eigenwertes und des zugehörigen Eigenvektors

Es sei eine EWA (7.2) vorgelegt mit einer diagonalähnlichen reellen Matrix $\boldsymbol{A}$, d. h. einer Matrix mit n linear unabhängigen EVen $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, ..., $\boldsymbol{x}_n \in \mathbb{R}^n$. Man beginnt mit einem beliebigen reellen Vektor $\boldsymbol{z}^{(0)} \neq \boldsymbol{0}$ und bildet mit der Matrix $\boldsymbol{A}$ die iterierten Vektoren $\boldsymbol{z}^{(\nu)}$ nach der Vorschrift

$$\boldsymbol{z}^{(\nu+1)} := \boldsymbol{A}\boldsymbol{z}^{(\nu)}, \quad \boldsymbol{z}^{(\nu)} = \begin{pmatrix} z_1^{(\nu)} \\ z_2^{(\nu)} \\ \vdots \\ z_n^{(\nu)} \end{pmatrix}, \quad \nu = 0, 1, 2, \ldots \tag{7.8}$$

Nach Satz 7.1 lässt sich $\boldsymbol{z}^{(0)}$ als Linearkombination der n EVen $\boldsymbol{x}_i$, $i = 1(1)n$, darstellen

$$\boldsymbol{z}^{(0)} = \sum_{i=1}^{n} c_i \boldsymbol{x}_i \tag{7.9}$$

mit $c_i \neq 0$ für mindestens ein i, so dass wegen (7.5) mit (7.8) und (7.9) folgt

$$\boldsymbol{z}^{(\nu)} = c_1\lambda_1^{\nu}\boldsymbol{x}_1 + c_2\lambda_2^{\nu}\boldsymbol{x}_2 + \ldots + c_n\lambda_n^{\nu}\boldsymbol{x}_n.$$

Nun werden die Quotienten $q_i^{(\nu)}$ der i-ten Komponenten der Vektoren $\boldsymbol{z}^{(\nu+1)}$ und $\boldsymbol{z}^{(\nu)}$ gebildet

$$q_i^{(\nu)} := \frac{z_i^{(\nu+1)}}{z_i^{(\nu)}} = \frac{c_1\lambda_1^{\nu+1}x_{1,i} + c_2\lambda_2^{\nu+1}x_{2,i} + \ldots + c_n\lambda_n^{\nu+1}x_{n,i}}{c_1\lambda_1^{\nu}x_{1,i} + c_2\lambda_2^{\nu}x_{2,i} + \ldots + c_n\lambda_n^{\nu}x_{n,i}}$$

Die Weiterbehandlung erfordert folgende *Fallunterscheidungen*:

1. $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \ldots \geq |\lambda_n|$:

a) $c_1 \neq 0$, $x_{1,i} \neq 0$: Für die Quotienten $q_i^{(\nu)}$ gilt

$$q_i^{(\nu)} = \lambda_1 + O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^{\nu}\right) \quad \text{bzw.} \quad \lim_{\nu\to\infty} q_i^{(\nu)} = \lambda_1.$$

Die Voraussetzung $x_{1,i} \neq 0$ ist für mindestens ein i erfüllt. Es strebt also mindestens einer der Quotienten $q_i^{(\nu)}$ gegen λ_1, für die übrigen λ_i, $i = 2(1)n$, vgl. unter b).

Für genügend große ν ist $q_i^{(\nu)}$ eine Näherung für den betragsgrößten EW λ_1. Bezeichnet man mit λ_i^* die Näherungen für λ_i, so gilt hier

$$\lambda_1^* = q_i^{(\nu)} \approx \lambda_1. \tag{7.10}$$

Bei der praktischen Durchführung des Verfahrens wird gerechnet, bis für die $q_i^{(\nu)}$ mit einer vorgegebenen Genauigkeit gleichmäßig für alle i mit $x_{1,i} \neq 0$ (7.10) gilt.

Der Vektor $\boldsymbol{z}^{(\nu)}$ hat für große ν annähernd die Richtung von $\boldsymbol{x}_1$. Für $\nu \to \infty$ erhält man das folgende asymptotische Verhalten

$$\boldsymbol{z}^{(\nu)} \sim \lambda_1^{\nu} c_1 \boldsymbol{x}_1, \quad \boldsymbol{z}^{(\nu)} \sim \lambda_1 \boldsymbol{z}^{(\nu-1)}.$$

Sind die EVen $\boldsymbol{x}_i$ normiert und bezeichnet man mit $\boldsymbol{x}_i^*$ die Näherungen für $\boldsymbol{x}_i$, so gilt mit (7.6) für hinreichend großes ν

$$\boldsymbol{x}_1^* = \frac{\boldsymbol{z}^{(\nu)}}{|\boldsymbol{z}^{(\nu)}|} \approx \boldsymbol{x}_1.$$

b) $c_1 = 0$ oder $x_{1,i} = 0,\ c_2 \neq 0,\ x_{2,i} \neq 0,\ |\lambda_2| > |\lambda_3| \geq \cdots \geq |\lambda_n|$:

Der Fall $c_1 = 0$ tritt dann ein, wenn der Ausgangsvektor $\boldsymbol{z}^{(0)}$ keine Komponente in Richtung von $\boldsymbol{x}_1$ besitzt. Im Falle symmetrischer Matrizen ist $c_1 = 0$, wenn $\boldsymbol{z}^{(0)}$ orthogonal ist zu $\boldsymbol{x}_1$ wegen $\boldsymbol{x}_i^{\mathsf{T}} \boldsymbol{x}_k = 0$ für $i \neq k$; dann gilt

$$q_i^{(\nu)} = \lambda_2 + O\left(\left|\frac{\lambda_3}{\lambda_2}\right|^{\nu}\right) \quad \text{bzw.} \quad \lim_{\nu\to\infty} q_i^{(\nu)} = \lambda_2.$$

$$\boldsymbol{z}^{(\nu)} \sim \begin{cases} c_1 \lambda_1^{\nu} \boldsymbol{x}_1 & \text{für} \quad c_1 \neq 0, \\ c_2 \lambda_2^{\nu} \boldsymbol{x}_2 & \text{für} \quad c_1 = 0. \end{cases}$$

Für hinreichend großes ν erhält man die Beziehungen

$$\boldsymbol{x}_1^* = \frac{\boldsymbol{z}^{(\nu)}}{|\boldsymbol{z}^{(\nu)}|} \approx \boldsymbol{x}_1 \quad \text{für} \quad c_1 \neq 0, \quad \boldsymbol{x}_2^* = \frac{\boldsymbol{z}^{(\nu)}}{|\boldsymbol{z}^{(\nu)}|} \approx \boldsymbol{x}_2 \quad \text{für} \quad c_1 = 0,$$

$$\lambda_2^* = q_i^{(\nu)} \approx \lambda_2 \begin{cases} \text{für alle} \quad i = 1(1)n, \ \text{falls} \ c_1 = 0, c_2 \neq 0, x_{2,i} \neq 0 \ \text{ist}, \\ \text{für alle} \quad i \quad \text{mit} \quad x_{1,i} = 0, \quad \text{falls} \quad c_1 \neq 0 \quad \text{ist}. \end{cases}$$

Es kann also vorkommen, dass die $q_i^{(\nu)}$ für verschiedene i gegen verschiedene EWe streben.

c) $c_i = 0$ für $i = 1(1)j,\ c_{j+1} \neq 0,\ x_{j+1,i} \neq 0,\ |\lambda_{j+1}| > |\lambda_{j+2}| \geq \cdots \geq |\lambda_n|$:

Man erhält hier für hinreichend großes ν die Beziehungen

$$\lambda_{j+1}^* = q_i^{(\nu)} \approx \lambda_{j+1}, \quad \boldsymbol{x}_{j+1}^* = \frac{\boldsymbol{z}^{(\nu)}}{|\boldsymbol{z}^{(\nu)}|} \approx \boldsymbol{x}_{j+1}.$$

Gilt hier $x_{j+1,i} = 0$ für ein i, so strebt das zugehörige $q_i^{(\nu)}$ gegen λ_{j+2}.

Rechenschema 7.2. (*Verfahren nach v. Mises*: $\boldsymbol{A}\boldsymbol{z}^{(\nu)} = \boldsymbol{z}^{(\nu+1)}$).

$\boldsymbol{A}$				$\boldsymbol{z}^{(0)}$	$\boldsymbol{z}^{(1)}$	$\boldsymbol{z}^{(2)}$	...
a_{11}	a_{12}	...	a_{1n}	$z_1^{(0)}$	$z_1^{(1)}$	$z_1^{(2)}$	
a_{21}	a_{22}	...	a_{2n}	$z_2^{(0)}$	$z_2^{(1)}$	$z_2^{(2)}$	
$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$	$\vdots$	
a_{n1}	a_{n2}	...	a_{nn}	$z_n^{(0)}$	$z_n^{(1)}$	$z_n^{(2)}$	

Bei der praktischen Durchführung berechnet man nicht nur die Vektoren $\boldsymbol{z}^{(\nu)}$, sondern normiert jeden Vektor $\boldsymbol{z}^{(\nu)}$ dadurch, dass man jede seiner Komponenten durch die betragsgrößte Komponente dividiert, so dass diese gleich 1 wird. Bezeichnet man den normierten Vektor mit $\boldsymbol{z}_n^{(\nu)}$, so wird $\boldsymbol{z}^{(\nu+1)}$ nach der Vorschrift $\boldsymbol{A}\boldsymbol{z}_n^{(\nu)} = \boldsymbol{z}^{(\nu+1)}$ bestimmt. Eine andere Möglichkeit ist, jeden Vektor $\boldsymbol{z}^{(\nu)}$ auf Eins zu normieren, was jedoch mehr Rechenzeit erfordert. Durch die Normierung wird ein zu starkes Anwachsen der Werte $z_i^{(\nu)}$ (und auch der Rundungsfehler) vermieden.

Beispiel 7.3.

Gegeben: Die Matrix

$$\boldsymbol{A} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

Gesucht: Die Näherungen λ_1^* und $\boldsymbol{x}_1^*$ für den betragsgrößten EW λ_1 und den zugehörigen EV $\boldsymbol{x}_1$ nach dem Verfahren von v. Mises.

Lösung: Anhand linear unabhängiger Vektoren $\boldsymbol{z}^{(0)}$ als Ausgangsvektoren wird demonstriert, welche Fälle auftreten können.

a) Wahl des Ausgangsvektors $\boldsymbol{z}^{(0)} = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}$

$\boldsymbol{A}$			$\boldsymbol{z}^{(0)}$	$\boldsymbol{z}^{(1)}$	$\boldsymbol{z}^{(2)}$	$\boldsymbol{z}^{(3)}$	$\boldsymbol{z}^{(4)}$	$\boldsymbol{z}^{(5)}$	...
1	1	0	1	3	6	12	24	48	
1	1	0	2	3	6	12	24	48	
0	0	−1	1	−1	1	−1	1	−1	
2	2	−1	—	5	13	23	49	95	

Hier gilt exakt: $q_1^{(\nu)} = q_2^{(\nu)} = 2 = \lambda_1$; $q_3^{(\nu)} = -1 = \lambda_2$ für alle ν. Es liegt also der Fall 1.b) mit $x_{1,3} = 0$ vor. Für $\boldsymbol{x}_1^*$ erhält man, wenn man bei $\nu = 5$ abbricht:

$$\boldsymbol{x}_1^* = \frac{z^{(5)}}{|z^{(5)}|} = \frac{1}{\sqrt{48^2 + 48^2 + 1}} \begin{pmatrix} 48 \\ 48 \\ -1 \end{pmatrix} = \frac{48}{\sqrt{4609}} \begin{pmatrix} 1 \\ 1 \\ -0.0208 \end{pmatrix} \approx \boldsymbol{x}_1.$$

b) Wählt man als Ausgangsvektor $\boldsymbol{z}^{(0)} = \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}$, so folgt

$\boldsymbol{A}$			$\boldsymbol{z}^{(0)}$	$\boldsymbol{z}^{(1)}$	$\boldsymbol{z}^{(2)}$	...
1	1	0	1	0	0	
1	1	0	−1	0	0	
0	0	−1	1	−1	1	
2	2	−1	—	−1	1	

und es gilt $q_3^{(\nu)} = -1 = \lambda_2$, $\boldsymbol{x}_2^* = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \approx \boldsymbol{x}_2$.
Im vorliegenden Fall ist $c_1 = 0$ wegen $\boldsymbol{z}^{(0)\mathsf{T}}\boldsymbol{x}_1 = 0$ (vgl. 1.b).

c) Wählt man als Ausgangsvektor $\boldsymbol{z}^{(0)} = \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix}$, so folgt aus $\boldsymbol{A}\boldsymbol{z}^{(0)} = \boldsymbol{z}^{(1)} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$,

bereits der EV $\boldsymbol{x}_3$ zum EW $\lambda_3 = 0$. Es liegt der Fall 1.c) vor mit $c_1 = c_2 = 0$. Ist $\boldsymbol{z}^{(0)}$ selbst EV zu λ_j, so gilt exakt $\boldsymbol{A}\boldsymbol{z}^{(\nu)} = \lambda_j \boldsymbol{z}^{(\nu)}$ für alle ν. □

Da die exakten Werte der EWe und EVen nicht bekannt sind, muss zur Sicherheit die Rechnung mit mehreren (theoretisch mit n) linear unabhängigen Ausgangsvektoren $\boldsymbol{z}^{(0)}$ durchgeführt werden, um aus den Ergebnissen auf den jeweils vorliegenden Fall schließen zu können. Für die Praxis gilt das jedoch nicht, denn mit wachsendem n wird die Wahrscheinlichkeit immer geringer, dass man zufällig ein $\boldsymbol{z}^{(0)}$ wählt, das z. B. keine Komponente in Richtung von $\boldsymbol{x}_1$ hat oder etwa bereits selbst ein EV ist.

Beispiel 7.4.

Gegeben: Die Matrix

$$\boldsymbol{A} = \begin{pmatrix} 1.2 & 2.3 & 3.2 \\ 2.3 & 3.2 & 4.4 \\ 3.2 & 4.4 & 5.1 \end{pmatrix}$$

Gesucht: Die Näherungen λ_1^* und $\boldsymbol{x}_1^*$ für den betragsgrößten EW λ_1 und den zugehörigen EV $\boldsymbol{x}_1$ nach dem Verfahren von v. Mises.

Lösung: (Die Rechnung wurde mit 15-stelliger Mantisse durchgeführt. Um die Rechnung besser nachvollziehen zu können, wurde auf eine Normierung der jeweiligen Vektoren verzichtet.)

$\boldsymbol{A}$			$\boldsymbol{z}^{(0)}$	$\boldsymbol{z}^{(1)}$	...	$\boldsymbol{z}^{(13)}$	$\boldsymbol{z}^{(14)}$
1.2	2.3	3.2	1	6.7		10359856639376.3	107173823041282
2.3	3.2	4.4	1	9.9		14949921641116.7	154658535558899
3.2	4.4	5.1	1	12.7		18861617281082.0	195125444600435

Man erhält hier $q_1^{(13)} = q_2^{(13)} = q_3^{(13)} = 10.3451067685560 = \lambda_1^* \approx \lambda_1$ und

$$\boldsymbol{x}_1^* = \frac{\boldsymbol{z}^{(14)}}{|\boldsymbol{z}^{(14)}|} = \begin{pmatrix} 0.395371930487755 \\ 0.570546445345872 \\ 0.719831779157857 \end{pmatrix} \approx \boldsymbol{x}_1.$$

□

2. $\lambda_1 = \lambda_2 = \ldots = \lambda_p, |\lambda_1| > |\lambda_{p+1}| \geq \ldots \geq |\lambda_n|$ **(mehrfacher EW)**:

Für $c_1x_{1,i} + c_2x_{2,i} + \ldots + c_px_{p,i} \neq 0$ ergeben sich zu p linear unabhängigen Ausgangsvektoren $\boldsymbol{z}^{(0)}$ die Beziehungen

$$q_i^{(\nu)} = \lambda_1 + O\left(\left|\frac{\lambda_{p+1}}{\lambda_1}\right|^{\nu}\right) \quad \text{bzw.} \quad \lim_{\nu\to\infty} q_i^{(\nu)} = \lambda_1.$$

$$\boldsymbol{z}^{(\nu)} \sim \lambda_1^{\nu}(c_1^{(r)}\boldsymbol{x}_1 + c_2^{(r)}\boldsymbol{x}_2 + \ldots + c_p^{(r)}\boldsymbol{x}_p) = \boldsymbol{y}_r, \quad r = 1(1)p,\ \nu = 0, 1, 2, \ldots$$

Die p Vektoren $\boldsymbol{y}_r$ sind linear unabhängig und spannen den sogenannten *Eigenraum* zu λ_1 auf; d. h. sie bilden eine Basis des Eigenraumes zu λ_1 (s. dazu [ZURM1997], S. 151). Als Näherung für λ_1 nimmt man für hinreichend großes ν wieder $\lambda_1^* = q_i^{(\nu)}$, für die EVen $\boldsymbol{x}_i$, i=1(1)p, erhält man hier keine Näherungen, sondern nur die Linearkombinationen $\boldsymbol{y}_r$.

Beispiel 7.5.

Gegeben: Die Matrix

$$\boldsymbol{A} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 0 & 1 \end{pmatrix}.$$

Gesucht: Die λ_i und $\boldsymbol{x}_i$ nach dem Verfahren von v. Mises. Exakte Lösungen sind: $\lambda_1 = \lambda_2 = 2$, $\lambda_3 = 0$;

$$\boldsymbol{x}_1 = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, \quad \boldsymbol{x}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad \boldsymbol{x}_3 = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}.$$

Lösung:

a) Wahl des Ausgangsvektors $\boldsymbol{z}^{(0)} = \begin{pmatrix} 2 \\ 3 \\ 4 \end{pmatrix}$

$\boldsymbol{A}$			$\boldsymbol{z}^{(0)}$	$\boldsymbol{z}^{(1)}$	$\boldsymbol{z}^{(2)}$	$\boldsymbol{z}^{(3)}$	...
1	0	1	2	6	12	24	
0	2	0	3	6	12	24	
1	0	1	4	6	12	24	
2	2	2	—	18	36	72	

So fortfahrend erhält man $q_i^{(\nu)} = 2$ für $i = 1, 2, 3$, $\nu = 1, 2, \ldots$.
Um weitere Aussagen machen zu können, wird zunächst die Rechnung mit einem neuen Ausgangsvektor wiederholt, der vom ersten linear unabhängig ist.

b) Wahl des Ausgangsvektors $\boldsymbol{z}^{(0)} = \begin{pmatrix} 2 \\ 3 \\ -1 \end{pmatrix}$

$\boldsymbol{A}$			$\boldsymbol{z}^{(0)}$	$\boldsymbol{z}^{(1)}$	$\boldsymbol{z}^{(2)}$	$\boldsymbol{z}^{(3)}$	...
1	0	1	2	1	2	4	
0	2	0	3	6	12	24	
1	0	1	−1	1	2	4	
2	2	2	—	8	16	32	

Man erhält wie unter a) $q_i^{(\nu)} = 2$ für $i = 1, 2, 3$, $\nu = 1, 2, \ldots$.
Allerdings streben die $\boldsymbol{z}^{(\nu)}$ hier gegen einen von den $\boldsymbol{z}^{(\nu)}$ in a) linear unabhängigen Vektor.

Es gilt somit $\lambda_1 = \lambda_2 = 2$; wenn die Rechnung bei $\nu = 3$ abbricht, erhält man

$$\boldsymbol{y}_1 = 24 \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \qquad \boldsymbol{y}_2 = 24 \begin{pmatrix} 0.1\overline{6} \\ 1 \\ 0.1\overline{6} \end{pmatrix};$$

$\boldsymbol{y}_1$ und $\boldsymbol{y}_2$ sind linear unabhängig. Es zeigt sich, dass hier zufällig $\boldsymbol{y}_1$ dem EV $\boldsymbol{x}_1$ entspricht und im Falle b) die normierten $\boldsymbol{z}^{(\nu)}$ gegen $c \cdot \boldsymbol{x}_2$ strebt. □

3. $\lambda_1 = -\lambda_2,\ |\lambda_1| > |\lambda_3| \geq \ldots \geq |\lambda_n|$:
Man bildet die Quotienten $\widetilde{q}_i^{(\nu)}$ der i-ten Komponenten der Vektoren $\boldsymbol{z}^{(\nu+2)}$ und $\boldsymbol{z}^{(\nu)}$

$$\widetilde{q}_i^{(\nu)} := \frac{z_i^{(\nu+2)}}{z_i^{(\nu)}}$$

und erhält mit $c_1 x_{1,i} + (-1)^\nu c_2 x_{2,i} \neq 0$

$$\widetilde{q}_i^{(\nu)} = \lambda_1^2 + O\left(\left|\frac{\lambda_3}{\lambda_1}\right|^\nu\right) \quad \text{bzw.} \quad \lim_{\nu\to\infty} \widetilde{q}_i^{(\nu)} = \lambda_1^2. \tag{7.11}$$

Für $\nu \to \infty$ ergibt sich das folgende asymptotische Verhalten

$$\begin{aligned} \boldsymbol{x}_1 &\sim \boldsymbol{z}^{(\nu+1)} + \lambda_1 \boldsymbol{z}^{(\nu)}, \\ \boldsymbol{x}_2 &\sim \boldsymbol{z}^{(\nu+1)} - \lambda_1 \boldsymbol{z}^{(\nu)}. \end{aligned}$$

Man erhält somit als Näherungen λ_1^*, λ_2^* für λ_1 und λ_2 für hinreichend großes ν wegen (7.11)

$$\lambda_{1,2}^* = \pm\sqrt{\widetilde{q}_i^{(\nu)}} \approx \lambda_{1,2}$$

und als Näherungen für $\boldsymbol{x}_1$ und $\boldsymbol{x}_2$

$$\begin{aligned} \boldsymbol{x}_1^* &= \frac{\boldsymbol{z}^{(\nu+1)} + \lambda_1^* \boldsymbol{z}^{(\nu)}}{\|\boldsymbol{z}^{(\nu+1)} + \lambda_1^* \boldsymbol{z}^{(\nu)}\|_2} \approx \boldsymbol{x}_1, \\ \boldsymbol{x}_2^* &= \frac{\boldsymbol{z}^{(\nu+1)} - \lambda_1^* \boldsymbol{z}^{(\nu)}}{\|\boldsymbol{z}^{(\nu+1)} - \lambda_1^* \boldsymbol{z}^{(\nu)}\|_2} \approx \boldsymbol{x}_2. \end{aligned}$$

Bei der praktischen Durchführung macht sich das Auftreten dieses Falles dadurch bemerkbar, dass gleiches Konvergenzverhalten nur für solche Quotienten eintritt, bei denen die zum Zähler und Nenner gehörigen Spalten durch genau eine Spalte des Rechenschemas getrennt sind. Die Fälle 2 und 3 gelten auch für betragsnahe EWe $|\lambda_i| \approx |\lambda_j|$ für $i \neq j$.

Beispiel 7.6.

Gegeben: Die Matrix

$$\boldsymbol{A} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & -2 & 0 \\ 1 & 0 & 1 \end{pmatrix}.$$

Gesucht: Der betragsgrößte EW und die zugehörigen EVen näherungsweise nach dem Verfahren von v. Mises. Exakte Lösungen: $\lambda_1 = 2$, $\lambda_2 = -2$, $\lambda_3 = 0$;

$$\boldsymbol{x}_1 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \frac{1}{\sqrt{2}}, \quad \boldsymbol{x}_2 = \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix}, \boldsymbol{x}_3 = \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix} \frac{1}{\sqrt{2}}.$$

Lösung:

$\boldsymbol{A}$			$\boldsymbol{z}^{(0)}$	$\boldsymbol{z}^{(1)}$	$\boldsymbol{z}^{(2)}$	$\boldsymbol{z}^{(3)}$	$\boldsymbol{z}^{(4)}$	...
1	0	1	2	1	2	4	8	
0	−2	0	1	−2	4	−8	16	
1	0	1	−1	1	2	4	8	
2	−2	2	—	0	8	0	32	

$\widetilde{q}_i^{(2)} = \frac{z_i^{(4)}}{z_i^{(2)}} = 4 = \lambda_1^{*2} = \lambda_1^2$, für $i = 1, 2, 3$, d. h. $\lambda_1 = 2, \lambda_2 = -2$.

Für die zugehörigen EVen erhält man

$$\boldsymbol{x}_1^* = \frac{\boldsymbol{z}^{(3)} + 2\boldsymbol{z}^{(2)}}{|\boldsymbol{z}^{(3)} + 2\boldsymbol{z}^{(2)}|} = \frac{1}{8\sqrt{2}} \begin{pmatrix} 8 \\ 0 \\ 8 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = \boldsymbol{x}_1,$$

$$\boldsymbol{x}_2^* = \frac{\boldsymbol{z}^{(3)} - 2\boldsymbol{z}^{(2)}}{|\boldsymbol{z}^{(3)} - 2\boldsymbol{z}^{(2)}|} = \frac{1}{16} \begin{pmatrix} 0 \\ -16 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix} = \boldsymbol{x}_2.$$

Im vorliegenden Fall hat man die exakten Werte der EWe und EVen erhalten. □

7.3.2 Bestimmung des betragskleinsten Eigenwertes

In (7.1) wird $\lambda = 1/\kappa \neq 0$ gesetzt. Dann lautet die transformierte EWA

$$\boldsymbol{A}^{-1}\boldsymbol{x} = \kappa \boldsymbol{x}.$$

Mit dem Verfahren nach v. Mises bestimmt man nach der Vorschrift

$$\boldsymbol{z}^{(\nu+1)} = \boldsymbol{A}^{-1}\boldsymbol{z}^{(\nu)} \tag{7.12}$$

den betragsgrößten EW $\widehat{\kappa}$ von $\boldsymbol{A}^{-1}$. Für den betragskleinsten EW $\widehat{\lambda}$ von $\boldsymbol{A}$ erhält man so die Beziehung

$$|\widehat{\lambda}| = \frac{1}{|\widehat{\kappa}|}.$$

Zur Bestimmung von $\boldsymbol{A}^{-1}$ kann der Gaußsche Algorithmus verwendet werden (Abschnitt 4.6). Die Berechnung von $\boldsymbol{A}^{-1}$ sollte aber besser umgangen werden, indem die Vektoren $\boldsymbol{z}^{(\nu+1)}$ jeweils aus der Beziehung $\boldsymbol{A}\boldsymbol{z}^{(\nu+1)} = \boldsymbol{z}^{(\nu)}$, die aus (7.12) folgt, berechnet werden – etwa mit Hilfe des Gaußschen Algorithmus, wobei die Zerlegung von $\boldsymbol{A}$ nur einmal, die Vorwärts- und Rückwärtselimination in jedem Iterationsschritt durchgeführt wird. Ist $\boldsymbol{A}$ symmetrisch und $\det \boldsymbol{A} = 0$, so verschwindet mindestens ein EW, so dass $\widehat{\lambda} = 0$ ist.

7.3.3 Bestimmung weiterer Eigenwerte und Eigenvektoren

$\boldsymbol{A}$ sei eine symmetrische Matrix, die EVen $\boldsymbol{x}_i$ seien orthonormiert. Dann gilt mit (7.9) $c_1 = \boldsymbol{z}^{(0)\mathsf{T}}\boldsymbol{x}_1$. Man bildet

$$\boldsymbol{y}^{(0)} := \boldsymbol{z}^{(0)} - c_1\boldsymbol{x}_1 = c_2\boldsymbol{x}_2 + c_3\boldsymbol{x}_3 + \ldots + c_n\boldsymbol{x}_n$$

und verwendet $\boldsymbol{y}^{(0)}$ als Ausgangsvektor für das Verfahren von v. Mises. Wegen

$$\boldsymbol{y}^{(0)\mathsf{T}}\boldsymbol{x}_1 = \boldsymbol{z}^{(0)\mathsf{T}}\boldsymbol{x}_1 - c_1 = 0$$

ist $\boldsymbol{y}^{(0)}$ orthogonal zu $\boldsymbol{x}_1$, und der Fall 1.b) des Abschnittes 7.3.1 tritt ein, d. h. die Quotienten $q_i^{(\nu)}$ streben gegen λ_2. Da $\boldsymbol{x}_1$ nur näherungsweise bestimmt wurde, wird $\boldsymbol{y}^{(0)}$ nicht vollständig frei von Komponenten in Richtung von $\boldsymbol{x}_1$ sein, so dass man bei jedem Schritt des Verfahrens die $\boldsymbol{y}^{(\nu)}$ von Komponenten in Richtung $\boldsymbol{x}_1$ säubern muss. Das geschieht, indem man

$$\widetilde{\boldsymbol{y}}^{(\nu)} = \boldsymbol{y}^{(\nu)} - (\boldsymbol{y}^{(\nu)\mathsf{T}}\boldsymbol{x}_1)\boldsymbol{x}_1 \quad \text{mit} \quad \widetilde{\boldsymbol{y}}^{(\nu)\mathsf{T}} = (\widetilde{y}_1^{(\nu)}, \widetilde{y}_2^{(\nu)}, \ldots, \widetilde{y}_n^{(\nu)})$$

bildet und danach $\boldsymbol{y}^{(\nu+1)} = \boldsymbol{A}\widetilde{\boldsymbol{y}}^{(\nu)}$ berechnet. So fortfahrend erhält man für hinreichend großes ν Näherungswerte λ_2^* für λ_2 und $\boldsymbol{x}_2^*$ für $\boldsymbol{x}_2$

$$\lambda_2^* = q_i^{(\nu)} = \frac{\widetilde{y}_i^{(\nu+1)}}{\widetilde{y}_i^{(\nu)}} \approx \lambda_2; \quad \boldsymbol{x}_2^* = \frac{\widetilde{\boldsymbol{y}}^{(\nu)}}{|\widetilde{\boldsymbol{y}}^{(\nu)}|} \approx \boldsymbol{x}_2.$$

Zur Berechnung weiterer EWe und EVen wird ganz analog vorgegangen. Sollen etwa der im Betrag drittgrößte EW und der zugehörige EV bestimmt werden, so wird als Ausgangsvektor mit bekannten EVen $\boldsymbol{x}_1, \boldsymbol{x}_2$

$$\boldsymbol{y}^{(0)} := \boldsymbol{z}^{(0)} - c_1\boldsymbol{x}_1 - c_2\boldsymbol{x}_2$$

gebildet mit $c_1 = \boldsymbol{z}^{(0)\mathsf{T}}\boldsymbol{x}_1, c_2 = \boldsymbol{z}^{(0)\mathsf{T}}\boldsymbol{x}_2$.

Da $\boldsymbol{x}_1, \boldsymbol{x}_2$ wieder nur näherungsweise durch $\boldsymbol{x}_1^*$, $\boldsymbol{x}_2^*$ gegeben sind, müssen hier die $\boldsymbol{y}^{(\nu)}$ entsprechend von Komponenten in Richtung von $\boldsymbol{x}_1, \boldsymbol{x}_2$ gesäubert werden usw. (siehe auch Gram-Schmidtsches-Orthogonalisierungsverfahren).

Beispiel 7.7. (Fortsetzung von Beispiel 7.4)

Gegeben: Die Matrix $\boldsymbol{A}$ und die Näherungen λ_1^* für λ_1 und x_1^* für x_1.

Gesucht: Die Näherungen λ_2^* für λ_2 und x_2^* für x_2 mit Hilfe des Verfahrens von v. Mises.

Lösung: Rechnung mit 15-stelliger Mantisse.

$\boldsymbol{A}$			$\boldsymbol{z}^{(0)}$	$\boldsymbol{y}^{(0)}$	$\boldsymbol{y}^{(1)}$
1.2	2.3	3.2	1	0.333501706900982	−0.194996003169674
2.3	3.2	4.4	1	0.038201241328359	−0.049910848322856
3.2	4.4	5.1	1	−0.213456533283150	0.146662604183852

$\widetilde{\boldsymbol{y}}^{(1)}$	...	$\widetilde{\boldsymbol{y}}^{(30)}$	$\boldsymbol{y}^{(31)} = \widetilde{\boldsymbol{y}}^{(31)}$
−0.194996003169750		$3.83422614855210\cdot10^{-7}$	$-2.44100858931504\cdot10^{-7}$
−0.049910848322967		$1.28184681022692\cdot10^{-7}$	$-8.16070558365899\cdot10^{-8}$
0.146662604183712		$-3.12197738471860\cdot10^{-7}$	$1.98756497830030\cdot10^{-7}$

Hiermit ergibt sich: $q_1^{(30)} = q_2^{(30)} = q_3^{(30)} = -0.636636571433538 = \lambda_2^* \approx \lambda_2$,

$$\boldsymbol{x}_2^* = \frac{\widetilde{\boldsymbol{y}}^{(31)}}{|\widetilde{\boldsymbol{y}}^{(31)}|} = \begin{pmatrix} -0.750638776992176 \\ -0.250951270123549 \\ 0.611199547201295 \end{pmatrix} \approx \boldsymbol{x}_2.$$

Analog erhält man bereits nach drei Iterationsschritten $\lambda_3^* = -0.208470197122534 \approx \lambda_3$ und

$$\boldsymbol{x}_3^* = \begin{pmatrix} -0.529360428307665 \\ 0.781984791237377 \\ -0.329055197824559 \end{pmatrix} \approx \boldsymbol{x}_3.$$

□

7.4 Konvergenzverbesserung mit Hilfe des Rayleigh-Quotienten im Falle hermitescher Matrizen

Für den betragsgrößten EW λ_1 einer hermiteschen Matrix lässt sich bei nur unwesentlich erhöhtem Rechenaufwand eine gegenüber (7.10) verbesserte Näherung angeben. Man benötigt dazu den Rayleigh-Quotienten.

Definition 7.8. (*Rayleigh-Quotient*)
Ist $\boldsymbol{A}$ eine beliebige (n,n)-Matrix und $\boldsymbol{x} \in \mathbb{R}^n$, so heißt

$$R[\boldsymbol{x}] = \frac{\overline{\boldsymbol{x}}^{\mathsf{T}} \boldsymbol{A} \boldsymbol{x}}{\overline{\boldsymbol{x}}^{\mathsf{T}} \boldsymbol{x}}$$

Rayleigh-Quotient von $\boldsymbol{A}$ an der Stelle $\boldsymbol{x}$.

Wegen $\boldsymbol{A}\boldsymbol{x}_i = \lambda_i \boldsymbol{x}_i$ gilt $R[\boldsymbol{x}_i] = \lambda_i$, d. h. der Rayleigh-Quotient zu einem EV $\boldsymbol{x}_i$ ist gleich dem zugehörigen EW λ_i. Ist $\boldsymbol{A}$ hermitesch, so gilt der

Satz 7.9.
Der Rayleigh-Quotient nimmt für einen der n EVen einer hermiteschen Matrix $\boldsymbol{A}$ seinen Extremalwert an. Für $|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n|$ gilt $|R[\boldsymbol{x}]| \leq |\lambda_1|$.

Beweis. Sind die EWe dem Betrage nach geordnet

$$|\lambda_1| \geq |\lambda_2| \geq \ldots \geq |\lambda_n|,$$

so folgt wegen $R[\boldsymbol{x}_i] = \lambda_i$ die Ungleichungskette

$$|R[\boldsymbol{x}_1]| = |\lambda_1| \geq |R[\boldsymbol{x}_2]| \geq \ldots \geq |R[\boldsymbol{x}_n]| = |\lambda_n|.$$

□

Setzt man $\boldsymbol{x} = \boldsymbol{H}\boldsymbol{y}$, wobei $\boldsymbol{H}$ die unitäre Eigenvektormatrix ist mit $\boldsymbol{H}^{-1} = \overline{\boldsymbol{H}}^{\mathsf{T}}$ und $\overline{\boldsymbol{H}}^{\mathsf{T}} \boldsymbol{A} \boldsymbol{H} = \boldsymbol{J}$, so erhält man ($\boldsymbol{J}$ ist Diagonalmatrix)

$$R[\boldsymbol{x}] = \frac{\overline{\boldsymbol{x}}^{\mathsf{T}} \boldsymbol{A} \boldsymbol{x}}{\overline{\boldsymbol{x}}^{\mathsf{T}} \boldsymbol{x}} = R[\boldsymbol{H}\boldsymbol{y}] = \frac{\overline{\boldsymbol{y}}^{\mathsf{T}} \overline{\boldsymbol{H}}^{\mathsf{T}} \boldsymbol{A} \boldsymbol{H} \boldsymbol{y}}{\overline{\boldsymbol{y}}^{\mathsf{T}} \overline{\boldsymbol{H}}^{\mathsf{T}} \boldsymbol{H} \boldsymbol{y}} = \frac{\overline{\boldsymbol{y}}^{\mathsf{T}} \boldsymbol{J} \boldsymbol{y}}{\overline{\boldsymbol{y}}^{\mathsf{T}} \boldsymbol{y}} = \frac{\lambda_1 y_1^2 + \lambda_2 y_2^2 + \ldots + \lambda_n y_n^2}{y_1^2 + y_2^2 + \ldots + y_n^2}$$

und daraus

$$\begin{cases} & |R[\boldsymbol{x}]| \leq |\lambda_1| \quad \text{mit} \\ \max\limits_{\boldsymbol{x} \in \mathbb{R}^n} & |R[\boldsymbol{x}]| = |R[\boldsymbol{x}_1]| = |\lambda_1|. \end{cases}$$

Der Rayleigh-Quotient zu dem iterierten Vektor $\boldsymbol{z}^{(\nu)}$ lautet

$$R[\boldsymbol{z}^{(\nu)}] = \frac{\overline{\boldsymbol{z}}^{(\nu)\mathsf{T}} \boldsymbol{z}^{(\nu+1)}}{\overline{\boldsymbol{z}}^{(\nu)\mathsf{T}} \boldsymbol{z}^{(\nu)}}.$$

Wegen Satz 7.9 gilt die Ungleichung $|R[\boldsymbol{z}^{(\nu)}]| \leq |\lambda_1|$, so dass man mit $|R[\boldsymbol{z}^{(\nu)}]|$ eine *untere Schranke* für $|\lambda_1|$ erhält.

Der Rayleigh-Quotient, gebildet zu der Näherung $\boldsymbol{z}^{(\nu)}$ für den EV $\boldsymbol{x}_1$, liefert einen besseren Näherungswert für den zugehörigen EW λ_1 als die Quotienten $q_i^{(\nu)}$. Es gilt nämlich

$$R[\boldsymbol{z}^{(\nu)}] = \lambda_1 + O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^{2\nu}\right),$$

hier ist die Ordnung des Restgliedes $O(|\lambda_2/\lambda_1|^{2\nu})$ im Gegensatz zur Ordnung $O(|\lambda_2/\lambda_1|^{\nu})$ bei dem Quotienten $q_i^{(\nu)}$.

Beispiel 7.10. (Fortsetzung von Beispiel 7.4)

Gegeben: Die Matrix und die iterierten Vektoren $\boldsymbol{z}^{(\nu)}$, $\nu = 1(1)14$.

Gesucht: Mit Hilfe des Rayleigh-Quotienten gebe man untere Schranken für $|\lambda_1|$ an.

Lösung:

$$\begin{array}{lcll} R[\boldsymbol{z}^{(1)}] & = & \frac{\overline{\boldsymbol{z}}^{(1)\mathsf{T}}\boldsymbol{z}^{(2)}}{\overline{\boldsymbol{z}}^{(1)\mathsf{T}}\boldsymbol{z}^{(1)}} & = 10.3428679443768, \\ \vdots & & & \\ R[\boldsymbol{z}^{(6)}] & = & \ldots = R[\boldsymbol{z}^{(13)}] & = 10.3451067685560. \end{array}$$

Wegen Satz 7.9 gilt $R[\boldsymbol{z}^{(6)}] \leq \lambda_1$, $(\lambda_1 > 0)$. □

7.5 Das Verfahren von Krylov

Es sei eine EWA (7.2) vorgelegt mit einer diagonalähnlichen reellen Matrix $\boldsymbol{A}$ (über den Fall nicht diagonalähnlicher Matrizen s. [ZURM1997], S.175); gesucht sind sämtliche EWe und EVen.

7.5.1 Bestimmung der Eigenwerte

1. Fall. Sämtliche EWe λ_i, $i = 1(1)n$, seien einfach.
Das charakteristische Polynom $P(\lambda)$ der Matrix $\boldsymbol{A}$ sei in der Form

$$P(\lambda) = \sum_{j=0}^{n-1} a_j \lambda^j + (-1)^n \lambda^n \tag{7.13}$$

dargestellt. Dann können die a_j aus dem folgenden linearen Gleichungssystem bestimmt werden:

$$\boldsymbol{Z}\,\boldsymbol{a} + (-1)^n \boldsymbol{z}^{(n)} = \boldsymbol{0} \tag{7.14}$$

mit

$$\begin{array}{lcl} \boldsymbol{Z} & = & (\boldsymbol{z}^{(0)}, \boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(n-1)}), \\ \boldsymbol{z}^{(\nu)} & = & \boldsymbol{A}\boldsymbol{z}^{(\nu-1)}, \quad \nu = 1(1)n, \\ \boldsymbol{a}^{\mathsf{T}} & = & (a_0, a_1, \ldots, a_{n-1}). \end{array}$$

Dabei ist $\boldsymbol{z}^{(0)}$ ein Ausgangsvektor mit der Darstellung (7.9), der bis auf die folgenden Ausnahmen willkürlich ist:

(a) $c_i \neq 0$ für $i = 1(1)n$: Dann ist $\det \boldsymbol{Z} \neq 0$ und das System (7.14) ist eindeutig lösbar. Einschließlich $\boldsymbol{z}^{(0)}$ gibt es n linear unabhängige Vektoren $\boldsymbol{z}^{(\nu)}$, $\nu = 1(1)n-1$.

(b) $c_i = 0$ für $i = q+1(1)n$, wobei die c_i o. B. d. A. so nummeriert werden: Dann gilt

$$\boldsymbol{z}^{(0)} = c_1\boldsymbol{x}_1 + c_2\boldsymbol{x}_2 + \ldots + c_q\boldsymbol{x}_q \quad \text{mit} \quad c_i \neq 0 \quad \text{für} \quad i = 1(1)q,\ q < n.$$

Die $q+1$ Vektoren $\boldsymbol{z}^{(0)}$ und $\boldsymbol{z}^{(\nu+1)} = \boldsymbol{A}\boldsymbol{z}^{(\nu)}$, $\nu = 0(1)q-1$, sind linear abhängig, d. h. es gilt: $\det \boldsymbol{Z} = 0$. Die (n, q)-Matrix

$$\boldsymbol{Z}_q = (\boldsymbol{z}^{(0)},\ \boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(q-1)})$$

besitzt den Rang q, so dass sich mit

$$\boldsymbol{b} = (b_0, b_1, \ldots, b_{q-1})^{\mathsf{T}}$$

das inhomogene lineare Gleichungssystem von n Gleichungen für q Unbekannte b_j, $j = 0(1)q-1$, ergibt

$$\boldsymbol{Z}_q\boldsymbol{b} + \boldsymbol{z}^{(q)} = \boldsymbol{0}, \tag{7.15}$$

von denen q widerspruchsfrei sind und ausgewählt werden können. Die b_j, $j = 0(1)q-1, b_q = 1$ sind die Koeffizienten eines Teilpolynoms $P_q(\lambda)$ von $P(\lambda)$:

$$P_q(\lambda) = \sum_{j=0}^{q} b_j \lambda^j. \tag{7.16}$$

Aus $P_q(\lambda) = 0$ lassen sich q der insgesamt n EWe λ_i bestimmen. Um sämtliche voneinander verschiedenen λ_i zu erhalten, muss das gleiche Verfahren für verschiedene (höchstens n) linear unabhängige $\boldsymbol{z}^{(0)}$ durchgeführt werden.

2. Fall. Es treten mehrfache EWe auf.
$\boldsymbol{A}$ besitze s verschiedene EWe λ_j, $j = 1(1)s$, $s < n$, der Vielfachheiten k_j mit $k_1 + k_2 + \ldots + k_s = n$; dann geht man so vor: Zunächst ist festzustellen, wieviele linear unabhängige iterierte Vektoren $\boldsymbol{z}^{(\nu+1)} = \boldsymbol{A}\boldsymbol{z}^{(\nu)}$, $\nu = 0, 1, 2, \ldots$, zu einem willkürlich gewählten Ausgangsvektor der Darstellung

$$\boldsymbol{z}^{(0)} = c_1\boldsymbol{x}_1 + c_2\boldsymbol{x}_2 + \ldots + c_s\boldsymbol{x}_s, \quad \boldsymbol{x}_r \quad \text{EV zu} \quad \lambda_r, \tag{7.17}$$

bestimmt werden können. Sind etwa

$$\boldsymbol{z}^{(0)}, \boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(s)}$$

linear unabhängig, so liefert das lineare Gleichungssystem von n Gleichungen für $s < n$ Unbekannte $\widehat{b}_j$

$$\left\{ \begin{array}{l} \widehat{\boldsymbol{Z}}\,\widehat{\boldsymbol{b}} + \boldsymbol{z}^{(s)} = \boldsymbol{0} \quad \text{mit} \quad \widehat{\boldsymbol{b}} = (\widehat{b}_0, \widehat{b}_1, \ldots, \widehat{b}_{s-1})^{\mathsf{T}} \\ \text{und} \quad \widehat{\boldsymbol{Z}} = (\boldsymbol{z}^{(0)}, \boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(s-1)}) \end{array} \right. \tag{7.18}$$

die Koeffizienten $\widehat{b}_j$ des Minimalpolynoms

$$m(\lambda) = \sum_{j=0}^{s-1} \widehat{b}_j \lambda^j + \lambda^s = \prod_{k=1}^{s} (\lambda - \lambda_k). \tag{7.19}$$

$m(\lambda)$ hat die s verschiedenen EWe von $\boldsymbol{A}$ als einfache Nullstellen. Sind in (7.17) einige der $c_i = 0$, so ist analog zu 1(b) vorzugehen.

7.5.2 Bestimmung der Eigenvektoren

1. Fall. Sämtliche EW λ_i, $i = 1(1)n$, seien einfach. Die EVen lassen sich als Linearkombinationen der iterierten Vektoren $\boldsymbol{z}^{(\nu)}$ gewinnen. Es gilt

$$\boldsymbol{x}_i = \sum_{j=0}^{n-1} \widetilde{a}_{ij} \boldsymbol{z}^{(j)},$$

wobei die $\widetilde{a}_{ij}$ die Koeffizienten des Polynoms

$$P_i(\lambda) = \frac{(-1)^n P(\lambda)}{\lambda - \lambda_i} = \sum_{j=0}^{n-1} \widetilde{a}_{ij} \lambda^j$$

sind. Die $\widetilde{a}_{ij}$ lassen sich leicht mit dem einfachen Horner-Schema bestimmen.

2. Fall. Es treten mehrfache EWe auf.
Das eben beschriebene Verfahren ist auch dann noch anwendbar. Hier erhält man jedoch zu einem Ausgangsvektor $\boldsymbol{z}^{(0)}$ jeweils nur einen EV, d. h. die Vielfachheit bleibt unberücksichtigt. Man muss deshalb entsprechend der Vielfachheit k_j des EWes λ_j genau k_j linear unabhängige Ausgangsvektoren $\boldsymbol{z}^{(0)}$ wählen und erhält damit alle k_j EVen zu λ_j.

Das Verfahren von Krylov sollte nur angewandt werden, wenn die Systeme (7.14), (7.15) und (7.18) gut konditioniert sind, da sonst Ungenauigkeiten bei der Bestimmung der Koeffizienten in (7.13), (7.16) und (7.19) zu wesentlichen Fehlern bei der Bestimmung der λ_j führen.

7.6 Bestimmung der Eigenwerte positiv definiter, symmetrischer, tridiagonaler Matrizen mit Hilfe des QD-Algorithmus

Für positiv definite *symmetrische* tridiagonale Matrizen $\boldsymbol{A}$ (vgl. Abschnitt 4.9.2) mit den Diagonalelementen d_i und den oberen Diagonalelementen $c_i \neq 0$, $i = 1(1)n{-}1$, lassen sich die Eigenwerte mit Hilfe des QD-Algorithmus bestimmen. Das *QD-Schema* ist zeilenweise auszufüllen und hat die Form:

ν	$e_0^{(\nu)}$	$q_1^{(\nu)}$	$e_1^{(\nu)}$	$q_2^{(\nu)}$	$e_2^{(\nu)}$	$q_3^{(\nu)}$	$e_3^{(\nu)}$	$\cdots$	$q_n^{(\nu)}$	$e_n^{(\nu)}$
1		$q_1^{(1)}$		0		0		$\cdots$	0	
	0		$e_1^{(1)}$		$e_2^{(1)}$		$e_3^{(1)}$	$\cdots$		0
2		$q_1^{(2)}$		$q_2^{(2)}$		$q_3^{(2)}$		$\cdots$	$q_n^{(2)}$	
	0		$e_1^{(2)}$		$e_2^{(2)}$		$e_3^{(2)}$	$\cdots$		0
3		$q_1^{(3)}$		$q_2^{(3)}$		$q_3^{(3)}$		$\cdots$	$q_n^{(3)}$	
	0		$e_1^{(3)}$		$e_2^{(3)}$		$e_3^{(3)}$	$\cdots$		0
$\downarrow$ ∞	$\downarrow$ 0	$\downarrow$ λ_1	$\downarrow$ 0	$\downarrow$ λ_2	$\downarrow$ 0	$\downarrow$ λ_3	$\downarrow$ 0	$\cdots$	$\downarrow$ λ_n	$\downarrow$ 0

Setzt man das QD-Schema mit den Werten

$$q_1^{(1)} = d_1, \quad e_k^{(1)} = c_k^2/q_k^{(1)}, \quad q_{k+1}^{(1)} = d_{k+1} - e_k^{(1)}, \quad k = 1(1)n{-}1,$$

für die beiden ersten Zeilen an und setzt $\mathrm{e}_0^{(\nu)} = \mathrm{e}_n^{(\nu)} = 0$, so erhält man die weiteren Zeilen des Schemas nach den Regeln

$$e_k^{(\nu+1)} = q_{k+1}^{(\nu)} e_k^{(\nu)} / q_k^{(\nu+1)}, \quad q_k^{(\nu+1)} = e_k^{(\nu)} + q_k^{(\nu)} - e_{k-1}^{(\nu+1)}.$$

Hierbei berechnet man für festes ν nacheinander

$$q_1^{(\nu)}, \quad e_1^{(\nu)}, \quad q_2^{(\nu)}, \quad e_2^{(\nu)}, \quad \ldots, \quad e_{n-1}^{(\nu)}, \quad q_n^{(\nu)}.$$

Dann sind durch $\lim\limits_{\nu\to\infty} q_k^{(\nu)} = \lambda_k$ die der Größe nach geordneten EWe von $\boldsymbol{A}$ gegeben. Es gilt auch $\lim\limits_{\nu\to\infty} \mathrm{e}_k^{(\nu)} = 0$. Die Matrix $\boldsymbol{A}$ hat lauter positive verschiedene EWe λ_i ([SCHW1972], S.139 und 168).

Bemerkung. Eine für DVA geeignete direkte Methode stellt die Jakobi-Methode in der ihr durch Neumann gegebenen Form dar ([RALS1979] Bd. I, Kap. 7; ferner [BERE1971] Bd. 2, §8.8; [FADD1979], §81; [SCHW1972], 4.4; [COLL1973] I, S.56 ff.; [SELD1979], 5.5).

7.7 Transformationen auf Hessenbergform, LR- und QR-Verfahren

LR- und QR-Verfahren dienen zur gleichzeitigen Berechnung sämtlicher Eigenwerte einer (n,n)-Matrix $\boldsymbol{A}$. Die Durchführbarkeit des LR-Verfahrens ist im Allgemeinen nicht gesichert, dagegen ist das QR-Verfahren immer durchführbar, wenn auch mit sehr großem Rechenaufwand.

Für beide Verfahren nimmt der Rechenaufwand stark ab, wenn die Matrix $\boldsymbol{A}$ vor Anwendung der Verfahren auf obere Hessenbergform transformiert wird. Die Hessenbergmatrix hat dann die gleichen Eigenwerte wie $\boldsymbol{A}$; nach Wilkinson werden bei der Transformation die Eigenwerte durch Rundungsfehler kaum gestört.

7.7.1 Transformation einer Matrix auf obere Hessenbergform

Jede (n,n)-Matrix $\boldsymbol{A} = (a_{ik}), a_{ik} \in \mathbb{R}$, lässt sich mit Hilfe von symmetrischen, orthogonalen Householdermatrizen (vgl. Satz 20) auf obere Hessenbergform $\widetilde{\boldsymbol{A}}$ transformieren:

$$\widetilde{\boldsymbol{A}} = \begin{pmatrix} * & * & * & \cdots & * \\ * & * & * & \cdots & * \\ 0 & * & * & \cdots & * \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & * & * \end{pmatrix} = (\widetilde{a}_{ik})$$

mit $\widetilde{a}_{ik} = 0$ für $i \geq k+2$.
Ist $\boldsymbol{A}$ symmetrisch, d. h. $\boldsymbol{A}^{\mathsf{T}} = \boldsymbol{A}$, so ist die zugehörige Hessenbergmatrix symmetrisch und tridiagonal.

Durchführung des Verfahrens

Man setzt zunächst $\boldsymbol{A}_1 := \boldsymbol{A} = (a_{ik}^{(1)})$, $i,k = 1(1)n$.
Als erste Transformationsmatrix wählt man

$$\boldsymbol{H}_1 := \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & & & \\ \vdots & & \widetilde{\boldsymbol{H}}_1 & \\ 0 & & & \end{pmatrix}$$

mit einer $(n–1, n–1)$-Householdermatrix $\widetilde{\boldsymbol{H}}_1$, die gemäß Satz 4.20 wie folgt gebildet wird:

$$\widetilde{\boldsymbol{H}}_1 = \boldsymbol{E}_{n-1} - \frac{2}{\|\boldsymbol{v}_1\|^2}\boldsymbol{v}_1\boldsymbol{v}_1^{\mathsf{T}}$$

mit

$$\boldsymbol{v}_1 = \begin{pmatrix} a_{21}^{(1)} + \operatorname{sign}(a_{21}^{(1)})\|\boldsymbol{a}_1^{(1)}\| \\ a_{31}^{(1)} \\ \vdots \\ a_{n1}^{(1)} \end{pmatrix}, \quad \boldsymbol{a}_1^{(1)} = \begin{pmatrix} a_{21}^{(1)} \\ a_{31}^{(1)} \\ \vdots \\ a_{n1}^{(1)} \end{pmatrix},$$

$\boldsymbol{v}_1, \boldsymbol{a}_1^{(1)} \in \mathbb{R}^{n-1}$, $\boldsymbol{a}_1^{(1)}$ ist die erste Spalte der Ausgangsmatrix $\boldsymbol{A}_1$ ohne $a_{11}^{(1)}$.
Mit

$$\boldsymbol{A}_1 = \begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \dots & a_{1n}^{(1)} \\ a_{21}^{(1)} & & & \\ \vdots & & \widetilde{\boldsymbol{A}}_1 & \\ a_{n1}^{(1)} & & & \end{pmatrix}$$

ergibt sich dann eine Matrix

$$\boldsymbol{A}_2 := \boldsymbol{H}_1\boldsymbol{A}_1\boldsymbol{H}_1 = \begin{pmatrix} a_{11}^{(1)} & * & \dots & * \\ * & & & \\ 0 & & \widetilde{\boldsymbol{H}}_1\widetilde{\boldsymbol{A}}_1\widetilde{\boldsymbol{H}}_1 & \\ \vdots & & & \\ 0 & & & \end{pmatrix}.$$

Auf $\widetilde{\boldsymbol{H}}_1\widetilde{\boldsymbol{A}}_1\widetilde{\boldsymbol{H}}_1$ wird das Verfahren erneut angewandt, und man erhält nach n–2 Transformationen die obere Hessenbergform. Es wird nach dem folgenden Algorithmus verfahren (s. auch [NIEM1987], 8.5):

Algorithmus 7.11. (*Transformation auf Hessenbergform*)

Gegeben: $\boldsymbol{A} =: \boldsymbol{A}_1 = (a_{ik}^{(1)}), \quad i,k = 1(1)n,$

Gesucht: $\boldsymbol{A}$ auf obere Hessenbergform transformiert .

Für jedes $i = 1(1)n{-}2$ sind dann folgende Schritte auszuführen:

1. Berechnung der $(n{-}i)$-reihigen Householdermatrix $\boldsymbol{H}_i$ nach der Vorschrift

$$\widetilde{\boldsymbol{H}}_i = \boldsymbol{E}_{n-i} - \frac{2}{\|\boldsymbol{v}_i\|^2}\boldsymbol{v}_i\boldsymbol{v}_i^{\mathsf{T}}$$

mit

$$\boldsymbol{v}_i = \begin{pmatrix} a^{(i)}_{i+1,i} + \operatorname{sign}(a^{(i)}_{i+1,i})\|\boldsymbol{a}^{(i)}_i\| \\ a^{(i)}_{i+2,i} \\ \vdots \\ a^{(i)}_{n,i} \end{pmatrix}, \quad \boldsymbol{a}^{(i)}_i = \begin{pmatrix} a^{(i)}_{i+1,i} \\ a^{(i)}_{i+2,i} \\ \vdots \\ a^{(i)}_{n,i} \end{pmatrix}.$$

2. Man setze

$$\boldsymbol{H}_i = \begin{pmatrix} \boldsymbol{E}_i & 0 \\ 0 & \widetilde{\boldsymbol{H}}_i \end{pmatrix} \begin{matrix} \} \\ \} \end{matrix} \begin{matrix} i \text{ Zeilen} \\ n-i \text{ Zeilen} \end{matrix}$$

und berechne $\boldsymbol{A}_{i+1} := \boldsymbol{H}_i \boldsymbol{A}_i \boldsymbol{H}_i = (a^{(i+1)}_{jk})$

Dann besitzt $\boldsymbol{A}_{n-1}$ obere Hessenbergform, und der Rechenaufwand beträgt nach [WERN1993] $\frac{5}{3}n^3 + O(n^2)$ Punktoperationen.

Beispiel 7.12.

Gegeben: Die (4,4)-Matrix $\boldsymbol{A}$

Gesucht: Die auf Hessenbergform transformierte Matrix zu $\boldsymbol{A}$

Lösung (mit Algorithmus 7.11):

$$\begin{aligned}
\boldsymbol{A} = \boldsymbol{A}_1 &= \begin{pmatrix} 1 & 2 & 3 & -4 \\ 4 & -5 & 6 & -3 \\ 1 & 0 & -3 & 1 \\ 2 & -2 & 0 & -3 \end{pmatrix} \\
\boldsymbol{H}_1 &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -0.8729 & -0.2182 & -0.4364 \\ 0 & -0.2182 & 0.9746 & -0.0509 \\ 0 & -0.4364 & -0.0509 & 0.8983 \end{pmatrix} \\
\boldsymbol{A}_2 &= \begin{pmatrix} 1 & -0.6547 & 2.6907 & -4.6186 \\ -4.5826 & -5.1905 & -5.7976 & 1.2797 \\ 0 & -0.8951 & -4.4763 & 1.2952 \\ 0 & 0.8285 & -2.4145 & -1.3332 \end{pmatrix} \\
\boldsymbol{H}_2 &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -0.7339 & 0.6793 \\ 0 & 0 & 0.6793 & 0.7339 \end{pmatrix} \\
\boldsymbol{A}_3 &= \begin{pmatrix} 1 & -0.6547 & -5.1120 & -1.5617 \\ -4.5826 & -5.1905 & 5.1240 & -2.9990 \\ 0 & 1.2196 & -2.4681 & -0.2448 \\ 0 & 0 & 3.4649 & -3.3415 \end{pmatrix}
\end{aligned}$$

Die Matrix $\boldsymbol{A}_3$ ist die auf obere Hessenbergform transformierte Matrix zu $\boldsymbol{A}$. □

Beispiel 7.13.

Gegeben: Die symmetrische (5,5)-Matrix $\boldsymbol{A}$

Gesucht: Die auf Hessenbergform transformierte Matrix zu $\boldsymbol{A}$

Lösung (mit Algorithmus 7.11):

$$\boldsymbol{A} = \boldsymbol{A}_1 = \begin{pmatrix} -7 & 4 & 3 & -4 & 5 \\ 4 & -5 & 0 & -3 & 4 \\ 3 & 0 & -2 & 1 & 3 \\ -4 & -3 & 1 & -3 & 2 \\ 5 & 4 & 3 & 2 & 1 \end{pmatrix}$$

$$\boldsymbol{H}_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & -0.4924 & -0.3693 & 0.4924 & -0.6155 \\ 0 & -0.3693 & 0.9086 & 0.1218 & -0.1523 \\ 0 & 0.4924 & 0.1218 & 0.8376 & 0.2031 \\ 0 & -0.6155 & -0.1523 & 0.2031 & 0.7462 \end{pmatrix}$$

$$\boldsymbol{A}_2 = \begin{pmatrix} -7 & -8.1240 & 0 & 0 & 0 \\ -8.1240 & 1.8333 & 0.2037 & -2.6924 & -1.3428 \\ 0 & 0.2037 & -2.3176 & 1.5668 & 1.0647 \\ 0 & -2.6924 & 1.5668 & -3.9468 & 4.8193 \\ 0 & -1.3428 & 1.0647 & 4.8193 & -4.5690 \end{pmatrix}$$

$$\boldsymbol{H}_2 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -0.0675 & 0.8928 & 0.4453 \\ 0 & 0 & 0.8928 & 0.2533 & -0.3724 \\ 0 & 0 & 0.4453 & -0.3724 & 0.8143 \end{pmatrix}$$

$$\boldsymbol{A}_3 = \begin{pmatrix} -7 & -8.1240 & 0 & 0 & 0 \\ -8.1240 & 1.8333 & -3.0155 & 0 & 0 \\ 0 & -3.0155 & -0.4837 & 0.6182 & 3.2447 \\ 0 & 0 & 0.6182 & -3.6429 & 2.7519 \\ 0 & 0 & 3.2447 & 2.7519 & -6.7067 \end{pmatrix}$$

$$\boldsymbol{H}_3 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -0.1872 & -0.9823 \\ 0 & 0 & 0 & -0.9823 & 0.1872 \end{pmatrix}$$

$$\boldsymbol{A}_4 = \begin{pmatrix} -7 & -8.1240 & 0 & 0 & 0 \\ -8.1240 & 1.8333 & -3.0155 & 0 & 0 \\ 0 & -3.0155 & -0.4837 & -3.3031 & 0 \\ 0 & 0 & -3.3031 & -5.5875 & 3.1224 \\ 0 & 0 & 0 & 3.1224 & -4.7622 \end{pmatrix}$$

Die Matrix $\boldsymbol{A}_4$ ist die auf obere Hessenbergform transformierte Matrix zu $\boldsymbol{A}$.
Da $\boldsymbol{A}$ symmetrisch ist, ist $\boldsymbol{A}_4$ tridiagonal. □

7.7.2 LR-Verfahren

Das LR-Verfahren läuft nach dem folgenden Algorithmus ab.

Algorithmus 7.14. (*LR-Verfahren*)

Gegeben: (n,n)-Matrix $\boldsymbol{A}$

Gesucht: Sämtliche Eigenwerte $\lambda_i, i = 1(1)n$, von $\boldsymbol{A}$

1. Setze $\boldsymbol{A}_1 := \boldsymbol{A}$
2. Führe für jedes $i = 1, 2, 3, ...$ durch:

   2.1 Die Faktorisierung $\boldsymbol{A}_i = \boldsymbol{L}_i \boldsymbol{R}_i$ (sofern durchführbar) mit einer normierten unteren Dreiecksmatrix $\boldsymbol{L}_i$ und einer oberen Dreiecksmatrix $\boldsymbol{R}_i$ (vgl. Abschnitt 4.2, Algorithmus 11).

   2.2 Die Matrizenmultiplikation $\boldsymbol{A}_{i+1} = \boldsymbol{R}_i \boldsymbol{L}_i$.
   Die Matrizen $\boldsymbol{A}_i$ sind ähnlich zu $\boldsymbol{A}$.

Dann gilt unter gewissen Voraussetzungen

$$\lim_{i\to\infty} \boldsymbol{A}_i = \lim_{i\to\infty} \boldsymbol{R}_i = \begin{pmatrix} \lambda_1 & \cdots & * \\ & \ddots & \vdots \\ 0 & & \lambda_n \end{pmatrix}, \lim_{i\to\infty} \boldsymbol{L}_i = \boldsymbol{E}.$$

Satz 7.15.
Sei $\boldsymbol{A}$ eine (n,n)-Matrix. Die LR-Zerlegung sei durchführbar und für ihre Eigenwerte gelte

$$|\lambda_1| > |\lambda_2| > \ldots > |\lambda_n| > 0 \, .$$

Dann gilt für die zu $\boldsymbol{A}$ ähnlichen Matrizen $\boldsymbol{A}_i$

$$\lim_{i\to\infty} \boldsymbol{A}_i = \lim_{i\to\infty} \boldsymbol{R}_i = \begin{pmatrix} \lambda_1 & \ldots & * \\ & \ddots & \vdots \\ 0 & & \lambda_n \end{pmatrix} \quad \text{und} \quad \lim_{i\to\infty} \boldsymbol{L}_i = \boldsymbol{E}.$$

Beispiel 7.16.

Gegeben: Die symmetrische (5,5)-Matrix $\boldsymbol{A}$

Gesucht: $\lim\limits_{i\to\infty} \boldsymbol{A}_i$

Lösung (mit Algorithmus 7.14):

$$
\begin{aligned}
\boldsymbol{A}_1 = \boldsymbol{A} &= \begin{pmatrix} -7 & 4 & 3 & -4 & 5 \\ 4 & -5 & 0 & -3 & 4 \\ 3 & 0 & -2 & 1 & 3 \\ -4 & -3 & 1 & -3 & 2 \\ 5 & 4 & 3 & 2 & 1 \end{pmatrix} \\
\boldsymbol{L}_1 &= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ -0.5714 & 1 & 0 & 0 & 0 \\ -0.4286 & -0.6316 & 1 & 0 & 0 \\ 0.5714 & 1.9474 & -11 & 1 & 0 \\ -0.7143 & -2.5263 & 25.7143 & -2.5714 & 1 \end{pmatrix} \\
\boldsymbol{R}_1 &= \begin{pmatrix} -7 & 4 & 3 & -4 & 5 \\ 0 & -2.7143 & 1.7143 & -5.2857 & 6.8571 \\ 0 & 0 & 0.3684 & -4.0526 & 9.4737 \\ 0 & 0 & 0 & -35 & 90 \\ 0 & 0 & 0 & 0 & 9.7143 \end{pmatrix} \\
\boldsymbol{A}_2 = \boldsymbol{R}_1\boldsymbol{L}_1 &= \begin{pmatrix} -16.4286 & -18.3158 & 175.5714 & -16.8571 & 5 \\ -7.1020 & -31.4135 & 236.1837 & -22.9184 & 6.8571 \\ -9.2406 & -32.0582 & 288.5564 & -28.4135 & 9.4737 \\ -84.2857 & -295.5263 & 2699.2857 & -266.4286 & 90 \\ -6.9388 & -24.5414 & 249.7959 & -24.9796 & 9.7143 \end{pmatrix} \\
\boldsymbol{L}_2 &= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0.4323 & 1 & 0 & 0 & 0 \\ 0.5625 & 0.9260 & 1 & 0 & 0 \\ 5.1304 & 8.5785 & 10.2336 & 1 & 0 \\ 0.4224 & 0.7153 & 1.4739 & 0.4738 & 1 \end{pmatrix} \\
\boldsymbol{R}_2 &= \begin{pmatrix} -16.4286 & -18.3158 & 175.5714 & -16.8571 & 5 \\ 0 & -23.4957 & 160.2845 & -15.6311 & 4.6957 \\ 0 & 0 & 41.3853 & -4.4581 & 2.3133 \\ 0 & 0 & 0 & -0.2297 & 0.3922 \\ 0 & 0 & 0 & 0 & 0.6485 \end{pmatrix} \\
\boldsymbol{A}_3 = \boldsymbol{R}_2\boldsymbol{L}_2 &= \begin{pmatrix} -9.9652 & 3.2232 & 10.4310 & -14.4881 & 5 \\ 1.7874 & -5.8114 & 7.2428 & -13.4062 & 4.6957 \\ 1.3830 & 1.7316 & -0.8279 & -3.3621 & 2.3133 \\ -1.0130 & -1.6903 & -1.7730 & -0.0439 & 0.3922 \\ 0.2739 & 0.4638 & 0.9558 & 0.3073 & 0.6485 \end{pmatrix} \\
&\vdots \\
\boldsymbol{L}_{69} &= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \\
\boldsymbol{R}_{69} &= \begin{pmatrix} -12.0902 & -0.6089 & 24.2068 & -16.4308 & 5 \\ 0 & -9.0643 & 20.4006 & -15.4969 & 4.7767 \\ 0 & 0 & 7.6109 & -7.8187 & 3.4037 \\ 0 & 0 & 0 & -3.3166 & 2.2114 \\ 0 & 0 & 0 & 0 & 0.8604 \end{pmatrix} \\
\boldsymbol{A}_{70} = \boldsymbol{R}_{69}\boldsymbol{L}_{69} &= \begin{pmatrix} -12.0902 & -0.6090 & 24.2068 & -16.4308 & 5 \\ 0 & -9.0643 & 20.4006 & -15.4969 & 4.7767 \\ 0 & 0 & 7.6109 & -7.8187 & 3.4037 \\ 0 & 0 & 0 & -3.3166 & 2.2114 \\ 0 & 0 & 0 & 0 & 0.8604 \end{pmatrix}
\end{aligned}
$$

Ab Matrix $\boldsymbol{A}_{70}$ ändern sich die Diagonalelemente nicht mehr. □

Die Voraussetzungen des Satzes 7.15 sind sehr stark. Nicht einmal für reguläre Matrizen ist eine LR-Zerlegung gesichert, da keine Zeilenvertauschungen zugelassen sind.
Deshalb ist es zu empfehlen, die Matrix zunächst auf Hessenbergform zu transformieren und dann den LR-Algorithmus anzuwenden. Die Matrizen $\boldsymbol{A}_i$ haben dann alle Hessenbergform, der Rechenaufwand für jede LR-Zerlegung wird deutlich geringer ($O(n^2)$ Punktoperationen); er wird besonders gering, wenn $\boldsymbol{A}$ symmetrisch ist, da dann die Hessenbergmatrizen symmetrisch und tridiagonal sind (Beispiel 7.16 wurde ohne Hessenbergform gerechnet).

7.7.3 QR-Verfahren

Die QR-Zerlegung einer Matrix $\boldsymbol{A}$ mit Hilfe der Householder-Transformation wurde in Abschnitt 4.13 besprochen; sie ist Grundlage für das QR-Verfahren. Die QR-Zerlegung ist dann eindeutig, wenn die (n,n)-Matrix $\boldsymbol{A}$ nicht singulär ist und die Vorzeichen der Diagonalelemente der Superdiagonalmatrix $\boldsymbol{R}$ fest vorgeschrieben sind (vgl. [WERN1993]). Will man die QR-Zerlegung auch für allgemeinere Eigenwertprobleme verwenden, etwa für komplexe Matrizen $\boldsymbol{A}$, so muss $\boldsymbol{Q}$ nicht nur orthogonal ($\boldsymbol{Q}^{-1} = \boldsymbol{Q}^{\mathsf{T}}$), sondern unitär vorausgesetzt werden ($\boldsymbol{Q}^{-1} = \overline{\boldsymbol{Q}}^{\mathsf{T}}$).
Das QR-Verfahren verläuft nun analog zum LR-Verfahren, lediglich wird die LR-Zerlegung durch eine QR-Zerlegung ersetzt. Auch hier ist es empfehlenswert, vor Anwendung des Verfahrens die (n,n)-Matrix $\boldsymbol{A}$ auf obere Hessenbergform zu transformieren, um den erheblichen Rechenaufwand herabsetzen zu können.

Algorithmus 7.17. (*QR-Verfahren von Rutishauser*)

Gegeben: (n,n)-Matrix $\boldsymbol{A}$

Gesucht: Sämtliche Eigenwerte von $\boldsymbol{A}$

1. Setze $\boldsymbol{A}_1 := \boldsymbol{A}$
2. Führe für jedes $i = 1, 2, 3, ...$ folgende Schritte durch:

   2.1 Faktorisierung $\boldsymbol{A}_i = \boldsymbol{Q}_i \boldsymbol{R}_i$ mit der unitären Matrix $\boldsymbol{Q}_i$ (d. h. $\boldsymbol{Q}_i^{-1} = \overline{\boldsymbol{Q}}_i^{\mathsf{T}}$) und der oberen Dreiecksmatrix $\boldsymbol{R}_i$.

   2.2 Die Matrizenmultiplikation $\boldsymbol{A}_{i+1} = \boldsymbol{R}_i \boldsymbol{Q}_i$.

Dann gilt unter gewissen Voraussetzungen (etwa für $|\lambda_1| > |\lambda_2| > ... > |\lambda_n| > 0$).

$$\lim_{i \to \infty} \boldsymbol{A}_i = \begin{pmatrix} \lambda_1 & \dots & * \\ & \ddots & \vdots \\ 0 & & \lambda_n \end{pmatrix}.$$

Konvergenzsätze zum QR-Verfahren siehe in [NIEM1987], 8.; [WERN1993], IV.; [WILK1996]. Zum QR-Verfahren nach Peters und Wilkinson s. Abschnitt 7.8.

7.8 Ermittlung der Eigenwerte und Eigenvektoren einer Matrix nach den Verfahren von Martin, Parlett, Peters, Reinsch und Wilkinson

Besitzt die Matrix $\boldsymbol{A} = (a_{ik})$, $i,k = 1(1)n$, keine spezielle Struktur, so kann man sie durch sukzessive auszuführende Transformationen in eine Form bringen, die eine leichte Bestimmung der Eigenwerte und Eigenvektoren zulässt.
Unter Verwendung der Arbeiten [MART1968], [PARL1969], [PETE1970] ergibt sich ein Algorithmus, der im Wesentlichen die folgenden Schritte beinhaltet:

1. Schritt. Vorbehandlung der Matrix $\boldsymbol{A}$ zur Konditionsverbesserung nach einem von B.N. Parlett und C. Reinsch angegebenen Verfahren [PARL1969].

2. Schritt. Transformation der Matrix $\boldsymbol{A}$ auf obere *Hessenbergform* $\boldsymbol{B}$ (s. [WERN1993], S.223) mit

$$\boldsymbol{B} = (b_{ik}) = \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ & \ddots & \ddots & \vdots \\ & & b_{nn-1} & b_{nn} \end{pmatrix}, \tag{7.20}$$

d. h. $b_{ik} = 0$ für $i > k+1$, nach einem Verfahren von R.S. Martin und J.H. Wilkinson [MART1968].
Gesucht ist zu der gegebenen Matrix eine nicht singuläre Matrix $\boldsymbol{C}$, so dass gilt

$$\boldsymbol{B} = \boldsymbol{C}^{-1}\boldsymbol{A}\boldsymbol{C}. \tag{7.21}$$

Diese Transformation gelingt durch Überführung des Systems (7.1) $\boldsymbol{A}\boldsymbol{x} = \lambda\boldsymbol{x}$ in ein dazu äquivalentes gestaffeltes System

$$\boldsymbol{B}\boldsymbol{y} = \lambda\boldsymbol{y} \qquad \text{mit} \qquad \boldsymbol{y} = \boldsymbol{C}^{-1}\boldsymbol{x} \tag{7.22}$$

in einer Weise, die dem Gaußschen Algorithmus, angewandt auf (4.2), entspricht. An die Stelle des bekannten Vektors $\boldsymbol{a}$ in (4.2) tritt hier der unbekannte Vektor $\lambda\boldsymbol{x}$.
Mit (7.21) folgt

$$\det(\boldsymbol{B} - \lambda\boldsymbol{E}) = \det(\boldsymbol{A} - \lambda\boldsymbol{E}),$$

d. h. $\boldsymbol{B}$ und $\boldsymbol{A}$ besitzen dieselben Eigenwerte λ_i.

Wegen der einfachen Gestalt (7.20) von $\boldsymbol{B}$ lassen sich die λ_i damit leichter bestimmen.

3. Schritt. Die Bestimmung der Eigenwerte λ_i wird nun mit dem *QR-Algorithmus* nach G. Peters und J.H. Wilkinson [PETE1970] vorgenommen. Ausgehend von $\boldsymbol{B}_1 := \boldsymbol{B}$ wird eine Folge $\{\boldsymbol{B}_s\}$, $s = 1,2,3,...$, von oberen Hessenbergmatrizen konstruiert, die gegen eine obere Dreiecksmatrix $\boldsymbol{R} = (r_{ik})$, $i,k = 1(1)n$, konvergiert (Konvergenzbedingungen s. [WERN1993], S.255). Es gilt dann für alle i die Beziehung: $r_{ii} = \lambda_i$.

Mit $\boldsymbol{B}_1 := \boldsymbol{B}$ lautet die *Konstruktionsvorschrift* für jedes $s = 1, 2, 3, ...$:

(i) $\boldsymbol{B}_s - k_s\boldsymbol{E} = \boldsymbol{Q}_s\boldsymbol{R}_s$,

(ii) $\boldsymbol{B}_{s+1} = \boldsymbol{R}_s\boldsymbol{Q}_s + k_s\boldsymbol{E}$.

Die Vorschrift (i) beinhaltet die Zerlegung der Hessenbergmatrix $\boldsymbol{B}_s - k_s\boldsymbol{E}$ in das Produkt aus einer Orthogonalmatrix $\boldsymbol{Q}_s(\boldsymbol{Q}_s^{\mathsf{T}} = \boldsymbol{Q}_s^{-1})$ und einer oberen Dreiecksmatrix $\boldsymbol{R}_s$. Danach wird $\boldsymbol{B}_{s+1}$ nach der Vorschrift (ii) gebildet, $\boldsymbol{B}_{s+1}$ anstelle von $\boldsymbol{B}_s$ gesetzt und zu (i) zurückgegangen. Durch geeignete Wahl des sogenannten *Verschiebungsparameters* k_s wird eine erhebliche Konvergenzbeschleunigung erreicht. Mit $k_s = 0$ für alle s ergibt sich der QR-Algorithmus von Rutishauser in Abschnitt 7.7.3.

4. Schritt. Die Bestimmung der Eigenvektoren erfolgt ebenfalls nach [PETE1970]. Wegen (7.22) gilt

$$\boldsymbol{B}\boldsymbol{y}_i = \lambda_i\boldsymbol{y}_i \quad \text{mit} \quad \boldsymbol{x}_i = \boldsymbol{C}\boldsymbol{y}_i.$$

Zu jedem λ_i lassen sich daraus rekursiv bei willkürlich gegebenem y_{in} die Komponenten y_{ik}, $k = n-1(-1)1$, von $\boldsymbol{y}_i$ berechnen. Mit $\boldsymbol{x}_i = \boldsymbol{C}\boldsymbol{y}_i$ ergeben sich die gesuchten Eigenvektoren $\boldsymbol{x}_i$, $i = 1(1)n$.

5. Schritt. Normierung der Eigenvektoren $\boldsymbol{x}_i$.

7.9 Entscheidungshilfen

Das Verfahren von v. Mises kann dann verwendet werden, wenn man im Falle diagonalähnlicher Matrizen nur etwa den betragsgrößten oder den betragskleinsten Eigenwert und den zugehörigen Eigenvektor zu ermitteln hat. Will man jedoch sämtliche Eigenwerte und Eigenvektoren einer Matrix berechnen, so ist die Transformation der Matrix auf obere Hessenbergform mit anschließender Anwendung des QR-Verfahrens zu empfehlen.

7.10 Anwendungsbeispiel

Beispiel 7.18.

Gegeben: Der 2-Massen-Schwinger:

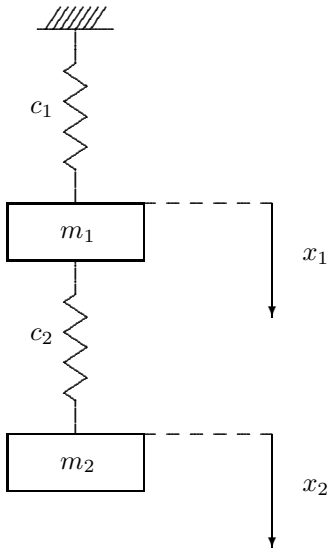


Abb. 7.1. Zweimassenschwinger

Bei Auslenkung einer oder beider Massen treten gekoppelte Schwingungen auf. Wird z. B. nur m_2 ausgelenkt, so wandert die Energie der Masse m_2 auf m_1, von dort allmählich wieder auf m_2 etc.

Gesucht: Die Eigenwerte der Bewegungsgleichungen.

Lösung:

Die Differentialgleichungen für die ungedämpfte Bewegung lauten:

$$\begin{array}{rcrcrcl} m_1\ddot{x}_1 & + & (c_1+c_2)x_1 & - & c_2x_2 & = & 0 \\ m_2\ddot{x}_2 & - & c_2x_1 & + & c_2x_2 & = & 0 \end{array}$$

$$\begin{pmatrix} m_1 & 0 \\ 0 & m_2 \end{pmatrix}\begin{pmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{pmatrix} + \begin{pmatrix} c_1+c_2 & -c_2 \\ -c_2 & c_2 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} ,$$

bzw. abgekürzt mit den beiden (2,2)-Matrizen $\boldsymbol{M}$ und $\boldsymbol{C}$ und den Vektoren $\boldsymbol{x}$ und $\ddot{\boldsymbol{x}}$

$$\boldsymbol{M}\,\ddot{\boldsymbol{x}} + \boldsymbol{C}\boldsymbol{x} = \boldsymbol{0}\,. \tag{7.23}$$

Als Lösungsansatz wird

$$\boldsymbol{x}(t) = \boldsymbol{a}\sin(\omega t)$$

gewählt. Differentiation ergibt

$$\dot{\boldsymbol{x}}\,(t) = \boldsymbol{a}\omega\cos(\omega t)$$

und

$$\ddot{\boldsymbol{x}}\,(t) = -\boldsymbol{a}\omega^2\sin(\omega t) = -\omega^2\boldsymbol{x}(t) \quad =: \quad -\lambda\,\boldsymbol{x}(t).$$

Die Vektoren $\boldsymbol{x}$ und $\ddot{\boldsymbol{x}}$ werden in (18) eingesetzt und führen auf

$$-\lambda\,\boldsymbol{M}\boldsymbol{x}(t) + \boldsymbol{C}\boldsymbol{x}(t) = \boldsymbol{0} \quad \text{bzw.} \quad (\boldsymbol{C} - \lambda\,\boldsymbol{M})\,\boldsymbol{x}(t) = \boldsymbol{0}.$$

Nur für $\det(\boldsymbol{C} - \lambda\,\boldsymbol{M}) = 0$ gibt es nichttriviale Lösungen. Es sind daher die entsprechenden Werte für λ zu bestimmen.

$$\det(\boldsymbol{C} - \lambda\,\boldsymbol{M}) = \begin{vmatrix} c_1 + c_2 - \lambda\,m_1 & -c_2 \\ -c_2 & c_2 - \lambda\,m_2 \end{vmatrix} = 0$$

$$\Longleftrightarrow \quad \lambda^2 - \left(\frac{c_1 + c_2}{m_1} + \frac{c_2}{m_2}\right)\lambda + \frac{c_1 + c_2}{m_1}\cdot\frac{c_2}{m_2} - \frac{c_2^2}{m_1 m_2} = 0$$

Mit

$$\frac{c_2^2}{m_1\,m_2} = \underbrace{\frac{c_1 + c_2}{m_1}}_{\omega_1^{*2}}\,\underbrace{\frac{c_2}{m_2}}_{\omega_2^{*2}}\,\frac{c_2}{c_1 + c_2}$$

und der Abkürzung

$$K_A^2 = \frac{c_2}{c_1 + c_2},$$

wobei aufgrund des physikalischen Hintergrunds (positive Federkonstanten)

$$0 \le K_A^2 \le 1$$

gilt, ergeben sich die Lösungen

$$\left.\begin{matrix}\omega_1^2 \\ \omega_2^2\end{matrix}\right\} = \frac{1}{2}\left\{\omega_1^{*2} + \omega_2^{*2} \pm \sqrt{(\omega_1^{*2} - \omega_2^{*2})^2 + 4K_A^2\omega_1^{*2}\omega_2^{*2}}\right\}.$$

Hieraus ergeben sich sofort die Beziehungen

$$0 \le \omega_2^2 \le \min(\omega_1^{*2}, \omega_2^{*2}) \quad \text{und} \quad \omega_1^2 + \omega_2^2 = \omega_1^{*2} + \omega_2^{*2}.$$

Die Schwingung wird beschrieben durch

$$\boldsymbol{x}(t) = \boldsymbol{a}\sin(\omega_1 t) + \boldsymbol{b}\sin(\omega_2 t)$$

mit den beiden Komponenten

$$x_1(t) = a_1 \sin(\omega_1 t) + b_1 \sin(\omega_2 t)$$

$$x_2(t) = a_2 \sin(\omega_1 t) + b_2 \sin(\omega_2 t)\,.$$

Die vier Konstanten a_i, b_i, die Amplituden der jeweiligen Elementarschwingungen, ergeben sich aus den Anfangsbedingungen, d. h. aus dem Anfangszustand des Massenschwingers.

Die Größen ω_1^{*2} und ω_2^{*2}, die die Frequenzen des Systems festlegen, haben folgende physikalische Bedeutung:

ω_1^{*2}	ist die Kreis-Eigenfrequenz der Masse m_1 bei festgehaltener Masse m_2 (d. h. für $x_2 = 0$) und
ω_2^{*2}	ist die Kreis-Eigenfrequenz der Masse m_2 bei festgehaltener Masse m_1 (d. h. für $x_1 = 0$).

Bei ω_1 $(< \omega_2)$ schwingen beide Massen im Gleichtakt.
Bei ω_2 schwingen die Massen im Gegentakt. □

Es gibt noch einen anderen Lösungsweg mit Hilfe der Cholesky-Zerlegung:

$$\boldsymbol{C}\,\boldsymbol{x} = \lambda\,\boldsymbol{M}\,\boldsymbol{x} \qquad \text{mit} \qquad \lambda = \omega^2\,.$$

Da $\boldsymbol{M}$ nicht singulär und eigentlich positiv definit ist, lässt sich eine Cholesky-Zerlegung anwenden.

$$\begin{aligned}
\boldsymbol{M} &= \boldsymbol{R}^{\mathsf{T}}\,\boldsymbol{R} \quad \text{mit} \quad \boldsymbol{R} = \begin{pmatrix} r_{11} & \cdots & r_{1n} \\ & \ddots & r_{nn} \end{pmatrix} \\
\boldsymbol{C}\,\boldsymbol{x} &= \lambda\,\boldsymbol{R}^{\mathsf{T}}\,\underbrace{\boldsymbol{R}\,\boldsymbol{x}}_{=:\boldsymbol{z}} = \lambda\,\boldsymbol{R}^{\mathsf{T}}\,\boldsymbol{z} \qquad (2) \\
&\qquad\qquad \Downarrow \\
&\qquad\quad \boldsymbol{x} = \boldsymbol{R}^{-1}\,\boldsymbol{z} \qquad (1) \\
\Rightarrow \boldsymbol{C}\,\boldsymbol{x} &\overset{(1)}{=} \boldsymbol{C}\,\boldsymbol{R}^{-1}\,\boldsymbol{z} \overset{(2)}{=} \lambda\,\boldsymbol{R}^{\mathsf{T}}\,\boldsymbol{z} \\
&\underbrace{(\boldsymbol{R}^{\mathsf{T}})^{-1}\,\boldsymbol{C}\,\boldsymbol{R}^{-1}}_{\boldsymbol{A}}\,\boldsymbol{z} = \lambda\,\underbrace{(\boldsymbol{R}^{\mathsf{T}})^{-1}\,\boldsymbol{R}^{\mathsf{T}}}_{\boldsymbol{E}}\,\boldsymbol{z} = \lambda\,\boldsymbol{z} \\
&\boldsymbol{A}\,\boldsymbol{z} = \lambda\,\boldsymbol{z}
\end{aligned}$$

Dies ist eine spezielle Eigenwertaufgabe (EWA).

Algorithmus 7.19. (*n-Massen-Schwinger*)

Gegeben: $\boldsymbol{M}$, $\boldsymbol{C}$, $\boldsymbol{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$.

$\boldsymbol{M}$, $\boldsymbol{C}$ sind symmetrische $(n \times n)$–Matrizen, $\boldsymbol{M}$ ist positiv definit.

Gesucht: $\boldsymbol{x}(t)$

DGL-System: $\boldsymbol{M}\ \ddot{\boldsymbol{x}} + \boldsymbol{C}\,\boldsymbol{x} = \boldsymbol{0}$

Mit $\boldsymbol{x} = \boldsymbol{a} \sin \omega t \Rightarrow$ allgemeine EWA

$$\boldsymbol{C}\,\boldsymbol{x} = \lambda\,\boldsymbol{M}\,\boldsymbol{x} \quad \text{mit} \quad \lambda = \omega^2$$

Cholesky-Zerlegung: $\boldsymbol{M} = \boldsymbol{R}^{\mathsf{T}}\,\boldsymbol{R}$ mit $\boldsymbol{R} = \begin{pmatrix} r_{11} & \cdots & r_{1n} \\ & \ddots & r_{nn} \end{pmatrix}$

Spezielle EWA:

$$\boldsymbol{A}\,\boldsymbol{z} = \lambda\,\boldsymbol{z} \quad \text{mit} \quad \boldsymbol{z} = \boldsymbol{R}\,\boldsymbol{x}\,, \quad \boldsymbol{A} = (\boldsymbol{R}^{\mathsf{T}})^{-1}\,\boldsymbol{C}\,\boldsymbol{R}^{-1}$$

Lösung:

$$\boldsymbol{x}(t) = \sum_{k=1}^{n} \boldsymbol{c_k} \sin \omega_k\, t$$

Bei Schwingerkette mit n Massen ist $\boldsymbol{A}$ eine symmetrische tridiagonale Matrix mit positiven Diagonalelementen und negativen Nebendiagonalelementen, die diagonaldominant ist. Daraus folgt, dass die Matrix positiv definit ist.

Beispiel 7.20. (Fortsetzung von Beispiel 7.18; Lösung nach Algorithmus 7.19)

Das Eigenwertproblem ist

$$\boldsymbol{C}\,\boldsymbol{x} = \lambda\,\boldsymbol{M}\,\boldsymbol{x}\,.$$

Zerlegt man die Matrix $\boldsymbol{M}$ in

$$\begin{array}{ccccc} \boldsymbol{M} & = & \boldsymbol{R}^{\mathsf{T}} & & \boldsymbol{R} \\ \begin{pmatrix} m_1 & 0 \\ 0 & m_2 \end{pmatrix} & = & \begin{pmatrix} \sqrt{m_1} & 0 \\ 0 & \sqrt{m_2} \end{pmatrix} & & \begin{pmatrix} \sqrt{m_1} & 0 \\ 0 & \sqrt{m_2} \end{pmatrix}, \end{array}$$

so wird mit $(\boldsymbol{R}^{\mathsf{T}})^{-1} = \boldsymbol{R}^{-1} = \frac{1}{\sqrt{m_1\,m_2}} \begin{pmatrix} \sqrt{m_2} & 0 \\ 0 & \sqrt{m_1} \end{pmatrix}$ das Eigenwertproblem zu $\boldsymbol{A}\,\boldsymbol{x} = \lambda\,\boldsymbol{x}$ mit

$$
\begin{aligned}
\boldsymbol{A} &= (\boldsymbol{R}^{\mathsf{T}})^{-1}\,\boldsymbol{C}\,\boldsymbol{R}^{-1} \\
&= \frac{1}{\sqrt{m_1\,m_2}} \begin{pmatrix} \sqrt{m_2} & 0 \\ 0 & \sqrt{m_1} \end{pmatrix} \begin{pmatrix} c_1+c_2 & -c_2 \\ -c_2 & c_2 \end{pmatrix} \frac{1}{\sqrt{m_1\,m_2}} \begin{pmatrix} \sqrt{m_2} & 0 \\ 0 & \sqrt{m_1} \end{pmatrix} \\
&= \frac{1}{m_1\,m_2} \begin{pmatrix} \sqrt{m_2} & 0 \\ 0 & \sqrt{m_1} \end{pmatrix} \begin{pmatrix} c_1+c_2 & -c_2 \\ -c_2 & c_2 \end{pmatrix} \begin{pmatrix} \sqrt{m_2} & 0 \\ 0 & \sqrt{m_1} \end{pmatrix} \\
&= \frac{1}{m_1\,m_2} \begin{pmatrix} \sqrt{m_2} & 0 \\ 0 & \sqrt{m_1} \end{pmatrix} \begin{pmatrix} (c_1+c_2)\sqrt{m_2} & -c_2\sqrt{m_1} \\ -c_2\sqrt{m_2} & c_2\sqrt{m_1} \end{pmatrix} \\
&= \frac{1}{m_1\,m_2} \begin{pmatrix} (c_1+c_2)\,m_2 & -c_2\sqrt{m_1\,m_2} \\ -c_2\sqrt{m_1\,m_2} & c_2\,m_1 \end{pmatrix} \\
&= \begin{pmatrix} \dfrac{c_1+c_2}{m_1} & -\dfrac{c_2}{\sqrt{m_1\,m_2}} \\ -\dfrac{c_2}{\sqrt{m_1\,m_2}} & \dfrac{c_2}{m_2} \end{pmatrix}
\end{aligned}
$$

Daraus folgt:

$$
\begin{aligned}
\det(\boldsymbol{A}-\lambda\,\boldsymbol{E}) &= \left(\frac{c_1+c_2}{m_1}-\lambda\right)\left(\frac{c_2}{m_2}-\lambda\right) - \frac{c_2^2}{m_1\,m_2} \\
&= \frac{c_2}{m_2}\,\frac{c_1+c_2}{m_1} - \lambda\,\frac{c_2}{m_2} - \lambda\,\frac{c_1+c_2}{m_1} + \lambda^2 - \frac{c_2^2}{m_1\,m_2} \\
&= \lambda^2 - \left(\frac{c_2}{m_2}+\frac{c_1+c_2}{m_1}\right)\lambda + \frac{c_1+c_2}{m_1}\,\frac{c_2}{m_2} - \frac{c_2^2}{m_1\,m_2} \;=\; 0\,.
\end{aligned}
$$

Man erhält also dieselben Eigenwerte wie auf die vorige Weise! □

Ergänzende Literatur zu Kapitel 7

[GOLU1996]; [GOOS1988]; [HAMM1994], 3.; [KELL1990]; [MAES1985], 4, 4.4; [OPFE2002], Kap.9; [PLAT2000], Kap.12,13; [PREU2001], Kap.5; [QUAR2002], Kap.5; [RALS2001], 10.3, 10.5; [SCHW1997], 6.; [SPEL1985], 2.1, 2.5, S.102-105; [STOE1990], 6.1; [STOE1999] Bd.2, Kap.6; [TORN1990] Bd.2, TEIL IV; [WERN1980].

Kapitel 8

Lineare und nichtlineare Approximation

8.1 Aufgabenstellung und Motivation

Es sei $f : [a, b] \to \mathbf{R}$ eine auf $[a, b]$ stetige Funktion, die durch eine sogenannte *Approximationsfunktion* $\Phi \in C[a, b]$ angenähert werden soll. Φ ist von $x \in [a, b]$ und von $n + 1$ freien Parametern $c_0, c_1, ..., c_n$ abhängig:

$$\Phi(x) := \Phi(x, c_0, c_1, \ldots, c_n) = \Phi(x, \boldsymbol{c}), \quad \boldsymbol{c} = (c_0, c_1, \ldots, c_n)^{\mathsf{T}}.$$

Die Parameter $c_0, c_1, ..., c_n$ sind so zu bestimmen, dass der Abstand zwischen f und Φ in noch vorzuschreibender Weise minimiert wird. Auf die Stetigkeit von f und Φ kann meist verzichtet werden, etwa wenn Eindeutigkeitsaussagen keine Rolle spielen.

Es werden zwei Aufgabenstellungen unterschieden:

1. Eine gegebene Funktion f ist durch eine Funktion Φ zu ersetzen, deren formelmäßiger Aufbau für den geforderten Zweck besser geeignet ist, d. h. die sich z. B. einfacher differenzieren oder integrieren lässt oder deren Funktionswerte leichter berechenbar sind. Man spricht hier von *kontinuierlicher* Approximation.

2. Eine empirisch gegebene Funktion f, von der endlich viele Wertepaare $\big(x_i, f(x_i)\big)$ an den (paarweise verschiedenen) diskreten Stützstellen x_i bekannt sind, ist durch eine formelmäßig gegebene Funktion Φ zu ersetzen. Hier spricht man von *diskreter* Approximation.

Je nach Art des Ansatzes für die Approximationsfunktion Φ wird außerdem zwischen *linearer* und *nichtlinearer* Approximation unterschieden. Die Approximation heißt linear, wenn der Ansatz für Φ die Form

$$\Phi(x, \boldsymbol{c}) = c_0\varphi_0(x) + c_1\varphi_1(x) + \ldots + c_n\varphi_n(x) = \sum_{k=0}^{n} c_k\varphi_k(x)$$

mit gegebenen, linear unabhängigen Funktionen $\varphi_k \in C[a, b]$ besitzt, andernfalls heißt sie nichtlinear. Die Funktion

$$\Phi_1(x, c_0, c_1, c_2, c_3) = c_0 + c_1\,\mathrm{e}^{2x} + c_2\,\ln\,x + c_3\,(x^2 + 1)$$

beschreibt einen linearen Approximationsansatz, dagegen lässt sich z. B. die Funktion

$$\Phi_2(x, c_0, c_1, c_2, c_3, c_4) = c_0 \cosh(c_1 x) + c_2 \, \mathrm{e}^{-c_3(x-c_4)^3}$$

nicht in der Form $\Sigma c_k \varphi_k(x)$ darstellen; es handelt sich bei Φ_2 um einen nichtlinearen Approximationsansatz (oder ein nichtlineares Modell). Die lineare Approximation ist wesentlich einfacher handzuhaben und wird deshalb bevorzugt eingesetzt.

Behandelt werden im Folgenden die kontinuierliche Approximation und die diskrete Approximation im quadratischen Mittel ((diskrete) L_2-Approximation oder (diskrete) Gaußsche Fehlerquadratmethode oder (diskreter) Ausgleich genannt) und die gleichmäßige Approximation durch Tschebyscheff-Polynome.

Beim diskreten Ausgleich wird zusätzlich der Einsatz der Householder-Transformation behandelt, um eine direkte Lösung der oft schlecht konditionierten Normalgleichungen zu umgehen und damit eine größere Genauigkeit der ermittelten Lösungen erwarten zu können.

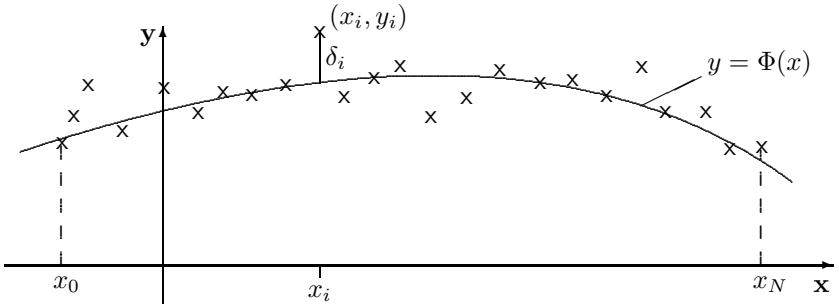


Abb. 8.1. Über $N+1$ Wertepaare $(x_0, y_0), \ldots, (x_N, y_N)$ mit $y_k = f(x_k)$ gegebene Funktion f mit Approximationsfunktion Φ

Beispiel 8.1. (*Anwendungsbeispiel aus der Kunststofftechnik*)

Gegeben: Ein T-Träger aus Plexiglas wird einer Vierpunktbiegung ausgesetzt. Bei der Biegung werden die Dehnungen auf der Gurtplatte in der Mitte (ε_M) und am Rand (ε_R) mittels Dehnmessstreifen (DMS) gemessen.

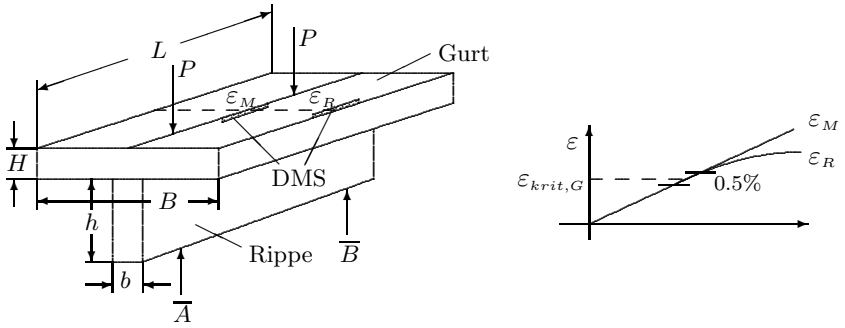


Abb. 8.2. Plexiglas–T-Träger, kritische Dehnung

Gesucht: Es interessiert nun die kritische Dehnung $\varepsilon_{krit,G} = \varepsilon_M - \varepsilon_R$ im Gurt; sie darf eine vorgegebene Toleranzgrenze nicht überschreiten. Die kritische Dehnung ist eine Funktion der geometrischen Trägergrößen Gurthöhe (H), Gurtbreite (B), Trägerlänge (L), Rippenhöhe (h) und Rippenbreite (b). Diese kritische Dehnung entsteht durch gleichzeitige Querverwölbung neben der Längsdurchbiegung durch die Einwirkung der Kräfte P, $\overline{A}$, $\overline{B}$.

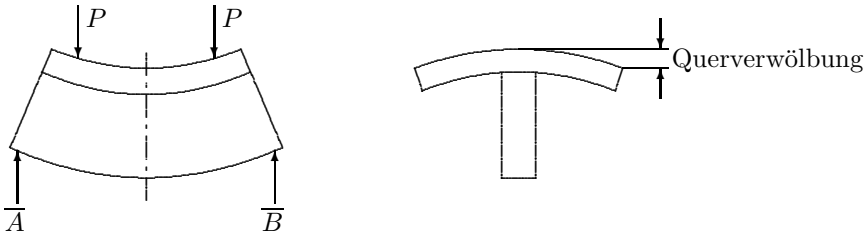


Abb. 8.3. Längsdurchbiegung mit Querverwölbung

Lösung: In Versuchen wird z. B. $\varepsilon_{krit,G}$ jeweils bei Variation von nur einer geometrischen Größe und Festhalten aller übrigen gemessen. Die Ergebnisse werden in Diagrammen festgehalten. So erhält man das Diagramm 1 (Abb. 8.4(a)) mit Kurven $H = b = \text{const.}$, wenn nur die Gurtbreite B variiert und $h = 50\,\text{mm}$, $L = 600\,\text{mm}$ festhalten werden. Das Diagramm 2 (Abb. 8.4(b)) ergibt sich bei Variation der Gurthöhe H = Rippenbreite b mit Kurven $B = \text{const.}$ bei festem $h = 50\,\text{mm}$, $L = 600\,\text{mm}$.[1]

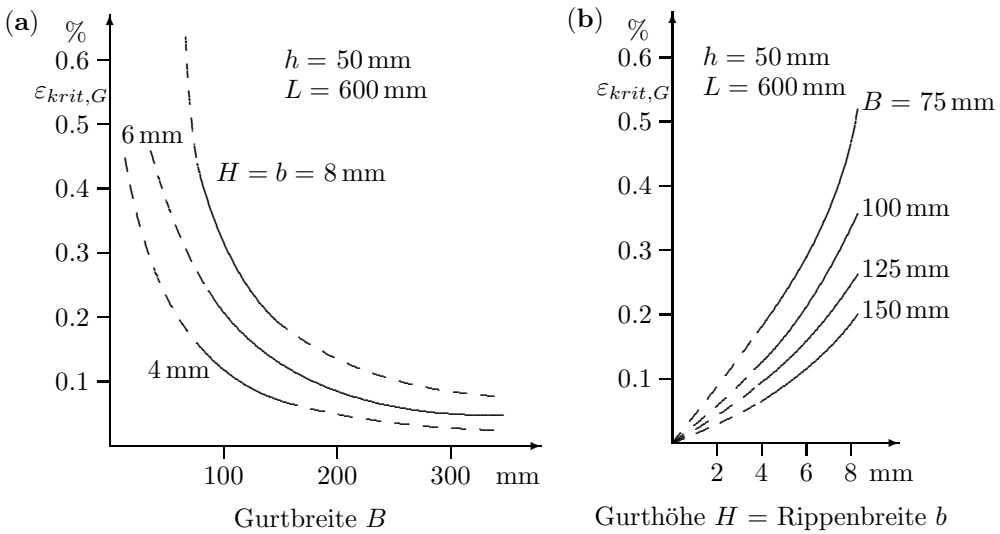


Abb. 8.4a,b. Diagramme, $\varepsilon_{krit,G}$ = kritische Gurtdehnung

[1] Die Messungen wurden am Lehrstuhl für Kunststoffverarbeitung der RWTH Aachen durchgeführt.

Gesucht sind nun funktionale Zusammenhänge Φ für die verschiedenen Messkurven. Die Gestalt der Kurven im jeweiligen Diagramm lässt auf die Form des Ansatzes für die Modellfunktion Φ schließen (vgl. dazu Abschnitt 8.2.1 und Beispiel 8.14 in Abschnitt 8.2.3.1). □

8.2 Lineare Approximation

8.2.1 Approximationsaufgabe und beste Approximation

Jeder Funktion $f \in C[a,b]$ wird eine reelle nichtnegative Zahl $\|f\|$, genannt *Norm* von f, zugeordnet, die den folgenden *Normaxiomen* genügt:

$$\left\{\begin{array}{ll} 1. & \|f\| \geq 0\,. \\ 2. & \|f\| = 0 \quad \text{genau dann, wenn } f = 0 \text{ überall in } [a,b]\,. \\ 3. & \|\alpha f\| = |\alpha|\,\|f\| \text{ für beliebige Zahlen } \alpha \in \mathbf{R}\,. \\ 4. & \|f+g\| \leq \|f\| + \|g\| \text{ für } f,g \in C[a,b]\,. \end{array}\right. \tag{8.1}$$

Für je zwei Funktionen $f_1, f_2 \in C[a,b]$ kann mit Hilfe einer Norm ein *Abstand*

$$\varrho\,(f_1,f_2) := \|f_1 - f_2\|$$

erklärt werden, für den die folgenden *Abstandsaxiome* gelten:

1. $\varrho\,(f_1,f_2) \geq 0\,.$
2. $\varrho\,(f_1,f_2) = 0$ genau dann, wenn $f_1 = f_2$ überall in $[a,b]\,.$
3. $\varrho\,(f_1,f_2) = \varrho\,(f_2,f_1)\,.$
4. $\varrho\,(f_1,f_3) \leq \varrho\,(f_1,f_2) + \varrho\,(f_2,f_3)$ für $f_1,f_2,f_3 \in C[a,b]\,.$

Nun wird ein System von $n+1$ linear unabhängigen Funktionen $\varphi_0,\varphi_1,\ldots,\varphi_n \in C[a,b]$ vorgegeben. Die Funktionen $\varphi_0,\varphi_1,\ldots,\varphi_n$ heißen *linear abhängig*, wenn es Zahlen $c_0,c_1,...,c_n$ gibt, die nicht alle Null sind, so dass für alle $x \in [a,b]$ gilt

$$c_0\varphi_0(x) + c_1\varphi_1(x) + \ldots + c_n\varphi_n(x) = 0\,;$$

andernfalls heißen $\varphi_0,\varphi_1,\ldots,\varphi_n$ *linear unabhängig*. Mit den Funktionen φ_k, $k = 0(1)n$, werden als Approximationsfunktionen die Linearkombinationen

$$\left\{\begin{array}{l} \Phi(x) := \Phi(x,c_0,c_1,\ldots,c_n) = \Phi(x,\boldsymbol{c}) = \displaystyle\sum_{k=0}^{n} c_k\varphi_k(x)\,, \\ x \in [a,b]\,, \quad c_k \in \mathbf{R}\,, \quad c_k = \text{const.}\,, \quad \boldsymbol{c} = (c_0,c_1,\ldots,c_n)^{\mathsf{T}}\,, \end{array}\right. \tag{8.2}$$

gebildet. Φ heißt *Approximationsfunktion* oder *Modellfunktion*, ein Ansatz der Form (8.2) heißt *lineare Approximation*.

$\overline{C}$ sei die Menge aller zulässigen Φ nach (8.2). Jede Linearkombination Φ ist durch das $(n+1)$-Tupel $(c_0, c_1, ..., c_n)$ ihrer Koeffizienten bestimmt. Der Abstand einer Funktion $f \in C[a,b]$ von Φ hängt bei festgehaltenem f nur von $\boldsymbol{c}$ ab; es ist

$$\varrho\,(f,\Phi) = \|f-\Phi\| =: D(c_0, c_1, \ldots, c_n)\,. \tag{8.3}$$

Häufig verwendete Funktionensysteme $\varphi_0, \varphi_1, \ldots, \varphi_n$ sind:

1. $\varphi_0 = 1,\ \varphi_1 = x,\ \varphi_2 = x^2, ..., \varphi_n = x^n$; die Approximationsfunktionen Φ sind dann algebraische Polynome vom Höchstgrad n.

2. $\varphi_0 = 1,\ \varphi_1 = \cos x,\ \varphi_2 = \sin x,\ \varphi_3 = \cos 2x,\ \varphi_4 = \sin 2x,\ ...$; die Approximationsfunktionen Φ sind dann 2π-periodische trigonometrische Polynome.

3. $\varphi_0 = 1,\ \varphi_1 = \mathrm{e}^{\alpha_1 x},\ \varphi_2 = \mathrm{e}^{\alpha_2 x}, \ldots, \varphi_n = \mathrm{e}^{\alpha_n x}$ mit paarweise verschiedenen reellen Zahlen α_i.

4. $\varphi_0 = 1,\ \varphi_1 = \frac{1}{(x-\alpha_1)^{p_1}},\ \varphi_2 = \frac{1}{(x-\alpha_2)^{p_2}},\ \ldots,\ \varphi_n = \frac{1}{(x-\alpha_n)^{p_n}},\quad \alpha_i \in \mathbb{R},\ p_i \in \mathbb{N}$; alle Paare (α_i, p_i) müssen untereinander verschieden sein. Die Approximationsfunktionen Φ sind dann spezielle rationale Funktionen (s. auch Bemerkung 8.3).

5. Orthogonale Funktionensysteme, s. dazu Sonderfälle in Abschnitt 8.2.2 und Abschnitt 8.2.3.2.

Ein Kriterium für die lineare Unabhängigkeit eines Funktionensystems $\varphi_0, \varphi_1, \ \ldots, \ \varphi_n \in C^n[a,b]$ ist das Nichtverschwinden der *Wronskischen Determinante* für $x \in [a,b]$

$$W(\varphi_0, \varphi_1, \ldots, \varphi_n) = \begin{vmatrix} \varphi_0 & \varphi_1 & \cdots & \varphi_n \\ \varphi_0' & \varphi_1' & \cdots & \varphi_n' \\ \vdots & \vdots & & \vdots \\ \varphi_0^{(n)} & \varphi_1^{(n)} & \cdots & \varphi_n^{(n)} \end{vmatrix} \not\equiv 0\,.$$

Approximationsaufgabe

Zu einer auf einem Intervall $[a,b]$ definierten Funktion f und zu einem auf $[a,b]$ vorgegebenen Funktionensystem $\varphi_0, \varphi_1, ..., \varphi_n$ ist unter allen Funktionen $\Phi \in \overline{C}$ der Gestalt (8.2) eine Funktion

$$\Phi^{(0)}(x) := \Phi^{(0)}(x, c_0^{(0)}, c_1^{(0)}, \ldots, c_n^{(0)}) = \sum_{k=0}^{n} c_k^{(0)} \varphi_k(x) \tag{8.4}$$

zu bestimmen mit der Eigenschaft

$$\left\{ \begin{array}{lcl} D(c_0^{(0)}, c_1^{(0)}, \ldots, c_n^{(0)}) = \|f - \Phi^{(0)}\| & = & \min\limits_{\Phi \in \overline{C}} \|f-\Phi\| \\ & = & \min\limits_{c_i \in \mathbf{R}} D(c_0, c_1, \ldots, c_n)\,. \end{array} \right. \tag{8.5}$$

$\Phi^{(0)}$ heißt *beste Approximation* von f bezüglich des vorgegebenen Systems $\varphi_0, \varphi_1, ..., \varphi_n$ und im Sinne der gewählten Norm $\|.\|$.

Satz 8.2. (*Existenzsatz*)
Zu jeder Funktion $f \in C[a,b]$ existiert für jedes System linear unabhängiger Funktionen $\varphi_0, \varphi_1, ..., \varphi_n \in C[a,b]$ und jede Norm $\|.\|$ mindestens eine beste Approximation $\Phi^{(0)}$ der Gestalt (8.4) mit der Eigenschaft (8.5).

Das Funktionensystem $\varphi_0, \varphi_1, ..., \varphi_n$ wird im Hinblick auf die jeweilige Aufgabenstellung gewählt, z. B. sind zur Bestimmung einer besten Approximation für eine periodische Funktion trigonometrische Polynome als Modellfunktionen zweckmäßig (s. Abschnitt 8.2.5).

Bemerkung 8.3. (*Rationale Approximation*)
Bei manchen Aufgabenstellungen, z. B. dann, wenn bekannt ist, dass f für bestimmte Werte Pole besitzt, empfiehlt sich als Approximationsfunktion eine Funktion der Gestalt

$$\Psi(x) = \frac{\sum\limits_{k=0}^{m} a_k \varphi_k(x)}{\sum\limits_{k=0}^{j} b_k \widetilde{\varphi}_k(x)}, \quad \varphi_k, \widetilde{\varphi}_k \in C[a,b]\,. \tag{8.6}$$

Für $\varphi_k(x) = \widetilde{\varphi}_k(x) = x^k$ liefert der Ansatz eine rationale Funktion, deren Zähler den Höchstgrad m und deren Nenner den Höchstgrad j besitzt. Wird $b_j = 1$ gesetzt, was bei passender Wahl von j o. B. d. A. möglich ist, so ist unter allen Funktionen Ψ aus der zulässigen Menge $\overline{\overline{C}}$ der Gestalt (8.6) eine beste Approximation $\Psi^{(0)}$ mit der Eigenschaft

$$\begin{aligned} D(a_0^{(0)}, a_1^{(0)}, \ldots, a_m^{(0)}, b_0^{(0)}, b_1^{(0)}, \ldots, b_{j-1}^{(0)}) &= \|f - \Psi^{(0)}\| \\ &= \min_{\Psi \in \overline{\overline{C}}} \|f - \Psi\| = \min_{a_i, b_j \in \mathbf{R}} D(a_0, a_1, \ldots, a_m, b_0, b_1, \ldots, b_{j-1}) \end{aligned}$$

zu bestimmen (siehe auch Abschnitt 8.3).

8.2.2 Kontinuierliche lineare Approximation im quadratischen Mittel

Man legt für eine auf $[a,b]$ definierte Funktion g die folgende L_2-Norm zugrunde

$$\|g\|_2 = \left(\int_a^b w(x) g^2(x)\, \mathrm{d}x \right)^{\frac{1}{2}}; \tag{8.7}$$

dabei ist $w(x) > 0$ eine gegebene, auf $[a,b]$ integrierbare *Gewichtsfunktion*. Die Norm (8.7) lässt sich mit Hilfe des *Skalarproduktes*

$$(g,h) := \int_a^b w(x)\; g(x)\; h(x)\, \mathrm{d}x$$

erklären durch $\|g\|_2 := (g,g)^{1/2}$. Eigenschaften des Skalarproduktes (s. [BERE1971] Bd. 1, S.317 ff.) sind:

1. $(g_1, g_2) = (g_2, g_1)$;
2. $(\alpha_1 g_1 + \alpha_2 g_2, g_3) = \alpha_1(g_1, g_3) + \alpha_2(g_2, g_3)$ für beliebige $\alpha_1, \alpha_2 \in \mathbb{R}$;
3. $(g,g) \geq 0$ und $(g,g) = 0$ genau dann, wenn $g(x) = 0$ für alle $x \in [a,b]$.

Setzt man $g = f - \Phi$ und betrachtet das Quadrat des Abstandes (8.3), das die gleichen Extremaleigenschaften wie der Abstand selbst hat, so lautet die (8.5) entsprechende Bedingung

$$\begin{aligned} \|f - \Phi^{(0)}\|_2^2 = \min_{\Phi \in \overline{C}} \|f - \Phi\|_2^2 &= \min_{\Phi \in \overline{C}} \int_a^b w(x)\big(f(x) - \Phi(x)\big)^2 \,\mathrm{d}x \\ &= \min_{c_i \in \mathbf{R}} D^2(c_0, c_1, \ldots, c_n)\,, \end{aligned} \tag{8.8}$$

d. h. das Integral über die gewichteten Fehlerquadrate ist zum Minimum zu machen. Die dafür notwendigen Bedingungen $\frac{\partial D^2}{\partial c_j}(c_0^{(0)}, \ldots, c_n^{(0)}) = 0$ für $j = 0(1)n$ liefern mit (8.2) und $\partial\Phi/\partial c_j = \varphi_j(x)$ insgesamt $n+1$ lineare Gleichungen zur Bestimmung der $n+1$ Koeffizienten $c_k^{(0)}$ einer besten Approximation (8.4):

$$\sum_{k=0}^{n} c_k^{(0)} \int_a^b w(x)\varphi_j(x)\varphi_k(x)\,\mathrm{d}x = \int_a^b w(x) f(x)\varphi_j(x)\,\mathrm{d}x\,, \quad j = 0(1)n\,, \tag{8.9}$$

oder mit den Abkürzungen (Skalarprodukten)

$$(\varphi_k, \varphi_j) := \int_a^b w(x)\varphi_k(x)\varphi_j(x)\,\mathrm{d}x\,; \quad (f, \varphi_j) := \int_a^b w(x) f(x)\varphi_j(x)\,\mathrm{d}x\,,$$

in Matrizenform

$$\left\{ \begin{array}{l} \boldsymbol{G}\boldsymbol{c}^{(0)} = \boldsymbol{a} \quad \text{mit} \\ \\ \boldsymbol{G} := \begin{pmatrix} (\varphi_0,\varphi_0) & (\varphi_0,\varphi_1) & \cdots & (\varphi_0,\varphi_n) \\ (\varphi_1,\varphi_0) & (\varphi_1,\varphi_1) & \cdots & (\varphi_1,\varphi_n) \\ \vdots & \vdots & & \vdots \\ (\varphi_n,\varphi_0) & (\varphi_n,\varphi_1) & \cdots & (\varphi_n,\varphi_n) \end{pmatrix}, \\ \\ \boldsymbol{c}^{(0)} = \begin{pmatrix} c_0^{(0)} \\ c_1^{(0)} \\ \vdots \\ c_n^{(0)} \end{pmatrix}, \; \boldsymbol{a} = \begin{pmatrix} (f,\varphi_0) \\ (f,\varphi_1) \\ \vdots \\ (f,\varphi_n) \end{pmatrix}. \end{array} \right. \tag{8.10}$$

Die Gleichungen (8.9) bzw. (8.10) heißen *Gaußsche Normalgleichungen*, die resultierende Funktion $\Phi^{(0)}(x) = \sum_{k=0}^{n} c_k^{(0)} \varphi_k(x)$ aufgrund der gewählten Norm (8.7) *beste Approximation im quadratischen Mittel* und die gesamte Vorgehensweise auch *Gaußsche Fehlerquadratmethode*.

Wegen $(\varphi_j, \varphi_k) = (\varphi_k, \varphi_j)$ gilt $\boldsymbol{G} = \boldsymbol{G}^{\mathsf{T}}$. Die Determinante det $\boldsymbol{G}$ des Gleichungssystems (8.10) heißt *Gramsche Determinante* des Systems $\varphi_0, \varphi_1, ..., \varphi_n$. Es gilt der

Hilfssatz 8.4.
Ein Funktionensystem $\varphi_0, \varphi_1, \ldots, \varphi_n \in C[a,b]$ ist genau dann linear abhängig, wenn seine Gramsche Determinante verschwindet (s. [BERE1971] Bd.1, S.319; [STUM1982], S.133).

Nach Voraussetzung ist das System $\varphi_0, \varphi_1, \ldots, \varphi_n$ linear unabhängig, die Gramsche Determinante also verschieden von Null, so dass die $c_0^{(0)}, c_1^{(0)}, \ldots, c_n^{(0)}$ durch (8.9) bzw. (8.10) *eindeutig* bestimmt sind. Es folgt daher mit Satz 8.2 und Hilfssatz 8.4 der

Satz 8.5.
Zu jeder Funktion $f \in C[a,b]$ existiert für jedes System linear unabhängiger Funktionen $\varphi_0, \varphi_1, \ldots, \varphi_n \in C[a,b]$ und die Norm (8.7) genau eine beste Approximation $\Phi^{(0)}$ der Gestalt (8.4) mit der Eigenschaft (8.8), deren Koeffizienten $c_k^{(0)}$ sich aus (8.9) bzw. (8.10) ergeben.

Sonderfälle

1. Algebraische Polynome
Die Approximationsfunktionen Φ sind mit $\varphi_k(x) = x^k$ algebraische Polynome vom Höchstgrad n

$$\Phi(x, \boldsymbol{c}) = \sum_{k=0}^{n} c_k x^k \,. \tag{8.11}$$

Die Normalgleichungen (8.9) zur Bestimmung der $c_k^{(0)}$ lauten hier

$$\sum_{k=0}^{n} c_k^{(0)} \int_a^b w(x) x^{j+k} \,\mathrm{d}x = \int_a^b w(x) f(x) x^j \,\mathrm{d}x \,, \qquad j = 0(1)n \,, \tag{8.12}$$

und mit $w(x) = 1$ für alle $x \in [a,b]$ (vgl. Bemerkung 8.6) ergibt sich für (8.10)

$$\boldsymbol{G}\boldsymbol{c}^{(0)} = \boldsymbol{a} \quad \text{mit} \tag{8.13}$$

$$\boldsymbol{G} = \begin{pmatrix} \int_a^b \mathrm{d}x & \int_a^b x\,\mathrm{d}x & \int_a^b x^2\,\mathrm{d}x & \cdots & \int_a^b x^n\,\mathrm{d}x \\ \int_a^b x\,\mathrm{d}x & \int_a^b x^2\,\mathrm{d}x & \int_a^b x^3\,\mathrm{d}x & \cdots & \int_a^b x^{n+1}\,\mathrm{d}x \\ \int_a^b x^2\,\mathrm{d}x & \int_a^b x^3\,\mathrm{d}x & \int_a^b x^4\,\mathrm{d}x & \cdots & \int_a^b x^{n+2}\,\mathrm{d}x \\ \vdots & \vdots & \vdots & & \vdots \\ \int_a^b x^n\,\mathrm{d}x & \int_a^b x^{n+1}\,\mathrm{d}x & \int_a^b x^{n+2}\,\mathrm{d}x & \cdots & \int_a^b x^{2n}\,\mathrm{d}x \end{pmatrix},$$

$$\boldsymbol{c}^{(0)} = \begin{pmatrix} c_0^{(0)} \\ c_1^{(0)} \\ c_2^{(0)} \\ \vdots \\ c_n^{(0)} \end{pmatrix}, \quad \boldsymbol{a} = \begin{pmatrix} \int_a^b f(x)\,\mathrm{d}x \\ \int_a^b f(x)x\,\mathrm{d}x \\ \int_a^b f(x)x^2\,\mathrm{d}x \\ \vdots \\ \int_a^b f(x)x^n\,\mathrm{d}x \end{pmatrix}.$$

Für $[a,b] = [0,1]$ stimmt $\boldsymbol{G}$ mit der sehr schlecht konditionierten Hilbert-Matrix $\boldsymbol{H}_n = (h_{ij})$, $h_{ij} = 1/(i+j+1)$ für $i,j = 0(1)n$, überein, was eine brauchbare direkte Lösungsbestimmung normalerweise zunichte macht. Ein Ausweg kann über den Einsatz der Householder-Transformation geboten werden (siehe Abschnitt 8.2.3.4 für den analogen diskreten Fall), ein anderer über besser geeignete Funktionensysteme, beispielsweise das sich nach dem weiter unten beschriebenen Schmidtschen Orthogonalisierungsverfahren ergebende System.

2. Orthogonale Funktionensysteme

Die Funktionen φ_k bilden ein orthogonales System, wenn gilt

$$(\varphi_j, \varphi_k) = \int_a^b w(x)\,\varphi_j(x)\,\varphi_k(x)\,\mathrm{d}x = 0 \quad \text{für} \quad j \neq k\,.$$

Dann erhält (8.9) bzw. (8.10) die besonders einfache Gestalt

$$(\varphi_j, \varphi_j) c_j^{(0)} = (f, \varphi_j)\,, \quad j = 0(1)n\,. \tag{8.14}$$

Bei einer Erhöhung von n auf $n+1$ im Ansatz (8.2) bleiben also hier im Gegensatz zu nicht orthogonalen Funktionensystemen die $c_j^{(0)}$ für $j = 0(1)n$ unverändert und $c_{n+1}^{(0)}$ errechnet sich aus (8.14) für $j = n+1$.

Zum diskreten Ausgleich durch orthogonale Polynome siehe Abschnitt 8.2.3.2.

Beispiele orthogonaler Funktionensysteme

a) $\varphi_k(x) = \cos kx\,,\ x \in [0, 2\pi]\,,\ k = 0(1)n\,,\ w(x) = 1\,.$

b) $\varphi_k(x) = \sin kx\,,\ x \in [0, 2\pi]\,,\ k = 1(1)n\,,\ w(x) = 1\,.$

c) $\varphi_0(x) = 1\,,\ \varphi_1(x) = \cos \omega x\,,\ \varphi_2(x) = \sin \omega x\,,\ \varphi_3(x) = \cos 2\omega x\,,\ \varphi_4(x) = \sin 2\omega x\,, \ldots$ für $x \in [0, L]\,,\ w(x) = 1$ und $\omega = \frac{2\pi}{L}$ (L-periodische trigonometrische Polynome, vgl. Abschnitt 8.2.5).

d) Legendresche Polynome P_k für $x \in [-1, +1]$ mit
$P_{k+1}(x) = \frac{1}{k+1}\big((2k+1)xP_k(x) - kP_{k-1}(x)\big)$, $k = 1, 2, 3, \ldots,$
$P_0(x) = 1, \quad P_1(x) = x, \quad w(x) = 1.$

e) Tschebyscheffsche Polynome T_k für $x \in [-1, +1]$ mit
$T_{k+1}(x) = 2x\,T_k(x) - T_{k-1}(x), \quad k = 1, 2, 3, \ldots,$
$T_0(x) = 1, \quad T_1(x) = x, \quad w(x) = 1/\sqrt{1-x^2}$ (vgl. Abschnitte 8.2.4.2 und 8.2.6.1).

Orthogonalisierungsverfahren von E. Schmidt

Es seien $\varphi_0, \varphi_1, ..., \varphi_n \in C[a, b]$ $n+1$ vorgegebene, linear unabhängige Funktionen. Dann lässt sich ein diesem System gleichwertiges orthogonales Funktionensystem $\widetilde{\varphi}_0, \widetilde{\varphi}_1, \ldots, \widetilde{\varphi}_n \in C[a, b]$ konstruieren. Man bildet dazu, ausgehend von $\widetilde{\varphi}_0 = \varphi_0$, die Linearkombinationen

$$\widetilde{\varphi}_k = a_{k0}\widetilde{\varphi}_0 + a_{k1}\widetilde{\varphi}_1 + \ldots + a_{k,k-1}\widetilde{\varphi}_{k-1} + \varphi_k\,, \quad k = 1(1)n\,,$$

und bestimmt die konstanten Koeffizienten a_{kj} der Reihe nach so, dass die Orthogonalitätsrelationen $(\widetilde{\varphi}_j, \widetilde{\varphi}_k) = 0$ für $j < k$ erfüllt sind; man erhält auf diese Weise

$$a_{kj} = -(\varphi_k, \widetilde{\varphi}_j)/(\widetilde{\varphi}_j, \widetilde{\varphi}_j)\,, \quad k = 1(1)n\,, \quad j = 0(1)k{-}1\,.$$

Für $\varphi_k(x) = x^k\,,\ x \in [-1, +1]$, und $w(x) = 1$ liefert das Verfahren die Legendreschen Polynome.

Bemerkung 8.6.
Als Gewichtsfunktion wird in vielen Fällen $w(x) = 1$ für alle $x \in [a, b]$ gewählt; dadurch wird kein Teilbereich des Intervalls $[a, b]$ gegenüber einem anderen besonders betont oder vernachlässigt. Bei manchen Problemen sind jedoch andere Gewichtsfunktionen sinnvoll. Erhält man z. B. mit $w(x) = 1$ eine beste Approximation $\Phi^{(0)}$, für die $\big(f(x) - \Phi^{(0)}(x)\big)^2$ etwa in der Umgebung von $x = a$ und $x = b$ besonders groß wird, so wählt man statt $w(x) = 1$ ein $\widetilde{w}(x)$, das für $x \to a$ und $x \to b$ besonders groß wird. Dann ergibt sich eine zu dieser Gewichtsfunktion $\widetilde{w}(x)$ gehörige beste Approximation $\widetilde{\Phi}^{(0)}$, für die $\big(f(x) - \widetilde{\Phi}^{(0)}(x)\big)^2$ für $x \to a$ und $x \to b$ klein wird. Für $a = -1$ und $b = +1$ kann z. B. $\widetilde{w}(x) = 1/\sqrt{1-x^2}$ eine solche Gewichtsfunktion sein.

Algorithmus 8.7. (*Kontinuierliche Gaußsche Fehlerquadratmethode*)
Gegeben sei eine auf dem Intervall $[a, b]$ definierte Funktion f; gesucht ist für f die beste Approximation $\Phi^{(0)}$ nach der kontinuierlichen Gaußschen Fehlerquadratmethode.

1. Schritt. Wahl eines geeigneten Funktionensystems $\varphi_0, \varphi_1, ..., \varphi_n$ zur Konstruktion der Approximationsfunktion Φ.
2. Schritt. Wahl einer geeigneten Gewichtsfunktion $w(x) > 0$; vgl. dazu Bemerkung 8.6.
3. Schritt. Aufstellung und Lösung des linearen Gleichungssystems (8.9) bzw. (8.10) für die Koeffizienten $c_k^{(0)}$ der besten Approximation (8.4). Sind die Approximationsfunktionen $\varphi_k = x^k$ speziell algebraische Polynome, so ist das System (8.12) bzw. für $w(x) \equiv 1$ das System (8.13) zu lösen; bilden die φ_k ein orthogonales System, so ist über (8.14) die Lösung gegeben.

Beispiel 8.8.

Gegeben: Die Funktion $f(x) = 1/(1 + x^2)$ für $x \in [-1, +1]$.

Gesucht: Die beste Approximation $\Phi^{(0)}$ für f unter allen quadratischen Polynomen $\Phi(x) = c_0 + c_1 x + c_2 x^2$ im Sinne von (8.8).

Lösung: Die Vorgehensweise erfolgt nach Algorithmus 8.7.

1. Schritt. Das Funktionensystem $\varphi_0(x) = 1,\ \varphi_1(x) = x,\ \varphi_2(x) = x^2$ ist hier naheliegend, da als Approximationsfunktionen alle quadratischen Polynome Φ mit $\Phi(x) = c_0 + c_1 x + c_2 x^2$ für $x \in [-1, +1]$ dienen sollen.

2. Schritt. Es wird $w(x) \equiv 1$ gesetzt, da nichts darüber ausgesagt ist, dass bestimmte Intervallanteile anders zu gewichten sind als andere.

3. Schritt. Das Gleichungssystem (8.13) für die $c_k^{(0)}$ lautet dann

$$\begin{pmatrix} \int_{-1}^{+1} \mathrm{d}x & \int_{-1}^{+1} x\mathrm{d}x & \int_{-1}^{+1} x^2\,\mathrm{d}x \\ \int_{-1}^{+1} x\,\mathrm{d}x & \int_{-1}^{+1} x^2\,\mathrm{d}x & \int_{-1}^{+1} x^3\,\mathrm{d}x \\ \int_{-1}^{+1} x^2\,\mathrm{d}x & \int_{-1}^{+1} x^3\,\mathrm{d}x & \int_{-1}^{+1} x^4\,\mathrm{d}x \end{pmatrix} \begin{pmatrix} c_0^{(0)} \\ c_1^{(0)} \\ c_2^{(0)} \end{pmatrix} = \begin{pmatrix} \int_{-1}^{+1} \frac{1}{1+x^2}\,\mathrm{d}x \\ \int_{-1}^{+1} \frac{x}{1+x^2}\,\mathrm{d}x \\ \int_{-1}^{+1} \frac{x^2}{1+x^2}\,\mathrm{d}x \end{pmatrix}$$

bzw.

$$\begin{pmatrix} 2 & 0 & \frac{2}{3} \\ 0 & \frac{2}{3} & 0 \\ \frac{2}{3} & 0 & \frac{2}{5} \end{pmatrix} \begin{pmatrix} c_0^{(0)} \\ c_1^{(0)} \\ c_2^{(0)} \end{pmatrix} = \begin{pmatrix} 2\arctan 1 \\ 0 \\ 2 - 2\arctan 1 \end{pmatrix};$$

es besitzt die eindeutige, auf 4 Mantissenstellen angegebene Lösung

$$c_0^{(0)} = 0.9624; \quad c_1^{(0)} = 0; \quad c_2^{(0)} = -0.5310.$$

Unter allen quadratischen Polynomen Φ ist somit

$$\Phi^{(0)}(x) = \sum_{k=0}^{2} c_k^{(0)} x^k = 0.9624 - 0.5310x^2 \quad \text{für} \quad x \in [-1, +1]$$

die beste kontinuierliche Approximation für f im quadratischen Mittel.

Für $x = 0.8 \in [-1, +1]$ erhält man auf vier Dezimalen genau den Wert $\Phi^{(0)}(0.8) = 0.6226$. Mit $f(0.8) = 1/1.64$ und $0.6097 < f(0.8) < 0.6098$ folgt für den absoluten Fehler die Abschätzung

$$|f(0.8) - \Phi(0.8)| \leq |0.6097 - 0.6226| = 1.29 \cdot 10^{-2}\,.$$

Bemerkung: Da f eine gerade Funktion ist, war $c_1^{(0)} = 0$ zu erwarten; siehe auch Anmerkung unter Satz 8.23. □

8.2.3 Diskrete lineare Approximation im quadratischen Mittel

8.2.3.1 Normalgleichungen für den diskreten linearen Ausgleich

Hier wird eine beste Approximation $\Phi^{(0)}$ der Gestalt (8.4) für eine auf $[a, b]$ definierte Funktion f gesucht, von der an $N + 1$ diskreten Stellen $x_i \in [a, b]$, $i = 0(1)N$, $N \geq n$, die Funktionswerte $f(x_i)$ gegeben sind. Es wird für eine Funktion g in Analogie zum kontinuierlichen Fall mit (8.7) die Seminorm

$$\|g\|_{d,2} = \left(\sum_{i=0}^{N} w_i g^2(x_i) \right)^{\frac{1}{2}}$$

zugrunde gelegt mit den Zahlen $w_i > 0$ als Gewichte. Für eine Seminorm gelten die Axiome (8.1) mit Ausnahme von 2. Diese Norm lässt sich mit Hilfe des Skalarproduktes

$$(g, h) := \sum_{i=0}^{N} w_i g(x_i) h(x_i)\,, \quad x_i \in [a, b]\,,$$

erklären durch $\|g\|_{d,2} := (g, g)^{1/2}$.

Setzt man $g = f - \Phi$ und betrachtet wieder vorteilhafterweise das Quadrat des Abstandes (8.3), so lautet die (8.5) entsprechende Bedingung für eine beste Approximation unter Verwendung der Seminorm

$$\begin{aligned} \|f - \Phi^{(0)}\|_{d,2}^2 &= \min_{\Phi \in \overline{C}} \|f - \Phi\|_{d,2}^2 = \min_{\Phi \in \overline{C}} \sum_{i=0}^{N} w_i \big(f(x_i) - \Phi(x_i, \boldsymbol{c})\big)^2 \qquad (8.15) \\ &= \min_{c_i \in \mathbf{R}} D^2(c_0, c_1, \ldots, c_n)\,, \end{aligned}$$

d. h. die Summe der gewichteten Fehlerquadrate ist zum Minimum zu machen (*diskrete Gaußsche Fehlerquadratmethode*).

Die notwendigen Bedingungen

$$\frac{\partial D^2}{\partial c_j}(c_0^{(0)},\ldots,c_n^{(0)}) = -2\sum_{i=0}^{N} w_i\big(f(x_i)-\Phi(x_i,\boldsymbol{c}^{(0)})\big)\frac{\partial\Phi(x_i,\boldsymbol{c}^{(0)})}{\partial c_j} = 0\,,\quad j=0(1)n\,,$$

liefern $n+1$ lineare Gleichungen zur Bestimmung der $n+1$ Koeffizienten $c_k^{(0)}$ einer besten Approximation.

Mit

$$\Phi(x_i,\boldsymbol{c}) = \Phi(x_i,c_0,\ldots,c_n) = \sum_{k=0}^{n} c_k\varphi_k(x_i)\,,\quad \frac{\partial\Phi(x_i,\boldsymbol{c})}{\partial c_j} = \varphi_j(x_i)$$

erhält man als lineares Gleichungssystem die *Gaußschen Normalgleichungen*

$$\sum_{k=0}^{n} c_k^{(0)}\sum_{i=0}^{N} w_i\varphi_j(x_i)\varphi_k(x_i) = \sum_{i=0}^{N} w_i f(x_i)\varphi_j(x_i)\,,\ j=0(1)n\,,\ N\geq n\,, \tag{8.16}$$

die unter Verwendung der Skalarprodukte

$$\left\{\begin{array}{lcl}(\varphi_j,\varphi_k) & := & \displaystyle\sum_{i=0}^{N} w_i\varphi_j(x_i)\varphi_k(x_i)\,,\\[2ex] (f,\varphi_j) & := & \displaystyle\sum_{i=0}^{N} w_i f(x_i)\varphi_j(x_i)\end{array}\right. \tag{8.17}$$

die Form besitzen (vgl. 8.10):

$$\left\{\begin{array}{l}\boldsymbol{G}\boldsymbol{c}^{(0)} = \boldsymbol{a}\quad\text{mit}\\ \boldsymbol{G} = \begin{pmatrix}(\varphi_0,\varphi_0) & \cdots & (\varphi_0,\varphi_n)\\ \vdots & & \vdots\\ (\varphi_n,\varphi_0) & \cdots & (\varphi_n,\varphi_n)\end{pmatrix},\ \boldsymbol{c}^{(0)} = \begin{pmatrix}c_0^{(0)}\\ \vdots\\ c_n^{(0)}\end{pmatrix},\ \boldsymbol{a} = \begin{pmatrix}(f,\varphi_0)\\ \vdots\\ (f,\varphi_n)\end{pmatrix}.\end{array}\right. \tag{8.18}$$

Wegen $(\varphi_j,\varphi_k) = (\varphi_k,\varphi_j)$ gilt $\boldsymbol{G} = \boldsymbol{G}^{\mathsf{T}}$.

Die Normalgleichungen (8.18) sind oft schlecht konditioniert. Die direkte numerische Berechnung der Lösungen kann zu einer Vergrößerung der Rundungsfehler führen und damit zur Verfälschung der Ergebnisse. In diesen Fällen sollte man z. B. mit der Householder-Transformation arbeiten (s. Abschnitt 8.2.3.4).

Definition 8.9. (abkürzende Schreibweise)
Jeder Funktion $\varphi_k\in C[a,b]$ wird im Falle $w_i=1$ für alle i der Vektor

$$\boldsymbol{\varphi}_k := \big(\varphi_k(x_0),\varphi_k(x_1),\ldots,\varphi_k(x_N)\big)^{\mathsf{T}}$$

zugeordnet bzw. im Falle beliebiger $w_i>0$ der Vektor

$$\boldsymbol{\varphi}_k := \big(\sqrt{w_0}\varphi_k(x_0),\sqrt{w_1}\varphi_k(x_1),\ldots,\sqrt{w_N}\varphi_k(x_N)\big)^{\mathsf{T}}.$$

Mit dieser Definition gilt für die Skalarprodukte (8.17)

$$(\varphi_j, \varphi_k) = \boldsymbol{\varphi}_j^{\mathsf{T}} \boldsymbol{\varphi}_k, \quad (f, \varphi_j) = \boldsymbol{f}^{\mathsf{T}} \boldsymbol{\varphi}_j \tag{8.19}$$

unter Verwendung von $\boldsymbol{f} := \big(f(x_0), f(x_1), ..., f(x_N)\big)^{\mathsf{T}}$ bzw.

$$\boldsymbol{f} := \big(\sqrt{w_0} f(x_0), \ldots, \sqrt{w_N} f(x_N)\big)^{\mathsf{T}}.$$

Satz 8.10.
Die Gaußschen Normalgleichungen (8.16) sind genau dann eindeutig lösbar, wenn die Vektoren $\boldsymbol{\varphi}_k$ linear unabhängig sind. Notwendig dafür ist die Bedingung $n \leq N$.

Im Falle $n > N$ sind die Vektoren $\boldsymbol{\varphi}_k$ immer linear abhängig, so dass die Eindeutigkeitsaussage entfällt. Im Falle $n = N$ liegt Interpolation vor, d. h. die Summe der Fehlerquadrate (8.15) führt auf den denkbar kleinsten Wert 0.

Bemerkung 8.11.
Häufig werden gleiche Gewichte ($w_i = 1$ für alle i) gewählt. Eine andere Wahl ist sinnvoll, wenn bekannt ist, dass die Werte $f(x_i)$ für verschiedene x_i unterschiedlich genau sind. Dann werden im Allgemeinen den weniger genauen Funktionswerten kleinere Gewichte zugeordnet. Normiert man die Gewichte w_i außerdem so, dass $\sum w_i = 1$ ist, kann man sie als die Wahrscheinlichkeiten für das Auftreten von Werten der Funktion f an den Stellen x_i deuten. Man kann auch in (8.15) für die Gewichte $w_i = 1/f^2(x_i)$ setzen. Dies ist gleichbedeutend damit, dass man die Quadratsumme der relativen Fehler minimiert:

$$\sum_{i=0}^{N} \left(\frac{f(x_i) - \Phi(x_i)}{f(x_i)} \right)^2 \stackrel{!}{=} \text{Min.} \qquad \big(f(x_i) \neq 0\big).$$

Algorithmus 8.12. (*Diskrete Gaußsche Fehlerquadratmethode*)

Von einer auf dem Intervall $[a, b]$ definierten Funktion f seien an $N+1$ untereinander verschiedenen Stellen $x_i \in [a, b]$, $i = 0(1)N$, die Werte $f(x_i)$ gegeben. Gesucht ist für f die beste Approximation $\Phi^{(0)}$ nach der diskreten Gaußschen Fehlerquadratmethode.

1. Schritt: Wahl eines geeigneten Funktionensystems $\varphi_0, \varphi_1, ..., \varphi_n$ zur Konstruktion der Approximationsfunktion Φ, wobei $n \leq N$ gelten muss.
2. Schritt: Festlegen der geeigneten Gewichte $w_i > 0$, vgl. dazu Bemerkung 8.11.
3. Schritt: Aufstellen der Normalgleichungen (8.16) bzw. (8.18) zur Berechnung der Koeffizienten $c_k^{(0)}$ der besten Approximation $\Phi^{(0)}$.

Die Forderung $x_i \neq x_k$ für $i \neq k$ kann fallengelassen werden, sofern $N'+1$ Stützstellen x_i untereinander verschieden sind und $n \leq N' \leq N$ gilt.

Beispiel 8.13. (vgl. Beispiel 8.8)

Gegeben: Die Wertetabelle der Funktion $f \in C[-1,+1]$; $f(x) = 1/(1+x^2)$

i	0	1	2	3	4
x_i	-1.0	-0.5	0	0.5	1
$f(x_i)$	0.5	0.8	1	0.8	0.5

Gesucht: Für f die beste Approximation $\Phi^{(0)}$ unter allen quadratischen Polynomen $\Phi(x) = c_0 + c_1\,x + c_2\,x^2$ nach der diskreten Gaußschen Fehlerquadratmethode.

Lösung: Die Vorgehensweise erfolgt gemäß Algorithmus 8.12.

1. Schritt: Durch die Aufgabenstellung ist das Funktionensystem $\varphi_0(x) = 1, \varphi_1(x) = x$, $\varphi_2(x) = x^2$ naheliegend.

2. Schritt: Alle Funktionswerte $f(x_i)$ sind mit gleicher Genauigkeit angegeben; deshalb wird $w_i = 1$ für alle i gesetzt.

3. Schritt: Das System der Normalgleichungen lautet mit $n = 2, N = 4$ bei Rundung auf 4-stellige Mantisse

$$\begin{pmatrix} 5.000 & 0.000 & 2.500 \\ 0.000 & 2.500 & 0.000 \\ 2.500 & 0.000 & 2.125 \end{pmatrix} \begin{pmatrix} c_0^{(0)} \\ c_1^{(0)} \\ c_2^{(0)} \end{pmatrix} = \begin{pmatrix} 3.600 \\ 0.000 \\ 1.400 \end{pmatrix}.$$

Es hat die Lösung

$$c_0^{(0)} = 0.9486,\ c_1^{(0)} = 0,\ c_2^{(0)} = -0.4571.$$

Unter allen quadratischen Polynomen Φ ist das Polynom $\Phi^{(0)}(x) = 0.9486 - 0.4571x^2$ (*Ausgleichsparabel*) für $x \in [-1,+1]$ die beste Approximation für die über die Wertetabelle gegebene Funktion f nach der diskreten Gaußschen Fehlerquadratmethode.

Für $x = 0.8 \in [-1,+1]$ erhält man $\Phi^{(0)}(0.8) = 0.6561$. Da f bekannt ist, kann man den absoluten Fehler abschätzen. Mit $0.6097 < f(0.8) = 1/1.64 < 0.6098$ folgt

$$|f(0.8) - \Phi^{(0)}(0.8)| \le |0.6097 - 0.6561| = 4.64 \cdot 10^{-2}.$$

Natürlich unterscheidet sich das hier ermittelte diskrete Ausgleichspolynom von dem kontinuierlichen Ausgleichspolynom derselben Funktion f aus Beispiel 8.8, da von f nur noch die Information an fünf diskreten Stellen in den Ausgleich eingeflossen ist. Aber bezüglich des zugrunde liegenden diskreten Fehlermaßes (8.15) hat das diskrete Ausgleichspolynom eine kleinere Fehlerquadratsumme!

Bemerkung. Da f eine gerade Funktion ist, war $c_1^{(0)} = 0$ zu erwarten; siehe auch Anmerkung unter Satz 8.25. □

Beispiel 8.14.

Gegeben: Die Wertetabelle $\big(x_i, f(x_i)\big)$, $i = 0(1)N$, $N = 3$

i	0	1	2	3
x_i	0.02	0.10	0.50	1.00
$f(x_i)$	50	10	1	0

Gesucht: Unter allen $\Phi \in C[0.02, 1.00]$ mit $\Phi(x) = \sum c_k \varphi_k(x)$ die beste Approximation $\Phi^{(0)}$ im Sinne der diskreten Gaußschen Fehlerquadratmethode ($w_i = 1$ für alle i) mit folgenden Ansatzfunktionen:

(i) $\varphi_0 = 1 \quad \varphi_1 = x$ (Ausgleichsgerade)

(ii) $\varphi_0 = 1 \quad \varphi_1 = x \quad \varphi_2 = x^2$ (quadratische Ausgleichsparabel)

(iii) $\varphi_0 = 1 \quad \varphi_1 = x \quad \varphi_2 = x^2 \quad \varphi_3 = x^3$ (kubische Ausgleichsparabel bzw. wegen $N = n = 3$ interpolierende kubische Parabel)

(iv) $\varphi_0 = 1 \quad \varphi_1 = \frac{1}{x}$ (rationale Approximation)

Lösung zu (ii): Man erhält das lineare Gleichungssystem (exakt)

$$\begin{pmatrix} 4.00000000 & 1.62000000 & 1.26040000 \\ 1.62000000 & 1.26040000 & 1.12600800 \\ 1.26040000 & 1.12600800 & 1.06260016 \end{pmatrix} \begin{pmatrix} c_0^{(0)} \\ c_1^{(0)} \\ c_2^{(0)} \end{pmatrix} = \begin{pmatrix} 61 \\ 2.5 \\ 0.37 \end{pmatrix}.$$

Bei 14-stelliger dezimaler Rechnung und Rundung der Ergebnisse auf 6-stellige Mantisse erhält man:

$$c_0^{(0)} = 39.6789, \quad c_1^{(0)} = -136.551, \quad c_2^{(0)} = 97.9830,$$
$$\Phi_2^{(0)}(x) = 39.6789 - 136.551x + 97.9830x^2,$$

$\Phi_2^{(0)}$ ist die quadratische Ausgleichsparabel zu den vorgegebenen Punkten.

Lösung zu (iv): Hier erhält man mit $\sum := \sum_{i=0}^{3}$ das System

$$\begin{pmatrix} \sum 1 & \sum \frac{1}{x_i} \\ \sum \frac{1}{x_i} & \sum \frac{1}{x_i^2} \end{pmatrix} \begin{pmatrix} c_0^{(0)} \\ c_1^{(0)} \end{pmatrix} = \begin{pmatrix} \sum f(x_i) \\ \sum \frac{1}{x_i} f(x_i) \end{pmatrix},$$

$$\Rightarrow \quad \begin{pmatrix} 4 & 63 \\ 63 & 2605 \end{pmatrix} \begin{pmatrix} c_0^{(0)} \\ c_1^{(0)} \end{pmatrix} = \begin{pmatrix} 61 \\ 2602 \end{pmatrix}.$$

Man erhält:

$$c_0^{(0)} = -0.778329, \quad c_1^{(0)} = 1.01767,$$
$$\Phi_{rat}^{(0)}(x) = -0.778329 + 1.01767 \cdot \tfrac{1}{x}.$$

Die folgende Abbildung 8.5(a) enthält die Graphen der besten Approximationsfunktionen zu den Polynomansätzen (i), (ii), (iii), Abbildung 8.5(b) zeigt den Graphen der besten Approximation zu dem rationalen Ansatz (iv). Auf sechs Dezimalen gerundet lauten diese Approximationsfunktionen:

(i) $\Phi_1^{(0)}(x) = 30.131723 - 36.744994x,$

(ii) $\Phi_2^{(0)}(x) = 39.678891 - 136.551407x + 97.982954x^2,$

(iii) $\Phi_3^{(0)}(x) = 62.981434 - 680.869756x + 1609.739229x^2 - 991.850907x^3,$

(iv) $\Phi_{rat}^{(0)}(x) = -0.778329 + 1.017672 \cdot \tfrac{1}{x}$.

Man erkennt unschwer, dass der rationale Ansatz hier der einzig geeignete Ansatz unter den vier Modellfunktionen (i), (ii), (iii), (iv) ist.

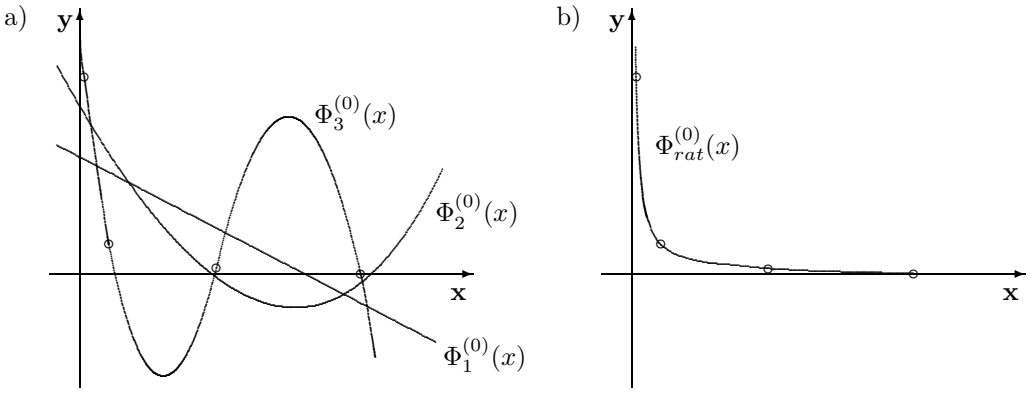


Abb. 8.5a,b. Graph von $\Phi_{rat}^{(0)}$

Nur unter der Voraussetzung, dass der Ansatz für die Modellfunktion Φ im Sinne der Anwendung vernünftig ist, ist mit der Fehlerquadratmethode eine gute Approximation zu erwarten. Ist keine Modellvorstellung vorhanden, so sollten Ausgleichssplines (siehe Kapitel 10) verwendet werden. Liegen mehrere Modelle vor, so kann man nicht generell unter den besten Approximationen $\Phi^{(0)}$ diejenige auswählen, die die kleinste Fehlerquadratsumme besitzt: In unserem Beispiel hat $\Phi_3^{(0)}$ als Interpolationspolynom die kleinstmögliche Fehlerquadratsumme 0 und diesbezüglich einen besseren Wert als $\Phi_{rat}^{(0)}$! □

8.2.3.2 Diskreter Ausgleich durch algebraische Polynome unter Verwendung orthogonaler Polynome

Werden als Approximationsfunktionen Φ algebraische Polynome (8.11) verwendet, dann lautet (8.16) mit $\varphi_k(x_i) = x_i^k$

$$\sum_{k=0}^{n} c_k^{(0)} \sum_{i=0}^{N} w_i x_i^{k+j} = \sum_{i=0}^{N} w_i f(x_i) x_i^j \,, \quad j = 0(1)n \,. \tag{8.20}$$

Für gleiche Gewichte $w_i = 1$ gilt speziell

$$\begin{pmatrix} N+1 & \sum x_i & \sum x_i^2 & \cdots & \sum x_i^n \\ \sum x_i & \sum x_i^2 & \sum x_i^3 & \cdots & \sum x_i^{n+1} \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 & \cdots & \sum x_i^{n+2} \\ \vdots & \vdots & \vdots & & \vdots \\ \sum x_i^n & \sum x_i^{n+1} & \sum x_i^{n+2} & \cdots & \sum x_i^{2n} \end{pmatrix} \begin{pmatrix} c_0^{(0)} \\ c_1^{(0)} \\ c_2^{(0)} \\ \vdots \\ c_n^{(0)} \end{pmatrix} = \begin{pmatrix} \sum f(x_i) \\ \sum f(x_i) x_i \\ \sum f(x_i) x_i^2 \\ \vdots \\ \sum f(x_i) x_i^n \end{pmatrix}, \tag{8.21}$$

wobei jede Summe über $i = 0(1)N$ läuft. Die Matrix in (8.20) bzw. (8.21) ist oft schlecht konditioniert. Dann ergeben sich bei Rechnung mit endlicher Stellenzahl im Allgemeinen total verfälschte Lösungen. Deshalb sollte man grundsätzlich nicht das System (8.21) direkt lösen, sondern zumindest die Householder-Transformation einsetzen (vgl. Abschnitt 8.2.3.4) oder mit einem Ansatz über diskrete orthogonale Polynome $\varphi_k \equiv Q_k$ genau k-ten Grades für die Approximationsfunktion Φ arbeiten:

$$\begin{aligned} \Phi(x) &= \sum_{k=0}^{n} c_k \, Q_k(x) \\ &= c_0 \, Q_0(x) + c_1 \, Q_1(x) + \ldots + c_n \, Q_n(x) \,. \end{aligned}$$

Wegen $(\varphi_j, \varphi_k) \equiv (Q_j, Q_k) = 0$ für $j \neq k$ haben dann die Gaußschen Normalgleichungen (8.16) Diagonalgestalt, und man kann direkt nach den $c_j^{(0)}$ auflösen:

$$c_j^{(0)} = \frac{(f, Q_j)}{(Q_j, Q_j)} \,, \qquad j = 0(1)n \,,$$

mit den Skalarprodukten

$$\begin{aligned} (f, Q_j) &:= \sum_{i=0}^{N} w_i \, f(x_i) \, Q_j(x_i) \,, \\ (Q_j, Q_j) &:= \sum_{i=0}^{N} w_i \, Q_j(x_i) \, Q_j(x_i) \,. \end{aligned}$$

Die orthogonalen Polynome Q_k sind abhängig von den gegebenen Stellen x_i und den Gewichten w_i und lassen sich rekursiv wie folgt berechnen:

$$
\begin{aligned}
Q_0(x) &= 1 \\
Q_1(x) &= x - b_1 \\
Q_k(x) &= (x - b_k)\, Q_{k-1}(x) - d_k\, Q_{k-2}(x)\,, \quad k \geq 2 \quad \text{mit} \\
b_k &= \frac{(xQ_{k-1}, Q_{k-1})}{(Q_{k-1}, Q_{k-1})}\,, \qquad k \geq 1\,, \\
d_k &= \frac{(Q_{k-1}, Q_{k-1})}{(Q_{k-2}, Q_{k-2})}\,, \qquad k \geq 2\,,
\end{aligned}
$$

wobei für das Skalarprodukt im Zähler von b_k gilt

$$(xQ_{k-1}, Q_{k-1}) := \sum_{i=0}^{N} w_i\, x_i\, Q_{k-1}(x_i)\, Q_{k-1}(x_i)\,,$$

die übrigen Skalarprodukte ergeben sich gemäß der Definition für die (Q_j, Q_j). Mit Hilfe der b_k, d_k und $c_k^{(0)} = (f, Q_k)/(Q_k, Q_k)$ kann man das Ausgleichspolynom $\Phi^{(0)}$ an jeder beliebigen Stelle x (hornerartig) wie folgt berechnen:

1. $s_n = c_n^{(0)}$
   $s_{n-1} = c_{n-1}^{(0)} + s_n(x - b_n)$
2. Für jedes $k = n-2, n-3, ..., 0$
   $s_k = c_k^{(0)} + s_{k+1}\,(x - b_{k+1}) - s_{k+2}\, d_{k+2}$

$\Rightarrow\ \Phi^{(0)}(x) = s_0.$

Ein Vorteil dieser Art des polynomialen Ausgleichs ist die Tatsache, dass man mit dem Ausgleichspolynom $\Phi^{(0)}(x) = \Phi_n^{(0)}(x)$ n-ten Grades auch jedes Ausgleichspolynom $\Phi_m^{(0)}$ m-ten Grades mit $m \leq n$ kennt:

$$\Phi_m^{(0)}(x) = \sum_{k=0}^{m} c_k^{(0)} Q_k(x)\,;$$

es hat dieselben Koeffizienten $c_k^{(0)}$ wie $\Phi_n^{(0)}$.

Beispiel 8.15. (vgl. Beispiel 8.14)

Gegeben: Die Wertetabelle $\big(x_i, f(x_i)\big)$, $i = 0(1)N$, $N = 3$

i	0	1	2	3
x_i	0.02	0.10	0.50	1.00
$f(x_i)$	50	10	1	0

Gesucht: Unter allen $\Phi \in C[0.02, 1.00]$ mit $\Phi(x) = \sum c_k \varphi_k(x)$ die beste Approximation $\Phi^{(0)}$ mit algebraischen Polynomen unter Verwendung orthogonaler Polynome.

Lösung: Man erhält die Q_j mit 14-stelliger dezimaler Rechnung und Rundung der Ergebnisse auf 6-stellige Mantissen:

$$\begin{array}{lll} Q_0(x) = & 1, & b_1 = 0.405 \\ Q_1(x) = & x - b_1, & b_2 = 0.613610, \quad d_2 = 0.151075 \\ Q_2(x) = & (x-b_2)(x-b_1) - d_2, & b_3 = 0.505567, \quad d_3 = 0.0636221 \\ Q_3(x) = & (x-b_3)(x-b_2)(x-b_1) - & \\ & d_2(x-b_3) - d_3(x-b_1) & \end{array}$$

Als Koeffizienten $c_j^{(0)}$ ergeben sich aus der Formel $c_j^{(0)} = \frac{(f,Q_j)}{(Q_j,Q_j)}$ bei gleicher Rechnung:

$$\begin{array}{lll} c_0^{(0)} & = & 15.2500 \\ c_1^{(0)} & = & -36.7450 \\ c_2^{(0)} & = & 97.9830 \\ c_3^{(0)} & = & -991.852 \end{array}$$

Damit erhält man gleichzeitig als Ausgleichskurven:

$$\begin{array}{ll} \Phi_1^{(0)}(x) = c_0^{(0)} + c_1^{(0)} Q_1(x) & \text{(Gerade)} \\ \Phi_2^{(0)}(x) = c_0^{(0)} + c_1^{(0)} Q_1(x) + c_2^{(0)} Q_2(x) & \text{(quadr. Parabel)} \\ \Phi_3^{(0)}(x) = c_0^{(0)} + c_1^{(0)} Q_1(x) + c_2^{(0)} Q_2(x) + c_3^{(0)} Q_3(x) & \text{(kub. Parabel)} \end{array}$$

Diese Funktionen stimmen mit denen aus Beispiel 8.14 überein! □

8.2.3.3 Lineare Regression – Ausgleich durch lineare algebraische Polynome

Als eine wichtige Anwendung für den diskreten Ausgleich durch lineare algebraische Polynome wird die lineare Regression (Statistik) behandelt.
Gegeben sind in der x,y-Ebene $N+1$ Punkte (x_i, y_i), $i = 0(1)N$, die $N+1$ Ausprägungen der Merkmale x, y in der Merkmalsebene darstellen. Gesucht sind zur Beschreibung des Zusammenhangs beider Merkmale in der Merkmalsebene zwei Regressionsgeraden, je eine für die Abhängigkeit des Merkmals y von x bzw. x von y, mit den Gleichungen

$$\begin{array}{lll} g_1 : y & = & \Phi^{(0)}(x) = c_0^{(0)} + c_1^{(0)} x \qquad \text{(Regression von } y \text{ auf } x), \\ g_2 : x & = & \widetilde{\Phi}^{(0)}(y) = \widetilde{c}_0^{(0)} + \widetilde{c}_1^{(0)} y \qquad \text{(Regression von } x \text{ auf } y). \end{array}$$

Die Koeffizienten $c_0^{(0)}$, $c_1^{(0)}$ zu g_1 ergeben sich aus den Normalgleichungen (8.16) mit $w_i = 1$, $\varphi_0(x) = 1$, $\varphi_1(x) = x$ bzw. (8.21) für $n = 1$.

Die Normalgleichungen zur Berechnung von $c_0^{(0)}$ und $c_1^{(0)}$ lauten demnach:

$$\begin{pmatrix} N+1 & \sum x_i \\ \sum x_i & \sum x_i^2 \end{pmatrix} \begin{pmatrix} c_0^{(0)} \\ c_1^{(0)} \end{pmatrix} = \begin{pmatrix} \sum y_i \\ \sum x_i y_i \end{pmatrix} \quad \text{mit} \quad \sum := \sum_{i=0}^{N}. \tag{8.22}$$

Da sie im Allgemeinen schlecht konditioniert sind, sollte man die in Abschnitt 8.2.3.2 beschriebene Methode verwenden.

Fasst man nun y als unabhängige und x als abhängige Variable auf, so ergeben sich entsprechend die Koeffizienten $\widetilde{c}_0^{(0)}$, $\widetilde{c}_1^{(0)}$ für g_2. Der Schwerpunkt $(\overline{x}, \overline{y})$ mit

$$\overline{x} = \frac{1}{N+1} \sum_{i=0}^{N} x_i, \quad \overline{y} = \frac{1}{N+1} \sum_{i=0}^{N} y_i$$

ist stets der Schnittpunkt der beiden Regressionsgeraden. Eine geringe Abweichung der Geraden voneinander ist dafür maßgebend, ob mit Recht näherungsweise von einem linearen Zusammenhang der Merkmale x, y gesprochen werden kann (ohne deren Kausalität festzulegen).

Beispiel 8.16.

Gegeben: Von 24 Schülern einer Klasse werden ihre Körperlänge und ihr Gewicht erfasst:

Nr.	x Körperlänge [cm]	y Körpergewicht [kg]	Nr.	x Körperlänge [cm]	y Körpergewicht [kg]
1	162	50	13	168	52
2	155	49	14	164	65
3	172	68	15	163	55
4	163	45	16	164	52
5	163	50	17	160	50
6	166	49	18	170	53
7	168	58	19	163	57
8	170	61	20	159	50
9	159	52	21	168	58.5
10	155	47	22	168	67
11	172	61	23	157	47
12	169	55	24	163	47

Gesucht: Eine Regressionsgerade von Körpergewicht auf -länge und eine Regressionsgerade von Körperlänge auf -gewicht.

Lösung:

1. Regression von Körpergewicht (y) auf -länge (x):

   Das zugehörige lineare Gleichungssystem (8.22) lautet

$$\begin{pmatrix} 24 & 3941 \\ 3941 & 647723 \end{pmatrix} \begin{pmatrix} c_0^{(0)} \\ c_1^{(0)} \end{pmatrix} = \begin{pmatrix} 1298.5 \\ 213728 \end{pmatrix}.$$

   Bezeichnet $\boldsymbol{A}$ die Matrix dieses Gleichungssystems, dann ist deren Konditionszahl (siehe Kapitel 4) gegeben durch

$$\mathrm{cond}_\infty(\boldsymbol{A}) = \|\boldsymbol{A}\|_\infty \cdot \|\boldsymbol{A}^{-1}\|_\infty \approx 3 \cdot 10^7,$$

   so dass eine direkte Auflösung dieses Systems mit auf Rechnern gebräuchlicher etwa sechsstelliger dezimaler Arithmetik zu keiner unbedingt sicheren Stelle im Ergebnis mehr führt!

   Deshalb sollte man besser die Methode aus dem letzten Abschnitt anwenden. Man erhält schließlich:

$$\text{Körpergewicht:} \quad \Phi^{(0)}(x) = 0.871134x - 88.9433\,.$$

2. Regression von Körperlänge (x) auf -gewicht (y):

   Durch Vertauschung der Rollen von x und y folgt analog

$$\text{Körperlänge:} \quad \widetilde{\Phi}^{(0)}(y) = 0.510633y + 136.581\,.$$

Die beiden Geraden schneiden sich im Punkt P (164.208333; 54.1041666), siehe folgende Graphen ($\Phi^{(0)}$ verläuft steiler als $\widetilde{\Phi}^{(0)}$):

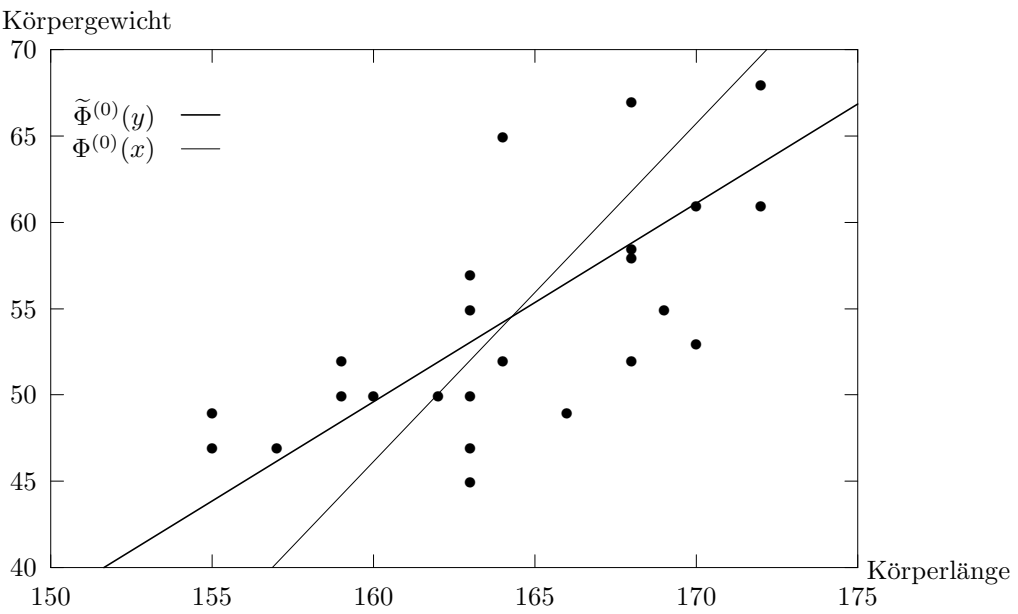


Abb. 8.6. Darstellung der beiden Regressionsgeraden mit den gegebenen Wertepaaren

Nur wenn alle gegebenen Punkte (x_i, y_i) auf einer Gerade lägen, wären beide Regressionsgeraden deckungsgleich. So besteht zwar kein strenger linearer Zusammenhang zwischen den Merkmalen Körperlänge und Körpergewicht (zur Information: der empirische Korrelationskoeffizient ist $r = 0.667$), wohl aber eine deutlich gleichsinnige Tendenz. □

8.2.3.4 Householder-Transformation zur Lösung des linearen Ausgleichsproblems

Gegeben seien $N+1$ Wertepaare $\big(x_i, f(x_i)\big)$, $i = 0(1)N$, und $N+1$ Gewichte $w_i > 0$. Gesucht sind die Koeffizienten $c_0, \ldots, c_n$ im linearen Approximationsmodell

$$\Phi(x, \boldsymbol{c}) = c_0\varphi_0(x) + c_1\varphi_1(x) + \ldots + c_n\varphi_n(x)\,,$$

so dass die gewichtete Fehlerquadratsumme

$$\sum_{i=0}^{N} w_i\big(f(x_i) - \Phi(x_i, c_0, c_1, \ldots, c_n)\big)^2$$

minimiert wird (vgl. 8.15).

Ohne Beschränkung der Allgemeinheit wird zwecks einfacherer Darstellung für alle i $w_i = 1$ gesetzt. Mit $\boldsymbol{c} = (c_0, c_1, \ldots, c_n)^\mathsf{T}$ und wegen

$$\Phi(x_i, \boldsymbol{c}) = c_0\varphi_0(x_i) + c_1\varphi_1(x_i) + \ldots + c_n\varphi_n(x_i)$$

für $i = 0(1)N$ gilt

$$\begin{pmatrix} \Phi(x_0, \boldsymbol{c}) \\ \Phi(x_1, \boldsymbol{c}) \\ \vdots \\ \Phi(x_N, \boldsymbol{c}) \end{pmatrix} = \begin{pmatrix} \varphi_0(x_0) & \varphi_1(x_0) & \cdots & \varphi_n(x_0) \\ \varphi_0(x_1) & \varphi_1(x_1) & \cdots & \varphi_n(x_1) \\ \vdots & \vdots & & \vdots \\ \varphi_0(x_N) & \varphi_1(x_N) & \cdots & \varphi_n(x_N) \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{pmatrix} =: \boldsymbol{A}\boldsymbol{c} \tag{8.23}$$

mit der $(N+1, n+1)$-Matrix $\boldsymbol{A}$, die spaltenweise aus den Vektoren

$$\boldsymbol{\varphi}_j := \big(\varphi_j(x_0), \varphi_j(x_1), \ldots, \varphi_j(x_N)\big)^\mathsf{T}, \quad j = 0(1)n$$

aufgebaut ist. Mit

$$\boldsymbol{f} := \big(f(x_0), f(x_1), \ldots, f(x_N)\big)^\mathsf{T}$$

gelten für die Skalarprodukte (8.17) mit $w_i = 1$ die Aussagen in (8.19), und es ergibt sich für die Normalgleichungen (8.18) wegen $\boldsymbol{G} = \boldsymbol{A}^\mathsf{T}\boldsymbol{A}$ und $\boldsymbol{a} = \boldsymbol{A}^\mathsf{T}\boldsymbol{f}$

$$\boldsymbol{A}^\mathsf{T}\boldsymbol{A}\boldsymbol{c} = \boldsymbol{A}^\mathsf{T}\boldsymbol{f}. \tag{8.24}$$

Im Falle beliebiger $w_i > 0$ müssen in (8.23), (8.24) die Vektoren

$$\boldsymbol{\varphi}_j := \begin{pmatrix} \sqrt{w_0}\,\varphi_j(x_0) \\ \vdots \\ \sqrt{w_N}\,\varphi_j(x_N) \end{pmatrix}, \quad \boldsymbol{f} := \begin{pmatrix} \sqrt{w_0}\,f(x_0) \\ \vdots \\ \sqrt{w_N}\,f(x_N) \end{pmatrix}$$

verwendet werden. Es ist nun

$$\sum_{i=0}^{N} \big(f(x_i) - \Phi(x_i, \boldsymbol{c})\big)^2 \quad \stackrel{!}{=} \quad \text{Min}$$

gleichbedeutend mit

$$\|\boldsymbol{f} - \boldsymbol{A}\boldsymbol{c}\|_{d,2}^2 \quad \stackrel{!}{=} \quad \text{Min}, \tag{8.25}$$

d. h. $\boldsymbol{c}$ ist genau dann Lösung der Normalgleichungen (8.24), wenn $\boldsymbol{c}$ auch Optimallösung für (8.25) ist. (Beweis siehe [STOE1989]).

Dann ergibt sich das Problem, das überbestimmte lineare Gleichungssystem

$$\boldsymbol{A}\boldsymbol{c} = \boldsymbol{f}$$

in dieser Weise optimal zu lösen. Geschieht dies mit Hilfe der Householder-Transformation, kommen dabei die vorgenannten Konditionsprobleme nicht zur Wirkung, da bei der Überführung von $\boldsymbol{A}$ in eine obere Dreiecksmatrix, die zur Lösungsbestimmung direkt genutzt werden kann, keine Konditionsverschlechterung eintritt.

Mit dem folgenden Algorithmus wird hier eine kurze Beschreibung des Verfahrens angegeben.

Algorithmus 8.17. (*Linearer Ausgleich mit Householder-Transformation*)

Gegeben: $(N+1, n+1)$-Matrix $\boldsymbol{A}$, $N \geq n$,
$\boldsymbol{f} \in \mathbb{R}^{N+1}$, $\text{Rang}(\boldsymbol{A}) = n+1$, $\boldsymbol{A}$ gemäß (8.23)

Gesucht: Optimale Lösung $\boldsymbol{c} = \boldsymbol{c}^{(0)} \in \mathbb{R}^{n+1}$ des überbestimmten Systems $\boldsymbol{A}\boldsymbol{c} = \boldsymbol{f}$ bezüglich des mittleren quadratischen Fehlers (8.25).

Man setze $\boldsymbol{A}^{(0)} := \boldsymbol{A}$, $\boldsymbol{f}^{(0)} := \boldsymbol{f}$ und berechne für jedes $i = 0(1)n$

$$1. \quad \boldsymbol{A}^{(i+1)} = \boldsymbol{H}_i \boldsymbol{A}^{(i)} \qquad 2. \quad \boldsymbol{f}^{(i+1)} = \boldsymbol{H}_i \boldsymbol{f}^{(i)}$$

mit den Householder-Transformationen $\boldsymbol{H}_i$. $\boldsymbol{H}_i$ sind orthogonale $(N+1, N+1)$-Matrizen, die gemäß Abschnitt 4.13 konstruiert werden.
Es gilt wegen der unitären Transformationen

$$\|\boldsymbol{f} - \boldsymbol{A}\boldsymbol{c}\|_{d,2} = \|\boldsymbol{f}^{(n+1)} - \boldsymbol{A}^{(n+1)}\boldsymbol{c}\|_{d,2}.$$

Man erhält mit

$$\boldsymbol{f}^{(n+1)} = \begin{pmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{pmatrix}, \quad \boldsymbol{A}^{(n+1)} = \begin{pmatrix} \boldsymbol{B} \\ \boldsymbol{0} \end{pmatrix}$$

$$\boldsymbol{f}^{(n+1)} - \boldsymbol{A}^{(n+1)}\boldsymbol{c} = \begin{pmatrix} \boldsymbol{b}_1 - \boldsymbol{B}\boldsymbol{c} \\ \boldsymbol{b}_2 \end{pmatrix},$$

wobei $\boldsymbol{B}$ eine $(n+1, n+1)$-Matrix von oberer Dreiecksgestalt ist, $\boldsymbol{0}$ eine $(N-n, n+1)$-Nullmatrix, $\boldsymbol{b}_1 \in \mathbb{R}^{n+1}$, $\boldsymbol{b}_2 \in \mathbb{R}^{N-n}$.

$\|\boldsymbol{f} - \boldsymbol{A}\boldsymbol{c}\|_{d,2}$ und damit (8.25) wird minimiert, wenn $\boldsymbol{c} = \boldsymbol{c}^{(0)}$ so gewählt wird, dass

$$\boldsymbol{B}\boldsymbol{c}^{(0)} = \boldsymbol{b}_1$$

gilt. Dieses gestaffelte Gleichungssystem kann direkt rekursiv gelöst werden.

Wegen $\text{Rang}(\boldsymbol{A}) = n + 1$ gilt $\det(\boldsymbol{B}) \neq 0$, und es existiert immer eine eindeutige Lösung $\boldsymbol{c}^{(0)}$ des linearen Ausgleichsproblems.

Beispiel 8.18. (vgl. Beispiel 8.14)

Gegeben: Die Wertetabelle $\big(x_i, f(x_i)\big)$, $i = 0(1)N$, $N = 3$

i	0	1	2	3
x_i	0.02	0.10	0.50	1.00
$f(x_i)$	50	10	1	0

Gesucht: Unter allen $\Phi \in C[0.02, 1.00]$ mit der Modellfunktion $\Phi(x) = c_0\,\varphi_0(x) + c_1\,\varphi_1(x)$ mit $\varphi_0(x) = 1$, $\varphi_1(x) = 1/x$ die beste Approximation $\Phi^{(0)}$ mit Hilfe des linearen Ausgleichs durch die Householder-Transformation.

Lösung: Aufstellen des überbestimmten Gleichungssystems $\boldsymbol{A}\boldsymbol{c} = \boldsymbol{f}$:

$$\begin{pmatrix} \varphi_0(x_0) & \varphi_1(x_0) \\ \varphi_0(x_1) & \varphi_1(x_1) \\ \varphi_0(x_2) & \varphi_1(x_2) \\ \varphi_0(x_3) & \varphi_1(x_3) \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 50 \\ 1 & 10 \\ 1 & 2 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \end{pmatrix} = \begin{pmatrix} 50 \\ 10 \\ 1 \\ 0 \end{pmatrix}$$

Nach Anwenden des Algorithmus 8.17 erhält man folgende auf 6 Mantissenstellen angegebene Lösung:

$$\begin{aligned} c_0^{(0)} &= -0.778329 \\ c_1^{(0)} &= 1.01767, \end{aligned}$$

also $\Phi^{(0)}(x) = -0.778329 + 1.01767 \cdot \frac{1}{x}$.

Mit Hilfe der Gaußschen Fehlerquadratmethode erhält man die gleiche Lösung für dieses Beispiel (s. Beispiel 8.14. (iv)). Dies ist im Allgemeinen bei Rechnung mit endlicher Stellenzahl jedoch nicht der Fall, da der lineare Ausgleich mit Householder-Transformation nicht so wie das direkte Lösen der Gaußschen Normalgleichungen von der schlechten Kondition der Matrix des Gleichungssystems negativ beeinflusst wird. □

8.2.4 Approximation von Polynomen durch Tschebyscheff-Polynome

Wenn von einer Funktion $f \in C[a,b]$ viele Funktionswerte berechnet werden müssen, kann es zweckmäßig sein, sie durch eine Funktion Φ zu approximieren, deren Auswertung weniger aufwändig ist als die von f. Dann muss allerdings gewährleistet sein, dass für alle x aus $[a,b]$ für den absoluten Fehler $|f(x) - \Phi(x)| \leq \varepsilon$ gilt mit einer von x unabhängigen Schranke $\varepsilon > 0$. Bei der Approximation im quadratischen Mittel kann eine solche von x unabhängige Schranke für den absoluten Fehler nicht vorgegeben werden, dagegen ist dies bei der sogenannten *gleichmäßigen* oder *Tschebyscheffschen Approximation* möglich.

Hier wird nur der Fall der gleichmäßigen Approximation von Polynomen durch *Tschebyscheff-Polynome* angegeben. Auf diesen Fall lässt sich die gleichmäßige Approximation einer Funktion f durch eine Approximationsfunktion Φ wie folgt zurückführen:

Die nach einem bestimmten Glied abgebrochene Taylorentwicklung von f, deren Restglied im Intervall $[a,b]$ nach oben abgeschätzt wird und den Abbruchfehler liefert, stellt ein Polynom dar, das mit Hilfe einer Linearkombination von Tschebyscheff-Polynomen gleichmäßig approximiert werden kann. Abbruch- und Approximationsfehler sollen dabei die gleiche Größenordnung haben und eine unterhalb der vorgegebenen Schranke ε liegende Summe besitzen.

Der Grad des Approximationspolynoms ist in der Regel kleiner als der des Polynoms, das durch Abbrechen der Taylorentwicklung entsteht. Dann ist auch die Berechnung von Werten des Approximationspolynoms weniger aufwändig als die von Werten der abgebrochenen Taylorentwicklung. Die Ermittlung des Approximationspolynoms erfordert einen einmaligen Rechenaufwand, der sich allerdings nur dann lohnt, wenn zahlreiche Funktionswerte nach der geschilderten Methode berechnet werden sollen.

8.2.4.1 Beste gleichmäßige Approximation, Definition

Als Norm einer auf dem Intervall $[a,b]$ gegebenen Funktion g wird die sogenannte *Maximumnorm*

$$\|g\|_\infty = \max_{x\in[a,b]} |g(x)|\, w(x)$$

zugrunde gelegt mit der Gewichtsfunktion $w(x) > 0$; zur Gewichtsfunktion vergleiche Bemerkung 8.6. Mit $\overline{C}$ wird die Menge aller Linearkombinationen Φ der Gestalt (8.2) zu einem auf $[a,b]$ gegebenen System linear unabhängiger Funktionen $\varphi_0, \varphi_1, ..., \varphi_n$ bezeichnet. Eine beste Approximation $\Phi^{(0)}$ der Gestalt (8.4) unter allen Funktionen $\Phi \in \overline{C}$ besitzt gemäß (8.5) mit der Maximumnorm die Eigenschaft

$$\begin{aligned} \|f - \Phi^{(0)}\|_\infty &= \max_{x\in[a,b]} |f(x) - \Phi^{(0)}(x)|\, w(x) \\ &= \min_{\Phi\in\overline{C}} \Big(\max_{x\in[a,b]} |f(x) - \Phi(x)|\, w(x) \Big), \end{aligned} \tag{8.26}$$

so dass das Maximum des absoluten Fehlers $|f(x) - \Phi^{(0)}(x)|$, gewichtet mit der Funktion $w(x)$, auf dem ganzen Intervall $[a, b]$ minimal wird. Damit ist gewährleistet, dass der gewichtete absolute Fehler $w|f - \Phi^{(0)}|$ auf $[a, b]$ einen bestmöglichen Wert $\varepsilon > 0$ nicht überschreitet; es gilt also $w(x)|f(x) - \Phi^{(0)}(x)| \leq \varepsilon$ für alle $x \in [a, b]$, d. h. f wird durch $\Phi^{(0)}$ bezüglich w mit der *Genauigkeit* ε approximiert.

Eine beste Approximation $\Phi^{(0)}$ im Sinne der Maximumnorm heißt deshalb *beste gleichmäßige Approximation* für f in der Funktionenklasse $\overline{C}$.
Im Falle der gleichmäßigen Approximation einer beliebigen Funktion $f \in C[a, b]$ gibt es im Gegensatz zur Approximation im quadratischen Mittel kein allgemeines Verfahren zur Bestimmung der in $\Phi^{(0)}$ auftretenden Koeffizienten $c_k^{(0)}$ (über Näherungsverfahren vgl. [BERE1971] Bd.1, 4.5; [MEIN1967], §7; [WERN1993], II, §§4-6). Hier wird nur der für die Praxis wichtige Sonderfall der gleichmäßigen Approximation von Polynomen durch sogenannte Tschebyscheff-Polynome behandelt.

8.2.4.2 Approximation durch Tschebyscheff-Polynome

Jedes Intervall $[a, b]$ kann durch lineare Transformation in das Intervall $[-1, +1]$ überführt werden (vgl. auch die Anmerkung nach Algorithmus 8.21).

A. Einführung der Tschebyscheff-Polynome

Als Funktionensystem $\varphi_0, \varphi_1, ..., \varphi_n$ werden die *Tschebyscheff-Polynome* $T_0, T_1, \ldots, T_n$ mit

$$T_k(x) = \cos(k \arccos x), \quad k = 0(1)n, \quad x \in [-1, +1], \tag{8.27}$$

gewählt, es sind für $n = 5$

$$\begin{cases} T_0(x) = 1\,, & T_3(x) = 4x^3 - 3x\,, \\ T_1(x) = x\,, & T_4(x) = 8x^4 - 8x^2 + 1\,, \\ T_2(x) = 2x^2 - 1\,, & T_5(x) = 16x^5 - 20x^3 + 5x\,. \end{cases} \tag{8.28}$$

Allgemein lassen sich die Tschebyscheff-Polynome mit Hilfe der Rekursionsformel

$$T_{k+1} = 2xT_k - T_{k-1}, \quad T_0 = 1, \quad T_1 = x, \quad k = 1(1)n \;\; (n \in \mathbb{N}) \tag{8.29}$$

berechnen. Wichtige *Eigenschaften* der Tschebyscheff-Polynome sind:

1. T_k ist ein Polynom in x vom Grade k.
2. Der Koeffizient von x^k in T_k ist 2^{k-1}.
3. Für alle k und alle $x \in [-1, +1]$ gilt $|T_k(x)| \leq 1$.
4. Die Extremwerte $T_k(x_j) = \pm 1$ werden an den $k + 1$ Stellen $x_j = \cos \frac{j\pi}{k}$, $j = 0(1)k$, angenommen.
5. T_k besitzt in $[-1, +1]$ genau k reelle Nullstellen $x_j = \cos \frac{2j+1}{k} \frac{\pi}{2}$, $j = 0(1)k - 1$.

(Siehe auch [SAUE1969] Bd. III, S.356-360).

B. Darstellung von Polynomen als Linearkombination von Tschebyscheff-Polynomen

Die Potenzen von x lassen sich wegen (8.28) bzw. (8.29) als Linearkombinationen von Tschebyscheff-Polynomen schreiben (T_0, T_2, $T_4, \ldots$ sind gerade, T_1, T_3, $T_5, \ldots$ ungerade Funktionen). Es sind

$$\left\{\begin{array}{ll} 1 = T_0\,, & x^3 = 2^{-2}(3T_1 + T_3)\,, \\ x = T_1\,, & x^4 = 2^{-3}(3T_0 + 4T_2 + T_4)\,, \\ x^2 = 2^{-1}(T_0 + T_2)\,, & x^5 = 2^{-4}(10T_1 + 5T_3 + T_5)\,, \end{array}\right. \tag{8.30}$$

und allgemein gilt für $k = 0, 1, 2, \ldots$

$$x^k = 2^{1-k}(T_k + \binom{k}{1} T_{k-2} + \binom{k}{2} T_{k-4} + \ldots + T^*)\,, \tag{8.31}$$

wobei das letzte Glied T^* für ungerades k die Form

$$T^* = \left(\begin{array}{c} k \\ \frac{k-1}{2} \end{array} \right) T_1$$

besitzt und für gerades k die Form

$$T^* = \frac{1}{2} \left(\begin{array}{c} k \\ \frac{k}{2} \end{array} \right) T_0\,.$$

Jedes Polynom in x vom Grad m

$$P_m(x) = \sum_{i=0}^{m} a_i x^i\,, \qquad a_i \in \mathbb{R}\,, \tag{8.32}$$

lässt sich eindeutig als Linearkombination (T-Entwicklung)

$$P_m(x) = \sum_{j=0}^{m} b_j T_j(x) \tag{8.33}$$

von Tschebyscheff-Polynomen ausdrücken. Man erhält (8.33), indem man in (8.32) die Potenzen von x durch (8.30) bzw. (8.31) ersetzt.

Zur Bestimmung der Koeffizienten b_j der T-Entwicklung (8.33) aus den Koeffizienten a_i von (8.32) dient für $i, j = 0(1)10$ das folgende Rechenschema.

Rechenschema 8.19.

a_0	1					
$\frac{a_2}{2}$	1	1				
$\frac{a_4}{8}$	3	4	1			
$\frac{a_6}{32}$	10	15	6	1		
$\frac{a_8}{128}$	35	56	28	8	1	
$\frac{a_{10}}{512}$	126	210	120	45	10	1
	b_0	b_2	b_4	b_6	b_8	b_{10}

a_1	1				
$\frac{a_3}{4}$	3	1			
$\frac{a_5}{16}$	10	5	1		
$\frac{a_7}{64}$	35	21	7	1	
$\frac{a_9}{256}$	126	84	36	9	1
	b_1	b_3	b_5	b_7	b_9

Die (außer a_0) durch 2^{i-1} dividierten Koeffizienten a_i (linke Spalte) werden jeweils mit derjenigen Zahl in der zugehörigen Zeile multipliziert, die in der Spalte über dem gesuchten Koeffizienten b_j steht und auch dort eingetragen. Die Spaltensumme der eingetragenen Zahlen liefert dann den Koeffizienten b_j der T-Entwicklung.

So führt $P(x) = x^5$ mit $a_5 = 1,\ a_4 = a_3 = a_2 = a_1 = a_0 = 0$ auf die Koeffizienten $b_0 = 0,\ b_2 = 0,\ b_4 = 0$ und $b_1 = \frac{10}{16},\ b_3 = \frac{5}{16},\ b_5 = \frac{1}{16}$ (vgl. 8.30).

C. Beste gleichmäßige Approximation

Es ist zweckmäßig, neben der T-Entwicklung (8.33) auch deren Teilsummen

$$S_n(x) = \sum_{j=0}^{n} b_j T_j(x)\,, \quad n \le m\,,$$

zu betrachten. Insbesondere ist

$$P_m(x) = S_m(x) = S_{m-1}(x) + b_m T_m(x)\,.$$

Als Approximationsfunktionen für P_m werden die Linearkombinationen Φ mit

$$\Phi(x) = \sum_{k=0}^{n} c_k T_k(x)\,, \quad n < m\,,$$

gewählt. Die Frage nach einer besten gleichmäßigen Approximation $\Phi^{(0)}$ mit

$$\Phi^{(0)}(x) = \sum_{k=0}^{n} c_k^{(0)} T_k(x)\,, \quad n < m\,,$$

für P_m im Sinne von (8.26) beantwortet der folgende Satz.

Satz 8.20.
Die beste gleichmäßige Approximation $\Phi^{(0)}$ eines Polynoms P_m durch ein Polynom $(m-1)$-ten Grades im Intervall $[-1,+1]$ ist mit $c_k^{(0)} = b_k$ für $k = 0(1)m-1$ die eindeutig bestimmte Teilsumme

$$\Phi^{(0)}(x) = S_{m-1}(x) = \sum_{k=0}^{m-1} b_k T_k(x)$$

von dessen T-Entwicklung S_m. Für $w(x) \equiv 1$ gilt

$$\|P_m - S_{m-1}\|_\infty = \max_{x\in[-1,+1]} |P_m(x) - S_{m-1}(x)| \le |b_m|$$

(s. [DEMI1968] I, §12; [STIE1976], S.202).

Um $\Phi^{(0)}$ zu erhalten, streicht man also nur in der T-Entwicklung S_m das letzte Glied $b_m T_m$.

D. Gleichmäßige Approximation

Da die Koeffizienten b_j der T-Entwicklung mit wachsendem j in den meisten Fällen dem Betrage nach rasch abnehmen, wird auch beim Weglassen von mehr als einem Glied der T-Entwicklung S_m noch eine sehr gute Approximation des Polynoms P_m erreicht, die nur wenig von der besten gleichmäßigen Approximation S_{m-1} abweicht. Ist dann

$$S_n(x) = \sum_{j=0}^{n} b_j T_j(x)\,, \quad n \le m-1\,,$$

eine Teilsumme der T-Entwicklung S_m, so gilt wegen Eigenschaft 3 der T-Polynome

$$\|P_m - S_n\|_\infty = \max_{x\in[-1,+1]} |P_m(x) - S_n(x)| \le \sum_{j=n+1}^{m} |b_j| = \varepsilon_1\,.$$

Da ε_1 unabhängig von x ist, ist S_n eine gleichmäßige Approximation für P_m, für $n = m-1$ ist es die beste gleichmäßige Approximation.

Um für eine genügend oft differenzierbare Funktion f im Intervall $[-1,+1]$ eine entsprechende Approximationsfunktion Φ zu finden, geht man aus von ihrer Taylorentwicklung an der Stelle $x = 0$

$$f(x) = P_m(x) + R_{m+1}(x)\,,$$

die sich aus einem Polynom P_m und dem Restglied R_{m+1} zusammensetzt. Für alle $x \in [-1,+1]$ gelte mit dem von x unabhängigen *Abbruchfehler* ε_2

$$|R_{m+1}(x)| \le \varepsilon_2\,.$$

Als Approximationsfunktion für f wählt man die Teilsumme S_n der T-Entwicklung S_m für $P_m (n \le m-1)$. Dann ist

$$\begin{aligned} \max_{x\in[-1,+1]} |f(x) - S_n(x)| &= \|f - S_n\|_\infty = \|P_m + R_{m+1} - S_n\|_\infty \\ &\le \|P_m - S_n\|_\infty + \|R_{m+1}\|_\infty \le \varepsilon_1 + \varepsilon_2. \end{aligned}$$

Der maximale absolute Fehler bei der Approximation von f durch S_n setzt sich somit aus dem Fehler ε_1 bei der gleichmäßigen Approximation von P_m durch S_n und dem Abbruchfehler ε_2 zusammen. Wenn bei vorgegebener Genauigkeit ε die Ungleichung $\varepsilon_1 + \varepsilon_2 \le \varepsilon$ erfüllt ist, dann wird wegen $\|f - S_n\|_\infty \le \varepsilon$ die Funktion f durch das Polynom S_n im Intervall $[-1,+1]$ gleichmäßig approximiert.

Algorithmus 8.21. (*Gleichmäßige Approximation durch Tschebyscheff-Polynome*)

Gegeben: Eine für $x \in [-1,+1]$ genügend oft differenzierbare Funktion f.

Gesucht: Ein Approximationspolynom S_n für f mit $|f(x) - S_n(x)| \leq \varepsilon$ für alle $x \in [-1,+1]$ zu vorgegebener Genauigkeitsschranke $\varepsilon > 0$.

1. Schritt: Taylorentwicklung für f an der Stelle $x = 0$:

$$f(x) = P_m(x) + R_{m+1}(x) = \sum_{i=0}^{m} a_i x^i + R_{m+1}(x), \quad a_i = \frac{f^{(i)}(0)}{i!},$$

wobei sich das kleinste m aus der Forderung $|R_{m+1}(x)| \leq \varepsilon_2 < \varepsilon$ für alle $x \in [-1,+1]$ ergibt (beispielsweise mit $\varepsilon_2 = \frac{\varepsilon}{2}$).

2. Schritt: T-Entwicklung für P_m unter Verwendung des Rechenschemas 8.19:

$$P_m(x) = \sum_{j=0}^{m} b_j T_j(x) \equiv S_m(x).$$

3. Schritt: Wahl des kleinstmöglichen $n \leq m-1$, so dass gilt

$$|f(x) - S_n(x)| \leq \varepsilon_2 + |b_{n+1}| + |b_{n+2}| + \ldots + |b_m| \leq \varepsilon_2 + \varepsilon_1 \leq \varepsilon$$

$\big($beispielsweise mit $\varepsilon_1 = \varepsilon_2 = \frac{\varepsilon}{2}\big)$. S_n ist das gesuchte Approximationspolynom für f mit der für das ganze Intervall $[-1,+1]$ gültigen Genauigkeit ε.

Zur Berechnung von Näherungswerten für die Funktion f mit Hilfe von S_n könnte S_n mit (8.28) bzw. (8.29) nach Potenzen von x umgeordnet werden; man erhält dann

$$S_n(x) = \sum_{j=0}^{n} b_j T_j(x) \equiv \sum_{k=0}^{n} \widetilde{a}_k x^k = \widetilde{P}_n(x)\,.$$

Günstiger ist die folgende hornerartige Auswertung von $S_n(x)$ an jeder beliebigen Stelle x unter Ausnutzung von (8.29):

1. Setze: $t_0 = 1,\ t_1 = x,\ s_1 = b_1 x + b_0$
2. Führe für jedes $k = 2(1)n$ durch:

$$t_k = 2\,x\,t_{k-1} - t_{k-2}\,; \quad s_k = b_k\,t_k + s_{k-1}\,.$$

Dann ist $S_n(x) = s_n$.

Ein Intervall $[a,b] \neq [-1,+1]$ wird durch eine lineare Transformation in das Intervall $[-1,+1]$ übergeführt. Durch $x = (2x' - b - a)/(b-a)$ geht das x'-Intervall $[a,b]$ in das x-Intervall $[-1,+1]$ über.

Beispiel 8.22.

Gegeben: Die Funktion $f : f(x) = \mathrm{e}^x$.

Gesucht: Ein Approximationspolynom $\Phi : \Phi(x) = S_n(x)$, das f mit einer Genauigkeit von $\varepsilon = 5 \cdot 10^{-6}$ im ganzen Intervall $[-1, +1]$ approximiert.

Lösung: Die Lösung erfolgt nach Algorithmus 8.21.

1. Schritt: Taylorentwicklung für f an der Stelle $x = 0$:
Es gilt

$$f(x) = \mathrm{e}^x = P_9(x) + R_{10}(x)$$

mit

$$P_9(x) = \sum_{i=0}^{9} \frac{x^i}{i!}$$

und dem Restglied

$$|R_{10}| \leq \frac{\mathrm{e}}{10!} < 7.5 \cdot 10^{-7} = \varepsilon_2 \quad \text{für alle } x \in [-1, 1]\,.$$

2. Schritt: T-Entwicklung von P_9:
Es gilt

$$P_9(x) = \sum_{j=0}^{9} b_j T_j(x)\,.$$

Nach dem Rechenschema 8.19 ergeben sich die folgenden Koeffizienten (auf 8 Mantissenstellen genau):

$$\begin{array}{lcll} b_0 & = & \frac{186689}{147456} & = & 1.2660658 \\ b_1 & = & \frac{833361}{737280} & = & 1.1303182 \\ b_2 & = & \frac{25021}{92160} & = & 2.7149523 \cdot 10^{-1} \\ b_3 & = & \frac{49033}{1105920} & = & 4.4336842 \cdot 10^{-2} \\ b_4 & = & \frac{1009}{184320} & = & 5.4741753 \cdot 10^{-3} \\ b_5 & = & \frac{1401}{2580480} & = & 5.4292225 \cdot 10^{-4} \\ b_6 & = & \frac{29}{645120} & = & 4.4952877 \cdot 10^{-5} \\ b_7 & = & \frac{33}{10321920} & = & 3.1970796 \cdot 10^{-6} \\ b_8 & = & \frac{1}{5160960} & = & 1.9376240 \cdot 10^{-7} \\ b_9 & = & \frac{1}{92897280} & = & 1.0764578 \cdot 10^{-8} \end{array}$$

3. Schritt: Es ist $|b_7| + |b_8| + |b_9| = 3.4016066 \cdot 10^{-6} = \varepsilon_1$, damit folgt

$$\begin{array}{rcl} |f(x) - S_6(x)| & = & |f(x) - \sum\limits_{j=0}^{6} b_j T_j(x)| \leq \varepsilon_1 + \varepsilon_2 \leq 3.41 \cdot 10^{-6} + 7.5 \cdot 10^{-7} \\ & = & 4.16 \cdot 10^{-6} \leq \varepsilon = 5 \cdot 10^{-6}. \end{array}$$

Es reicht also ein Approximationspolynom 6. Grades aus, um die gewünschte Genauigkeit zu erzielen. In gewöhnlicher Polynomdarstellung erhält man dann

$$S_6(x) \equiv \widetilde{P}_6(x) = \sum_{k=0}^{6} \widetilde{a}_k x^k$$

mit den auf 8 Mantissenstellen angegebenen Koeffizienten

$$\begin{array}{llll} \widetilde{a}_0 = 1, & \widetilde{a}_1 = 1.0000223, & \widetilde{a}_2 = 5.0000620 \cdot 10^{-1}, & \widetilde{a}_3 = 1.6648892 \cdot 10^{-1}, \\ & \widetilde{a}_4 = 4.1635665 \cdot 10^{-2}, & \widetilde{a}_5 = 8.6867560 \cdot 10^{-3}, & \widetilde{a}_6 = 1.4384921 \cdot 10^{-3}. \end{array}$$

In der Form $\widetilde{P}_6(x)$ kann man nun einfach mit Hilfe des Horner-Schemas die Näherungswerte für f berechnen.

Wegen $|R_9| \leq \frac{e}{9!} < 7.5 \cdot 10^{-6}$ für $x \in [-1, +1]$ hätte man bei der Approximation von f durch eine abgebrochene Taylorentwicklung ein Polynom 8. Grades benötigt, um die geforderte Genauigkeit zu erreichen. Sind zahlreiche Funktionswerte ein und derselben Funktion f zu berechnen, so lohnt sich also die Ermittlung des Approximationspolynoms nach der angegebenen Methode. □

8.2.5 Approximation periodischer Funktionen

Eine Funktion f heißt *L-periodisch*, wenn es eine kleinste Zahl $L > 0$ gibt, für die $f(x \pm L) = f(x)$ für alle x gilt. L heißt die *Periode* der Funktion. So ist $\sin(x)$ 2π-periodisch, $\tan(x)$ π-periodisch und $\cos(\alpha x)$ für $\alpha > 0$ $\frac{2\pi}{\alpha}$-periodisch. Periodische Funktionen haben die beachtliche Eigenschaft, dass man ihren Verlauf auf ganz $\mathbf{R}$ kennt, wenn sie nur ausschnittsweise auf einem beliebigen Intervall der Länge einer Periode, beispielsweise $[-\frac{L}{2}, \frac{L}{2})$ oder $[0, L)$, gegeben sind.

Soll eine L-periodische Funktion f approximiert werden, so liegt es nahe, diese Eigenschaft schon im Ansatz für die Approximationsfunktion zu berücksichtigen. Dazu bietet sich ein einfach zu handhabendes *trigonometrisches, L-periodisches Polynom n-ten Grades* an, das durch

$$\Phi(x) \equiv \Phi(x, \boldsymbol{a}, \boldsymbol{b}) = \frac{a_0}{2} + \sum_{k=1}^{n} \{a_k \cos(k\omega x) + b_k \sin(k\omega x)\} \tag{8.34}$$

gegeben ist, wobei man den Faktor $\omega = \frac{2\pi}{L}$ als Kreisfrequenz bezeichnet; im Fall einer 2π-periodischen Funktion wird ω zu 1. Das absolute Glied in (8.34) wird als $\frac{a_0}{2}$, also mit Faktor $\frac{1}{2}$ versehen angegeben, weil dann im Folgenden bei den Berechnungen dieses Koeffizienten und der a_k für $k \geq 1$ keine Unterscheidung mehr getroffen werden muss.

8.2.5.1 Kontinuierliche Approximation periodischer Funktionen im quadratischen Mittel

Ist f 2π-periodisch und wählt man als Approximationsfunktion ein 2π-periodisches trigonometrisches Polynom n-ten Grades

$$\Phi(x) \equiv \Phi(x, \boldsymbol{a}, \boldsymbol{b}) = \frac{a_0}{2} + \sum_{k=1}^{n} \{a_k \cos(kx) + b_k \sin(kx)\}\,, \tag{8.35}$$

so besteht die Approximationsaufgabe darin, die Koeffizienten $\boldsymbol{a} = (a_0, a_1, \ldots, a_n)^{\mathsf{T}}$ und $\boldsymbol{b} = (b_1, \ldots, b_n)^{\mathsf{T}}$ so zu wählen, dass der mittlere quadratische Fehler über ein Periodenintervall, etwa $[-\pi, \pi)$ oder $[0, 2\pi)$, minimal wird (vgl. 8.8):

$$\|f - \Phi^{(0)}\|_2^2 = \min_{\Phi} \|f - \Phi\|_2^2 = \min_{a_i, b_j \in \mathbf{R}} D^2(a_0, \ldots, a_n, b_1, \ldots, b_n)\,.$$

Das trigonometrische Polynom (8.35) stellt einen linearen Approximationsansatz mit den linear unabhängigen Funktionen

$$\begin{array}{lll} \varphi_0(x) = 1\,, & \varphi_1(x) = \cos(x)\,, & \varphi_2(x) = \sin(x)\,, \\ & \varphi_3(x) = \cos(2x)\,, & \varphi_4(x) = \sin(2x)\,, \ldots \end{array}$$

dar, und die für die Gaußschen Normalgleichungen (8.9) bzw. (8.10) benötigten Skalarprodukte sind dann mit $w(x) \equiv 1$ gegeben durch

$$(\varphi_j, \varphi_k) = \int_{-\pi}^{\pi} \varphi_j(x)\varphi_k(x)\,\mathrm{d}x = \int_{0}^{2\pi} \varphi_j(x)\varphi_k(x)\,\mathrm{d}x = \left\{ \begin{array}{ll} 0\,, & j \neq k \\ \pi\,, & j = k \neq 0 \\ 2\pi\,, & j = k = 0\,. \end{array} \right.$$

Es handelt sich also im Fall der Gleichgewichtung um ein orthogonales Funktionensystem. Dies folgt aus den bekannten Identitäten

$$\begin{aligned} \int_0^{2\pi} \cos(jx)\cos(kx)\,\mathrm{d}x &= \left\{ \begin{array}{ll} 0\,, & j \neq k \\ \pi\,, & j = k \neq 0 \\ 2\pi\,, & j = k = 0 \end{array} \right\}\,, && j, k = 0(1)n\,, \\ \int_0^{2\pi} \sin(jx)\sin(kx)\,\mathrm{d}x &= \left\{ \begin{array}{ll} 0\,, & j \neq k \\ \pi\,, & j = k \end{array} \right\}\,, && j, k = 1(1)n\,, \\ \int_0^{2\pi} \sin(jx)\cos(kx)\,\mathrm{d}x &= 0\,, && j = 1(1)n,\ k = 0(1)n\,. \end{aligned}$$

Daher lassen sich die Gaußschen Normalgleichungen sofort auflösen, und man erhält nach (8.14) als Koeffizienten $\boldsymbol{a}^{(0)} = (a_0^{(0)}, a_1^{(0)}, \ldots, a_n^{(0)})$ und $\boldsymbol{b}^{(0)} = (b_1^{(0)}, \ldots, b_n^{(0)})$ der bestausgleichenden Funktion $\Phi^{(0)}$

$$\left\{ \begin{array}{lllll} a_k^{(0)} & = & \frac{1}{\pi} \int_{-\pi}^{\pi} f(x)\cos(kx)\,\mathrm{d}x & = & \frac{1}{\pi} \int_{0}^{2\pi} f(x)\cos(kx)\,\mathrm{d}x, \quad k = 0(1)n\,, \\ b_k^{(0)} & = & \frac{1}{\pi} \int_{-\pi}^{\pi} f(x)\sin(kx)\,\mathrm{d}x & = & \frac{1}{\pi} \int_{0}^{2\pi} f(x)\sin(kx)\,\mathrm{d}x, \quad k = 1(1)n\,. \end{array} \right.$$

Auf den Fall einer allgemeinen L-periodischen Funktion übertragen heißt das:

Satz 8.23.
Eine L-periodische Funktion $f \in C[0,L]$ besitzt unter allen trigonometrischen, L-periodischen Polynomen n-ten Grades als beste Approximation im quadratischen Mittel zur Gewichtsfunktion $w(x) \equiv 1$ das trigonometrische Polynom

$$\Phi^{(0)}(x) = \frac{a_0^{(0)}}{2} + \sum_{k=1}^{n} \{a_k^{(0)} \cos(k\omega x) + b_k^{(0)} \sin(k\omega x)\}, \quad \omega = \frac{2\pi}{L},$$

mit den Koeffizienten

$$a_k^{(0)} = \frac{2}{L} \int_{-\frac{L}{2}}^{\frac{L}{2}} f(x) \cos(k\omega x)\,\mathrm{d}x = \frac{2}{L} \int_0^L f(x) \cos(k\omega x)\,\mathrm{d}x\,, \quad k = 0(1)n\,,$$

$$b_k^{(0)} = \frac{2}{L} \int_{-\frac{L}{2}}^{\frac{L}{2}} f(x) \sin(k\omega x)\,\mathrm{d}x = \frac{2}{L} \int_0^L f(x) \sin(k\omega x)\,\mathrm{d}x\,, \quad k = 1(1)n\,.$$

Da der Kosinus eine gerade und der Sinus eine ungerade Funktion ist und sich die Bestimmung der Koeffizienten als Integration über ein zum Nullpunkt symmetrisches Intervall darstellen lässt, gilt weiter:

$$\begin{array}{lll} a_k^{(0)} = 0 & \text{für alle } k, \text{ wenn } f \text{ ungerade ist} & \big(f(-x) = -f(x)\big)\,, \\ b_k^{(0)} = 0 & \text{für alle } k, \text{ wenn } f \quad \text{gerade ist} & \big(f(-x) = f(x)\big)\,. \end{array}$$

Beispiel 8.24.

Die Funktion $f(x) = |\sin x|$ ist L-periodisch mit $L = \pi$ und eine gerade Funktion. Daher hat man $\omega = \frac{2\pi}{L} = 2$, $b_k^{(0)} = 0$ für alle k und mit zweimaliger partieller Integration:

$$\begin{aligned} a_k^{(0)} &= \frac{2}{\pi} \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} |\sin x| \cos(2kx)\,\mathrm{d}x = \frac{4}{\pi} \int_0^{\frac{\pi}{2}} \sin x \cos(2kx)\,\mathrm{d}x \\ &= -\frac{4}{\pi} \cos x \cos(2kx)\Big|_0^{\frac{\pi}{2}} - \frac{8k}{\pi} \int_0^{\frac{\pi}{2}} \cos x \sin(2kx)\,\mathrm{d}x \\ &= \frac{4}{\pi} - \frac{8k}{\pi} \left[\sin x \sin(2kx)\Big|_0^{\frac{\pi}{2}} - 2k \int_0^{\frac{\pi}{2}} \sin x \cos(2kx)\,\mathrm{d}x \right] \\ &= \frac{4}{\pi} + 4k^2 \frac{4}{\pi} \int_0^{\frac{\pi}{2}} \sin x \cos(2kx)\,\mathrm{d}x \;=\; \frac{4}{\pi} + 4k^2\, a_k^{(0)} \end{aligned}$$

Das Integral auf der rechten Seite ist bis auf den Faktor $4k^2$ der gesuchte Koeffizient $a_k^{(0)}$. Löst man diese Gleichung nach $a_k^{(0)}$ auf, so erhält man

$$(1-4k^2)a_k^{(0)} = \frac{4}{\pi} \quad \text{bzw.} \quad a_k^{(0)} = -\frac{4}{\pi}\cdot\frac{1}{4k^2-1}$$

Damit ist das bestausgleichende trigonometrische Polynom n-ten Grades gegeben durch

$$\Phi^{(0)}(x) = \frac{2}{\pi} - \frac{4}{\pi}\sum_{k=1}^{n}\frac{1}{4k^2-1}\cos(2kx)\,,$$

also speziell das 1. Grades

$$\Phi_1^{(0)}(x) = \frac{2}{\pi} - \frac{4}{3\pi}\cos(2x)$$

und das 2. Grades

$$\Phi_2^{(0)}(x) = \frac{2}{\pi} - \frac{4}{3\pi}\cos(2x) - \frac{4}{15\pi}\cos(4x)\,.$$

□

8.2.5.2 Diskrete Approximation periodischer Funktionen im quadratischen Mittel

Die Anwendung der im letzten Abschnitt beschriebenen Approximation periodischer Funktionen durch trigonometrische Polynome kann aus zweierlei Sicht problematisch sein: Zum einen lassen sich die Integrale zur Berechnung der optimalen Koeffizienten $a_k^{(0)}$ und $b_k^{(0)}$ nach Satz 8.23 meist nicht exakt berechnen, und zum anderen ist die zugrunde liegende periodische Funktion womöglich nur an diskreten Stellen (etwa über eine Messreihe) bekannt. In diesen Fällen hilft die diskrete lineare Approximation im quadratischen Mittel weiter.

Unter der Voraussetzung, dass $N+1$ Werte einer L-periodischen Funktion an im Periodenintervall, etwa $[0, L)$, gleichverteilten Stellen

$$x_i = i\frac{L}{N+1}\,, \qquad i = 0(1)N\,,$$

bekannt sind und keine unterschiedliche Gewichtung vorgenommen werden soll (d. h. $w_i = 1$ für alle i), werden die Gaußschen Normalgleichungen (8.16) bzw. (8.17) besonders einfach, wenn wieder trigonometrische Polynome als Approximationsfunktionen dienen sollen.

Das zugrunde liegende Funktionensystem mit $\omega = \frac{2\pi}{L}$

$$\begin{aligned}\varphi_0(x) = 1\,, \quad & \varphi_1(x) = \cos(\omega x)\,, \quad && \varphi_2(x) = \sin(\omega x)\,,\\ & \varphi_3(x) = \cos(2\omega x)\,, && \varphi_4(x) = \sin(2\omega x)\,,\ldots,\end{aligned}$$

das anzahlmäßig höchstens $N+1$ Funktionen umfassen darf, bildet dann ein orthogonales System, weil für die in der Matrix der Gaußschen Normalgleichungen benötigten Skalarprodukte gemäß (8.17) gilt:

$$(\varphi_j, \varphi_k) = \sum_{i=0}^{N}\varphi_j(x_i)\,\varphi_k(x_i) = \begin{cases} 0\,, & j \neq k \\ \frac{N+1}{2}\,, & j = k \neq 0 \\ N+1\,, & j = k = 0 \end{cases}$$

für $j, k = 0(1)N+1$, wobei sich diese Formel für ungerades N und $j = k = N+1$ auf den Wert $N+1$ korrigiert. Dies führt zur Unterscheidung, ob die Anzahl der Wertepaare ungerade oder gerade ist.

Satz 8.25.

(a) Sind die Funktionswerte $f(x_i)$ einer L-periodischen Funktion f an im Periodenintervall äquidistant verteilten Stellen (ungerade Anzahl, $M = \frac{N}{2}$)

$$x_i = i\frac{L}{2M+1}\,, \quad i = 0, 1, \ldots, 2M$$

bekannt, so wird für jede Teilsumme Φ des trigonometrischen Polynoms

$$\Psi(x) = \frac{a_0}{2} + \sum_{k=1}^{M} \{a_k \cos(k\omega x) + b_k \sin(k\omega x)\}\,,\ \omega = \frac{2\pi}{L}\,,$$

die Fehlerquadratsumme

$$\|f - \Phi\|_{d,2}^2 = \sum_{i=0}^{2M} \big(f(x_i) - \Phi(x_i)\big)^2$$

genau dann minimal, wenn die benötigten Koeffizienten gemäß

$$\begin{aligned} a_k^{(0)} &= \tfrac{2}{2M+1} \sum_{i=0}^{2M} f(x_i)\cos(k\omega x_i)\,, \quad k \geq 0\,, \\ b_k^{(0)} &= \tfrac{2}{2M+1} \sum_{i=0}^{2M} f(x_i)\sin(k\omega x_i)\,, \quad k \geq 1\,, \end{aligned}$$

gewählt werden. Im Fall $\Phi^{(0)} = \Psi^{(0)}$, d. h. mit $2M+1$ Koeffizienten, wird die Fehlerquadratsumme Null, es liegt *trigonometrische Interpolation* vor.

(b) Sind die Funktionswerte $f(x_i)$ einer L-periodischen Funktion f an im Periodenintervall äquidistant verteilten Stellen (gerade Anzahl, $M = \frac{N+1}{2}$)

$$x_i = i\frac{L}{2M}\,, \quad i = 0, 1, \ldots, 2M{-}1$$

bekannt, so wird für jede Teilsumme Φ des trigonometrischen Polynoms

$$\Psi(x) = \frac{a_0}{2} + \sum_{k=1}^{M-1} \{a_k \cos(k\omega x) + b_k \sin(k\omega x)\} + \frac{a_M}{2}\cos(M\omega x)\,,\ \omega = \frac{2\pi}{L}\,,$$

die Fehlerquadratsumme

$$\|f - \Phi\|_{d,2}^2 = \sum_{i=0}^{2M-1} \big(f(x_i) - \Phi(x_i)\big)^2$$

genau dann minimal, wenn die benötigten Koeffizienten gemäß

$$\begin{aligned} a_k^{(0)} &= \tfrac{1}{M} \sum_{i=0}^{2M-1} f(x_i)\cos(k\omega x_i)\,, \quad k \geq 0\,, \\ b_k^{(0)} &= \tfrac{1}{M} \sum_{i=0}^{2M-1} f(x_i)\sin(k\omega x_i)\,, \quad k \geq 1\,, \end{aligned}$$

gewählt werden. Im Fall $\Phi^{(0)} = \Psi^{(0)}$, d. h. mit $2M$ Koeffizienten, wird die Fehlerquadratsumme Null, es liegt *trigonometrische Interpolation* vor.

s. [NIED1987].

Wie im kontinuierlichen Fall liegt auch hier wieder die vorteilhafte Situation vor, dass sich die beteiligten Koeffizienten in der Approximationsfunktion nicht ändern, wenn der Grad des trigonometrischen Ausgleichspolynoms modifiziert wird. Die im nächsten Abschnitt angesprochene FFT liefert nämlich auf sehr effiziente Weise alle Koeffizienten für die trigonometrische Interpolation $\Psi^{(0)}$ auf einmal, und man hat damit die Informationen für alle denkbaren trigonometrischen Ausgleichspolynome $\Phi^{(0)}$.

Ebenfalls in Analogie zum kontinuierlichen Sachverhalt wird die Eigenschaft einer geraden bzw. ungeraden Funktion auf die Approximationsfunktion übertragen, wenn man sie auf die im Periodenintervall gegebenen gleichverteilten Wertepaare

$$\big(x_i, f(x_i)\big) \quad \text{mit} \quad x_i = i\frac{L}{N+1}\,, \quad i = 0(1)N$$

bezieht: Es gilt

$$\begin{array}{lll} a_k^{(0)} = 0 & \text{für alle k, falls} & f(x_{N+1-i}) = -f(x_i) \quad \text{für alle } i\,, \\ b_k^{(0)} = 0 & \text{für alle k, falls} & f(x_{N+1-i}) = f(x_i) \quad \text{für alle } i\,. \end{array}$$

Beispiel 8.26. (vgl. Beispiel 8.24)

Von der π-periodischen Funktion $f(x) = |\sin x|$ seien nur $N+1 = 16$ Funktionswerte an im Periodenintervall gleichverteilten Stellen $x_i = i\frac{\pi}{16}$, $i = 0(1)15$, bekannt. Man erhält dann nach Satz 8.25(b) mit $M = 8$, $L = \pi$ und der Eigenschaft von f, eine gerade Funktion zu sein,

$$a_k^{(0)} = \frac{1}{8}\sum_{i=0}^{15} f(x_i)\cos(k\cdot 2\cdot x_i)\,, \quad b_k^{(0)} = 0\,.$$

Auf 6 Mantissenstellen angegeben lauten die berechenbaren Koeffizienten:

$$\begin{array}{lll} a_0^{(0)} = 1.26915\,, & a_1^{(0)} = -4.28538\cdot 10^{-1}\,, & a_2^{(0)} = -8.91056\cdot 10^{-2}\,, \\ a_3^{(0)} = -4.07728\cdot 10^{-2}\,, & a_4^{(0)} = -2.48640\cdot 10^{-2}\,, & a_5^{(0)} = -1.78855\cdot 10^{-2}\,, \\ a_6^{(0)} = -1.44478\cdot 10^{-2}\,, & a_7^{(0)} = -1.28035\cdot 10^{-2}\,, & a_8^{(0)} = -1.23114\cdot 10^{-2}\,. \end{array}$$

Die trigonometrischen Ausgleichspolynome 1. und 2. Grades sind damit gegeben durch

$$\begin{aligned} \Phi_1^{(0)}(x) &= \tfrac{a_0^{(0)}}{2} + a_1^{(0)}\cos(2x)\,, \\ \Phi_2^{(0)}(x) &= \tfrac{a_0^{(0)}}{2} + a_1^{(0)}\cos(2x) + a_2^{(0)}\cos(4x)\,, \end{aligned}$$

und ein Vergleich mit Beispiel 8.24 zeigt, dass sich diese trigonometrischen Ausgleichspolynome im kontinuierlichen und im diskreten Fall kaum unterscheiden – eine Tatsache, die sehr allgemein gilt (vgl. Abschnitt 8.2.6.2). Bezüglich des kontinuierlichen Fehlermaßes (8.8) (mit $w(x) = 1$) führt die gemäß Beispiel 8.24 ermittelte Approximationsfunktion auf einen kleineren Wert, im Hinblick auf das diskrete Fehlermaß (8.15) (mit $w_i = 1$) hingegen ist jede der hier ermittelten Approximationen der entsprechenden aus Beispiel

8.24 überlegen. Mit den berechneten Koeffizienten kann auch das trigonometrische Interpolationspolynom angegeben werden:

$$\Psi^{(0)}(x) = \frac{a_0^{(0)}}{2} + \sum_{k=1}^{7} a_k^{(0)} \cos(2kx) + \frac{a_8^{(0)}}{2} \cos(16x)\,.$$

□

8.2.5.3 Fourier-Transformation und FFT

Für die Approximation einer L-periodischen Funktion f kann auch ein anderer Zugang gewählt werden: Man zerlegt $f(x)$ in ihre zu den harmonischen Schwingungen $\cos(k\omega x)$ und $\sin(k\omega x)$, $\omega = \frac{2\pi}{L}$, gehörenden Bestandteile, also solche, die ganzzahlige Vielfache der Grundfrequenz $\frac{1}{L}$ sind. Berücksichtigt man alle diese Anteile, die sich übrigens nach Satz 8.23 berechnen lassen, so gilt unter wenig einschränkenden Bedingungen

$$f(x) = \frac{a_0^{(0)}}{2} + \sum_{k=1}^{\infty} \{a_k^{(0)} \cos(k\omega x) + b_k^{(0)} \sin(k\omega x)\} \tag{8.36}$$

für alle x; die rechte Seite wird *Fourierreihe von f* genannt, ihre Koeffizienten nach Satz 8.23 heißen dementsprechend auch *Fourierkoeffizienten von f* und deren Bestimmung (*endliche* bzw. *periodische*) *Fourier-Transformation* oder *harmonische Analyse*. Man kann zeigen, dass der mittlere quadratische Fehler, der dazu in Abschnitt 8.2.5.1 betrachtet wurde, gegen den kleinstmöglichen Wert Null strebt, wenn der Grad n der *Fourierteilsummen von f*

$$\frac{a_0^{(0)}}{2} + \sum_{k=1}^{n} \{a_k^{(0)} \cos(k\omega x) + b_k^{(0)} \sin(k\omega x)\} \tag{8.37}$$

gegen Unendlich strebt. Daraus folgt noch nicht unbedingt die Identität (8.36)! (8.36) gilt auf jeden Fall für alle differenzierbaren periodischen Funktionen f, aber sogar auch an jeder Unstetigkeitsstelle, wenn dort die Funktion f den arithmetischen Mittelwert annimmt und die links- und rechtsseitige Ableitung existieren.

Die Gleichung (8.36) sagt aus, dass zur vollständigen Wiedergabe der periodischen Funktion f auf ganz $\mathbb{R}$ nur abzählbar viele Koeffizienten $a_k^{(0)}$, $b_k^{(0)}$, die Fourierkoeffizienten von f, benötigt werden. Diese vortreffliche Eigenschaft zeichnet periodische Funktionen aus und geht bei einer Erweiterung der Fourier-Transformation auf nichtperiodische Prozesse verloren (s. [NIED1984]).

Viele Phänomene aus Natur und Technik lassen sich im Detail durch eine harmonische Analyse erfassen. So können etwa Luftdruckschwankungen durch Schallwellen – analog zum menschlichen Gehörvorgang – in reine Töne mit bestimmten Frequenzen $k\omega$ zerlegt werden; dabei wird über die Größen $a_k^{(0)}$ und $b_k^{(0)}$ jedem reinen Teilton des Geräusches eine anteilige Stärke oder Amplitude zugeordnet. Equalizer-Anzeigen von HiFi-Verstärkern veranschaulichen solche Intensitätsspektren: Tiefe Töne gehören zu niedrigen Frequenzen,

hohe Töne setzen sich vorwiegend aus höheren Frequenzen zusammen. Störendes Rauschen etwa als systembedingte Folge einer Übertragung (elektronisches oder thermisches Rauschen) zeichnet sich durch einen geringen, bei allen Frequenzen etwa gleichstarken Beitrag aus und lässt sich, wenn man die anteiligen Amplituden kennt, leicht herausfiltern. Bei einem solchen Filtervorgang werden diejenigen Summanden aus (8.37) ignoriert, die zu Amplituden $a_k^{(0)}$, $b_k^{(0)}$ gehören, deren Betrag unterhalb eines vorgegebenen Schwellenwertes liegt (siehe Abbildung 8.7).

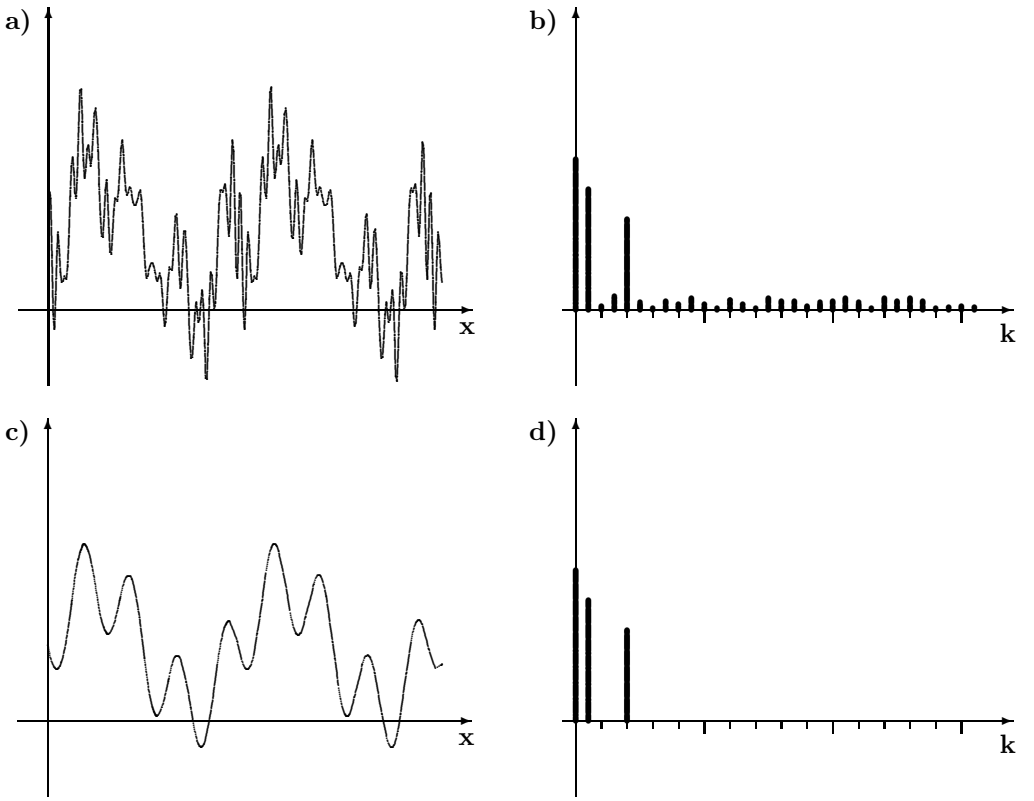


Abb. 8.7. Verrauschtes Signal (**a**) mit zugehörigem Amplitudenspektrum (**b**), das über $\left| a_0^{(0)} / 2 \right|$ und $\sqrt{(a_k^{(0)})^2 + (b_k^{(0)})^2}$ für $k \geq 1$ gegeben ist, sowie gefiltertes Spektrum (**d**), das alle Amplitudenanteile unterhalb eines Schwellenwertes unterdrückt, mit zugehörigem Signal (**c**)

Die nach Satz 8.25 berechneten Koeffizienten zur Approximation einer diskret gegebenen Funktion gehen aus den Fourierkoeffizienten von f nach Satz 8.23 hervor, wenn man auf deren Integrale die äquidistant zusammengesetzte Trapezregel anwendet. Sie heißen deshalb auch *diskrete Fourierkoeffizienten*, und das damit gebildete trigonometrische Polynom (8.36) heißt *diskrete Fourierteilsumme von f*.

Der komplette Satz der diskreten Fourierkoeffizienten $a_k^{(0)}$ und $b_k^{(0)}$ nach Satz 8.25 kann mit Hilfe der *Schnellen Fourier-Transformation (Fast Fourier Transform: FFT)* auf ein-

mal berechnet werden; sie nutzt gewisse Eigenschaften der komplexen Einheitswurzeln vorteilhaft aus und arbeitet in dem Fall, dass die Anzahl $N+1$ der Wertepaare mit einer Zweierpotenz $2^\tau (\tau \in \mathbb{N})$ übereinstimmt, am effektivsten. Die Anzahl der nötigen Operationen reduziert sich damit von der Größenordnung $2^\tau \cdot 2^\tau$ auf die Größenordnung $\tau \cdot 2^\tau$. Die Effizienz erhöht sich also mit wachsendem τ: Bei $2^6 = 64$ Wertepaaren wird der Aufwand gegenüber einer direkten Berechnung mit gut 4000 (komplexen) Multiplikationen und gleichvielen (komplexen) Additionen auf knapp 23,5 % gedrückt, wohingegen bei $2^{12} = 4096$ Wertepaaren die direkte Berechnung jeweils rund $2^{24} \approx 16,8$ Millionen derartige Multiplikationen und Additionen erfordert, die FFT aber insgesamt nur etwa 233.500 Operationen, also nur noch 0,7 % davon! Die FFT bedient sich dazu einer komplexen Schreibweise der diskreten Fourierkoeffizienten: Unter Ausnutzung der Eulerschen Formel

$$\mathrm{e}^{\mathrm{i}x} = \cos x + \mathrm{i} \sin x \,,$$

wobei i mit $\mathrm{i}^2 = -1$ hierbei die imaginäre Einheit darstellt, können die Koeffizienten $a_k^{(0)}$ und $b_k^{(0)}$ mit Hilfe der komplexen Größen (komplexe Fourierkoeffizienten)

$$\begin{aligned} c_k^{(0)} &= \tfrac{1}{N+1} \sum_{\ell=0}^{N} f(x_\ell) \mathrm{e}^{-\mathrm{i}\, k\omega x_\ell} \\ &= \tfrac{1}{N+1} \sum_{\ell=0}^{N} f(x_\ell) \Big(\mathrm{e}^{-\mathrm{i}\, 2\pi \frac{k}{N+1}} \Big)^{\ell}, \qquad \text{mit} \quad x_\ell = \ell \, \tfrac{L}{N+1} \end{aligned}$$

über deren Real- und Imaginärteile folgendermaßen angegeben werden:

$$\begin{aligned} a_k^{(0)} &= 2 \cdot \mathrm{Re}(c_k^{(0)}) \\ b_k^{(0)} &= -2 \cdot \mathrm{Im}(c_k^{(0)}) \end{aligned}$$

Durch fortwährend geschicktes Zerlegen und erneutes Zusammenfassen auftretender Faktoren bei der Berechnung aller benötigten komplexen Fourierkoeffizienten lässt sich der Rechenaufwand wie oben beschrieben erheblich reduzieren. Im Fall einer geraden Anzahl von Wertepaaren, also $N+1 = 2M$, werden dazu die $2M$ Summen für die komplexen Fourierkoeffizienten

$$c_k^{(0)} = \frac{1}{2M} \sum_{\ell=0}^{2M-1} f_\ell \cdot \Omega_{2M}^{k\ell} \qquad \text{mit} \qquad f_\ell = f(x_\ell) \,, \; \Omega_{2M} = \mathrm{e}^{-\mathrm{i}\frac{2\pi}{2M}} \,, \tag{8.38}$$

$k = 0(1)2M-1$, auf genausoviele Summen gleicher Struktur zurückgeführt, die jeweils nur noch halb so viele Summanden enthalten. Es gilt nämlich für die Fourierkoeffizienten mit **geraden** Inidzes ($k = 2j$)

$$c_{2j}^{(0)} = \frac{1}{2M} \sum_{\ell=0}^{2M-1} f_\ell \cdot \Omega_{2M}^{2j\ell} = \frac{1}{2M} \sum_{\ell=0}^{M-1} \left\{ f_\ell \cdot \Omega_{2M}^{2j\ell} + f_{M+\ell} \cdot \Omega_{2M}^{2j(M+\ell)} \right\} .$$

Wegen

$$\begin{aligned} \Omega_{2M}^{2j\ell} &= \mathrm{e}^{-\mathrm{i}\frac{2\pi}{2M} \cdot 2j\ell} = \mathrm{e}^{-\mathrm{i}\frac{2\pi}{M} j\ell} = \Omega_M^{j\ell}, \\ \Omega_{2M}^{2j(M+\ell)} &= \Omega_{2M}^{2jM} \cdot \Omega_{2M}^{2j\ell} = \mathrm{e}^{-\mathrm{i}\, 2\pi j} \cdot \Omega_M^{j\ell} = 1 \cdot \Omega_M^{j\ell} \end{aligned}$$

folgt daraus

$$c_{2j}^{(0)} = \frac{1}{2M} \sum_{\ell=0}^{M-1} \left\{ f_\ell \cdot \Omega_M^{j\ell} + f_{M+\ell} \cdot \Omega_M^{j\ell} \right\} ,$$

was einer Summe ursprünglichen Typs, jedoch nur halb so umfangreich, entspricht, wenn man

$$f'_\ell = f_\ell + f_{M+\ell} , \qquad \ell = 0(1)M{-}1 ,$$

abkürzt:

$$c_{2j}^{(0)} = \frac{1}{2M} \sum_{\ell=0}^{M-1} f'_\ell \cdot \Omega_M^{j\ell} , \qquad j = 0(1)M{-}1 . \tag{8.39}$$

Analog ergibt sich für die Fourierkoeffizienten mit **ungeraden** Indizes ($k = 2j+1$)

$$c_{2j+1}^{(0)} = \frac{1}{2M} \sum_{\ell=0}^{2M-1} f_\ell \cdot \Omega_{2M}^{(2j+1)\ell} = \frac{1}{2M} \sum_{\ell=0}^{M-1} \left\{ f_\ell \cdot \Omega_{2M}^{(2j+1)\ell} + f_{M+\ell} \cdot \Omega_{2M}^{(2j+1)(M+\ell)} \right\} .$$

Dies führt mit

$$\begin{aligned} \Omega_{2M}^{(2j+1)\ell} &= \mathrm{e}^{-\mathrm{i}\frac{2\pi}{2M}(2j+1)\ell} = \mathrm{e}^{-\mathrm{i}\frac{2\pi}{M}j\ell} \cdot \mathrm{e}^{-\mathrm{i}\frac{2\pi}{2M}\ell} = \Omega_M^{j\ell} \cdot \Omega_{2M}^{\ell}, \\ \Omega_{2M}^{(2j+1)(M+\ell)} &= \Omega_{2M}^{(2j+1)M} \cdot \Omega_{2M}^{(2j+1)\ell} = \mathrm{e}^{-\mathrm{i}\,\pi\,(2j+1)} \cdot \Omega_M^{j\ell} \cdot \Omega_{2M}^{\ell} = -\Omega_M^{j\ell}\, \Omega_{2M}^{\ell} \end{aligned}$$

auf

$$c_{2j+1}^{(0)} = \frac{1}{2M} \sum_{\ell=0}^{M-1} \left\{ f_\ell \cdot \Omega_{2M}^{\ell}\, \Omega_M^{j\ell} - f_{M+\ell} \cdot \Omega_{2M}^{\ell}\, \Omega_M^{j\ell} \right\}$$

und somit über die Abkürzung

$$f''_\ell = (f_\ell - f_{M+\ell})\, \Omega_{2M}^{\ell}$$

ebenfalls auf eine halb so umfangreiche Summation gleicher Struktur:

$$c_{2j+1}^{(0)} = \frac{1}{2M} \sum_{\ell=0}^{M-1} f''_\ell \cdot \Omega_M^{j\ell} , \qquad j = 0(1)M{-}1 . \tag{8.40}$$

(8.39) und (8.40) sind zusammen völlig äquivalent zur Ausgangsformel (8.38) und lassen sich bei geradem M in gleicher Weise weiter zerlegen. Ist die Anzahl der gegebenen Wertepaare eine Zweierpotenz, $N+1 = 2^\tau$ für ein $\tau \in \mathbb{N}$, so gelingt dieser jeweilige Reduktionsvorgang so lange, bis schließlich die beteiligten Summen nur noch aus einem Summanden bestehen.

Im Fall $N+1 = 4 = 2^2$ lauten die Berechnungsgleichungen für die komplexen Fourierkoeffizienten demnach

$$\begin{aligned} c_0^{(0)} &= \tfrac{1}{4}\{f_0 + f_1 + f_2 + f_3\} &&= \tfrac{1}{4}\{f'_0 + f'_1\} \\ c_2^{(0)} &= \tfrac{1}{4}\{f_0 + f_1\Omega_4^2 + f_2\Omega_4^4 + f_3\Omega_4^6\} &&= \tfrac{1}{4}\{f'_0 + f'_1\Omega_2\} \\ c_1^{(0)} &= \tfrac{1}{4}\{f_0 + f_1\Omega_4 + f_2\Omega_4^2 + f_3\Omega_4^3\} &&= \tfrac{1}{4}\{f''_0 + f''_1\} \\ c_3^{(0)} &= \tfrac{1}{4}\{f_0 + f_1\Omega_4^3 + f_2\Omega_4^6 + f_3\Omega_4^9\} &&= \tfrac{1}{4}\{f''_0 + f''_1\Omega_2\} \end{aligned}$$

wobei

$$\begin{array}{rclcrcl} f_0' &=& f_0 + f_2\,, & & f_1' &=& f_1 + f_3\,, \\ f_0'' &=& f_0 - f_2\,, & & f_1'' &=& (f_1 - f_3)\Omega_4 \end{array}$$

gesetzt wurde. In Matrixschreibweise lässt sich das folgendermaßen übersichtlich zusammenfassen:

$$\begin{aligned} \begin{pmatrix} c_0^{(0)} \\ c_2^{(0)} \\ c_1^{(0)} \\ c_3^{(0)} \end{pmatrix} &= \tfrac{1}{4} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & \Omega_4^2 & 1 & \Omega_4^2 \\ 1 & \Omega_4 & \Omega_4^2 & \Omega_4^3 \\ 1 & \Omega_4^3 & \Omega_4^2 & \Omega_4 \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \end{pmatrix} \\ &= \tfrac{1}{4} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & \Omega_4^2 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & \Omega_4^2 \end{pmatrix} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & \Omega_4^2 & 0 \\ 0 & \Omega_4 & 0 & \Omega_4^3 \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \end{pmatrix} \\ &= \tfrac{1}{4} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & \Omega_2 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & \Omega_2 \end{pmatrix} \begin{pmatrix} f_0' \\ f_1' \\ f_0'' \\ f_1'' \end{pmatrix} \end{aligned}$$

Hierbei wurden die Identitäten $\Omega_4^4 = 1$, $\Omega_4^6 = \Omega_4^2 = \Omega_2 = -1$, $\Omega_4^9 = \Omega_4^5 = \Omega_4$ und $\Omega_4^3 = -\Omega_4$ ausgenutzt. Aus speicherplatzsparenden Gründen wird auf Computern üblicherweise nur ein eindimensionales Feld der Länge $N+1$ eingesetzt, das in diesem Fall ($N+1=4$) folgende Zwischenschritte durchläuft:

$$\begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \end{pmatrix} \longrightarrow \begin{pmatrix} f_0' \\ f_1' \\ f_0'' \\ f_1'' \end{pmatrix} \longrightarrow \begin{pmatrix} c_0^{(0)} \\ c_2^{(0)} \\ c_1^{(0)} \\ c_3^{(0)} \end{pmatrix}$$

Das Feld enthält schließlich die komplexen Fourierkoeffizienten, allerdings nicht in durchnummerierter Weise. Der Rückschluss gelingt über die sogenannte Bit-Umkehrfunktion: Man spiegelt die Dualzahldarstellung der Indizes

$$\begin{array}{ll} 0 = (00)_2\,, & \text{gespiegelt: } (00)_2 = 0\,, \\ 1 = (01)_2\,, & \text{gespiegelt: } (10)_2 = 2\,, \\ 2 = (10)_2\,, & \text{gespiegelt: } (01)_2 = 1\,, \\ 3 = (11)_2\,, & \text{gespiegelt: } (11)_2 = 3\,. \end{array}$$

Im Fall $N+1 = 8 = 2^3$ würde ein weiterer Summations-Zerlegungsschritt gleichen Typs sowie eine abschließende Umordnung nach der Bit-Umkehrfunktion auf die 8 komplexen Fourierkoeffizienten in durchnummerierter Reihenfolge führen. Geht man von $N+1 = 2^\tau$ Daten aus, so sorgen insgesamt τ gleichartige, jeweils aufwandshalbierende Schritte und eine einmalige Bit-Umkehrung am Ende für die schnelle Berechnung der Fourierkoeffizienten nach der FFT.

Die FFT ist selbst nicht für die Handrechnung geeignet und steht vielseitig in größeren Programmbibliotheken zur Verfügung. Diese nutzen in der Regel auch vielfältige weitere Eigenschaften (etwa im Fall reeller Daten oder wenn die Anzahl der Daten keine Zweierpotenz ist sowie bei Faltungs- und Korrelationsprozessen) in effizienter Weise aus.

Beispiel 8.27.

Bei einem Nachrichtenübertragungssystem wird sich in der Regel das übertragene Signal von dem zur Übertragung eingegebenen Signal unterscheiden; man spricht dann von einer durch das Übertragungssystem erzeugten Verzerrung. Ist die Verzerrung nichtlinear, so kann man sie nur schwer erfassen.

Bei nichtlinearen Verzerrungen enthält das übertragene Signal Frequenzkomponenten, die im Eingangssignal nicht vorkommen. Als Maß für die nichtlineare Verzerrung kennzeichnet der Klirrfaktor die Übertragungsgüte eines elektronischen Nachrichtenverarbeitungssystems bei einer bestimmten Frequenz. Der Klirrfaktor gibt vom empfangenen Signal das Verhältnis des Effektivwertes des nicht im Eingangssignal enthaltenen Schwingungsgehalts zum Effektivwert des Gesamtsignals an und liegt damit immer zwischen 0 und 1. Ist das empfangene Signal unangenehm beeinträchtigt, so wird der Klirrfaktor einen relativ großen Wert haben.

Nimmt man als Eingangssignal eine Sinusschwingung der Form

$$S_v(t) = v \cdot \sin(\omega t)\,,$$

$\omega = \frac{2\pi}{L}$, in Abhängigkeit der Eingangsspannung v, so ist für diese Frequenz der Klirrfaktor gegeben durch

$$K = \frac{\sqrt{\sum\limits_{k\geq 2}\left\{(a_k^{(0)})^2 + (b_k^{(0)})^2\right\}}}{\sqrt{\sum\limits_{k\geq 1}\left\{(a_k^{(0)})^2 + (b_k^{(0)})^2\right\}}}\,,$$

wobei $a_k^{(0)}$ und $b_k^{(0)}$ die Fourierkoeffizienten des übertragenen Signals $\widetilde{S}_v$ sind. Der Klirrfaktor ist von der Eingangsspannung abhängig: $K = K(v)$. Die für das Übertragungssystem optimale Aussteuerung v für das eingegebene Signal S_v minimiert den Klirrfaktor $K(v)$.
Das verzerrte Signal $\widetilde{S}_v$ kann nur in den seltensten Fällen durch einen analytischen Ausdruck beschrieben werden. Deshalb kennt man auch nicht die Fourierkoeffizienten $a_k^{(0)}$ und $b_k^{(0)}$ dieses Signals. Um dennoch den Klirrfaktor näherungsweise bestimmen zu können, ersetzt man die Fourierkoeffizienten durch hinreichend viele diskrete Fourierkoeffizienten, die man berechnen kann, wenn man das übertragene Signal zu möglichst vielen gleichabständigen Zeitpunkten abtastet.

Bei dem hier betrachteten Nachrichtenübertragungssystem zeigt das übertragene Signal $\widetilde{S}_v$ bei der eingegebenen Grundschwingung

$$S_v(t) = v \cdot \sin(\omega t)$$

für verschiedene Werte von v ein Verhalten wie in Abbildung 8.8.

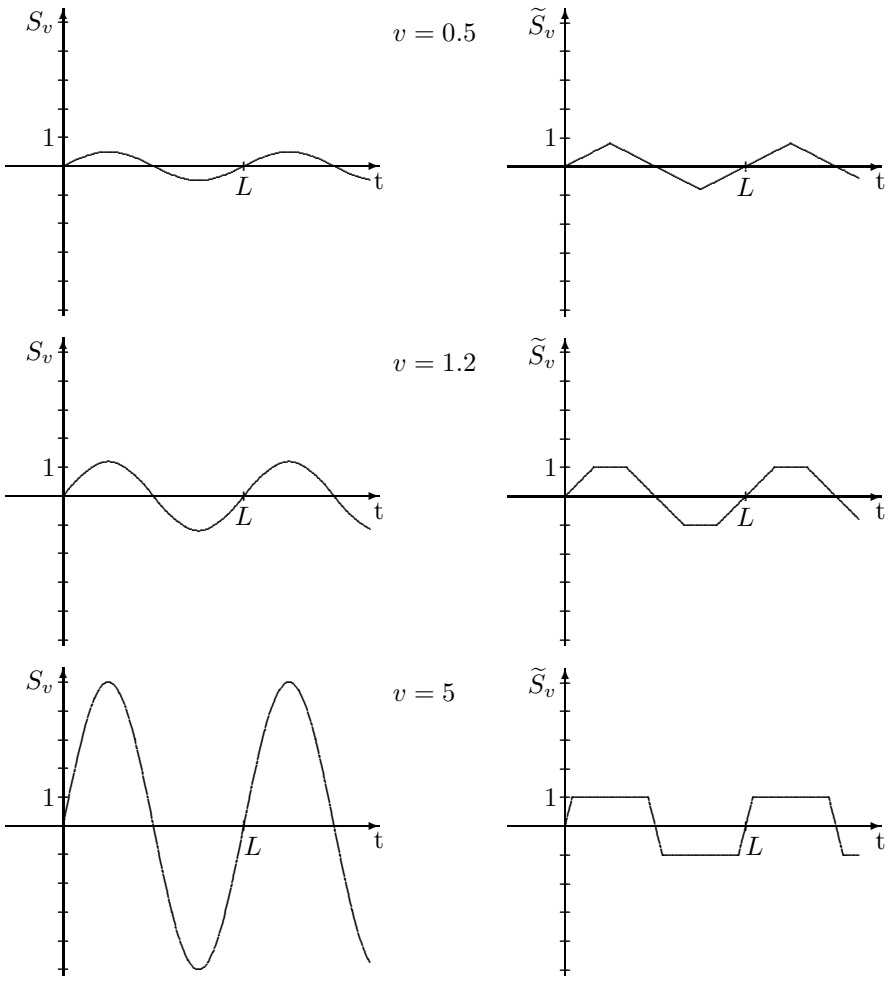


Abb. 8.8. Ausgangssignal S_v und übertragenes Signal $\widetilde{S}_v$ für verschiedene Werte von v. Für wachsendes v nimmt $\widetilde{S}_v$ immer mehr einen rechteckförmigen Verlauf an. Die Begrenzung der Spannung des übertragenen Signals, hier durch den Wert 1, ist für derartige Übertragungssysteme charakteristisch.

Das übertragene, stückweise lineare Signal $\widetilde{S}_v$ habe im Nullpunkt die Steigung v. In diesem Beispiel wurde das übertragene Signal $\widetilde{S}_v$ zu 64 im Periodenintervall $[0, L)$ äquidistant verteilten Zeitpunkten abgetastet und mit Hilfe der FFT die von v abhängigen diskreten reellen Fourierkoeffizienten $a_0^{(0)}, \ldots, a_{32}^{(0)}$ und $b_1^{(0)}, \ldots, b_{31}^{(0)}$ des Signals $\widetilde{S}_v$ gemäß Satz 8.25 (b) bestimmt. Wegen der Symmetrie des empfangenen Signals ist $a_k^{(0)} = 0$ für alle k; das verzerrte Signal enthält keine Kosinus-Schwingungsanteile.

Bestimmt man für Werte von v zwischen 0.1 und 10 in Abständen von 0.1 zum Eingangssignal

$$S_v(t) = v \cdot \sin(\omega t)$$

die von v abhängigen Fourierkoeffizienten $a_k^{(0)}$ und $b_k^{(0)}$ des empfangenen Signal $\widetilde{S}_v$ und damit den Klirrfaktor, der in diesem Beispiel durch

$$K(v) = \frac{\sqrt{\sum_{k=2}^{31} (b_k^{(0)})^2}}{\sqrt{\sum_{k=1}^{31} (b_k^{(0)})^2}}$$

gegeben ist, so erhält man den Verlauf für $K(v)$ aus Abbildung 8.9.

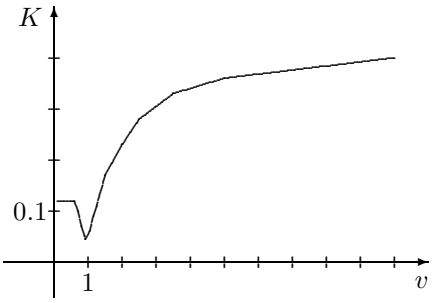


Abb. 8.9. Verhalten des Klirrfaktors in Abhängigkeit der Eingangsspannung v

Um diejenige Eingangsspannung v anzugeben, für die die Annäherung des empfangenen Signals $\widetilde{S}_v$ an das eingegebene Signal S_v am größten und damit der Anteil der durch das Übertragungssystem in das Signal $\widetilde{S}_v$ eingeführten „Oberwellen“ am geringsten ist, braucht man die Stelle, an der $K(v)$ sein absolutes Minimum annimmt. Über Interpolation gewinnt man den Wert $v = 0.92$ und damit für den Klirrfaktor die Größe 0.0445.

Abbildung 8.10 zeigt das eingegebene und das empfangene Signal für die so ermittelte optimale Aussteuerung $v = 0.92$.

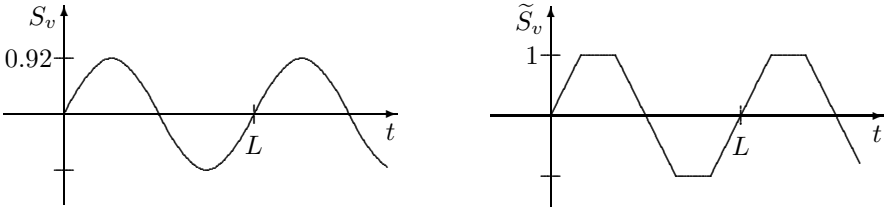


Abb. 8.10. Ausgangssignal S_v und übertragenes Signal $\widetilde{S}_v$ bei optimaler Aussteuerung ($v = 0.92$) □

8.2.6 Fehlerabschätzungen für lineare Approximationen

Eine sehr flexible Klasse von Funktionen bilden im allgemeinen Fall algebraische Polynome und im periodischen Fall trigonometrische Polynome. Solche Funktionen lassen sich

ohne viel Aufwand auswerten sowie formelmäßig leicht differenzieren und integrieren; sie sind deshalb sehr einfach zu handhaben. Die gleichmäßige Approximation beliebiger Funktionen f durch Polynome ist aus diesem Grunde inzwischen ausgiebig untersucht worden. Im Folgenden werden praktikable Ergebnisse dazu zusammengefasst.

8.2.6.1 Gleichmäßige Approximation durch algebraische Polynome

Legt man die Gewichtsfunktion $w(x) \equiv 1$ zugrunde und bildet $\overline{C}_n$ die Menge aller algebraischen Polynome

$$p_n(x) = c_0 + c_1 x + \ldots + c_n x^n = \sum_{k=0}^{n} c_k x^k \,, \quad c_k \in \mathbb{R} \,,$$

(höchstens) n-ten Grades, so besteht nach Abschnitt 8.2.4.1 die Bestimmung der besten gleichmäßigen Approximation für eine Funktion $f \in C[a,b]$ darin, Koeffizienten $c_0^*, \ldots, c_n^*$ eines Polynoms aus $\overline{C}_n$

$$p_n^*(x) = c_0^* + c_1^* x + \ldots + c_n^* x^n = \sum_{k=0}^{n} c_k^* x^k$$

zu finden, das die maximale absolute Abweichung minimiert $\big($vgl. (8.26)$\big)$:

$$\left\{ \begin{array}{lcl} \|f - p_n^*\|_\infty & = & \max\limits_{x\in[a,b]} |f(x) - p_n^*(x)| \\ & = & \min\limits_{p_n \in \overline{C}_n} \big(\max\limits_{x\in[a,b]} |f(x) - p_n(x)| \big) \end{array} \right. \tag{8.41}$$

Man kann zeigen, dass das in diesem Sinne optimale Polynom, das *Polynom der besten Approximation*, eindeutig bestimmt ist. Leider gibt es keine direkte Methode zur Berechnung der ausgezeichneten Koeffizienten $c_0^*, \ldots, c_n^*$, sondern nur ein aufwändiges Näherungsverfahren (Remez-Algorithmus, siehe z. B. [MUEL1995]). Daher liegt es nahe, fast bestmögliche Approximationspolynome zu betrachten, die konstruktiv angebbar sind. Man beurteilt ihre Güte durch Vergleich mit dem Optimum, das allgemein mit

$$E_n(f) = \|f - p_n^*\|_\infty = \min_{p_n \in \overline{C}_n} \big(\max_{x\in[a,b]} |f(x) - p_n(x)| \big) \tag{8.42}$$

bezeichnet wird und die folgenden Eigenschaften besitzt:

- **Approximationssatz von Weierstraß:**
  Jede Funktion $f \in C[a,b]$ lässt sich beliebig genau durch ein Polynom genügend hohen Grades gleichmäßig approximieren, d. h. es gilt
  $$\lim_{n\to\infty} E_n(f) = 0 \,.$$
- **Satz von Jackson:**
  Ist $f \in C^r[a,b]$ und $n > r$, so gilt
  $$E_n(f) \le \frac{K}{n^r} \|f^{(r)}\|_\infty$$
  mit einer nur von der Intervalllänge $b-a$ und r abhängigen Konstanten K.

Der Approximationssatz von Weierstraß sagt aus, dass es möglich ist, stetige Funktionen f durch Polynome beliebig genau anzunähern, und nach dem Satz von Jackson wird der kleinstmögliche Fehler $E_n(f)$ für wachsendes n schnell klein, wenn f entsprechend oft differenzierbar ist und die höheren Ableitungen von f betragsmäßig nicht zu groß sind.

Überraschenderweise können Polynome, die bestmöglich im quadratischen Mittel ausgleichen, auch hervorragende gleichmäßige Approximationseigenschaften haben. Für den kontinuierlichen Fall (Abschnitt 8.2.2) gilt nämlich:

Satz 8.28.
Für $f \in C[-1,1]$ und $w(x) = \frac{1}{\sqrt{1-x^2}}$ hat das den gewichteten mittleren quadratischen Fehler gemäß (8.8)

$$\|f - p_n^{(0)}\|_2^2 = \min_{p_n \in \overline{C}_n} \|f - p_n\|_2^2 = \min_{p_n \in \overline{C}_n} \int_{-1}^{1} w(x)\big(f(x) - p_n(x)\big)^2 \,\mathrm{d}x$$

minimierende Polynom n-ten Grades in der Darstellung mit den Tschebyscheff-Polynomen nach (8.27), (8.29)

$$p_n^{(0)}(x) = \sum_{k=0}^{n} c_k^{(0)} T_k(x)$$

die Koeffizienten

$$c_0^{(0)} = \frac{1}{\pi} \int_{-1}^{1} w(x) f(x) \,\mathrm{d}x\,, \quad c_k^{(0)} = \frac{2}{\pi} \int_{-1}^{1} w(x) T_k(x) f(x) \,\mathrm{d}x\,, \quad k = 1(1)n\,.$$

Dieses Polynom genügt der Abschätzung

$$\|f - p_n^{(0)}\|_\infty \le \big(\frac{4}{\pi^2} \ln n + 2.74\big) E_n(f)\,.$$

Der Faktor $\frac{4}{\pi^2} \ln n + 2.74$ wächst außerordentlich langsam mit n; für $n = 10$ hat er fast die Größe 3.7 und für $n = 100$ ungefähr 4.6. Mit anderen Worten: Verwendet man Polynome nicht über 100-ten Grades zur gleichmäßigen Approximation einer Funktion f auf dem Intervall $[-1,1]$, so hat das nach der Gaußschen Fehlerquadratmethode bestimmte Ausgleichspolynom $p_n^{(0)}$ auch fast optimale gleichmäßige Approximationseigenschaften; es ist nicht mal um den Faktor 5, also eine halbe Zehnerpotenz, schlechter als das theoretisch bestmögliche Ergebnis. Da die Tschebyscheff-Polynome auf $[-1,1]$ bezüglich der angegebenen Gewichtsfunktion ein orthogonales Funktionensystem bilden, sind außerdem die Koeffizienten $c_k^{(0)}$ unabhängig vom Grad n des Ausgleichspolynoms, d. h. bei Änderung des Grades bleiben die gemeinsamen Koeffizienten gleich. Für stetige Funktionen f auf einem beliebigen Intervall $[a,b]$ lautet das Polynom

$$p_n^{(0)}(x) = \sum_{k=0}^{n} c_k^{(0)} T_k\Big(\frac{2}{b-a}x - \frac{b+a}{b-a}\Big)$$

mit

$$c_0^{(0)} = \frac{1}{\pi}\int_{-1}^{1} w(x) f\Big(\frac{b-a}{2}x + \frac{b+a}{2}\Big)\,\mathrm{d}x\,, \quad c_k^{(0)} = \frac{2}{\pi}\int_{-1}^{1} w(x) T_k(x) f\Big(\frac{b-a}{2}x + \frac{b+a}{2}\Big)\,\mathrm{d}x\,.$$

Ein Nachteil des Ergebnisses von Satz 8.28 besteht darin, dass die Koeffizienten $c_k^{(0)}$ über Integrale definiert sind, die womöglich überhaupt nicht exakt angegeben werden können. Einen Ausweg bietet dann ein Zugang über die diskrete Approximation im quadratischen Mittel (Abschnitt 8.2.3), wenn der Polynomgrad n sich der Anzahl der gegebenen Wertepaare $\big(x_0, f(x_0)\big), \big(x_1, f(x_1)\big), \ldots, \big(x_N, f(x_N)\big)$ anpasst, d. h. der Grenzfall $n = N$ der Interpolation gewählt wird, und Einfluss auf die Wahl der Stellen $x_0, \ldots, x_N$ genommen werden kann:

Satz 8.29.
Ist $f \in [a,b]$ durch die Wertepaare $\big(x_0, f(x_0)\big), \big(x_1, f(x_1)\big), \ldots, \big(x_n, f(x_n)\big)$ mit $x_i = \frac{b-a}{2}\cos\big(i\frac{\pi}{n}\big) + \frac{a+b}{2}$, $i = 0(1)n$, gegeben, so erfüllt das nach der diskreten Gaußschen Fehlerquadratmethode bestimmte Ausgleichspolynom $p_n^{(0)}(x)$ n-ten Grades die Abschätzung

$$\|f - p_n^{(0)}\|_\infty \le \left(\frac{2}{\pi}\ln n + 2\right) E_n(f)\,.$$

Der Faktor $\frac{2}{\pi}\ln n + 2$ verhält sich ähnlich wie der im vorherigen Satz; für $n = 10$ liegt er unter 3.5, für $n = 100$ bei etwa 4.93 und selbst für $n = 1000$ noch knapp unter 6.4. Ist die zugrunde liegende Funktion entsprechend oft differenzierbar, so strebt nach dem Satz von Jackson $E_n(f)$ schnell gegen Null, und das nach Satz 8.29 bestimmte Polynom hat für größere n hervorragende, nahezu optimale gleichmäßige Approximationseigenschaften.

Da im Fall der Interpolation, d. h. $n = N$, das Ausgleichspolynom die Fehlerquadratsumme zu Null macht, hat eine besondere Wahl der Gewichte w_i im Fehlermaß (8.15) überhaupt keinen Einfluss; man kann sie ohne weiteres im Sinne von $w_i = 1$ für alle i vernachlässigen. Zweckmäßigerweise bestimmt man $p_n^{(0)}$ nach Abschnitt 8.2.3.2, also in der Darstellung

$$p_n^{(0)}(x) = \sum_{k=0}^{n} c_k^{(0)} Q_k(x)$$

mit zum gegebenen Sachverhalt orthogonalen Polynomen Q_k k-ten Grades.

Die in Satz 8.29 gewählten Stellen x_i stimmen mit den Extremalstellen des Tschebyscheff-Polynoms $T_n(x)$, die durch $\cos\Big(i\frac{\pi}{n}\Big)$, $i = 0(1)n$, gegeben sind (vgl. Abschnitt 8.2.4.2), überein, wenn man sie auf das Intervall $[a,b]$ transformiert. Statt dieser $n+1$ Extremalpunkte von T_n können auch entsprechend die $n+1$ Nullstellen $\cos\Big(\frac{2i+1}{n+1}\frac{\pi}{2}\Big)$, $i = 0(1)n$, des Polynoms T_{n+1} gewählt werden, für die ebenfalls eine Aussage wie im letzten Satz gilt.

8.2.6.2 Gleichmäßige Approximation durch trigonometrische Polynome

Betrachtet man im nichtperiodischen Fall die gleichmäßige Approximation einer Funktion $f \in C[a,b]$, so wird nichts über die Approximationsgüte für Punkte x außerhalb des Intervalls $[a,b]$ ausgesagt. Das ist beim periodischen Sachverhalt grundlegend anders: Wird die L-periodische Funktion $f \in C[0,L]$ im Intervall $[0,L]$ durch ein L-periodisches trigonometrisches Polynom vom Grad n

$$t_n(x) = \frac{a_0}{2} + \sum_{k=1}^{n} \{a_k \cos(k\omega x) + b_k \sin(k\omega x)\}\,, \quad \omega = \frac{2\pi}{L}\,, \tag{8.43}$$

mit einer Genauigkeit ε angenähert, so weicht $t_n(x)$ von $f(x)$ aufgrund derselben Periodizität beider Funktionen sogar für alle $x \in \mathbb{R}$ höchstens nur um ε ab. Somit besteht im Sinne einer gleichmäßigen Approximation analog zu (8.41) die Aufgabe darin, unter allen trigonometrischen Polynomen der Form (8.43) dasjenige $t_n^*(x)$ zu finden, das die maximale Abweichung zur Gewichtsfunktion $w(x) \equiv 1$ minimiert:

$$\begin{aligned} \|f - t_n^*\|_\infty &= \max_{x\in[0,L]} |f(x) - t_n^*(x)| = \max_{x\in\mathbf{R}} |f(x) - t_n^*(x)| \\ &= \min_{t_n} \|f - t_n\|_\infty \end{aligned}$$

Entsprechend (8.42) ist auch hier die Bezeichnung

$$E_n(f) = \|f - t_n^*\|_\infty = \min_{t_n} \big(\max_{x\in[0,L]} |f(x) - t_n(x)| \big)$$

gebräuchlich, und man hat sinngemäße Eigenschaften dafür:

- **Approximationssatz von Weierstraß:**
  Jede L-periodische Funktion $f \in C[0,L]$ lässt sich durch ein trigonometrisches Polynom genügend hohen Grades der Periode L beliebig genau gleichmäßig approximieren, d. h. es gilt
  $$\lim_{n\to\infty} E_n(f) = 0\,.$$
- **Satz von Jackson:**
  Ist $f \in C^r[0,L]$ L-periodisch und $n > r$, so gilt mit $\omega = \frac{2\pi}{L}$
  $$E_n(f) \le \frac{\pi}{2} \frac{1}{\omega^r (n+1)^r} \|f^{(r)}\|_\infty\,.$$

Die Interpretation dieser beiden zentralen Aussagen kann im Prinzip wortwörtlich vom algebraischen Fall des letzten Abschnitts übernommen werden; je glatter die zugrunde liegende Funktion ist, umso besser lässt sie sich durch trigonometrische Polynome gleichmäßig annähern.

Die in Abschnitt 8.2.5 behandelte Gaußsche Fehlerquadratmethode für periodische Funktionen bringt gleichzeitig überraschend gute gleichmäßige Approximationseigenschaften mit.

Satz 8.30.
Für L-periodische Funktionen $f \in C[0, L]$ und $w(x) \equiv 1$ hat das den mittleren quadratischen Fehler gemäß (8.8)

$$\|f - t_n^{(0)}\|_2^2 = \min_{t_n} \|f - t_n\|_2^2 = \min_{t_n \in \overline{C}} \int_0^L \big(f(x) - t_n(x)\big)^2 \, \mathrm{d}x$$

minimierende trigonometrische Polynom n-ten Grades der Periode L

$$t_n^{(0)}(x) = \frac{a_0^{(0)}}{2} + \sum_{k=1}^{n} \{a_k^{(0)} \cos(k\omega x) + b_k^{(0)} \sin(k\omega x)\} \ , \quad \omega = \frac{2\pi}{L} \ ,$$

nach Satz 8.23 die Koeffizienten (Fourierkoeffizienten von f)

$$\begin{aligned} a_k^{(0)} &= \tfrac{2}{L} \int_0^L f(x) \cos(k\omega x) \, \mathrm{d}x \ , \quad k = 0(1)n \ , \\ b_k^{(0)} &= \tfrac{2}{L} \int_0^L f(x) \sin(k\omega x) \, \mathrm{d}x \ , \quad k = 1(1)n \ . \end{aligned}$$

Dieses Polynom genügt der Abschätzung

$$\|f - t_n^{(0)}\|_\infty \le \left(\frac{4}{\pi^2} \ln n + 2.74 \right) E_n(f) \ .$$

Dieses Ergebnis gleicht völlig dem aus Satz 8.28 im algebraischen Fall; die den mittleren quadratischen Fehler minimierende Teilsumme $t_n^{(0)}$ der Fourierreihe von f hat gleichzeitig auch fast optimale gleichmäßige Approximationseigenschaften.

Die diskreten Fourierkoeffizienten nach Satz 8.25 gehen aus den Fourierkoeffizienten von f nach Satz 8.23 durch Anwendung der äquidistant zusammengesetzten Trapezregel auf deren Integrale hervor. Nach der Euler-MacLaurinschen Summenformel (vgl. Abschnitt 14.5) erweist sich diese Quadraturformel als die optimale beim vorliegenden periodischen Sachverhalt, so dass sich die diskreten Fourierkoeffizienten von den Fourierkoeffizienten von f um so wenig unterscheiden, wie es die Funktion f nur zulässt: Die Abweichungen liegen absolut genommen unter $2E_n(f)$. Damit stellt auch die diskrete Fourierteilsumme $\Psi^{(0)}$, die mit dem Grenzfall der trigonometrischen Interpolation nach Satz 8.25 übereinstimmt, ebenfalls eine hervorragende gleichmäßige Approximationsfunktion dar; sie erfüllt eine Abschätzung der Form

$$\|f - \Psi^{(0)}\|_\infty \le K \cdot \ln n \cdot E_{n-1}(f) \ ,$$

mit einer von n und f unabhängigen Konstante K, was qualitativ der Aussage aus Satz 8.30 entspricht. Bei der Bestimmung von $\Psi^{(0)}$ können keine Schwierigkeiten mehr auftreten – im Gegenteil: Ihre Berechnung kann mit Hilfe der FFT effizient erfolgen.

8.3 Diskrete nichtlineare Approximation

Ist die Approximationsfunktion Φ nicht von der Gestalt (8.2)

$$\Phi(x, c_0, c_1, \ldots, c_n) = c_0\varphi_0(x) + \ldots + c_n\varphi_n(x)$$

mit gegebenen, linear unabhängigen Funktionen φ_k, sondern hängen die φ_k ihrerseits wieder von freien Parametern ab $\big(\text{z. B. } \Phi(x, \boldsymbol{c}) = c_0 + c_1 \mathrm{e}^{c_2(x-c_3)^4} + c_4 \ln(c_5 x)\big)$, so führt die Minimierung

$$D^2(c_0, c_1, \ldots, c_n) = \sum_{i=0}^{N} w_i \big(f(x_i) - \Phi(x_i, \boldsymbol{c})\big)^2 \stackrel{!}{=} \quad \text{Min.} \tag{8.44}$$

im Allgemeinen auf ein nichtlineares Gleichungssystem für die optimalen Koeffizienten $c_k^{(0)}$:

$$\frac{\partial D^2}{\partial c_j}(c_0^{(0)}, \ldots, c_n^{(0)}) = 0 \quad \text{für} \quad j = 0(1)n. \tag{8.45}$$

In einigen Spezialfällen lässt sich die nichtlinare Modellfunktion Φ durch eine geeignete Transformation in ein lineares Modell der Gestalt (8.2) überführen (s. Abschnitt 8.3.1), in den anderen Fällen kann das nichtlineare System z. B. mit dem gedämpften Newton-Verfahren bzw. mit einer Kombination aus der Householder-Transformation und dem gedämpften Newton-Verfahren gelöst werden.

8.3.1 Transformationsmethode beim nichtlinearen Ausgleich

Liegt ein nichtlineares Modell $\Phi(x, \boldsymbol{c})$ z. B. der Gestalt

$$\Phi_1(x, c_0, c_1) = 1/(c_0 + c_1 \ln x)$$

oder

$$\Phi_2(x, \boldsymbol{c}) = \mathrm{e}^{c_0 + c_1 x + c_2 x^2 + c_3 x^3}$$

vor, dann würde die Minimierung (8.44) auf ein nichtlineares System für die optimalen Koeffizienten $c_k^{(0)}$ führen. Transformiert man jedoch das Modell Φ_1 mit $T(\Phi_1) = 1/\Phi_1 = \widetilde{\Phi}_1$, so erhält man das lineare Modell

$$\widetilde{\Phi}_1(x, c_0, c_1) = c_0 + c_1 \ln x\,;$$

transformiert man Φ_2 mit $T(\Phi_2) = \ln \Phi_2 = \widetilde{\Phi}_2$, so führt das auf das lineare Modell

$$\widetilde{\Phi}_2(x, \boldsymbol{c}) = c_0 + c_1 x + c_2 x^2 + c_3 x^3\,.$$

Statt der Fehlerquadratsumme

$$D^2(c_0, c_1, \ldots, c_n) = \sum_{i=0}^{N} w_i \big(f(x_i) - \Phi(x_i, \boldsymbol{c})\big)^2$$

mit der nichtlinearen Modellfunktion Φ minimiert man jetzt die transformierte Fehlerquadratsumme

$$\begin{aligned}\widetilde{D}^2(c_0,c_1,\ldots,c_n) &= \sum_{i=0}^{N} \widetilde{w}_i\Big(T\big(f(x_i)\big)-T\big(\Phi(x_i)\big)\Big)^2\\ &=: \sum_{i=0}^{N} \widetilde{w}_i\big(\widetilde{f}(x_i)-\widetilde{\Phi}(x_i,\boldsymbol{c})\big)^2\end{aligned}$$

mit der linearen Modellfunktion $\widetilde{\Phi}$, so dass sich die Koeffizienten $c_k^{(0)}$ der besten Approximation $\widetilde{\Phi}^{(0)}$ aus einem linearen Gleichungssystem ergeben. Um dabei die Forderung

$$D^2(c_0^{(0)},c_1^{(0)},\ldots,c_n^{(0)}) \stackrel{!}{=} \widetilde{D}^2(c_0^{(0)},c_1^{(0)},\ldots,c_n^{(0)})$$

zumindest näherungsweise zu erfüllen, muss man mit Gewichten $\widetilde{w}_i$ arbeiten, die sich aus der folgenden Formel ergeben (vgl. [SPAT1974]):

$$\widetilde{w}_i = \frac{w_i}{T'^2(\Phi)|_{\Phi=f(x_i)}}\,. \tag{8.46}$$

Algorithmus 8.31. (*Transformationsmethode*)

Gegeben: Wertepaare $\big(x_i,f(x_i)\big)$ und Gewichte $w_i>0$,
$i=0(1)N$, und eine nichtlineare Modellfunktion Φ.

Gesucht: Beste Approximation $\Phi^{(0)}$ mit
$D^2(c_0,c_1,\ldots,c_n)=\sum\limits_{i=0}^{N} w_i\big(f(x_i)-\Phi(x_i,\boldsymbol{c})\big)^2 \stackrel{!}{=}$ Min.

1. Schritt: Wahl einer Transformation T so, dass gilt
$T(\Phi)=:\widetilde{\Phi}$ mit $\widetilde{\Phi}(x)=\sum\limits_{k=0}^{N} c_k\widetilde{\varphi}_k(x)$ (lineares Modell).

2. Schritt: Berechnung der transformierten Gewichte $\widetilde{w}_i$ aus der Formel (8.46).

3. Schritt: Berechnung der Koeffizienten $c_k^{(0)}$ von $\widetilde{\Phi}^{(0)}$ aus den Normalgleichungen

$$\begin{pmatrix} (\widetilde{\varphi}_0,\widetilde{\varphi}_0) & (\widetilde{\varphi}_0,\widetilde{\varphi}_1) & \cdots & (\widetilde{\varphi}_0,\widetilde{\varphi}_n)\\ (\widetilde{\varphi}_1,\widetilde{\varphi}_0) & (\widetilde{\varphi}_1,\widetilde{\varphi}_1) & \cdots & (\widetilde{\varphi}_1,\widetilde{\varphi}_n)\\ \vdots & \vdots & & \vdots\\ (\widetilde{\varphi}_n,\widetilde{\varphi}_0) & (\widetilde{\varphi}_n,\widetilde{\varphi}_1) & \cdots & (\widetilde{\varphi}_n,\widetilde{\varphi}_n)\end{pmatrix}\begin{pmatrix} c_0^{(0)}\\ c_1^{(0)}\\ \vdots\\ c_n^{(0)}\end{pmatrix}=\begin{pmatrix} (\widetilde{f},\widetilde{\varphi}_0)\\ (\widetilde{f},\widetilde{\varphi}_1)\\ \vdots\\ (\widetilde{f},\widetilde{\varphi}_n)\end{pmatrix}$$

mit $(\widetilde{\varphi}_j,\widetilde{\varphi}_k):=\sum\limits_{i=0}^{N}\widetilde{w}_i\widetilde{\varphi}_j(x_i)\widetilde{\varphi}_k(x_i)$

$(\widetilde{f},\widetilde{\varphi}_j):=\sum\limits_{i=0}^{N}\widetilde{w}_i\widetilde{f}_i\widetilde{\varphi}_j(x_i)$ mit $\widetilde{f}_i=T\big(f(x_i)\big)$.

4. Schritt: Einsetzen der $c_k^{(0)}$ in $\Phi^{(0)}$.

Gegebenenfalls sollte die Lösung der Normalgleichungen im 3. Schritt mit Hilfe der Householder-Transformation (s. Abschnitt 8.2.3.4) erfolgen, um eine (das Ergebnis verfälschende) Anhäufung von Rundungsfehlern bei der direkten Lösung der Normalgleichungen zu vermeiden.

Weitere Spezialfälle zur Erzeugung eines linearen Ausgleichs sind in [SPAT1974] zu finden.

Beispiel 8.32.

Gegeben: Die folgende Wertetabelle

i	0	1	2
x_i	1	e	e^2
$f_i = f(x_i)$	1	2	3

und außerdem die Modellfunktion $\Phi(x; a, b) = \sqrt{a \ln(bx)}$.

Gesucht: Mit Hilfe der diskreten Gaußschen Fehlerquadratmethode die optimalen Werte für a und b in der Modellfunktion Φ.

Lösung: Die Vorgehensweise erfolgt nach Algorithmus 8.31.

1. $a \ln(bx) = a(\ln b + \ln x) = a \ln b + a \ln x = c_0 + c_1 \ln x$
   $$\Rightarrow \Phi(x; c_0, c_1) = \sqrt{c_0 + c_1 \ln x}$$
2. $T(\Phi) = \Phi^2 = \widetilde{\Phi} = c_0 + c_1 \ln x \quad$ mit $\quad \varphi_0(x) = 1,\ \varphi_1(x) = \ln x$
3. Normalgleichungen für $\widetilde{\Phi}$ aufstellen: Die Minimierung des (8.15) entsprechenden Ausdrucks
   $$\widetilde{D}^2 = \sum_{i=0}^{2} \widetilde{w}_i \left(\widetilde{f}_i - \widetilde{\Phi}(x_i) \right)$$
   führt im Fall der Gleichgewichtung ($w_i = 1$ für alle i) mit
   $$\widetilde{w}_i = \frac{w_i}{\left[\frac{\mathrm{d}}{\mathrm{d}\Phi} T(\Phi) \right]^2 \Big|_{\Phi = f_i}} = \frac{1}{4\Phi^2|_{\Phi=f_i}} = \frac{1}{4f_i^2}$$
   und $\varphi_0(x) = 1$, $\varphi_1(x) = \ln x$ gemäß (8.16) auf das lineare Gleichungssystem:
   $$\sum_{i=0}^{2} \frac{1}{4f_i^2} c_0^{(0)} + \sum_{i=0}^{2} \frac{\ln x_i}{4f_i^2} c_1^{(0)} = \sum_{i=0}^{2} \frac{f_i^2}{4f_i^2}$$
   $$\sum_{i=0}^{2} \frac{\ln x_i}{4f_i^2} c_0^{(0)} + \sum_{i=0}^{2} \frac{\ln^2 x_i}{4f_i^2} c_1^{(0)} = \sum_{i=0}^{2} \frac{f_i^2 \ln x_i}{4f_i^2}\,.$$
   Nach jeweiliger Multiplikation mit 4 lautet es in Matrixform, wenn die gegebenen Werte eingesetzt werden:

$$\begin{pmatrix} \frac{49}{36} & \frac{17}{36} \\ \frac{17}{36} & \frac{25}{36} \end{pmatrix} \begin{pmatrix} c_0^{(0)} \\ c_1^{(0)} \end{pmatrix} = \begin{pmatrix} 3 \\ 3 \end{pmatrix}.$$

Als Lösungen ergeben sich daraus (auf 4 Dezimalstellen genau):

$$c_0^{(0)} = \frac{108}{117} = 0.9231\,, \quad c_1^{(0)} = \frac{432}{117} = 3.6923$$

Damit erhält man als optimale Ausgleichsfunktion des nichtlinearen Modells über

$$a^{(0)} = c_1^{(0)}\,, \quad b^{(0)} = \mathrm{e}^{c_0^{(0)}/c_1^{(0)}}$$

die Funktion

$$\begin{aligned} \Phi^{(0)}(x) &= \sqrt{\tfrac{432}{117}\ln(\mathrm{e}^{1/4}x)} \\ &= \sqrt{\tfrac{432}{117}\left(\tfrac{1}{4}+\ln x\right)}\,. \end{aligned}$$

□

Beispiel 8.33.

Gegeben: Messwerte $(\Delta p_i, Q_i) =: (x_i, f_i)$ mit Δp_i = Druckdifferenz in bar und Q_i = Volumenstrom (in l/min) für eine Drosselventil-Kennlinie in der Hydraulik. Modellfunktion:

$$Q(\Delta p) \;=\; A\,\sqrt[B]{\Delta p} \;\Rightarrow\; \Phi(x;A,B) \;=\; A\,\sqrt[B]{x}$$

(Die Modellfunktion ist nichtlinear, da sie sich nicht in der Form $\Phi = c_1\varphi_1(x) + c_2\varphi_2(x)$ darstellen lässt.)

Wertetabelle mit den Messwerten: [2]

Q l/min	0	0.91	1.31	1.80	2.30	2.83	3.32	3.79	4.23	4.45
Δp bar	0	4.0	8.0	14.0	22.5	34.0	46.5	59.5	74.0	83.0

Gesucht: Optimale Werte für A, B zu dieser Wertetabelle.

Lösung: Es wird eine Transformation T gesucht, die den vorgegebenen Ansatz auf ein lineares Modell überführt. Dies gelingt über

$$T(\Phi) \;=\; \ln\Phi \;=\; \ln A + \frac{1}{B}\ln x \;=\; c_0 + c_1 \ln x$$

mit $A = \mathrm{e}^{c_0}$ und $B = 1/c_1$, wenn man das erste Wertepaar (0,0) außer Acht lässt. Das ist ohne Einschränkung möglich, da der Modellfunktionsansatz diesen Punkt schon gewährleistet. Die transformierte Approximationsfunktion lautet dann

$$\widetilde{\Phi}(x;c_0,c_1) \;=\; c_0 + c_1 \ln x\,.$$

[2] Die Messungen wurden im Hydraulik-Labor der Fachhochschule Aachen durchgeführt.

Statt

$$D^2(A,B) = \sum_{i=1}^{9} w_i \left(f_i - \Phi(x_i; A, B)\right)^2$$

wird nun der Ausdruck

$$\widetilde{D}^2(c_0, c_1) = \sum_{i=1}^{9} \widetilde{w}_i \left(T(f_i) - \underbrace{T(\Phi)}_{\widetilde{\Phi}}\right)^2 = \sum_{i=1}^{9} \widetilde{w}_i \left(\ln f_i - c_0 - c_1 \ln x_i\right)^2$$

minimiert. Die Berechnung der Gewichte $\widetilde{w}_i = \dfrac{w_i}{T'^2(\Phi)|_{\Phi=f_i}}$ erfolgt über

$$T(\Phi) = \ln \Phi \quad \Rightarrow \quad T'(\Phi) = \frac{1}{\Phi} \quad \Rightarrow \quad T'^2(\Phi) = \frac{1}{\Phi^2} \quad \Rightarrow \quad T'^2(\Phi)|_{\Phi=f_i} = \frac{1}{f_i^2}.$$

Damit erhält man

$$\widetilde{w}_i = \frac{w_i}{1/f_i^2} = f_i^2 \, w_i,$$

und mit $\varphi_0(x) = 1$, $\varphi_1(x) = \ln x$ lauten die zugehörigen Normalgleichungen gemäß (8.16) bzw. (8.18) in Matrixform

$$\begin{pmatrix} \sum\limits_{i=1}^{9} \widetilde{w}_i & \sum\limits_{i=1}^{9} \widetilde{w}_i \ln x_i \\ \sum\limits_{i=1}^{9} \widetilde{w}_i \ln x_i & \sum\limits_{i=1}^{9} \widetilde{w}_i (\ln x_i)^2 \end{pmatrix} \begin{pmatrix} c_0^{(0)} \\ c_1^{(0)} \end{pmatrix} = \begin{pmatrix} \sum\limits_{i=1}^{9} \widetilde{w}_i \ln f_i \\ \sum\limits_{i=1}^{9} \widetilde{w}_i \ln f_i \ln x_i \end{pmatrix}.$$

Im Folgenden sollen 2 Gewichtungen unterschieden werden:

(I) $w_i = \frac{1}{f_i^2}$ (d. h. es werden die aufsummierten relativen Fehlerquadrate minimiert).

Damit ergibt sich $\widetilde{w}_i = 1$ für alle i, und die Normalgleichungen vereinfachen sich zu

$$\begin{pmatrix} 9 & \sum \ln x_i \\ \sum \ln x_i & \sum (\ln x_i)^2 \end{pmatrix} \begin{pmatrix} c_0^{(0)} \\ c_1^{(0)} \end{pmatrix} = \begin{pmatrix} \sum \ln f_i \\ \sum \ln f_i \cdot \ln x_i \end{pmatrix}$$

$$\Rightarrow \quad \begin{cases} c_0^{(0)} = -0.812\,105\,862 \;\Rightarrow\; A^{(0)} = \mathrm{e}^{c_0^{(0)}} = 0.4439\,222\,422 \\ c_1^{(0)} = 0.524\,379\,166 \;\Rightarrow\; B^{(0)} = \frac{1}{c_1^{(0)}} = 1.907\,017\,030 \end{cases}$$

$$\Rightarrow \qquad \Phi\left(x; A^{(0)}, B^{(0)}\right) = 0.4439222422 \cdot \sqrt[1.907017030]{x}$$

$$\text{bzw.} \quad Q\left(\Delta p; A^{(0)}, B^{(0)}\right) = 0.44392 \cdot (\Delta p)^{0.52438}$$

(II) $w_i = 1$ (d. h. es werden alle Messungenauigkeiten gleichgewichtet). Daraus folgt $\widetilde{w}_i = f_i^2$ und somit für die Normalgleichungen

$$\begin{pmatrix} \sum f_i^2 & \sum f_i^2 \ln x_i \\ \sum f_i^2 \ln x_i & \sum f_i^2 (\ln x_i)^2 \end{pmatrix} \begin{pmatrix} c_0^{(0)} \\ c_1^{(0)} \end{pmatrix} = \begin{pmatrix} \sum f_i^2 \ln f_i \\ \sum f_i^2 \ln f_i \ln x_i \end{pmatrix}$$

$$\Rightarrow \begin{cases} c_0^{(0)} = -0.785\,062\,253 \Rightarrow A^{(0)} = \mathrm{e}^{c_0^{(0)}} = 0.456\,091\,308 \\ c_1^{(0)} = 0.517\,036\,160 \Rightarrow B^{(0)} = \frac{1}{c_1^{(0)}} = 1.934\,100\,700 \end{cases}$$

$$\Rightarrow \qquad \Phi\,(x; A^{(0)}, B^{(0)}) = 0.456091308 \cdot \sqrt[1.934100702]{x}$$

$$\text{bzw.} \quad Q\,(\Delta p; A^{(0)}, B^{(0)}) = 0.45609 \cdot (\Delta p)^{0.51704}$$

Die resultierende Approximationsfunktion unterscheidet sich, wie Abbildung 8.11 zeigt, zunächst nicht wesentlich von der zuvor ermittelten Approximation gleichen Typs, die die Summe der relativen Fehlerquadrate minimiert. Größere Differenzen werden allerdings dann sichtbar, wenn man größere Werte für Δp einsetzt (Extrapolation!).

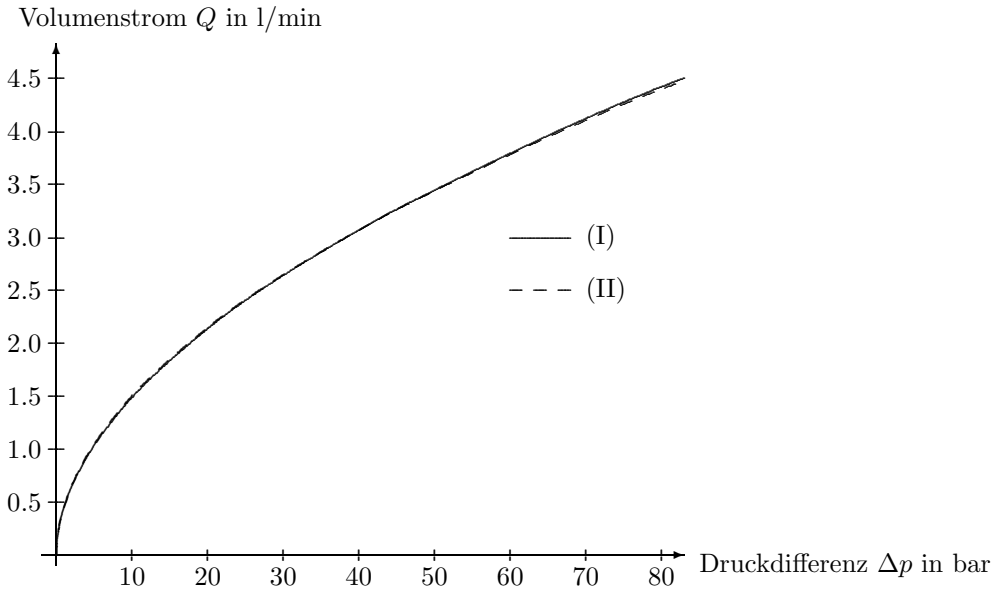


Abb. 8.11. Ausgleichskurven der Form $A \cdot \sqrt[B]{\Delta p}$ zu zwei unterschiedlichen Gewichtungen □

8.3.2 Nichtlinearer Ausgleich im quadratischen Mittel

Lässt sich eine geeignete Transformation der in den Koeffizienten c_k nichtlinearen Approximationsfunktion

$$\Phi(x, \boldsymbol{c}) = \Phi(x, c_0, c_1, \ldots, c_n)$$

gemäß Abschnitt 8.3.1 nicht finden, so führt bei gegebenen Wertepaaren $\big(x_i, f(x_i)\big)$, $i = 0(1)N$, $N \geq n$, die Minimierung (8.44) auf ein nichtlineares Gleichungssystem (8.45) für die optimalen Koeffizienten der nichtlinearen Approximationsfunktion Φ.

Dieses nichtlineare Gleichungssystem kann z. B. mit dem gedämpften Newton-Verfahren gelöst werden, wobei die dabei entstehenden linearen Systeme entweder mit dem Gaußschen Algorithmus oder mit der Householder-Transformation (vgl. Abschnitt 4.13) behandelt werden können. Der zweite Weg ist vorzuziehen, da durch die Householder-Transformation die Lösungsbestimmung durch eine mögliche schlechte Kondition der Matrix des linearen Systems nicht weiter beeinträchtigt wird.

Vom Anwender müssen $n+1$ Startwerte für die Koeffizienten $c_k^{(0)}$, $k = 0(1)n$, vorgegeben werden sowie $N+1$ Gewichte w_i zu den Wertepaaren $\big(x_i, f(x_i)\big)$, $i = 0(1)N$, sofern nicht alle $w_i = 1$ gesetzt werden sollen. Die partiellen Ableitungen, die für den Aufbau der Funktionalmatrix im Newton-Verfahren benötigt werden, können gegebenenfalls auch über einen Differenzenquotienten angenähert werden.

8.4 Entscheidungshilfen

Bei der diskreten linearen und nichtlinearen Approximation im quadratischen Mittel kommt es in erster Linie auf die Qualität der Modellfunktion an. Kann ein gutes Modell nicht mit ausreichender Sicherheit angegeben werden, so sollte man mit Ausgleichssplines arbeiten (s. Kapitel 10). In den Fällen, in denen sich algebraische Polynome als Modellfunktion eignen, sollte der Ausgleich unter Verwendung orthogonaler Polynome (vgl. Abschnitt 8.2.3.2) durchgeführt werden.

Im Falle der nichtlinearen Approximation kann die Lösung eines nichtlinearen Gleichungssystems umgangen werden, wenn sich die Modellfunktion durch eine Transformation in ein lineares Modell überführen lässt. In allen anderen Fällen muss nichtlinear gerechnet werden, möglichst unter Verwendung der Householder-Transformation (Abschnitt 8.3.2).

Im Fall periodischer Funktionen bieten sich trigonometrische Polynome als Approximationsfunktionen an (Abschnitt 8.2.5). Liegen diskrete Wertepaare an äquidistant verteilten Stützstellen vor, so gelingt diese Approximation sehr effizient mit Hilfe der Schnellen Fourier-Transformation (FFT, s. Abschnitt 8.2.5.3).

Um hervorragende gleichmäßige Approximationseigenschaften durch algebraische Polynome zu erzielen, ist die Wahl der Gewichtsfunktion bzw. der Stützstellen im diskreten Fall bedeutsam (Abschnitt 8.2.6.1). Analoge Aussagen für periodische Funktionen bleiben mit der FFT verbunden (Abschnitt 8.2.6.2).

Ergänzende Literatur zu Kapitel 8

[BOOR2001]; [BRIG1997]; [BUTZ2003]; [COLL1973], 3.3, 3.4; [CONT1987], 6.; [DEUF2002] Bd.1, Kap.3; [HAMM1978], Kap.1; [HAMM1994], 4.; [KRAB1975], III; [LOUI1998]; [MAES1988], 6.; [MUWI1999]; [NITS1968], II; [OPPE1992]; [PREU2001], Kap.6; [PLAT2000], Kap.3; [QUAR2002] Bd.2, Kap.10; [SCHW1997], 4.3, 7; [STOE2002]; [TORN1990] Bd. 2, 11.4-11.8.

Kapitel 9

Polynomiale Interpolation sowie Shepard-Interpolation

9.1 Aufgabenstellung

Gegeben sind $n+1$ *Wertepaare* (x_i, y_i) mit $x_i, y_i \in \mathbb{R}$, $i = 0(1)n$, in Form einer Wertetabelle:

i	0	1	2	...	n
x_i	x_0	x_1	x_2	...	x_n
y_i	y_0	y_1	y_2	...	y_n

Die *Stützstellen* x_i seien paarweise verschieden, aber nicht notwendig äquidistant und auch nicht notwendig in der Anordnung $x_0 < x_1 < x_2 \ldots < x_n$. Die Wertepaare (x_i, y_i) heißen *Interpolationsstellen*.

Gesucht ist ein algebraisches Polynom Φ möglichst niedrigen Grades, das an den Stützstellen x_i die zugehörigen *Stützwerte* y_i annimmt. Es gilt der

Satz 9.1. (*Existenz- und Eindeutigkeitssatz*)
Zu $n+1$ Interpolationsstellen (x_i, y_i) mit den paarweise verschiedenen Stützstellen $x_i, i = 0(1)n$, gibt es genau ein Polynom Φ:

$$\Phi(x) = \sum_{k=0}^{n} c_k x^k, \qquad c_k \in \mathbb{R}, \tag{9.1}$$

mit der Eigenschaft

$$\Phi(x_i) = \sum_{k=0}^{n} c_k x_i^k = y_i, \qquad i = 0(1)n. \tag{9.2}$$

Φ heißt das *Interpolationspolynom* zu dem gegebenen System von Interpolationsstellen; $\Phi(x_i) = y_i$ für $i = 0(1)n$ sind die *Interpolationsbedingungen*.

Beweis. (9.2) ist ein lineares inhomogenes Gleichungssystem für die c_k. Es lautet

$$\boldsymbol{A}\boldsymbol{c} = \boldsymbol{y}$$

mit

$$\boldsymbol{A} = \begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & & x_n^n \end{pmatrix}, \quad \boldsymbol{c} = \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{pmatrix}, \quad \boldsymbol{y} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}.$$

Die Determinante der Matrix $\boldsymbol{A}$ ist eine sogenannte *Vandermondesche Determinante*, die sich in der Form

$$\det \boldsymbol{A} = \prod_{i>j} (x_i - x_j)$$

darstellen lässt (siehe [ZURM1997], S.150). Da nach Voraussetzung die x_i paarweise verschieden sind, ist $\det \boldsymbol{A} \neq 0$. Also sind die c_k und damit auch das Polynom (9.1) eindeutig bestimmt. Falls alle $y_i = 0$ sind, hat das homogene System (9.2) wegen $\det \boldsymbol{A} \neq 0$ nur die triviale Lösung $c_k = 0$ für alle k, d. h. $\Phi(x) \equiv 0$. Damit ist der Satz bewiesen. □

Sind von einer (beispielsweise empirischen) Funktion $f \in C[a,b]$ an den $n+1$ Stützstellen $x_i \in [a,b]$ die Stützwerte $f(x_i)$ bekannt, und ist $\Phi \in C[a,b]$ das Interpolationspolynom zu den Interpolationsstellen $\big(x_i, y_i = f(x_i)\big)$, d. h. es gilt $\Phi(x_i) = f(x_i) = y_i$, so trifft man die Annahme, dass Φ die Funktion f in $[a,b]$ annähert. Die Ermittlung von Werten $\Phi(\overline{x})$ zu Argumenten $\overline{x} \in [a,b]$, $\overline{x} \neq x_i$, nennt man *Interpolation*; liegt $\overline{x}$ außerhalb $[a,b]$, so spricht man von *Extrapolation*.

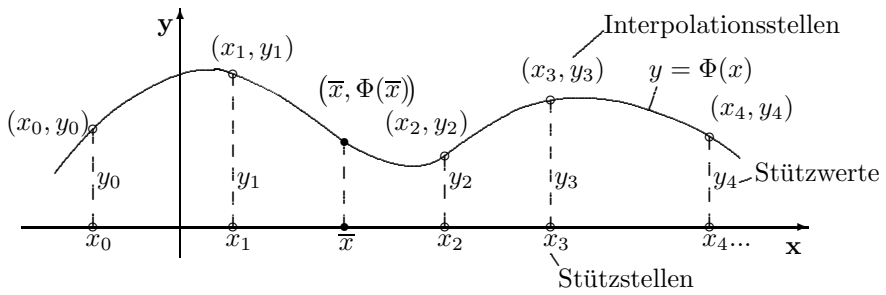


Abb. 9.1. Interpolation

Im Folgenden werden verschiedene Darstellungsformen (*Interpolationsformeln*) für das eindeutig bestimmte Interpolationspolynom zu $n+1$ Interpolationsstellen angegeben.

Bemerkung. (*Hermite-Interpolation*) Ist zu jedem $x_i, i = 0(1)n$, $x_i \in [a,b]$, statt des einen Stützwertes y_i ein (m_i+1)-Tupel von Zahlen $(y_i, y_i', \ldots, y_i^{(m_i)})$ gegeben, dann heißt das Interpolationspolynom H mit den k-ten Ableitungen

$$H^{(k)}(x_i) = y_i^{(k)} \qquad \text{für} \quad k = 0(1)m_i, \quad i = 0(1)n,$$

Hermitesches Interpolationspolynom (s. [WERN1993], S.7-16).

9.2 Interpolationsformeln von Lagrange

9.2.1 Lagrangesche Formel für beliebige Stützstellen

Φ wird mit von y_k unabhängigen L_k in der Form angesetzt

$$\Phi(x) \equiv L(x) = \sum_{k=0}^{n} L_k(x) y_k. \tag{9.3}$$

An den Stützstellen x_i muss wegen der Interpolationsbedingungen $\Phi(x_i) = y_i$, $i = 0(1)n$, gelten, also

$$L(x_i) = \sum_{k=0}^{n} L_k(x_i)\, y_k = y_i\,, \quad i = 0(1)n.$$

Daran erkennt man (durch Koeffizientenvergleich bei den y_k) die Beziehungen

$$L_k(x_i) = \begin{cases} 1 & \text{für} \quad k = i, \\ 0 & \text{für} \quad k \neq i. \end{cases}$$

Allgemein wird dies erfüllt mit den folgenden L_k

$$\begin{cases} L_k(x) = \dfrac{(x-x_0)(x-x_1)\ldots(x-x_{k-1})(x-x_{k+1})\ldots(x-x_n)}{(x_k-x_0)(x_k-x_1)\ldots(x_k-x_{k-1})(x_k-x_{k+1})\ldots(x_k-x_n)} \\ \qquad\quad = \displaystyle\prod_{\substack{i=0 \\ i\neq k}}^{n} \frac{x-x_i}{x_k-x_i}. \end{cases} \tag{9.4}$$

Die L_k sind Polynome vom Grad n, so dass $\Phi \equiv L$ ein Polynom vom Höchstgrad n ist. (9.3) ist die *Interpolationsformel von Lagrange für beliebige Stützstellen.*

Die Auswertung von $L_k(x)y_k$ erfordert n Multiplikationen. Insgesamt sind für $L(x)$ also $(n+1)n$ Multiplikationen auszuführen. Bei der Newton'schen Interpolationsformel (Abschnitt 9.5) genügen n Multiplikationen.

Algorithmus 9.2. (*Interpolationsformel von Lagrange*)

Gegeben: (x_i, y_i), $i = 0(1)n$, $x_i \neq x_k$ für $i \neq k$

Gesucht: Interpolationsformel von Lagrange

1. Schritt: Ermittlung der L_k, $k = 0(1)n$, nach Formel (9.4).

2. Schritt: Aufstellen der Interpolationsformel L gemäß Formel (9.3).

Bemerkung. Bei Hinzunahme einer Interpolationsstelle müssen alle $L_k(x)$ neu berechnet werden. Deshalb hat die Methode eher theoretische als praktische Bedeutung, s. Abschnitt 9.8.

Beispiel 9.3.

Gegeben: Die folgende Wertetabelle der Funktion $f : f(x) = 1/(1+x^2)$:

i	0	1	2
x_i	0	0.5	1
$f(x_i) = y_i$	1	0.8	0.5

Gesucht: Die zugehörige Interpolationsformel von Lagrange $L(x)$ und der Fehler $|f(0.8) - L(0.8)|$.

Lösung:

1. Schritt: Ermittlung der $L_k(x)$ nach Algorithmus 9.2 und Formel (9.4):

$$L_0(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} = \frac{(x-0.5)(x-1)}{0.5} = 2\,(x-0.5)(x-1),$$

$$L_1(x) = \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} = \frac{x(x-1)}{-(0.5)^2} = -4\,x\,(x-1),$$

$$L_2(x) = \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} = \frac{x(x-0.5)}{0.5} = 2\,x\,(x-0.5).$$

2. Schritt: Aufstellen der Interpolationsformel

$$L(x) = \sum_{k=0}^{2} L_k(x) y_k = 2(x-0.5)(x-1) - 3.2x(x-1) + x(x-0.5)\ .$$

Lösung: Berechnung von L(0.8) und von $|f(0.8) - L(0.8)|$.

$L(0.8) = -0.12 + 0.512 + 0.24 = 0.632$.

Da $f(x)$ im vorliegenden Fall bekannt ist, kann man den absoluten Fehler $|f(x) - L(x)|$ bestimmen. Für $x = 0.8$ ist
$|f(0.8) - L(0.8)| \leq |0.610 - 0.632| = 0.220 \cdot 10^{-1}$;

dieser Wert liegt zwischen den entsprechenden Werten, die sich bei der Approximation derselben Funktion nach der kontinuierlichen und der diskreten Fehlerquadratmethode ergaben (Beispiele 8.8 und 8.15). □

Lineare Interpolation

Für die Interpolationsstellen (x_0, y_0), (x_1, y_1) wird die Interpolationsformel von Lagrange mit dem Höchstgrad $n = 1$ bestimmt. Mit (9.4) wird

$$L_0(x) = \frac{x-x_1}{x_0-x_1}, \qquad L_1(x) = \frac{x-x_0}{x_1-x_0},$$

so dass die Interpolationsformel lautet

$$L(x) = \sum_{k=0}^{1} L_k(x) y_k = \frac{x - x_1}{x_0 - x_1} y_0 + \frac{x - x_0}{x_1 - x_0} y_1 = \frac{\begin{vmatrix} y_0 & x_0 - x \\ y_1 & x_1 - x \end{vmatrix}}{x_1 - x_0}. \tag{9.5}$$

9.2.2 Lagrangesche Formel für äquidistante Stützstellen

Die Stützstellen x_i seien äquidistant mit der festen *Schrittweite* $h = x_{i+1} - x_i$, $i = 0(1)n{-}1$. Dann ist $x_i = x_0 + hi$, $i = 0(1)n$, und es wird gesetzt

$$x = x_0 + ht, \qquad t \in [0, n].$$

Damit erhält man für (9.4)

$$L_k(x) = \prod_{\substack{i=0 \\ i \neq k}}^{n} \frac{t - i}{k - i} =: \widetilde{L}_k(t) = \frac{t(t-1)\dots(t-k+1)(t-k-1)\dots(t-n)}{k!(-1)^{n-k}(n-k)!}. \tag{9.6}$$

Die *Interpolationsformel von Lagrange für äquidistante Stützstellen* lautet somit

$$\widetilde{L}(t) = \sum_{k=0}^{n} \widetilde{L}_k(t) y_k = \left(\prod_{i=0}^{n} (t - i) \right) \left(\sum_{k=0}^{n} \frac{(-1)^{n-k} y_k}{k!(n-k)!(t-k)} \right).$$

Beispiel 9.4. (Fortsetzung von Beispiel 9.3)

Gegeben: Die folgende Wertetabelle

i	0	1	2
x_i	0	0.5	1
y_i	1	0.8	0.5

Gesucht: Der Wert $L(0.8)$ nach der Lagrange-Interpolationsformel $\widetilde{L}(t)$ für äquidistante Stützstellen.

Lösung: Mit $h = 0.5$, $t \in [0, 2]$, ergeben sich die $\widetilde{L}_k$ nach (9.6) wie folgt:

$$\widetilde{L}_0(t) = \frac{t-1}{0-1}\,\frac{t-2}{0-2} = \tfrac{1}{2}\,(t-1)\,(t-2),$$

$$\widetilde{L}_1(t) = \frac{t-0}{1-0}\,\frac{t-2}{1-2} = -t\,(t-2),$$

$$\widetilde{L}_2(t) = \frac{t-0}{2-0}\,\frac{t-1}{2-1} = \tfrac{1}{2}\,t\,(t-1).$$

Für das Lagrangesche Polynom erhält man

$$\widetilde{L}(t) = \sum_{k=0}^{2} \widetilde{L}_k(t)\, y_k = 0.5\,(t-1)\,(t-2) - 0.8\,t\,(t-2) + 0.25\,t\,(t-1)\,.$$

Wegen $x = x_0 + ht$ gilt $t = \dfrac{x - x_0}{h} = \dfrac{x-0}{0.5} = 2x$,
so dass man für $x = 0.8$ und mit $h = 0.5$ den Wert $t = 1.6$ erhält.

Eingesetzt in $\widetilde{L}(t)$ folgt $L(0.8) = \widetilde{L}(1.6) = 0.632$, wie es auch nach Beispiel 9.3 zu erwarten war. □

9.3 Aitken-Interpolationsschema für beliebige Stützstellen

Wenn zu $n+1$ gegebenen Interpolationsstellen (x_i, y_i) mit nicht notwendig äquidistanten Stützstellen x_i nicht das Interpolationspolynom Φ selbst, sondern nur sein Wert $\Phi(\overline{x})$ an einer Stelle $\overline{x}$ benötigt wird, so benutzt man zu dessen Berechnung zweckmäßig das *Interpolationsschema von Aitken.*

Den Wert $\Phi(\overline{x})$ des Interpolationspolynoms findet man durch fortgesetzte Anwendung der linearen Interpolation (9.5). Das zu (x_0, y_0) und (x_1, y_1) gehörige lineare Interpolationspolynom wird mit P_{01} bezeichnet. Es ist

$$P_{01}(x) = \frac{1}{x_1 - x_0} \begin{vmatrix} y_0 & x_0 - x \\ y_1 & x_1 - x \end{vmatrix}.$$

Sind x_0, x_i zwei verschiedene Stützstellen, so gilt für das zugehörige lineare Interpolationspolynom P_{0i} :

$$P_{0i}(x) = \frac{1}{x_i - x_0} \begin{vmatrix} y_0 & x_0 - x \\ y_i & x_i - x \end{vmatrix} = P_{i0}(x), \quad i = 1(1)n,\ i \text{ fest}, \tag{9.7}$$

und es sind $P_{0i}(x_0) = y_0, P_{0i}(x_i) = y_i$, d. h. P_{0i} löst die Interpolationsaufgabe für die beiden Wertepaare (x_0, y_0), (x_i, y_i).

Unter Verwendung zweier linearer Polynome P_{01} und P_{0i} für $i \geq 2$ werden Polynome P_{01i} vom Höchstgrad zwei erzeugt mit

$$P_{01i}(x) = \frac{1}{x_i - x_1} \begin{vmatrix} P_{01}(x) & x_1 - x \\ P_{0i}(x) & x_i - x \end{vmatrix}, \quad i = 2(1)n,\ i \text{ fest}. \tag{9.8}$$

P_{01i} ist das Interpolationspolynom, das die Interpolationsaufgabe für die drei Interpolationsstellen $(x_0, y_0), (x_1, y_1), (x_i, y_i)$ löst. Die fortgesetzte Anwendung der linearen Interpolation führt auf Interpolationspolynome schrittweise wachsenden Grades. Das Interpolationspolynom vom Höchstgrad n zu $n+1$ Interpolationsstellen erhält man durch lineare

Interpolation, angewandt auf zwei verschiedene Interpolationspolynome vom Höchstgrad $n-1$, von denen jedes für n der gegebenen $n+1$ Stützstellen aufgestellt ist. Allgemein berechnet man bei bekannten Funktionswerten der Polynome $P_{012\ldots(k-1)i}$ vom Grad $k-1$ die Funktionswerte der Polynome $P_{012\ldots ki}$ vom Grad k nach der Formel

$$P_{012\ldots(k-1)ki}(x) = \frac{1}{x_i - x_k} \begin{vmatrix} P_{012\ldots(k-1)k}(x) & x_k - x \\ P_{012\ldots(k-1)i}(x) & x_i - x \end{vmatrix}, \quad \begin{matrix} k=0(1)n-1, \\ i=(k+1)(1)n. \end{matrix} \tag{9.9}$$

Dabei lösen die Polynome $P_{012\ldots ki}$ vom Grad k die Interpolationsaufgabe zu den Interpolationsstellen (x_0, y_0), $(x_1, y_1), \ldots, (x_k, y_k), (x_i, y_i)$.

Rechenschema 9.5. (*Interpolationsschema von Aitken*)

i	x_i	y_i	$P_{0i}(\overline{x})$	$P_{01i}(\overline{x})$	$P_{012i}(\overline{x})$	$\cdots$	$P_{0123\ldots n}(\overline{x})$	$x_i - \overline{x}$
0	x_0	y_0						$x_0 - \overline{x}$
1	x_1	y_1	P_{01}					$x_1 - \overline{x}$
2	x_2	y_2	P_{02}	P_{012}				$x_2 - \overline{x}$
3	x_3	y_3	P_{03}	P_{013}	P_{0123}			$x_3 - \overline{x}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$			$\vdots$
k	x_k	y_k	P_{0k}	P_{01k}	P_{012k}			$x_k - \overline{x}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$		$\vdots$
n	x_n	y_n	P_{0n}	P_{01n}	P_{012n}	$\cdots$	$P_{0123\ldots n}$	$x_n - \overline{x}$

$P_{012\ldots n}$ löst die Interpolationsaufgabe zu den $n+1$ Interpolationsstellen (x_i, y_i), $i = 0(1)n$. Im obigen Schema erhält man den Wert $P_{012\ldots n}(\overline{x})$ an einer festen Stelle $\overline{x}$.

Algorithmus 9.6. (*Interpolationsschema von Aitken*)

Gegeben: (x_i, y_i), $i = 0(1)n$, $x_i \neq x_k$ für $i \neq k$

Gesucht: Wert des zugehörigen Interpolationspolynoms $\Phi(\overline{x}) = P_{012\ldots n}(\overline{x})$ an einer (nichttabellierten) Stelle $\overline{x} \neq x_i$.

1. Schritt: In dem Rechenschema 9.5 sind zunächst für $i = 0(1)n$ die Spalte der x_i, die der y_i und die der $x_i - \overline{x}$ auszufüllen.
2. Schritt: Berechnung der $P_{0i}(\overline{x})$ nach Formel (9.7) für $i = 1(1)n$ und $x = \overline{x}$.
3. Schritt: Berechnung der $P_{01i}(\overline{x})$ nach Formel (9.8) für $i = 2(1)n$ und $x = \overline{x}$.
4. Schritt: Berechnung aller weiteren $P_{012\ldots ki}(\overline{x})$ nach Formel (9.9) für $k = 2(1)n-1$ und $i = (k+1)(1)n$ bis zum Wert $P_{0123\ldots n}(\overline{x}) = \Phi(\overline{x})$.

Wenn die Stützwerte y_i die Werte einer empirischen Funktion f sind, $y_i = f(x_i)$, dann ist $\Phi(\overline{x})$ ein Näherungswert für $f(\overline{x})$.

Nützlich für die praktische Anwendung des Aitken-Schemas ist, dass nicht im Voraus entschieden werden muss, mit wievielen Interpolationsstellen (x_i, y_i) gearbeitet wird. Es ist möglich, stufenweise neue Interpolationsstellen hinzuzunehmen, das Schema also *zeilenweise* auszufüllen. Die Stützstellen müssen nicht monoton angeordnet sein (d. h. es muss nicht gelten $x_0 < x_1 < \ldots < x_n$).

Beispiel 9.7. (Fortsetzung von Beispiel 9.3)

Gegeben: Die folgende Wertetabelle

i	0	1	2
x_i	0	0.5	1
y_i	1	0.8	0.5

mit $y_i = f(x_i) = 1/(1 + x_i^2)$.

Gesucht: Der Wert $\Phi(\overline{x})$ für $\overline{x} = 0.8$ nach dem Interpolationsschema von Aitken.

Lösung: Nach Algorithmus 9.6.

1. Schritt:

x_i	y_i	$P_{0j}(\overline{x})$	$P_{01j}(\overline{x})$	$x_i - \overline{x}$
0	1			−0.8
0.5	0.8	0.680		−0.3
1	0.5	0.600	0.632	0.2

2. Schritt: Berechnung der $P_{0j}(\overline{x})$ und Eintragen in das Schema des 1. Schrittes

$$P_{01}(\overline{x}) = \frac{1}{x_1 - x_0} \begin{vmatrix} y_0 & x_0 - \overline{x} \\ y_1 & x_1 - \overline{x} \end{vmatrix} = \frac{1}{0.5} \begin{vmatrix} 1 & -0.8 \\ 0.8 & -0.3 \end{vmatrix} = 0.680,$$

$$P_{02}(\overline{x}) = \frac{1}{x_2 - x_0} \begin{vmatrix} y_0 & x_0 - \overline{x} \\ y_2 & x_2 - \overline{x} \end{vmatrix} = \begin{vmatrix} 1 & -0.8 \\ 0.5 & 0.2 \end{vmatrix} = 0.600.$$

3. Schritt: Berechnung von $P_{012}(\overline{x})$ und Eintragen in das Schema des 1. Schrittes

$$P_{012}(\overline{x}) = \frac{1}{x_2 - x_1} \begin{vmatrix} P_{01} & x_1 - \overline{x} \\ P_{02} & x_2 - \overline{x} \end{vmatrix} = \frac{1}{0.5} \begin{vmatrix} 0.680 & -0.3 \\ 0.600 & 0.2 \end{vmatrix} = 0.632.$$

Die Ergebnisse der Schritte 2 und 3 werden sofort in das Schema des Schrittes 1 eingetragen. Also ist

$$\Phi(0.8) = P_{012}(0.8) = 0.632.$$

Vergleich mit Beispiel 9.3:

$$L(0.8) = P_{012}(0.8) = 0.632;$$

wegen Satz 9.1 war dieses Ergebnis zu erwarten. Für die Anzahl erforderlicher Punktoperationen ergeben sich bei Lagrange $3 \cdot 5 = 15$, bei Aitken $3 \cdot 3 = 9$.

P_{02} ist das Interpolationspolynom 1. Grades zu den Interpolationsstellen $(x_0, y_0), (x_2, y_2)$ und P_{012} das Interpolationspolynom 2. Grades zu den Interpolationsstellen (x_0, y_0), (x_1, y_1), (x_2, y_2). Bei Rundung auf drei Dezimalen gilt $f(0.8) = 0.610$, $P_{02}(0.8) = 0.600$, $P_{012}(0.8) = 0.632$, d. h. an der Stelle $\overline{x} = 0.8$ weicht das quadratische Polynom P_{012} mehr von dem wahren Funktionswert $f(0.8)$ ab als das lineare Polynom P_{02}. Abbildung 9.2 veranschaulicht dies mit Hilfe der Graphen von P_{02}, P_{012} und f. Außerdem zeigt die Abbildung, dass z. B. an der Stelle $\overline{x} = 0.4$ im Gegensatz zu $\overline{x} = 0.8$ die Funktion f durch P_{012} besser angenähert wird als durch P_{02}. Aus dem Grad des Interpolationspolynoms kann also *nicht* auf die Güte der Annäherung geschlossen werden (siehe dazu auch Abschnitt 9.8 und Abbildung 9.3).

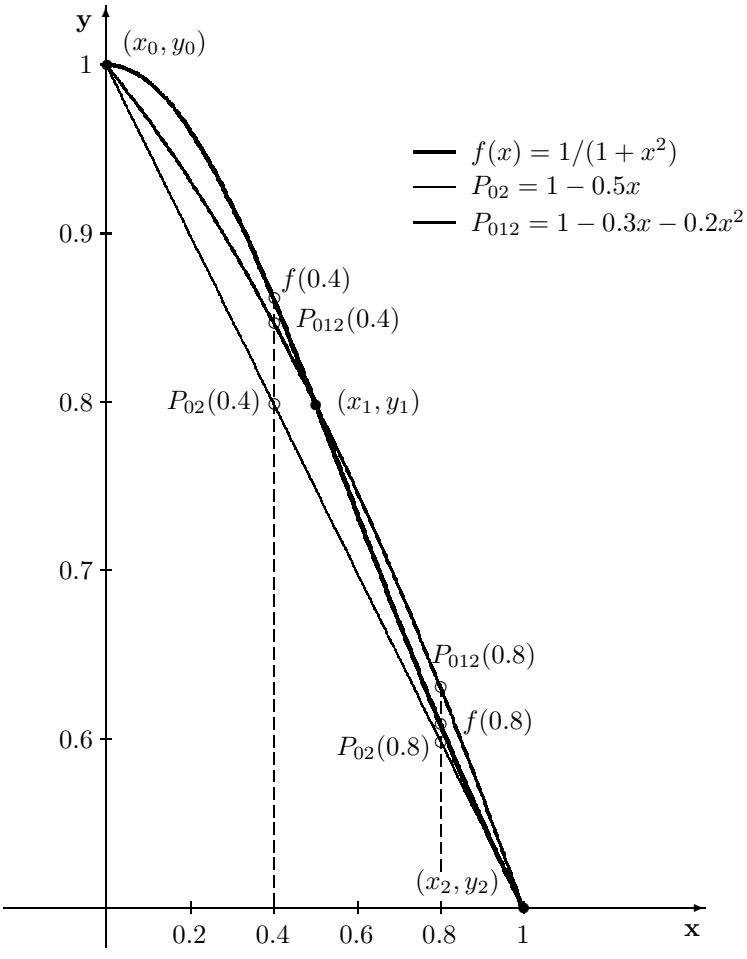


Abb. 9.2.

□

9.4 Inverse Interpolation nach Aitken

Ist für eine in Form einer Wertetabelle $\big(x_i, y_i = f(x_i)\big)$ vorliegende Funktion $f \in C[a,b]$ zu einem nichttabellierten Wert $\overline{y} = f(\overline{x})$ das Argument $\overline{x}$ zu bestimmen oder sind die Nullstellen einer tabellierten Funktion zu bestimmen, d. h. die zu $\overline{y} = 0$ gehörigen Argumente $\overline{x}$, so kann das Aitken-Schema verwendet werden, indem man dort die Rollen von x und y vertauscht. Voraussetzung dafür ist, dass die Umkehrfunktion $x = f^{-1}(y)$ existiert, d. h. f in $[a,b]$ streng monoton ist.

Man bestimmt dann den Wert $\overline{x} = \Phi^*(\overline{y})$ des Interpolationspolynoms Φ^* zu den Interpolationsstellen $\big(y_i, x_i = f^{-1}(y_i)\big)$.

Rechenschema 9.8. (*Inverse Interpolation nach Aitken*)

i	y_i	x_i	x_{0i}	x_{01i}	$\cdots$	$x_{012\ldots n}$	$y_i - \overline{y}$
0	y_0	x_0					$y_0 - \overline{y}$
1	y_1	x_1	x_{01}				$y_1 - \overline{y}$
2	y_2	x_2	x_{02}	x_{012}			$y_2 - \overline{y}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$		$\vdots$
n	y_n	x_n	x_{0n}	x_{01n}	$\cdots$	$x_{012\ldots n}$	$y_n - \overline{y}$

Man geht nach Algorithmus 9.6 vor, indem dort x und y vertauscht, sowie P_{0i} durch x_{0i}, P_{01i} durch x_{01i} usw. ersetzt werden.

Ist f nicht streng monoton, so stellt man das Interpolationspolynom Φ zu den Stellen (x_i, y_i) auf, setzt $\overline{y} = \Phi(\overline{x})$ und löst diese Beziehung nach $\overline{x}$ auf. Dieses Vorgehen erfordert im Allgemeinen das Auflösen einer algebraischen Gleichung hohen Grades.

Beispiel 9.9.

Gegeben: Eine Wertetabelle

i	0	1	2
x_i	0.2	0.4	0.6
y_i	0.962	0.862	0.735

mit auf 3 Dezimalstellen gerundeten Werten $y_i = f(x_i)$ der Funktion $f(x) = 1/(1+x^2)$ und ein Wert $\overline{y} = f(\overline{x}) = 0.8$.

Gesucht: Ein Näherungswert für $\overline{x}$ mittels inverser Interpolation.

Lösung: Die Ergebnisse der Schritte 2 und 3 werden sofort in das Schema des Schrittes 1 eingetragen.

1. Schritt:

y_i	x_i	x_{0j}	x_{01j}	$y_i - \overline{y}$
0.962	0.2			0.162
0.862	0.4	0.524		0.062
0.735	0.6	0.485	0.504	−0.065

2. Schritt:

$$x_{01} = \frac{1}{y_1 - y_0} \begin{vmatrix} x_0 & y_0 - \overline{y} \\ x_1 & y_1 - \overline{y} \end{vmatrix} = -\frac{1}{0.1} \begin{vmatrix} 0.2 & 0.162 \\ 0.4 & 0.062 \end{vmatrix} = 0.524,$$

$$x_{02} = \frac{1}{y_2 - y_0} \begin{vmatrix} x_0 & y_0 - \overline{y} \\ x_2 & y_2 - \overline{y} \end{vmatrix} = -\frac{1}{0.227} \begin{vmatrix} 0.2 & 0.162 \\ 0.6 & -0.065 \end{vmatrix} = 0.485.$$

3. Schritt:

$$x_{012} = \frac{1}{y_2 - y_1} \begin{vmatrix} x_{01} & y_1 - \overline{y} \\ x_{02} & y_2 - \overline{y} \end{vmatrix} = -\frac{1}{0.127} \begin{vmatrix} 0.524 & 0.062 \\ 0.485 & -0.065 \end{vmatrix} = 0.505.$$

Wegen $f(0.5) = 0.8$ ist $\overline{x} = 0.5$, und für den absoluten Fehler ergibt sich $|\overline{x} - x_{012}| = |0.5 - 0.505| = 0.5 \cdot 10^{-2}$.

Für das Interpolationspolynom zu den durch die Wertetabelle gegebenen Interpolationsstellen erhält man bei Rundung auf 4 sichere Dezimalen

$$\Phi : \Phi(x) = 1.0350 - 0.2975x - 0.3375x^2.$$

Daraus folgt $\Phi(0.505) = 0.799$, d. h. $|f(0.505) - \Phi(0.505)| = 1 \cdot 10^{-3}$. □

Beispiel 9.10.

Gegeben: Die Besselfunktion J_0 hat eine Nullstelle $\overline{x}$ mit $2 < \overline{x} < 3$, d. h. $J_0(\overline{x}) = 0$.

Gesucht: Mit Hilfe der Wertetabelle

i	0	1	2	3	4
x_i	2.0	2.2	2.4	2.6	2.8
$y_i = J_0(x_i)$	0.2239	0.1104	0.0025	−0.0968	−0.1850

soll die Nullstelle $\overline{x}$ näherungsweise durch inverse Interpolation bestimmt werden.

Lösung: Nach Rechenschema 9.8.

y_i	x_i	x_{0j}	x_{01j}	x_{012j}	x_{0123j}	$y_i - J_0(\overline{x})$
0.2239	2.0					0.2239
0.1104	2.2	2.3945				0.1104
0.0025	2.4	2.4045	2.4047			0.0025
−0.0968	2.6	2.4189	2.4075	2.4048		−0.0968
−0.1850	2.8	2.4381	2.4108	2.4048	2.4048	−0.1850

Das Verfahren kommt also in diesem Beispiel bei 2.4048 innerhalb der verwendeten Stellenzahl zum Stehen; 2.4048 ist der gesuchte Näherungswert für die Nullstelle $\overline{x}$.

Falls f *nicht monoton* ist, muss der folgende Weg eingeschlagen werden:
Man stellt ohne Vertauschung der Rollen das Interpolationspolynom Φ zu den gegebenen Interpolationsstellen (x_i, y_i) auf. Dann setzt man $\overline{y} = \Phi(\overline{x})$ und löst nach $\overline{x}$ auf. Falls Φ ein Interpolationspolynom hohen Grades ist, erfordert diese Methode natürlich das Auflösen einer algebraischen Gleichung hohen Grades (z. B. dem mit Verfahren von Muller, siehe Abschnitt 3.3.2). Also sollte man dann die Nullstelle besser nach einer anderen Methode berechnen. □

9.5 Interpolationsformeln von Newton

9.5.1 Newtonsche Formel für beliebige Stützstellen

Sind $n+1$ Interpolationsstellen (x_i, y_i), $i = 0(1)n$, mit paarweise verschiedenen Stützstellen x_i gegeben, so lautet der Ansatz für das Newtonsche Interpolationspolynom N:

$$\begin{aligned} \Phi(x) \equiv N(x) = b_0 \quad &+ \quad b_1(x - x_0) + b_2(x - x_0)(x - x_1) + \ldots + \\ &+ \quad b_n(x - x_0)(x - x_1)(x - x_2)\ldots(x - x_{n-1}); \end{aligned} \tag{9.10}$$

es ist ein Polynom (höchstens) n-ten Grades in x.

Während die Auswertung von $N(x)$ in dieser Form $\frac{1}{2}\, n\,(n+1)$ Multiplikationen erfordert, sind mit der zum Horner-Verfahren analogen Darstellung

$$N(x) = \Big\{ \Big[\ldots \big[b_n\,(x - x_{n-1}) + b_{n-1} \big]\,(x - x_{n-2}) + \ldots b_2 \Big]\,(x - x_1) + b_1 \Big\}\,(x - x_0) + b_0$$

nur n Multiplikationen auszuführen.

Die $n+1$ Interpolationsbedingungen

$$\Phi(x_i) \equiv N(x_i) = y_i \quad \text{für} \quad i = 0(1)n$$

bilden ein gestaffeltes System aus $n+1$ linearen Gleichungen für die $n+1$ Koeffizienten $b_0, b_1, \ldots, b_n$.

Die Koeffizienten lassen sich darstellen mit Hilfe der sogenannten dividierten Differenzen

$$\begin{aligned} [x_i\, x_k] \quad &:= \quad \frac{y_i - y_k}{x_i - x_k}\,, \\ [x_i\, x_k\, x_h] \quad &:= \quad \frac{[x_i\, x_k] - [x_k\, x_h]}{x_i - x_h}\,, \\ [x_i\, x_k\, x_h\, x_m] \quad &:= \quad \frac{[x_i\, x_k\, x_h] - [x_k\, x_h\, x_m]}{x_i - x_m}\,, \ldots, \end{aligned}$$

die bei jeder Permutation der beteiligten Interpolationsstellen unverändert bleiben ([WILL1971], S.65 f.).

Als Beispiel werden b_0, b_1, b_2 berechnet aus $N(x_i) = y_i$ für $i = 0, 1, 2$ mit Verwendung von (9.10).

$$
\begin{aligned}
i = 0: \quad & N(x_0) = y_0 = b_0 + b_1\,(x_0 - x_0) + \ldots = b_0 \\
& \Rightarrow \quad b_0 = y_0 \\
i = 1: \quad & N(x_1) = y_1 = b_0 + b_1\,(x_1 - x_0) = y_0 + b_1\,(x_1 - x_0) \\
& \Rightarrow \quad b_1 = \tfrac{y_1 - y_0}{x_1 - x_0} = [x_1\,x_0] \\
i = 2: \quad & N(x_2) = y_2 = y_0 + [x_1\,x_0]\,(x_2 - x_0) + b_2\,(x_2 - x_0)\,(x_2 - x_1) \\
& \Rightarrow \quad b_2 = \left(\tfrac{y_2 - y_0}{x_2 - x_0} - [x_1\,x_0]\right)/(x_2 - x_1) \\
& \qquad\quad = \tfrac{[x_2\,x_0] - [x_0\,x_1]}{x_2 - x_1} = [x_2\,x_0\,x_1] \\
& \qquad\quad = [x_2\,x_1\,x_0]
\end{aligned}
$$

Hier wurde die Invarianz der dividierten Differenzen gegenüber Vertauschungen der Argumente benutzt.

Insgesamt ergeben sich für die Koeffizienten

$$
\left\{
\begin{aligned}
& b_0 = y_0\,, \\
& b_1 = [x_1 x_0] = \frac{y_1 - y_0}{x_1 - x_0}\,, \\
& b_2 = [x_2 x_1 x_0] = \frac{[x_2 x_1] - [x_1 x_0]}{x_2 - x_0}\,, \\
& b_3 = [x_3 x_2 x_1 x_0] = \frac{[x_3 x_2 x_1] - [x_2 x_1 x_0]}{x_3 - x_0}\,, \\
& \vdots \\
& b_n = [x_n x_{n-1} \ldots x_2 x_1 x_0] = \frac{[x_n x_{n-1} \ldots x_2 x_1] - [x_{n-1} x_{n-2} \ldots x_1 x_0]}{x_n - x_0}\,.
\end{aligned}
\right. \tag{9.11}
$$

Die b_k lassen sich besonders bequem mit dem folgenden Rechenschema bestimmen, dabei ist die Reihenfolge der Stützstellen x_i beliebig.

Rechenschema 9.11. (*Interpolation nach Newton*)

i	x_i	y_i				
0	x_0	$y_0 = \underline{b_0}$				
1	x_1	y_1	$[x_1 x_0] = \underline{b_1}$			
2	x_2	y_2	$[x_2 x_1]$	$[x_2 x_1 x_0] = \underline{b_2}$		
3	x_3	y_3	$[x_3 x_2]$	$[x_3 x_2 x_1]$	$[x_3 x_2 x_1 x_0] = \underline{b_3}$	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	

Algorithmus 9.12. (*Interpolationsformel von Newton*)

Gegeben: Die Interpolationsstellen (x_i, y_i), $i = 0(1)n$, $x_i \neq x_k$ für $i \neq k$

Gesucht: Das zugehörige Interpolationspolynom in der Form von Newton

1. Schritt: Berechnung der b_k mit dem Rechenschema 9.11 unter Verwendung von (9.11).

2. Schritt: Aufstellen der Interpolationsformel $N(x)$ gemäß (9.10).

Bemerkung. Bei Hinzunahme einer Interpolationsstelle können alle zuvor berechneten Werte verwendet werden, das Schema wird lediglich fortgesetzt. Deshalb eignet sich diese Methode zur praktischen Verwendung besser als die von Lagrange (s. Abschnitt 9.8).

Beispiel 9.13.

Gegeben: Die folgende Wertetabelle

i	0	1	2	3
x_i	0	1	2	4
y_i	-3	1	2	7

Gesucht: Das Newtonsche Interpolationspolynom N_3 zu (x_i, y_i), $i = 0(1)3$.

Lösung: Mit Hilfe des Rechenschemas 9.11 erhält man

i	x_i	y_i			
0	$x_0 = 0$	$y_0 = \underline{-3 = b_0}$	$\frac{1-(-3)}{1-0} = \underline{4 = b_1}$	$\frac{1-4}{2-0} = \underline{-1.5 = b_2}$	$\frac{0.5-(-1.5)}{4-0} = \underline{0.5 = b_3}$
1	$x_1 = 1$	$y_1 = 1$	$\frac{2-1}{2-1} = 1$	$\frac{2.5-1}{4-1} = 0.5$	
2	$x_2 = 2$	$y_2 = 2$	$\frac{7-2}{4-2} = 2.5$		
3	$x_3 = 4$	$y_3 = 7$			

und mit (9.10)

$$\begin{aligned} N_3(x) &= b_0 + b_1\,(x - x_0) + b_2\,(x - x_0)\,(x - x_1) + b_3\,(x - x_0)\,(x - x_1)\,(x - x_2) \\ &= -3 + 4\,(x - 0) - 1.5\,(x - 0)\,(x - 1) + 0.5\,(x - 0)\,(x - 1)\,(x - 2) \\ &= -3 + 4x - 1.5\,x\,(x - 1) + 0.5\,x\,(x - 1)\,(x - 2) \\ &= -3 + x\big\{4 + (x - 1)[-1.5 + (x - 2)\,0.5]\big\} \end{aligned}$$

□

Beispiel 9.14. (Fortsetzung von Beispiel 9.13)

Gegeben: Die folgende (um eine Spalte erweiterte) Wertetabelle

i	0	1	2	3	**4**
x_i	0	1	2	4	**1.5**
y_i	-3	1	2	7	**3**

Gesucht: Das Newtonsche Interpolationspolynom N_4 zu $(x_i,\, y_i)$, $i = 0(1)4$.

Lösung: Durch Ergänzung des Rechenschemas 9.11 in Beispiel 9.13 erhält man

i	x_i	y_i				
0	$x_0 = 0$	$y_0 = \underline{-3 = b_0}$	$\frac{1-(-3)}{1-0} = \underline{4 = b_1}$	$\frac{1-4}{2-0} = \underline{-1.5 = b_2}$	$\frac{0.5-(-1.5)}{4-0} = \underline{0.5 = b_3}$	$\frac{2.6-0.5}{1.5-0} = \underline{1.4 = b_4}$
1	$x_1 = 1$	$y_1 = 1$	$\frac{2-1}{2-1} = 1$	$\frac{2.5-1}{4-1} = 0.5$	$\frac{1.8-0.5}{1.5-1} = 2.6$	
2	$x_2 = 2$	$y_2 = 2$	$\frac{7-2}{4-2} = \frac{5}{2} = 2.5$	$\frac{1.6-2.5}{1.5-2} = \frac{9}{5} = 1.8$		
3	$x_3 = 4$	$y_3 = 7$	$\frac{3-7}{1.5-4} = \frac{-4}{-2.5} = 1.6$			
4	$x_4 = \frac{3}{2}$	$y_4 = 3$				

und mit (9.10)

$$
\begin{aligned}
N_4(x) &= b_0 + b_1\,(x - x_0) + b_2\,(x - x_0)\,(x - x_1) + b_3\,(x - x_0)\,(x - x_1)\,(x - x_2) \\
&\quad + \boxed{b_4\,(x - x_0)\,(x - x_1)\,(x - x_2)\,(x - x_3)} \\
&= N_3(x) + \boxed{b_4\,(x - x_0)\,(x - x_1)\,(x - x_2)\,(x - x_3)} \\
&= -3 + 4x - 1.5\,x\,(x-1) + 0.5\,x\,(x-1)\,(x-2) + \boxed{1.4\,x\,(x-1)\,(x-2)\,(x-4)}
\end{aligned}
$$

□

9.5.2 Newtonsche Formel für äquidistante Stützstellen

Sind die Stützstellen x_i äquidistant mit der festen Schrittweite $h = x_{i+1} - x_i$, $i = 0(1)n-1$, dann ist $x_i = x_0 + hi$, $i = 0(1)n$, und es wird gesetzt

$$x = x_0 + ht\,, \qquad t \in [0, n]\,.$$

Für die Koeffizienten b_i im Rechenschema 9.11 führt man mit sogenannten *Differenzen* Δ_i^k eine abkürzende Schreibweise ein. Dabei beziehen sich die Differenzen Δ_i^k hier

grundsätzlich auf y-Werte, so dass statt $\Delta_i^k y$ kurz Δ_i^k geschrieben wird. Die Differenzen sind wie folgt definiert:

$$\begin{aligned}
\Delta_i^0 &= y_i, \\
\Delta_{i+1/2}^{k+1} &= \Delta_{i+1}^k - \Delta_i^k, && k = 0, 2, 4, \ldots, \\
\Delta_i^{k+1} &= \Delta_{i+1/2}^k - \Delta_{i-1/2}^k, && k = 1, 3, 5, \ldots .
\end{aligned}$$

Dann sind z. B.

$$\begin{aligned}
\Delta_{i+1/2}^1 &= y_{i+1} - y_i, \quad \Delta_i^2 = \Delta_{i+1/2}^1 - \Delta_{i-1/2}^1 = y_{i+1} - 2y_i + y_{i-1}, \\
\Delta_{i+1/2}^3 &= \Delta_{i+1}^2 - \Delta_i^2 = y_{i+2} - 3y_{i+1} + 3y_i - y_{i-1}.
\end{aligned}$$

Die Differenzen Δ_i^k werden mit dem folgendem Rechenschema bestimmt.

Rechenschema 9.15. (*Differenzenschema*)

i	y_i	$\Delta_{i+1/2}^1$	Δ_i^2	$\Delta_{i+1/2}^3$	$\cdots$
0	y_0				
		$y_1 - y_0 = \Delta_{1/2}^1$			
1	y_1		$\Delta_{3/2}^1 - \Delta_{1/2}^1 = \Delta_1^2$		
		$y_2 - y_1 = \Delta_{3/2}^1$		$\Delta_2^2 - \Delta_1^2 = \Delta_{3/2}^3$	
2	y_2		$\Delta_{5/2}^1 - \Delta_{3/2}^1 = \Delta_2^2$	$\vdots$	
		$y_3 - y_2 = \Delta_{5/2}^1$	$\vdots$		
3	y_3	$\vdots$			
$\vdots$	$\vdots$				

Für die b_i gilt mit $h = x_{i+1} - x_i$

$$b_i = [x_i x_{i-1} \ldots x_1 x_0] = \frac{1}{i! h^i} \Delta_{i/2}^i, \qquad i = 1(1)n,$$

und für (9.10) unter Verwendung der Binomialkoeffizienten $\binom{t}{k}$

$$N(x) = \widetilde{N}(t) = y_0 + \binom{t}{1}\Delta_{1/2}^1 + \binom{t}{2}\Delta_1^2 + \ldots + \binom{t}{n}\Delta_{n/2}^n;$$

$N(x)$ bzw. $\widetilde{N}(t)$ ist die *Newtonsche Interpolationsformel für absteigende Differenzen* und wird mit $N_+(x)$ bzw. $\widetilde{N}_+(t)$ bezeichnet (sie wird in Abschnitt 14.5 angewendet).

Beispiel 9.16.

Gegeben: Die folgende Wertetabelle

i	0	1	2	3
x_i	-3	-1	1	3
y_i	5	4	0	1

mit der konstanten Schrittweite $h = 2$.

Gesucht: Die Newtonsche Interpolationsformel $N_3(x)$ für äquidistante Stützstellen zu $(x_i,\, y_i)$, $i = 0(1)3$.

Lösung: Mit Hilfe des Rechenschemas 9.15 erhält man

i	$y_i = \Delta^0_{i/2}$	$\Delta^1_{i+1/2}$	Δ^2_1	$\Delta^3_{i+1/2}$
0	5			
		-1		
1	4		-3	
		-4		8
2	0		5	
		1		
3	1			

$$\Rightarrow \quad N_3(x) = \widetilde{N}_3(t) = 5 + \binom{t}{1} \cdot (-1) + \binom{t}{2} \cdot (-3) + \binom{t}{3} \cdot 8\,,$$

bzw. mit

$$b_0 = \frac{1}{0!\,h^0}\,\Delta^0_{0/2} = \frac{5}{1} = 5\,, \quad b_1 = \frac{-1}{1!\,h^1} = -\frac{1}{2}\,, \quad b_2 = \frac{-3}{2!\,h^2} = -\frac{3}{8} \quad \text{und} \quad b_3 = \frac{8}{3!\,h^3} = \frac{1}{6}$$

$\Rightarrow$

$$\begin{aligned}
N_3(x) &= 5 - \tfrac{1}{2}\,(x+3) - \tfrac{3}{8}\,(x+3)\,(x+1) + \tfrac{1}{6}\,(x+3)\,(x+1)\,(x-1)\\
N_3(-3) &= 5\,,\\
N_3(-1) &= 5 - \tfrac{1}{2}\cdot 2 = 5 - 1 = 4\,,\\
N_3(1) &= 5 - \tfrac{4}{2} - \tfrac{3}{8}\cdot 4\cdot 2 = 5 - 2 - 3 = 0\,,\\
N_3(3) &= 5 - \tfrac{6}{2} - \tfrac{3}{8}\cdot 6\cdot 4 + \tfrac{1}{6}\cdot 6\cdot 4\cdot 2 = 5 - 3 - 9 + 8 = 1
\end{aligned}$$

□

9.6 Abschätzung und Schätzung des Interpolationsfehlers

Das Interpolationspolynom $\Phi \in C(I_x)$, gebildet zu $n+1$ Interpolationsstellen $\big(x_i, y_i = f(x_i)\big), x_i \in I_x$, $i = 0(1)n$, nimmt an den Stützstellen x_i die Stützwerte $f(x_i)$ an, während es im Allgemeinen an allen anderen Stellen $x \in I_x$ von $f \in C(I_x)$ abweicht. Dann ist R mit

$$R(x) := f(x) - \Phi(x), \quad x \in I_x\,,$$

der wahre *Interpolationsfehler*, und R heißt das *Restglied der Interpolation*. Während das Restglied R also an den Stützstellen verschwindet, kann man über seinen Verlauf in I_x für $x \neq x_i$ im Allgemeinen nichts aussagen, denn man kann f an den Stellen $x \neq x_i$ beliebig ändern, ohne damit Φ zu verändern (siehe Abbildung 9.3).

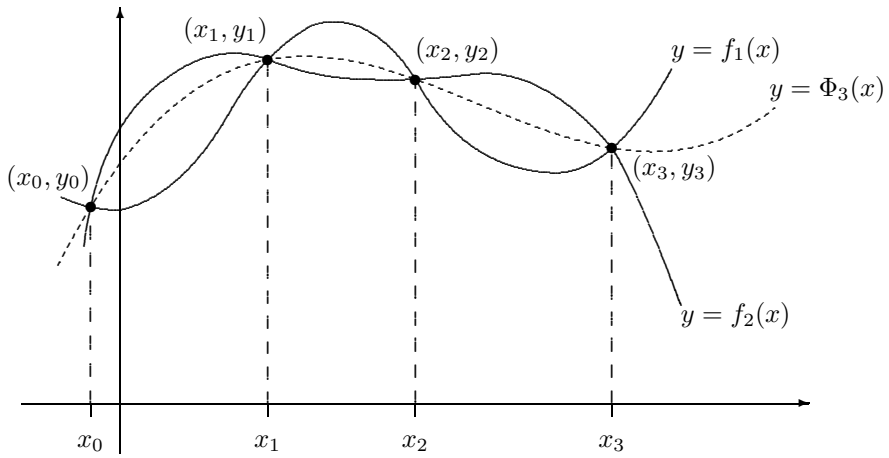


Abb. 9.3. Interpolationspolynom Φ zu verschiedenen Funktionen mit denselben Interpolationsstellen

Für die Funktionen f_1, f_2 ist $y = \Phi_3(x)$ das Interpolationspolynom zu den Interpolationsstellen (x_i, y_i), $i = 0(1)3$, durch die auch f_1 bzw. f_2 gehen. Also kann zwischen den Interpolationsstellen die Annäherung von Φ_3 an eine Funktion f_i beliebig schlecht sein. Für eine Abschätzung dieses Fehlers mit der *Lagrangeschen Restgliedformel* benötigt man die $(n+1)$-te Ableitung der zu interpolierenden Funktion f. Sie lautet

$$\left\{ \begin{array}{lcl} R(x) & = & \frac{1}{(n+1)!} f^{(n+1)}(\xi)\pi(x) \quad \text{mit} \quad \pi(x) = \prod_{i=0}^{n}(x - x_i), \\ \xi & = & \xi(x) \in I_x, \end{array} \right. \tag{9.12}$$

bzw. im Falle äquidistanter Stützstellen $x_i = x_0 + hi$, $x = x_0 + ht$, $t \in I_t = [0, n]$

$$R(x) = R(x_0 + ht) = h^{n+1} \tfrac{1}{(n+1)!} f^{(n+1)}(\widetilde{\xi}) \pi^*(t) =: \widetilde{R}(t) \quad \text{mit}$$

$$\pi^*(t) = \prod_{i=0}^{n} (t - i), \quad \widetilde{\xi} = \widetilde{\xi}(t) \in I_t.$$

Für die praktische Anwendung sind diese Formeln völlig unbrauchbar. Man kann jedoch im Fall äquidistanter Stützstellen aus dem Verlauf von $\pi^*(t)$ einige interessante Schlüsse ziehen.

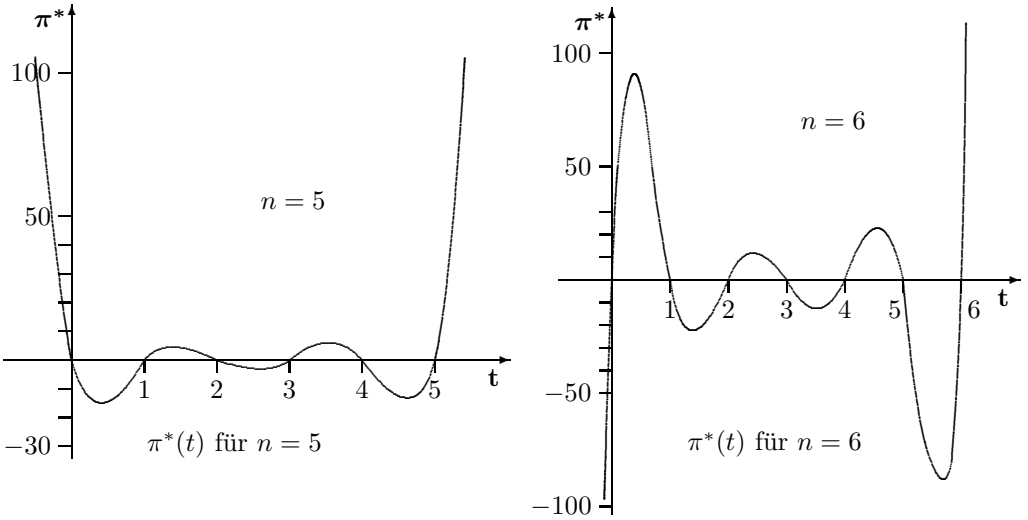


Abb. 9.4. Interpolation

Die Beträge der Extremwerte von $\pi^*(t)$ nehmen bis zur Mitte des Intervalls $[0, n]$ ab und danach wieder zu, sie wachsen außerhalb dieses Intervalls stark an. Man entnimmt daraus: $\widetilde{R}(t)$ wird besonders groß für Werte, die außerhalb des Interpolationsintervalls liegen (Extrapolation); das Interpolationsintervall $I_t = [0, n]$ erstreckt sich von der ersten bis zur letzten der zur Interpolation verwendeten Stützstellen. Diese Aussage ist von Bedeutung für die Auswahl der für eine bestimmte Aufgabe geeigneten Interpolationsstellen für das Interpolationspolynom. Man wählt die für ein Interpolationspolynom zu verwendenden Wertepaare (x_i, y_i) so aus, dass die Stelle $\overline{x}$ für eine Auswertung des Polynoms etwa in der Mitte des Interpolationsintervalls liegt.

Einzig und allein praktikabel sind hier *Fehlerschätzungen*, die die Kenntnis der $(n+1)$-ten Ableitung umgehen. Eine mögliche *Schätzung des Restgliedes* $R(x) = R(x_0 + th) = \widetilde{R}(t)$ für den Fall, dass auch außerhalb des Interpolationsintervalls Interpolationsstellen bekannt sind und somit die $(n+1)$-ten Differenzen Δ^{n+1} gebildet werden können, ist

$$\widetilde{R}(t) \approx \frac{1}{(n+1)!} \Delta^{n+1} \pi^*(t),$$

falls sich die Differenzen Δ^{n+1} nur wenig voneinander unterscheiden; es ist dann gleichgültig, welche der $(n+1)$-ten Differenzen verwendet wird, vgl. [POLO1964], S.136/137; [ZURM1965], S.218.

Als Schätzwert mit dem *Newton-Restglied für beliebige Stützstellen* erhält man für den Interpolationsfehler $R(\overline{x})$ an einer Stelle $\overline{x} \in [a,b]$

$$R(\overline{x}) \approx [x_{n+1}, x_n, x_{n-1}, \ldots, x_1, x_0] \prod_{i=0}^{n} (\overline{x} - x_i),$$

wenn außer den für Φ verwendeten $n+1$ Interpolationsstellen (x_i, y_i), $i = 0(1)n$, noch eine weitere Stelle (x_{n+1}, y_{n+1}) bekannt ist.

Beispiel 9.17.

Das zur Wertetabelle

i	0	1	2
x_i	0	$\pi/2$	π
$y_i = \sin x_i$	0	1	0

gehörige Interpolationspolynom wird mit der Newtonschen Interpolationsformel berechnet. Zusätzlich sei die Interpolationsstelle $(\frac{\pi}{6}, \frac{1}{2})$ bekannt. Somit ist eine Fehlerschätzung z. B. an der Stelle $\overline{x} = \frac{\pi}{4}$ wie folgt möglich.

i	x_i	y_i			
0	0	$0 = b_0$			
			$\frac{2}{\pi} = b_1$		
1	$\frac{\pi}{2}$	1		$-\frac{4}{\pi^2} = b_2$	
			$-\frac{2}{\pi}$		$-\frac{6}{5\pi^3} = [x_3\, x_2\, x_1\, x_0]$
2	π	0		$-\frac{21}{5\pi^2}$	
			$-\frac{3}{5\pi}$		
3	$\frac{\pi}{6}$	$\frac{1}{2}$			

$$N_2(x) = 0 + \frac{2}{\pi}(x-0) - \frac{4}{\pi^2}(x-0)(x-\frac{\pi}{2})$$

Für den Fehler von N_2 an der Stelle $\overline{x} = \frac{\pi}{4}$ ergibt sich der Schätzwert

$$\begin{aligned} R_2\left(\tfrac{\pi}{4}\right) &= f\left(\tfrac{\pi}{4}\right) - N_2\left(\tfrac{\pi}{4}\right) \approx [x_3\, x_2\, x_2\, x_1] \prod_{i=0}^{2} \left(\tfrac{\pi}{4} - x_i\right) \\ &= -\tfrac{6}{5\,\pi^3} \cdot \tfrac{\pi}{4} \cdot \left(-\tfrac{\pi}{4}\right) \left(-\tfrac{3\,\pi}{4}\right) = -\tfrac{9}{160} = -0.05625\,. \end{aligned}$$

Für den wahren Fehler gilt mit $f(\frac{\pi}{4}) = \sin(\frac{\pi}{4}) = \frac{1}{2}\sqrt{2}$ und $N_2(\frac{\pi}{4}) = 0.75$

$$R_2\left(\frac{\pi}{4}\right) = \frac{1}{2}\sqrt{2} - 0.75 = -0.042893219\,.$$

Das heißt der Schätzwert -0.05625 war recht gut, er liefert das richtige Vorzeichen und die richtige Größenordnung. □

Zur Konvergenz der Interpolation

Von einer Funktion $f : [a, b] \to \mathbb{R}$ seien an n+1 Stützstellen $x_i \in [a, b]$, $i = 0(1)n$, $n \in \mathbb{N}_0$, die Funktionswerte $f(x_i)$ bekannt. Es wird eine Folge $\{\Phi_n\}$ von Interpolationspolynomen $\Phi_n, n = 0, 1, 2, ...,$ konstruiert zu den $\left(x_i^{(n)}, f(x_i^{(n)})\right)$ mit den Stützstellen in der n-ten Zeile des Dreieckschemas

$$\begin{array}{lllll}
\Phi_0: & x_0^{(0)} & & & \\
\Phi_1: & x_0^{(1)}, & x_1^{(1)} & & \\
\Phi_2: & x_0^{(2)}, & x_1^{(2)}, & x_2^{(2)} & \\
\vdots & \vdots & \vdots & & \\
\Phi_n: & x_0^{(n)}, & x_1^{(n)}, & x_2^{(n)}, & \ldots \quad x_n^{(n)}
\end{array}$$

Der Interpolationsprozess heißt konvergent, wenn

$$\lim_{n\to\infty} \Phi_n(x) = f(x) \quad \text{für} \quad x \in [a, b]$$

gilt. Man könnte annehmen, dass die Konvergenz für stetige Funktionen f immer gewährleistet ist, wenn die Stützstellen, an denen die exakten Funktionswerte vorgegeben sind, immer dichter liegen; dies ist jedoch nicht der Fall. Auch bei beliebig oft differenzierbaren Funktionen braucht die Folge der Interpolationspolynome nicht gegen f zu konvergieren. Dazu ein

Beispiel 9.18.

Die Funktion $f : [-5, 5] \to \mathbb{R}$, $f(x) = 1/(1 + x^2)$ ist gegeben. Es werden die Interpolationspolynome Φ_n zu 3, 5, 9 und 17 äquidistanten Stützstellen aus [−5,5] konstruiert und mit der entsprechenden natürlichen kubischen Splinefunktion S (siehe Kapitel 10) und dem exakten Graphen von f verglichen. Aus Gründen der Darstellbarkeit wird die y-Achse gestreckt.

Man sieht, dass bei wachsender Anzahl der Stützstellen, zu denen ein Interpolationspolynom bestimmt wird, die Annäherung im mittleren Bereich zwar besser wird, aber im äußeren Bereich ($-5 \le x \le -2.5, 2.5 \le x \le 5$) immer schlechter. Es liegt hier keine Konvergenz der Interpolation vor im Gegensatz zu den Splines, die gegen f konvergieren (siehe Abbildung 9.5). Zur Theorie der Konvergenz von Interpolationspolynomen siehe [BERE1971] I, Abschnitt 2.9.

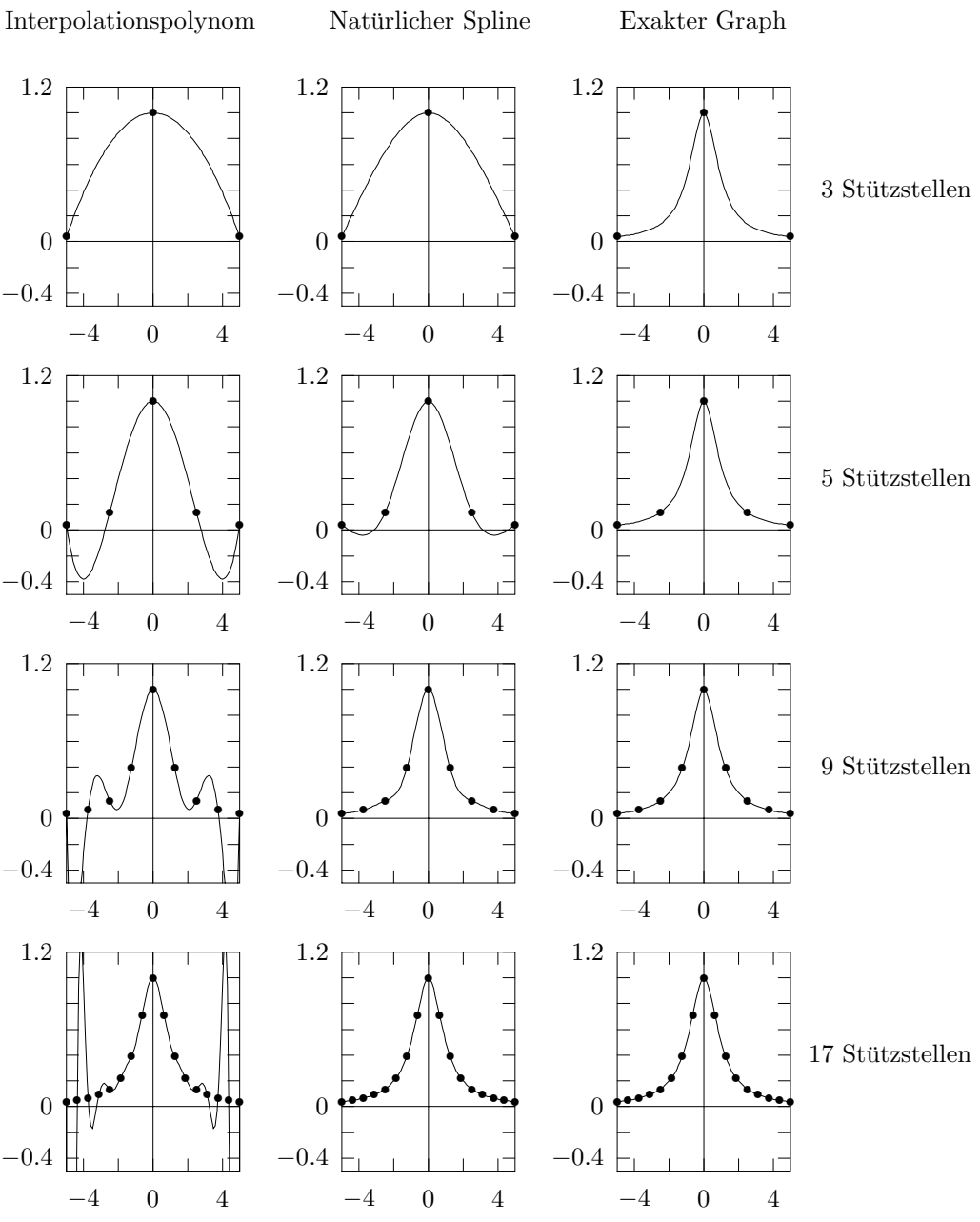


Abb. 9.5. Gegenüberstellung Interpolationspolynom, natürlicher Spline, exakter Graph zu $f(x) = 1/(1 + x^2)$ für $x \in [-5, 5]$, bei Verwendung von 3, 5, 9, 17 äquidistanten Stützstellen □

9.7 Zweidimensionale Interpolation

In Verallgemeinerung der bisher behandelten eindimensionalen Interpolation, bei der die Stützstellen x_i auf der x-Achse liegen, sind die Stützstellen jetzt Punkte (x_j, y_j) in der x, y-Ebene. Auch hier müssen die Stützstellen paarweise verschieden sein:

$$(x_j, y_j) \neq (x_k, y_k) \quad \text{für} \quad j \neq k\,.$$

Alle Stützstellen sollen in einem endlichen Bereich B der x, y-Ebene liegen:

$$(x_j, y_j) \in B \subset \mathbb{R}^2 \quad \text{für} \quad j = 0(1)N\,.$$

Jeder Stützstelle (x_j, y_j) sei genau ein Stützwert $z_j \in \mathbb{R}$ zugeordnet, so dass also $N{+}1$ Interpolationsstellen $(x_j, y_j, z_j) \in \mathbb{R}^3$ mit paarweise verschiedenen Stützstellen gegeben sind.
Gesucht wird eine stetige Funktion

$$\Phi : B \to \mathbb{R},\ (x, y) \mapsto z = \Phi(x, y),\ (x, y) \in B,$$

die den $N{+}1$ Interpolationsbedingungen

$$\Phi(x_j, y_j) = z_j, \quad j = 0(1)N,$$

genügt.

Der Graph der Funktion Φ

$$\left\{ (x, y, z) \,|\, (x, y) \in B,\ z = \Phi(x, y) \right\}$$

ist eine Fläche über der x, y-Ebene, auf der alle Interpolationsstellen, die Punkte (x_j, y_j, z_j), liegen (Abb 9.7).

Mit einer solchen Funktion Φ kann eine empirische Funktion $f : z = f(x, y)$, von der nur die $N{+}1$ Interpolationsstellen $\big(x_j, y_j, z_j = f(x_j, y_j)\big)$, $i = 0(1)N$, bekannt sind, angenähert werden.

Für Φ wird beispielsweise ein algebraisches Polynom

$$\Phi(x, y) \equiv P_r(x, y) = \sum_{p,q} a_{pq}\, x^p\, y^q$$

gewählt mit einem möglichst niedrigen Grad $r = \max(p + q)$. Es soll den $N{+}1$ Interpolationsbedingungen

$$P_r(x_j, y_j) = \sum_{p,q} a_{pq}\, x_j^p\, y_j^q = z_j, \quad j = 0(1)N,$$

genügen. Hier sind Existenz und Eindeutigkeit eines solchen Polynoms im Allgemeinen nicht gesichert ([BERE1971] Bd.1, S.130; [ISAA1973], Abschnitt 6.6; [SAUE1969] Bd.III, S.292). Zumindest müsste die Anzahl der Interpolationsstellen mit der Anzahl der zu bestimmenden Koeffizienten übereinstimmen.

Deshalb wird im Folgenden vorausgesetzt, dass die Stützstellen (x_j, y_j) die Punkte eines Rechteckgitters sind.

9.7.1 Zweidimensionale Interpolationsformel von Lagrange

Die Stützstellen seien jetzt die Gitterpunkte eines rechtwinkligen Netzes. Man bezeichnet sie mit (x_i, y_k), $i = 0(1)m$, $k = 0(1)n$, und die ihnen zugeordneten Stützwerte mit z_{ik}. Dann können die Interpolationsstellen in einer Wertetabelle angegeben werden:

	y_0	y_1	$\cdots$	y_n
x_0	z_{00}	z_{01}	$\cdots$	z_{0n}
x_1	z_{10}	z_{11}	$\cdots$	z_{1n}
$\vdots$	$\vdots$	$\vdots$		$\vdots$
x_m	z_{m0}	z_{m1}	$\cdots$	z_{mn}

Diese spezielle Interpolationsaufgabe ist eindeutig lösbar durch

$$\Phi(x,y) = \sum_{i=0}^{m}\sum_{k=0}^{n} a_{ik} x^i y^k$$

([SAUE1969] Bd.III, S.292). Die Interpolationsformel von Lagrange für die obige Stützstellenverteilung erhält mit

$$\left\{\begin{array}{lcl} L_i^{(1)}(x) & = & \dfrac{(x-x_0)\ldots(x-x_{i-1})(x-x_{i+1})\ldots(x-x_m)}{(x_i-x_0)\ldots(x_i-x_{i-1})(x_i-x_{i+1})\ldots(x_i-x_m)}, \\ L_k^{(2)}(y) & = & \dfrac{(y-y_0)\ldots(y-y_{k-1})(y-y_{k+1})\ldots(y-y_n)}{(y_k-y_0)\ldots(y_k-y_{k-1})(y_k-y_{k+1})\ldots(y_k-y_n)} \end{array}\right. \tag{9.13}$$

die Form

$$\Phi(x,y) \equiv L(x,y) = \sum_{i=0}^{m}\sum_{k=0}^{n} L_i^{(1)}(x) L_k^{(2)}(y) z_{ik}.$$

In (9.13) müssen die Stützstellen zwar nicht äquidistant sein, jedoch ist

$$\begin{array}{rcl} x_{i+1} - x_i & = & h_i^{(1)} = \quad \text{const. für alle} \quad y_k \quad \text{und festes} \quad i, \\ y_{k+1} - y_k & = & h_k^{(2)} = \quad \text{const. für alle} \quad x_i \quad \text{und festes} \quad k. \end{array}$$

Zur Approximation von Funktionen mehrerer Veränderlichen vgl. noch [COLL1968], §25; [SAUE1969] Band III, S. 348-350; die Verfahren sind weniger weit entwickelt als bei Funktionen einer Veränderlichen. Es empfiehlt sich die Verwendung mehrdimensionaler Splines (vgl. Kapitel 12).

Beispiel 9.19.

Gegeben: Folgende Wertetabelle für (x_i, y_k, z_{ik}), $i = 0(1)3$, $k = 0(1)2$

x_i \ y_k	$y_0 = 1.00$	$y_1 = 3.00$	$y_2 = 4.00$
$x_0 = -2.00$	0.00	5.00	7.00
$x_1 = 0.00$	5.00	7.00	4.00
$x_2 = 1.00$	7.00	4.00	−1.00
$x_3 = 5.00$	3.00	0.00	0.00

Gesucht: Die zugehörigen Interpolationspolynome $L_i^{(1)}(x)$ und $L_k^{(2)}(y)$ für die Interpolationsformel von Lagrange nach (9.13) und eine Wertetabelle, die außerdem die Werte $L(x,y)$ für

$x =$ $-1.00,$ $2.00,$ $3.00,$ 4.00 und
$y =$ $1.50,$ $2.00,$ $2.50,$ 3.50 enthält.

Lösung: Im Folgenden werden die Interpolationspolynome berechnet.

$$L_0^{(1)}(x) = \frac{(x-x_1)\,(x-x_2)\,(x-x_3)}{(x_0-x_1)\,(x_0-x_2)\,(x_0-x_3)} = \frac{(x-0)\,(x-1)\,(x-5)}{(-2)\cdot(-3)\cdot(-7)} = \frac{x\,(x-1)\,(x-5)}{-42}$$

$$L_1^{(1)}(x) = \frac{(x-x_0)\,(x-x_2)\,(x-x_3)}{(x_1-x_0)\,(x_1-x_2)\,(x_1-x_3)} = \frac{(x+2)\,(x-1)\,(x-5)}{2\cdot(-1)\cdot(-5)} = \frac{(x+2)\,(x-1)\,(x-5)}{10}$$

$$L_2^{(1)}(x) = \frac{(x-x_0)\,(x-x_1)\,(x-x_3)}{(x_2-x_0)\,(x_2-x_1)\,(x_2-x_3)} = \frac{(x+2)\,(x-0)\,(x-5)}{3\cdot 1\cdot(-4)} = \frac{(x+2)\,x\,(x-5)}{-12}$$

$$L_3^{(1)}(x) = \frac{(x-x_0)\,(x-x_1)\,(x-x_2)}{(x_3-x_0)\,(x_3-x_1)\,(x_3-x_2)} = \frac{(x+2)\,(x-0)\,(x-1)}{7\cdot 5\cdot 4} = \frac{(x+2)\,x\,(x-1)}{140}$$

$$L_0^{(2)}(y) = \frac{(y-y_1)\,(y-y_2)}{(y_0-y_1)\,(y_0-y_2)} = \frac{(y-3)\,(y-4)}{(-2)\cdot(-3)} = \frac{(y-3)\,(y-4)}{6}$$

$$L_1^{(2)}(y) = \frac{(y-y_0)\,(y-y_2)}{(y_1-y_0)\,(y_1-y_2)} = \frac{(y-1)\,(y-4)}{2\cdot(-1)} = \frac{(y-1)\,(y-4)}{-2}$$

$$L_2^{(2)}(y) = \frac{(y-y_0)\,(y-y_1)}{(y_2-y_0)\,(y_2-y_1)} = \frac{(y-1)\,(y-3)}{3\cdot 1} = \frac{(y-1)\,(y-3)}{3}$$

Die ergänzte Wertetabelle:

x \ y	1.00	1.50	2.00	2.50	3.00	3.50	4.00
−2.00	0.00	1.38	2.67	3.88	5.00	6.04	7.00
−1.00	2.54	4.66	6.24	7.30	7.83	7.83	7.30
0.00	5.00	6.50	7.33	7.50	7.00	5.83	4.00
1.00	7.00	7.12	6.67	5.62	4.00	1.79	−1.00
2.00	8.17	6.75	4.97	2.83	0.31	−2.56	−5.80
3.00	8.14	5.61	2.98	0.25	−2.57	−5.49	−8.50
4.00	6.54	3.91	1.41	−0.95	−3.17	−5.25	−7.20
5.00	3.00	1.88	1.00	0.38	0.00	−0.12	0.00

□

9.7.2 Shepard-Interpolation

Wenn die Stützstellen (x_j, y_j), $j = 0(1)N$, in einem Bereich $B \subset \mathbb{R}^2$ beliebig und ungeordnet verteilt sind, so empfiehlt sich der Einsatz der Methode von Shepard [SHEP1968]. Diese Methode wird gern bei der graphischen Darstellung einer empirischen Funktion $z = f(x, y)$ verwendet.

Shepard benutzt für die interpolierende Funktion $\Phi : B \to \mathbb{R}$ zu den Interpolationsstellen (x_j, y_j, z_j), $j = 0(1)N$, den Ansatz

$$(x, y) \mapsto z = \Phi(x, y) = \sum_{j=0}^{N} w_j(x, y)\, z_j. \tag{9.14}$$

Darin sind $w_j(x, y)$ Gewichte mit $w_j(x, y) \geq 0$ und $\sum\limits_{j=0}^{N} w_j(x, y) = 1$.

Die $N+1$ Interpolationsbedingungen

$$z_k = \Phi(x_k, y_k) = \sum_{j=0}^{N} w_j(x_k, y_k)\, z_j, \quad k = 0(1)N,$$

sind erfüllt mit

$$w_j(x_k, y_k) = \begin{cases} 1 \text{ für } k = j \\ 0 \text{ für } k \neq j. \end{cases}$$

Nun müssen die Gewichte $w_j(x, y)$ für eine Stelle (x, y), die keine Stützstelle ist, erklärt werden; es sei also $(x, y) \neq (x_j, y_j)$ für $j = 0(1)N$.

Die Gewichte werden mit dem Abstand

$$r_j(x, y) = \sqrt{(x - x_j)^2 + (y - y_j)^2} \tag{9.15}$$

der Stelle (x, y) von der Stelle (x_j, y_j) und mit einem Parameter μ, $0 < \mu < \infty$, definiert durch

$$\varphi_j(x, y) = \frac{1}{r_j(x, y)}, \tag{9.16}$$

$$w_j(x, y) = \frac{\varphi_j^{\mu}}{\sum\limits_{i=0}^{N} \varphi_i^{\mu}}, \quad j = 0(1)N. \tag{9.17}$$

Damit gelten $w_j(x, y) > 0$ und $\sum\limits_{j=0}^{N} w_j(x, y) = 1$.

Für die Shepard-Funktion (9.14) ergibt sich mit (9.17) die Darstellung

$$\Phi(x, y) = \sum_{j=0}^{N} \varphi_j^{\mu}\, z_j \;\Big/\; \sum_{i=0}^{N} \varphi_i^{\mu}.$$

Mit der Wahl (9.16) bleibt der Einfluss von Stützwerten z_j, deren Stützstelle (x_j, y_j) einen großen Abstand von der Auswertungsstelle (x, y) hat, gering gegenüber solchen Stützwerten, deren Stützstelle nahe bei der Auswertungsstelle liegen.

Der Exponent μ in (9.17) ist frei wählbar. An den Stützstellen (x_j, y_j) hat die Shepard-Funktion Φ für $0 < \mu \leq 1$ Spitzen und für $\mu > 1$ Flachpunkte mit einer zur x, y-Ebene parallelen Tangentialebene; μ wird auch Glättungsparameter genannt.

Der folgende Algorithmus liefert zu einer Auswertungsstelle (x, y) den Funktionswert $\Phi(x, y)$.

Algorithmus 9.20. (*Globale Shepard-Interpolation*)

Gegeben: N+1 Interpolationsstellen (x_j, y_j, z_j), $j = 0(1)N$, der Glättungsparameter μ, $0 < \mu < \infty$, und eine Auswertungsstelle (x, y).

Gesucht: Der Wert $\Phi(x, y)$ der globalen Shepard-Funktion.

1. Setze $Z := 0$, $S := 0$, $j := 0$.

2. Berechne
$$r := \sqrt{(x - x_j)^2 + (y - y_j)^2}.$$

   2.1 Wenn $r > 0$ ist, berechne
   $$w := r^{-\mu}$$
   und setze
   $$\begin{aligned} S &:= S + w, \\ Z &:= Z + w \cdot z_j, \\ j &:= j + 1. \end{aligned}$$

   2.2 Wenn $r = 0$ ist, setze
   $$\begin{aligned} S &:= 1, \\ Z &:= z_j, \\ j &:= N + 1. \end{aligned}$$

3. Wenn $j \leq N$ ist, weiter mit 2.
   Andernfalls berechne
   $$\Phi(x, y) = Z/S.$$

Beispiel 9.21.

Gegeben: Eine Wertetabelle mit 5 Interpolationsstellen

j	0	1	2	3	4
x_j	-2.5	1	2	0	-1
y_j	-0.5	-2	1	0	1.5
z_j	1.5811	2	2	3	2.3979

sowie $\mu = 2$ und die Auswertungsstelle $(x, y) = (-2, 1)$.

Gesucht: Der Wert $\Phi(x, y) = \Phi(-2, 1)$ der globalen Shepard–Funktion.

Lösung: Mit Algorithmus 9.20.

1. $Z := 0$, $S := 0$.

2. $j = 0$:

$$r = \sqrt{\big(-2-(-2.5)\big)^2 + \big(1-(-0,5)\big)^2} = \sqrt{2.5},$$
$$w = \frac{1}{r^2} = \frac{1}{2.5} = 0.4,$$
$$S := S + 0.4 = 0.4,$$
$$Z := Z + 0.4 \cdot 1.5811 = 0.63244.$$

$j = 1$:

$$r = \sqrt{(-2-1)^2 + \big(1-(-2)\big)^2} = \sqrt{18},$$
$$w = \frac{1}{r^2} = \frac{1}{18} = 0.055\,555\,56,$$
$$S := S + \frac{1}{18} = 0.455\,555\,56,$$
$$Z := Z + \frac{1}{18} \cdot 2 = 0.743\,551\,11.$$

$j = 2$:

$$r = \sqrt{(-2-2)^2 + (1-1)^2} = \sqrt{16},$$
$$w = \frac{1}{r^2} = \frac{1}{16} = 0.0625,$$
$$S := S + 0.0625 = 0.518\,055\,56,$$
$$Z := Z + 0.0625 \cdot 2 = 0.868\,551\,11.$$

$j = 3$:

$$r = \sqrt{(-2-0)^2 + (1-0)^2} = \sqrt{5},$$
$$w = \frac{1}{r^2} = \frac{1}{5} = 0.2,$$
$$S := S + 0.2 = 0.718\,055\,56,$$
$$Z := Z + 0.2 \cdot 3 = 1.468\,551\,11.$$

$j = 4$:

$$r = \sqrt{\big(-2-(-1)\big)^2 + (1-1.5)^2} = \sqrt{1.25},$$

$$w = \frac{1}{r^2} = \frac{1}{1.25} = 0.8,$$

$$S := S + 0.2 = 1.518\,055\,56,$$

$$Z := Z + 0.2 \cdot 2.3979 = 3.386\,871\,11.$$

3. $\Phi(-2,1) = \dfrac{Z}{S} = 2.2311.$ □

Für die Berechnung eines jeden Funktionswertes $\Phi(x,y)$ müssen alle $N+1$ Interpolationsstellen (x_j, y_j, z_j), $j = 0(1)N$, verwendet werden. Für eine große Anzahl N erfordert diese globale Methode einen erheblichen Rechenaufwand.

Mittels einer lokalen Variante lässt sich dieser Rechenaufwand stark vermindern. Dabei werden zur Berechnung eines Funktionswertes Φ nur solche Interpolationsstellen (x_j, y_j, z_j) herangezogen, deren Stützstellen (x_j, y_j) innerhalb eines Kreises vom Radius R um die Auswertungsstelle (x, y) liegen. Der Radius R dieses Kreises ist so zu wählen, dass genügend viele Stützstellen innerhalb des Kreises liegen.

Im Folgenden werden die Stützstellen (x_j, y_j), die von der Auswertungsstelle (x, y) einen Abstand $r_j(x,y) < R$ haben und somit innerhalb des Kreises liegen, mit (x_k, y_k), $k = 0(1)M$, $0 < M < N$, bezeichnet; dann ist also $r_k(x,y) < R$.

Bei den folgenden lokalen Methoden erhalten alle Stützstellen (x_j, y_j) mit $r_j(x,y) \geq R$ das Gewicht $w_j(x,y) = 0$. Dann ist

$$\Phi(x,y) = \sum_{k=0}^{M} w_k(x,y)\, z_k. \tag{9.18}$$

Die lokale Shepard-Methode verwendet mit

$$\psi_k(x,y) = \begin{cases} \dfrac{1}{r_k(x,y)} & \text{für} \quad 0 < r_k(x,y) < \dfrac{R}{3} \\ \dfrac{27}{4R} \cdot \left(\dfrac{r_k(x,y)}{R} - 1 \right)^2 & \text{für} \quad \dfrac{R}{3} \leq r_k(x,y) < R \end{cases}$$

die Gewichte

$$w_k(x,y) = \frac{\psi_k^{\mu}}{\sum\limits_{n=0}^{M} \psi_n^{\mu}}, \quad k = 0(1)M\,.$$

Damit ergibt sich die lokale Shepard-Funktion (9.18).

Eine sehr brauchbare lokale Variante der Shepard-Interpolation ergibt sich mit der Verwendung der Franke-Little-Gewichte ([FRAN1982], [HOSC1989], 9.).

Mit

$$\xi_k(x,y) = 1 - \frac{r_k(x,y)}{R} \quad \text{für} \quad 0 < r_k(x,y) < R$$

ergeben sich diese Gewichte

$$w_k(x,y) = \frac{\xi_k^\mu}{\sum\limits_{n=0}^{M} \xi_k^\mu}\,, \quad k = 0(1)M\,,$$

für die lokale Shepard-Funktion (9.18).

Der folgende Algorithmus liefert zu einer Auswertungstabelle (x,y) den Wert der lokalen Shepard-Funktion Φ.

Algorithmus 9.22. (*Lokale Shepard-Interpolation mit Franke-Little-Gewichten*)

Gegeben: $N{+}1$ Interpolationsstellen (x_j, y_j, z_j), $j = 0(1)N$, der Glättungsparameter μ, $0 < \mu < \infty$, der Radius R, $R > 0$, und eine Auswertungsstelle (x,y).

Gesucht: Der Wert $\Phi(x,y)$ der lokalen Shepard-Funktion mit Franke-Little-Gewichten.

1. Setze $Z := 0$, $S := 0$, $j := 0$.

2. Berechne
$$r := \sqrt{(x-x_j)^2 + (y-y_j)^2}.$$

   2.1 Wenn $0 < r < R$ ist, berechne
   $$w := \left(1 - \frac{r}{R}\right)^\mu$$
   und setze
   $$\begin{aligned} S &:= S + w, \\ Z &:= Z + w \cdot z_j, \\ j &:= j + 1. \end{aligned}$$

   2.2 Wenn $r \geq R$ ist, setze
   $$j := j + 1.$$

   2.3 Wenn $r = 0$ ist, setze
   $$\begin{aligned} S &:= 1, \\ Z &:= z_j, \\ j &:= N + 1. \end{aligned}$$

3. Wenn $j \leq N$ ist, weiter mit 2.
   Andernfalls weiter mit 4.

4. Wenn

   4.1 $S > 0$ ist, berechne
   $$z := Z/S;$$
   $\Phi(x, y) = z$ ist der gesuchte Funktionswert.

   4.2 $S = 0$ ist, dann liegen im Kreis um (x, y) mit dem Radius R keine Stützstellen (x_j, y_j); für alle $j = 0(1)N$ ist $r \geq R$. R muss größer gewählt werden.

Beispiel 9.23.

Gegeben: Die in Beispiel 9.21 verwendeten 5 Interpolationsstellen sowie $\mu = 2$, $R = 2$ und die Auswertungsstelle $(x, y) = (-2, 1)$.

Gesucht: Der Wert $\Phi(x, y) = \Phi(-2, 1)$ der lokalen Shepard-Funktion mit Franke-Little-Gewichten.

Lösung: Mit Algorithmus 9.22.
Die Abstände r für $j = 0(1)4$ werden aus dem Beispiel 9.21 übernommen.

1. $Z := 0$, $S := 0$.

2. $j = 0$:

$$r = \sqrt{2.5} < 2,$$
$$w = \left(1 - \frac{\sqrt{2.5}}{2}\right)^2 = 0.043\,861\,17,$$
$$S := S + 0.043\,861\,17 = 0.043\,861\,17,$$
$$Z := Z + 0.043\,861\,17 \cdot 1.5811 = 0.069\,348\,90.$$

$j = 1$:

$$r = \sqrt{18} > 2,$$

$j = 2$:

$$r = \sqrt{16} > 2,$$

$j = 3$:

$$r = \sqrt{5} > 2,$$

$j = 4$:

$$r = \sqrt{1.25} < 2,$$

$$w = \left(1 - \frac{\sqrt{1.25}}{2}\right)^2 = 0.194\,466\,01,$$

$$S := S + 0.194\,466\,01 = 0.238\,327\,18,$$

$$Z := Z + 0.194\,466\,01 \cdot 2.3979 = 0.535\,658\,94.$$

3. $\Phi(-2, 1) = \dfrac{Z}{S} = 2.2476.$

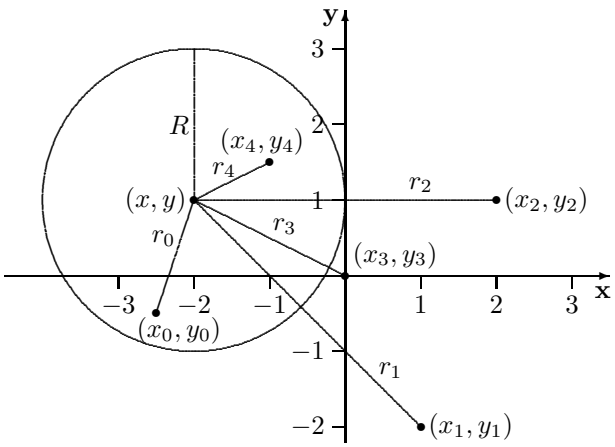


Abb. 9.6.

□

Bei der praktischen Anwendung der lokalen Shepard-Interpolation können sich Schwierigkeiten ergeben bezüglich der Wahl von R bei unterschiedlicher Skalierung auf den beiden Achsen. In [KUHN1990] wird deshalb folgende Variante zur Berechnung der r_j vorgeschlagen.

Als Bereich B, in dem alle Stützstellen (x_j, y_j) liegen, wird mit

$$x_{\min} = \min_{0 \le j \le N} x_j\,, \quad x_{\max} = \max_{0 \le j \le N} x_j\,,$$
$$y_{\min} = \min_{0 \le j \le N} y_j\,, \quad y_{\max} = \max_{0 \le j \le N} y_j\,,$$

das Rechteck

$$B = \left\{ (x, y) \,|\, x_{\min} \le x \le x_{\max},\, y_{\min} \le y \le y_{\max} \right\}$$

gewählt. Mit

$$x_e = x_{\max} - x_{\min}\,, \quad y_e = y_{\max} - y_{\min}$$

sei dann für $(x, y) \in B$

$$r_j(x, y) = \sqrt{\left(\frac{x - x_j}{x_e}\right)^2 + \left(\frac{y - y_j}{y_e}\right)^2}\,. \tag{9.19}$$

Damit gilt $0 \le r_j(x,y) \le \sqrt{2}$. Deshalb muss für die lokale Shepard-Interpolation ein Radius R mit $0 < R < \sqrt{2}$ gewählt werden.

Empfehlungen: Bezüglich des Rechenaufwandes ist die lokale Methode der globalen auf jeden Fall vorzuziehen. Um Spitzen der Funktion Φ an den Stützstellen zu vermeiden, wird empfohlen, einen Glättungsparameter $\mu \ge 2$ zu verwenden. Wenn die Abstände r_j nach (9.19) berechnet werden, ist R mit $0.3 \le R \le 0.5$ empfehlenswert.

Beispiel 9.24.

Gegeben ist eine Funktion T mit

$$\begin{aligned} T(x,y) \quad = \quad & \frac{3}{4}\mathrm{e}^{-\frac{1}{4}\left[(9x-2)^2+(9y-2)^2\right]} + \frac{3}{4}\mathrm{e}^{-\left[\frac{1}{49}(9x+1)^2+\frac{1}{10}(9y+1)\right]} \\ & -\frac{1}{5}\mathrm{e}^{\left[(9x-4)^2+(9y-7)^2\right]} + \frac{1}{2}\mathrm{e}^{-\frac{1}{4}\left[(9x-7)^2+(9y-3)^2\right]}, \\ & \text{wobei } (x,y) \in B = \left\{ (x,y) \,|\, 0 \le x \le 1.1,\, 0 \le y \le 1.1 \right\}. \end{aligned}$$

Der Graph dieser Funktion ist in der folgenden Abbildung wiedergegeben.

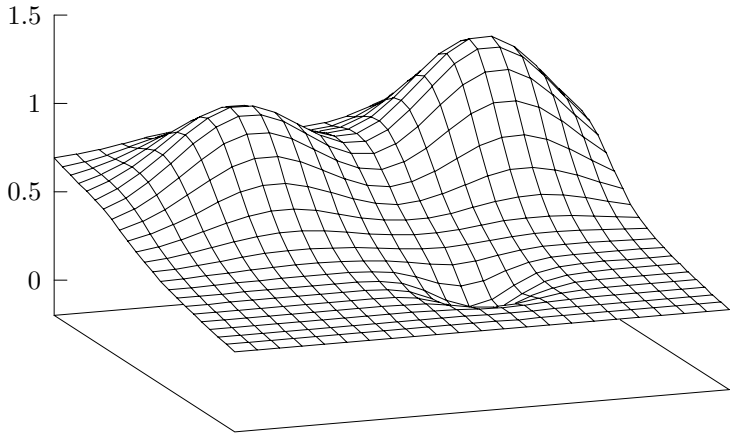


Abb. 9.7. Graph der Funktion T

Mit der Funktion T werden an 100 zufällig erzeugten Stützstellen (x_j, y_j) mit $0 \le x_j \le 1.1$ und $0 \le y_j \le 1.1$ die Interpolationsstellen $\big(x_j,\, y_j,\, z_j = T(x_j,y_j)\big)$, $j = 1(1)100$, bereitgestellt. Mit diesem Datensatz werden

die globale Shepard-Interpolation mit $\mu = 2$ (Abbildung 9.8),

die lokale Shepard-Interpolation mit Franke-Little-Gewichten mit $\mu = 2$ und $R = 0.4$ (Abbildung 9.9) sowie mit

$\mu = 2$ und $R = 0.2$ (Abbildung 9.10) ausgeführt.

Im letzteren Fall erhält man ein akzeptables Ergebnis, wobei mit einigen zusätzlichen Interpolationsstellen an den Rändern ein noch glatteres Erscheinungsbild erzielt werden kann.

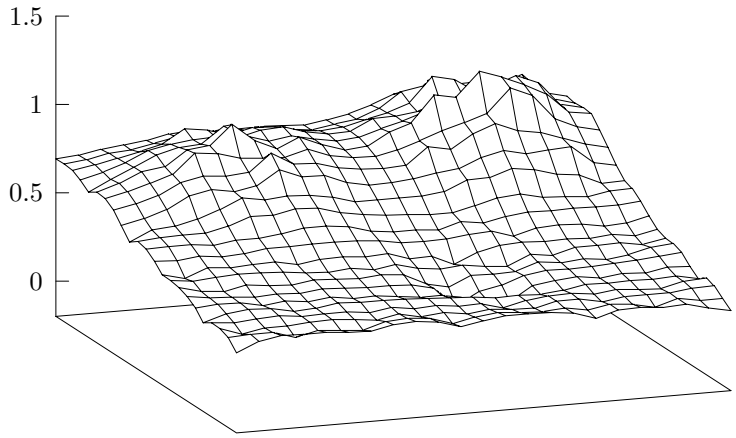


Abb. 9.8. Globale Shepard-Interpolation mit dem Glättungsparameter $\mu = 2$

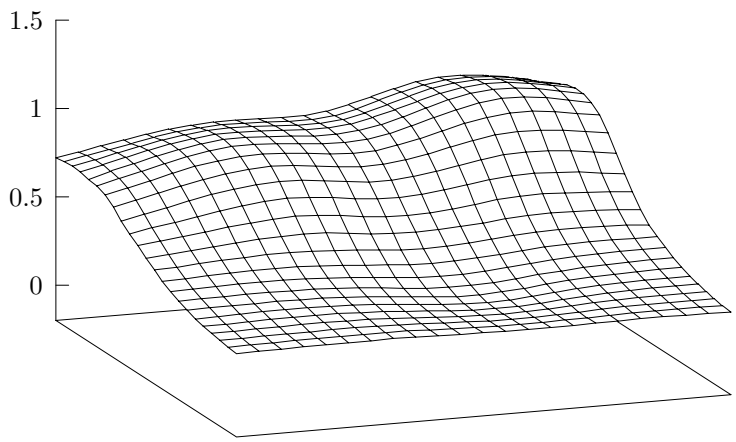


Abb. 9.9. Lokale Shepard-Interpolation mit Franke-Little-Gewichten, Glättungsparameter $\mu = 2$, Radius $R = 0.4$

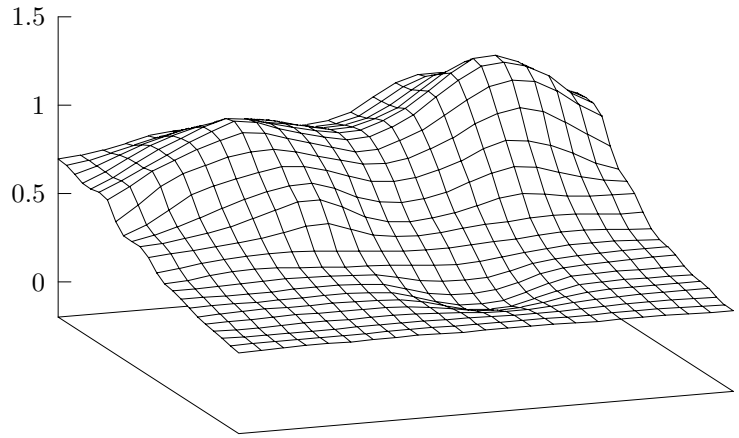


Abb. 9.10. Lokale Shepard-Interpolation mit Franke-Little-Gewichten, Glättungsparameter $\mu = 2$, Radius $R = 0.2$

Im Vergleich mit der Originalfunktion (Abb. 9.7) ergibt sich mit der lokalen Shepard-Interpolation mit Franke-Little-Gewichten ($\mu = 2$, $R = 0.2$) das beste Ergebnis. Die globale Methode (vgl. Abbildung 9.8) ist nicht zu empfehlen. □

9.8 Entscheidungshilfen

Grundsätzlich ist zu bemerken, dass die Interpolation mit Polynomen stark an Bedeutung verloren hat, seit es interpolierende Splines und Subsplines gibt, die nicht die zunehmende Welligkeit bei Polynomen wachsenden Grades zeigen und sich besser an die vorgegebene Wertemenge anpassen lassen. Wenn jedoch Interpolation durch Polynome gefragt ist, so sollte man algebraische Polynome bis höchstens zum fünften Grade verwenden, da Polynome höheren Grades zu stark schwanken.

Bemerkung. (zur Auswahl des geeigneten algebraischen Interpolationspolynoms)
Da es zu $n+1$ Interpolationsstellen (x_i, y_i), $i = 0(1)n$, genau ein Interpolationspolynom vom Grad $\leq n$ gibt, ist es im Grunde gleichgültig, in welcher Darstellung es benutzt wird. Die Darstellung in der Form von Lagrange hat mehr theoretische als praktische Bedeutung, weil bei Hinzunahme einer Interpolationsstelle alle $L_k(x)$ neu berechnet werden müssen; da ist die Newtonsche Interpolationsformel der Lagrangeschen unbedingt vorzuziehen. Interessiert nicht das Interpolationspolynom Φ in allgemeiner Gestalt, sondern nur sein Wert $\Phi(\overline{x})$ an einer (oder wenigen) Stelle(n) $\overline{x}$, so benutzt man zu dessen Berechnung zweckmäßig das Interpolationsschema von Aitken (Abschnitt 9.4). Das Verfahren erlaubt, stufenweise neue Interpolationsstellen hinzuzunehmen; dabei müssen die Stützstellen (wie auch bei den anderen Formeln) nicht monoton angeordnet sein.

Liegt eine umfangreiche Wertetabelle (x_i, y_i), $i = 0(1)n$, vor und will man durch ein Polynom m-ten Grades ($m \leq 5$) interpolieren, so wählt man in der Umgebung einer Stelle $\overline{x} \in [x_0, x_n]$, an der ein Näherungswert $\overline{y}$ für $f(\overline{x})$ gesucht ist, $m{+}1$ benachbarte Stützstellen x_i so aus, dass $\overline{x}$ etwa in der Mitte dieser Stützstellen liegt, weil dann der Interpolationsfehler am kleinsten wird (vgl. Abschnitt 9.6).

Die Anzahl der erforderlichen Multiplikationen bei der Auswertung der Interpolationsformel von Lagrange zu $n{+}1$ Interpolationsstellen ist für

$$\begin{aligned} L_k(x): &\quad n{-}1, \\ L_k(x)\, y_k: &\quad n, \\ \sum_{k=0}^{n} L_k(x)\, y_k: &\quad n\,(n{+}1). \end{aligned}$$

Für die Newtonsche Formel in der Gestalt (9.10) ist diese Anzahl

$$\tfrac{1}{2}\, n\,(n{+}1)\,,$$

und in der „Hornerschen“ Form ist sie n.

Auch wegen dieses Rechenaufwandes ist die Formel von Lagrange für die praktische Verwendung ungeeignet.

Im Falle mehrdimensionaler Interpolation sind Splinemethoden (Kapitel 12) zu empfehlen. Die Shepard-Interpolation (Abschnitt 9.7.2) eignet sich jedoch gut zur graphischen Darstellung von empirischen Funktionen $z = f(x, y)$.

Ergänzende Literatur zu Kapitel 9

[BOOR2001] I; [CARN1990], 1.6, 1.7; [CONT1987], 2, 3.1-3.4, 3.6; [DEUF2002] Bd.1, Kap.7; [ENGE1996], 9; [HAMM1994], 5; [HERM2001], Kap.6; [NIED1987]; [OPFE2002], Kap.3; [PREU2001], Kap.6; [QUAR2002], Kap.8; [RICE1993], Kap.5; [SCHU1992]; [SCHW1997], 3, 3.1-3.5; [STOE1989], 2, 2.1.1, 2.1.3, 2.2; [STUM1982], 3.1, 3.1.1, 3.1.2; [TORN1990] Bd.2, 11.1, 11.2; [UBER1995], Kap.9.

Kapitel 10

Interpolierende Polynom-Splines zur Konstruktion glatter Kurven

10.1 Polynom-Splines dritten Grades

Für die häufig gestellte Aufgabe, durch $n{+}1$ gegebene Punkte ($n \geq 2$)

$$P_0 = (x_0, y_0)\,,\ P_1 = (x_1, y_1)\,, \ldots,\ P_n = (x_n, y_n) \quad \text{mit} \quad x_0 < x_1 < \ldots < x_n$$

in dieser Reihenfolge eine glatte Kurve zu legen, wird ein geübter Zeichner mit Hilfe von Kurvenlinealen meist eine geeignete Lösung anbieten können. Es gibt allerdings viele verschiedene Kurven, die alle durch die gegebenen Punkte gehen; deshalb wird der Aufgabensteller entscheiden müssen, ob er eine ermittelte Kurve als Lösung der Aufgabe akzeptiert.

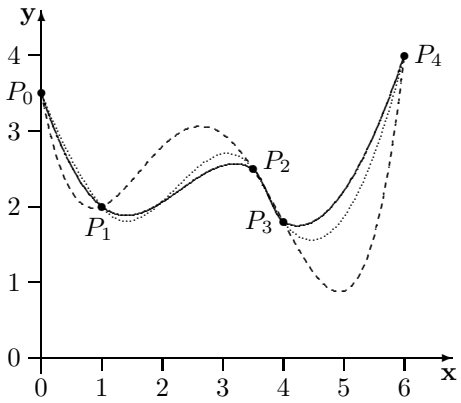


Abb. 10.1. Drei glatte Kurven, die durch fünf gegebene Punkte gehen

Eine rechnerische Lösung der gestellten Aufgabe liefert beispielsweise das Interpolationspolynom vom Höchstgrad n, das durch die gegebenen $n{+}1$ Punkte eindeutig bestimmt ist. Der Graph des Interpolationspolynoms im Intervall $[x_0, x_n]$ ist eine glatte Kurve, die die gegebenen Punkte verbindet.

Die Graphen von Interpolationspolynomen zeigen allerdings bei zunehmender Anzahl von Punkten und damit wachsendem Polynomgrad sowie in Abhängigkeit von der Lage der gegebenen Punkte oft starke Schwankungen und damit deutliche Abweichungen vom Verlauf einer Kurve, die den Erwartungen des Aufgabenstellers entspricht.

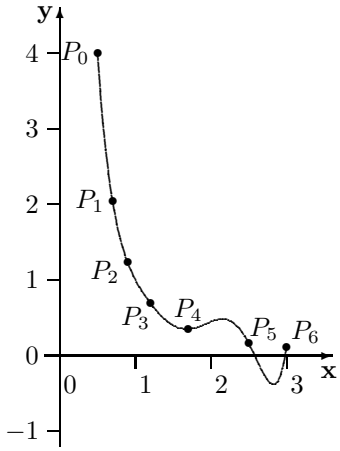


Abb. 10.2. Graph des Interpolationspolynoms zu sieben gegebenen Punkten

Um einen geeigneten Kurvenverlauf zu erzielen und um unerwünschte Effekte (wie die erwähnten starken Schwankungen) zu vermeiden, wird die zu erzeugende Kurve deshalb aus n Segmenten S_0, S_1, …, S_{n-1} zusammengesetzt. Das Segment S_0 gehört zum Teilintervall $[x_0, x_1]$, das Segment S_1 zum Teilintervall $[x_1, x_2]$ und schließlich S_{n-1} zu $[x_{n-1}, x_n]$. Für die Darstellung dieser Segmente werden Polynome niedrigen Grades, in erster Linie kubische Polynome, verwendet, und zwar so, dass insgesamt eine überall glatte, also mindestens einmal stetig differenzierbare Kurve entsteht. Diese Methode wird Spline-Interpolation genannt. Die aus den n Segmenten zusammengesetzte Funktion heißt Splinefunktion, und der Graph dieser Splinefunktion wird auch Splinekurve oder kurz Spline genannt.

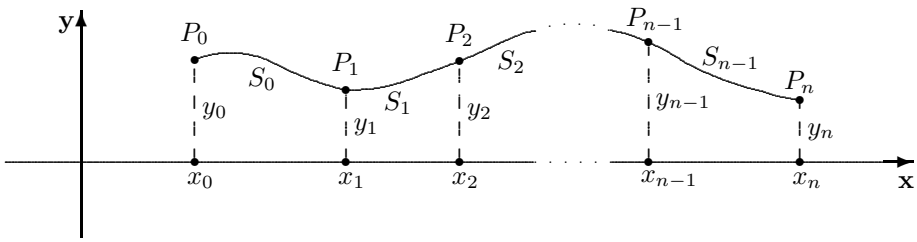


Abb. 10.3. Spline-Kurve mit den Segmenten S_0, S_1, …, S_{n-1} (vgl. Abb. 10.6)

Die verschiedenen Splineverfahren unterscheiden sich durch den gewählten Polynomgrad für die Segmente, die geforderte Differentiationsordnung und weitere Nebenbedingungen (beispielsweise Randbedingungen). Verschiedene Splineverfahren werden für dieselbe In-

terpolationsaufgabe auch verschiedene Lösungskurven anbieten. Dann kann und muss der Aufgabensteller, gestützt auf Erfahrung, eine geeignete Kurve auswählen.

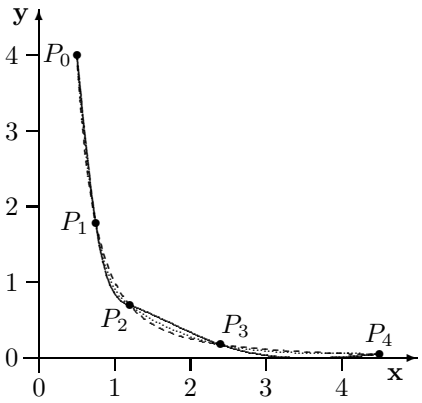


Abb. 10.4. Drei verschiedene Splinekurven als Lösungskurven für dieselbe Interpolationsaufgabe

Beispiel 10.1.

Zu sieben Punkten $P_i = (x_i, 1/x_i^2)$, $i = 0(1)6$, werden das Interpolationspolynom und eine interpolierende Splinefunktion erzeugt.

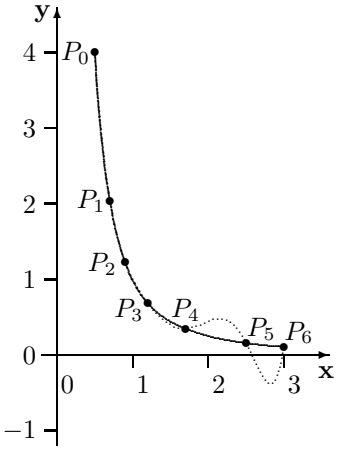


Abb. 10.5. Splinefunktion (durchgezogen) und Interpolationspolynom (gepunktet) zu sieben Punkten

Die Splinefunktion, deren 6 Segmente kubische Polynome sind, liefert für die Interpolationsaufgabe eine bessere Lösung als das Interpolationspolynom vom Grad 6. □

10.1.1 Aufgabenstellung

Gegeben seien $n+1$ Punkte, $n \geq 2$,

$$P_0, P_1, \ldots, P_n\,,$$

$P_i \in \mathbb{R}^m$, $i = 0(1)n$, $m = 2$ oder $m = 3$, also $P_i = (x_i, y_i)$ oder $P_i = (x_i, y_i, z_i)$. Diese Punkte werden auch *Stützpunkte* genannt. Je zwei aufeinander folgende Punkte seien verschieden:

$$P_i \neq P_{i+1} \quad \text{für} \quad i = 0(1)n{-}1\,.$$

Zulässig sind sowohl $P_n \neq P_0$ als auch $P_n = P_0$.

Die Punkte $P_0, \ldots, P_n$ sollen in der vorgegebenen Reihenfolge durch eine glatte Kurve miteinander verbunden werden.

Zur Lösung der gestellten Aufgabe wird die zu erzeugende Kurve aus n Segmenten S_i, $i = 0(1)n{-}1$, zusammengesetzt. Dabei soll das Segment S_i die Punkte P_i und P_{i+1} verbinden.

Damit die aus den Segmenten $S_0, S_1, \ldots, S_{n-1}$ zusammengesetzte Kurve glatt ist, müssen je zwei benachbarte Segmente S_i und S_{i+1} in dem gemeinsamen Punkt P_{i+1} dieselbe Tangente besitzen. Außerdem sollen sie dort dieselbe Krümmung haben.

Es werden nun zwei Fälle unterschieden.

Fall 1: Nichtparametrische, interpolierende, kubische Splinefunktion

In diesem Fall ist $m = 2$, also $P_i = (x_i, y_i)$, und es wird vorausgesetzt, dass die Stützstellen x_i, auch *Knoten* genannt, monoton angeordnet sind:

$$a := x_0 < x_1 < \ldots < x_n =: b\,.$$

Wegen dieser Voraussetzung kann man die zu erzeugende Kurve als Graph einer Funktion S mit $y = S(x)$, $x \in [a, b]$, auffassen. Die Funktion

$$S : x \mapsto y = S(x)\,, \quad x \in [a, b] = I$$

muss an den Stützstellen x_i die Stützwerte $S(x_i) = y_i$ annehmen und soll auf dem Intervall I zweimal stetig differenzierbar sein; es sei also $S \in C^2[a, b]$.

Zur Lösung der gestellten Aufgabe wird für das Segment S_i, das die Punkte P_i und P_{i+1} verbindet, ein spezielles Kurvenstück gewählt, dargestellt durch ein kubisches Polynom

$$S_i(x) := a_i + b_i\,(x - x_i) + c_i\,(x - x_i)^2 + d_i\,(x - x_i)^3\,, \quad x \in [x_i, x_{i+1}]\,;$$
$$a_i, b_i, c_i, d_i \in \mathbb{R}\,, \quad i = 0(1)n{-}1\,.$$

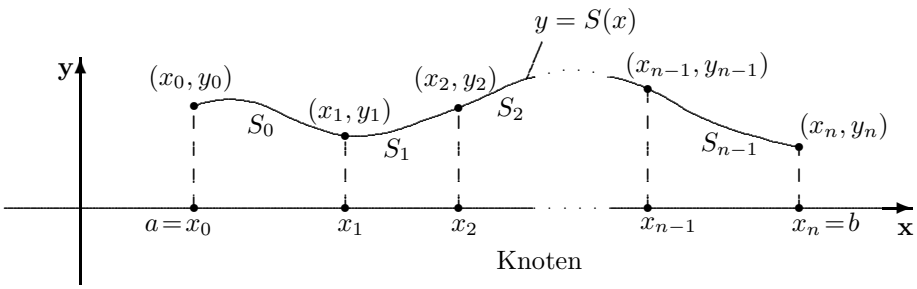


Abb. 10.6. Spline-Interpolation

Die Funktion S setzt sich also stückweise aus n kubischen Polynomen S_i zusammen und wird kubische Splinefunktion S genannt; der Graph von S geht durch die vorgegebenen Stützpunkte.
Das folgende Beispiel zeigt, dass sich mit diesem Ansatz die oben genannten Forderungen erfüllen lassen.

Beispiel 10.2.

Gegeben: Vier Stützpunkte $P_i = (x_i, y_i)$, $i = 0(1)3$, $a = x_0 < x_1 < x_2 < x_3 = b$.

Gesucht: Eine kubische Splinefunktion S, deren Graph durch diese vier Punkte geht.

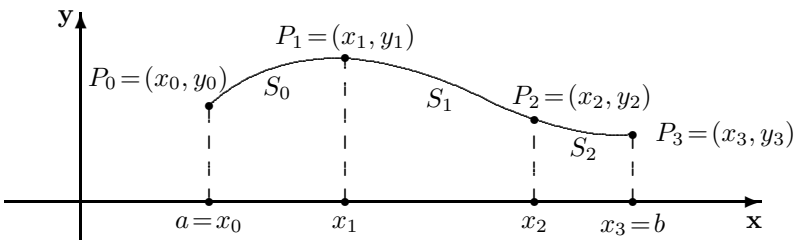


Abb. 10.7.

S setzt sich aus den drei kubischen Polynomen S_0, S_1, S_2, zusammen:

$$\begin{aligned} S_0(x) &= a_0 + b_0\,(x - x_0) + c_0\,(x - x_0)^2 + d_0\,(x - x_0)^3\,, \quad x \in [x_0, x_1]\,, \\ S_1(x) &= a_1 + b_1\,(x - x_1) + c_1\,(x - x_1)^2 + d_1\,(x - x_1)^3\,, \quad x \in [x_1, x_2]\,, \\ S_2(x) &= a_2 + b_2\,(x - x_2) + c_2\,(x - x_2)^2 + d_2\,(x - x_2)^3\,, \quad x \in [x_2, x_3]\,. \end{aligned}$$

Es sind also $3{\cdot}4 = 12$ Koeffizienten zu berechnen. Dafür sind 12 Bedingungen erforderlich. Mit den vier gegebenen Stützpunkten $P_i = (x_i, y_i)$ ergeben sich die vier Interpolationsbedingungen $S(x_i) = y_i$, $i = 0(1)3$:

$$
\begin{aligned}
&(1) && S_0(x_0) &&= && y_0\\
&(2) && S_1(x_1) &&= && y_1\\
&(3) && S_2(x_2) &&= && y_2\\
&(4) && S_2(x_3) &&= && y_3
\end{aligned}
$$

Wegen der Forderung $S \in C^2[x_0, x_3]$ müssen je zwei aufeinander folgende kubische Polynome zweimal stetig differenzierbar aneinander schließen; deshalb ergeben sich an den inneren Knoten x_1 und x_2 je drei Anschlussbedingungen

$$
\begin{aligned}
&(5) && S_0(x_1) &&= && S_1(x_1)\\
&(6) && S_0'(x_1) &&= && S_1'(x_1)\\
&(7) && S_0''(x_1) &&= && S_1''(x_1)\\
&(8) && S_1(x_2) &&= && S_2(x_2)\\
&(9) && S_1'(x_2) &&= && S_2'(x_2)\\
&(10) && S_1''(x_2) &&= && S_2''(x_2)
\end{aligned}
$$

Die Bedingungen (5) und (8) bedeuten, dass das Segment S_0 auch durch den Punkt $P_1 = (x_1, y_1)$ und das Segment S_1 auch durch den Punkt $P_2 = (x_2, y_2)$ gehen muss, dass also S_0 und S_1 bzw. S_1 und S_2 an den Anschlussknoten x_1 bzw. x_2 übereinstimmen. Man beachte, dass mit (6) und (9) sowie (7) und (10) gefordert wird, dass dort auch die ersten und zweiten Ableitungen benachbarter Polynome gleich und somit die Ableitungen S' und S'' der Splinefunktion S stetig sind. Auf die *Werte* der Ableitungen an den Knoten x_1 und x_2 hat man aber keinen Einfluss. Im Allgemeinen gelten $S_0'''(x_1) \neq S_1'''(x_1)$ und $S_1'''(x_2) \neq S_2'''(x_2)$. S ist also nur zweimal stetig differenzierbar.

Bisher hat man nur 10 Bedingungen für 12 Koeffizienten. Wenn zwei zusätzliche Bedingungen formuliert werden, hat man die gesuchten 12 Bedingungen. Zum Beispiel kann man als *Randbedingungen* die 1. Ableitungen von S in x_0 und x_3 vorschreiben:

$$
\begin{aligned}
&(11) && S'(x_0) = S_0'(x_0) &&= && y_0'\\
&(12) && S'(x_3) = S_2'(x_3) &&= && y_3'
\end{aligned}
$$

Auf diese Weise erhält man eine „kubische Splinefunktion mit vorgegebenen 1. Randableitungen“. Möchte man dagegen die Splinekurve links von x_0 und rechts von x_3 geradlinig fortsetzen, so muss man

$$
\begin{aligned}
&(11) && S''(x_0) = S_0''(x_0) &&= && 0\\
&(12) && S''(x_3) = S_2''(x_3) &&= && 0
\end{aligned}
$$

fordern, damit bei x_0 und x_3 die Krümmung von S verschwindet. Mit diesen Forderungen ergibt sich die „natürliche kubische Splinefunktion“. Zur Berechnung der Koeffizienten siehe die Abschnitte 10.1.4 und 10.1.5. □

Bemerkung 10.3. Wenn außer den bisher genannten noch weitere Forderungen an die Splinefunktion S gestellt werden sollen, können für einzelne oder für alle Segmente S_i auch Polynome höheren Grades angesetzt werden. Wichtig ist, dass die Anzahl der Koeffizienten mit der Anzahl der Bedingungen übereinstimmt.
Wird z. B. eine Splinefunktion S konstruiert, bei der $S_1, \ldots, S_{n-2}$ $(n \geq 3)$ kubische Polynome sind, S_0 und S_{n-1} dagegen Polynome 4. Grades, dann muss man für die beiden zusätzlichen Koeffizienten von S_0 und S_{n-1} auch zwei zusätzliche Randbedingungen aufstellen.
Im Beispiel 10.2 wären dann S_0 und S_2 Polynome 4. Grades, und man könnte z. B. die Werte $S_0'(x_0)$, $S_0''(x_0)$, $S_2'(x_3)$, $S_2''(x_3)$ vorschreiben, um 4 Randbedingungen aufzustellen (siehe Abschnitt 10.1.7).

Fall 2: Parametrischer, interpolierender, kubischer Spline

Jetzt wird der allgemeine Fall $m = 2$ und $m = 3$ behandelt. Dann sind für $m = 2$ die Stützstellen x_i im Allgemeinen nicht monoton angeordnet, wie z. B. bei einer geschlossenen Kurve mit $P_n = P_0$ (monoton angeordnete Stützstellen sind zulässig, doch wird von dieser Eigenschaft kein Gebrauch gemacht).

Eine ebene Kurve, die nicht als Graph einer Funktion aufgefasst werden kann, muss mit Hilfe eines Parameters dargestellt werden.

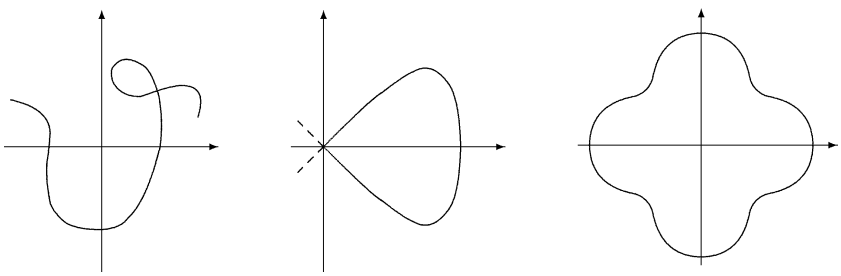


Abb. 10.8. Ebene Kurven, die nicht Graph einer Funktion sind

Beispielsweise besitzt der Einheitskreis die Darstellung mit dem Parameter t:

$$\boldsymbol{x}(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} \cos t \\ \sin t \end{pmatrix}, \quad t \in [0, 2\pi]$$

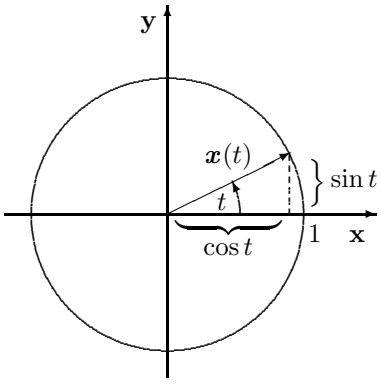


Abb. 10.9.

Durch je zwei aufeinander folgende Stützpunkte P_i und P_{i+1} soll ein Kurvensegment S_i mit der Parameterdarstellung

$$\begin{aligned}&\boldsymbol{S}_i(t) := \boldsymbol{a}_i + \boldsymbol{b}_i\,(t-t_i) + \boldsymbol{c}_i\,(t-t_i)^2 + \boldsymbol{d}_i\,(t-t_i)^3\,, \quad t \in [t_i, t_{i+1}]\,;\\ &\boldsymbol{a}_i, \boldsymbol{b}_i, \boldsymbol{c}_i, \boldsymbol{d}_i \in \mathbb{R}^m\,, \quad i = 0(1)n{-}1\end{aligned}$$

gehen. Die den Stützpunkten P_i zugeordneten Parameterwerte t_i, die Knoten, müssen monoton angeordnet sein:

$$a := t_0 < t_1 < \ldots < t_n =: b\,.$$

Dann hat der aus den n Segmenten S_i zusammengesetzte kubische Spline die Darstellung

$$\boldsymbol{S} : [a,b] \to \mathbb{R}^m\,, \quad t \mapsto \boldsymbol{x}(t) = \boldsymbol{S}(t) \equiv \boldsymbol{S}_i(t)\,, \quad t \in [a,b] = I_t\,.$$

Für die Parameterwerte $t_0, t_1, \ldots, t_n$ soll $\boldsymbol{S}$ mit den Stützpunkten übereinstimmen,

$$\boldsymbol{S}(t_i) = \boldsymbol{P}_i\,, \quad i = 0(1)n\,,$$

und die Vektorfunktion $\boldsymbol{S}$ soll auf dem Intervall I_t zweimal stetig differenzierbar sein. Zur Wahl der t_i siehe Abschnitt 10.1.6.

Für $m = 2$ erscheinen die skalaren Vektorkomponenten in der Schreibweise

$$\boldsymbol{x}(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \boldsymbol{S}(t) = \begin{pmatrix} S_x(t) \\ S_y(t) \end{pmatrix} \equiv \begin{pmatrix} S_{ix}(t) \\ S_{iy}(t) \end{pmatrix} = \boldsymbol{S}_i(t)\,, \tag{10.1}$$

worin z. B.

$$\begin{aligned} S_{ix}(t) \quad = \quad & a_{ix} + b_{ix}\,(t-t_i) + c_{ix}\,(t-t_i)^2 + d_{ix}\,(t-t_i)^3\,, \\ & t \in [t_i, t_{i+1}]\,, \quad a_{ix}, b_{ix}, c_{ix}, d_{ix} \in \mathbb{R} \end{aligned}$$

eine interpolierende kubische Splinefunktion ist. Analog gilt für $m = 3$

$$\boldsymbol{x}(t) = \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix} = \boldsymbol{S}(t) = \begin{pmatrix} S_x(t) \\ S_y(t) \\ S_z(t) \end{pmatrix} \equiv \begin{pmatrix} S_{ix}(t) \\ S_{iy}(t) \\ S_{iz}(t) \end{pmatrix} = \boldsymbol{S}_i(t)\,. \tag{10.2}$$

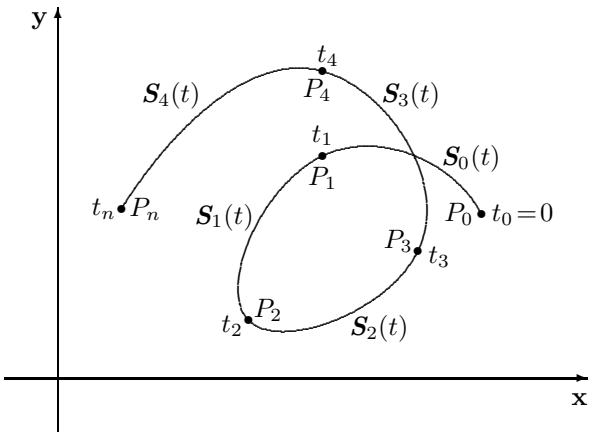


Abb. 10.10. Ebene Splinekurve ($n = 5$) mit den Segmenten S_0, S_1, ..., S_4

10.1.2 Woher kommen Splines? Mathematische Analyse

Bei der Interpolation mit Hilfe algebraischer Polynome (Kapitel 9) wurde durch $n+1$ Punkte genau ein Interpolationspolynom vom Höchstgrad n gelegt. Die Polynome schwanken jedoch stark mit wachsendem n in Abhängigkeit von der Lage der Punkte (vgl. Abb. 10.2). In der Praxis werden deshalb solche Interpolationspolynome nur auf wenige Punkte mit aufeinander folgenden Abszissen beschränkt, und für jeden Abszissenbereich werden neue niedriggradige Polynome benutzt. Das Verfahren mit dem kleinstmöglichen Polynomgrad 1, angewandt auf jeweils zwei Punkte, liefert einen Polygonzug. Hier ist zwar die Schwankung der interpolierenden Funktion minimal, aber die Unstetigkeit in den Knoten setzt schon bei der ersten Ableitung ein, und die Kurve ist nicht glatt.

Deshalb ist die Zusammensetzung niedriggradiger (und daher schwach schwankender) Polynome zu einer im ganzen Intervall $[a, b]$ möglichst oft differenzierbaren Funktion zweckmäßig; sie stellt einen Kompromiss zwischen Polygonzug und Interpolationspolynom höheren Grades dar. Diese Methode heißt *Spline-Interpolation*. Sie ist entstanden in Analogie zum manuellen Strakverfahren: Beim Straken wird ein altes, aus dem Schiffbau kommendes Werkzeug, die so genannte Straklatte (im Englischen *Spline*) benutzt, um beispielsweise die Längs- und Querversteifungen für die Beplankung eines Schiffsrumpfes zu zeichnen. Die Straklatte ist ein schlankes, biegsames Kurvenlineal konstanter Elastizität aus Holz, Metall oder Kunststoff, das verwendet wird, um durch eine Reihe vorgegebener Punkte eine möglichst glatte Kurve zu legen. Durch Anbringen von Gewichten, die senkrecht zur Straklatte wirken und keine Komponente in ihrer Richtung haben, wird die Straklatte fixiert und zur Erfüllung der Forderung minimaler Biegungsenergie entspannt.

Mit welcher Funktionenklasse lässt sich dieses Strakverfahren analytisch beschreiben? Man betrachtet dazu einen horizontalen schlanken Balken konstanter Elastizität und konstanten Querschnitts, der an den Enden drehbar gelagert ist.

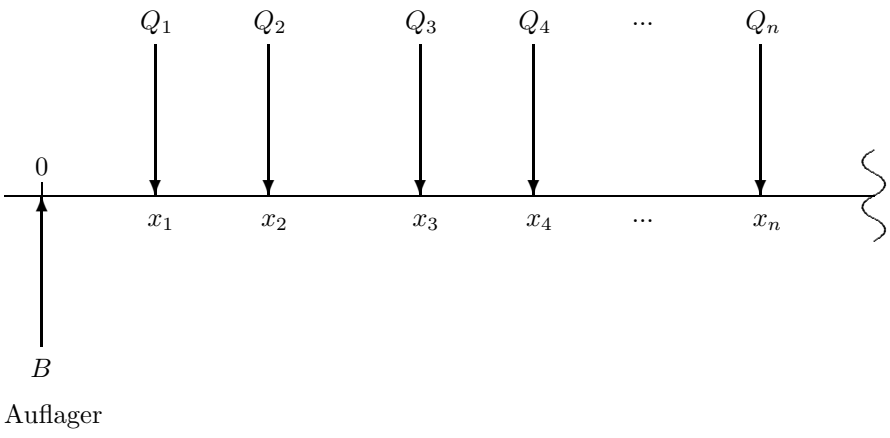


Abb. 10.11. Strakverfahren

Wenn vertikale Kräfte Q_1 bis Q_n an aufeinander folgenden Positionen x_1 bis x_n angreifen, setzt sich die Biegelinie $y = y(x)$ des Balkens stückweise aus kubischen Polynomen zusammen.

Die Differentialgleichung der Biegelinie $y = y(x)$ lautet

$$(*) \qquad EIy''(x) = -M(x)\,, \qquad x \geq 0\,;$$

dabei sind E die Elastizitätskonstante, I das Trägheitsmoment des Querschnitts bezüglich der Nulllinie und $M(x)$ das an der Stelle x wirkende Drehmoment.

Für das Drehmoment an der Stelle x ergibt sich die stetige Funktion

$$M(x) = -Bx + \sum_{i=1}^{n} Q_i\,(x - x_i)_+\,, \qquad x \geq 0\,, \text{ mit}$$

$$(**) \qquad (x - x_i)_+ = \begin{cases} 0 & \text{für} \quad x \leq x_i \\ x - x_i & \text{für} \quad x > x_i\,. \end{cases}$$

Damit lautet die Differentialgleichung $(*)$

$$EIy''(x) \quad = \quad Bx - \sum_{i=1}^{n} Q_i\,(x - x_i)_+\,.$$

Zweimalige Integration liefert

$$EIy'(x) \quad = \quad B\frac{x^2}{2} - \frac{1}{2}\sum_{i=1}^{n} Q_i\,(x - x_i)_+^2 + \overline{C}_1\,,$$

$$EIy(x) \quad = \quad B\frac{x^3}{2\cdot 3} - \frac{1}{2\cdot 3}\sum_{i=1}^{n} Q_i\,(x - x_i)_+^3 + \overline{C}_1 x + \overline{C}_0\,.$$

Die Biegelinie

$$y(x) \quad = \quad \frac{B}{6EI}\,x^3 + C_1 x + C_0 - \sum_{i=1}^{n} \frac{Q_i}{6EI}\,(x - x_i)_+^3$$

ist wegen (**) für jedes Intervall $[x_i, x_{i+1}]$ ein anderes kubisches Polynom, aber insgesamt eine zweimal stetig differenzierbare Funktion.

So entstand der Grundgedanke für die Konstruktion von Splinefunktionen, nämlich niedriggradige Polynome intervallweise zu einer glatten (zweimal stetig differenzierbaren) interpolierenden Gesamtfunktion zusammenzuschließen.

10.1.3 Anwendungsbeispiele

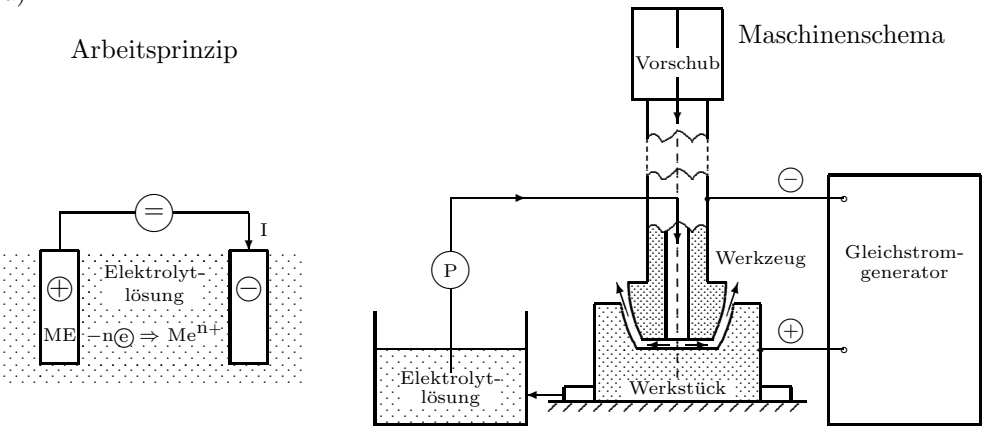
Überall dort, wo es gilt, durch vorgegebene Punkte in der Ebene oder im Raum eine glatte Kurve mit einer analytischen Darstellung zu bestimmen, werden Splines eingesetzt. Es ist kein CAD-Softwarepaket vorstellbar, in welchem nicht Splines die Berechnungsgrundlage bilden. Aber auch überall dort, wo Messwerte analytisch dargestellt werden sollen, werden Splines eingesetzt. Einige Beispiele seien in der Folge genannt.

1. *Das elektrochemische Senken*

Das elektrochemische Senken ist ein abbildendes Formgebungsverfahren. Bei ihm geht es darum, zeilen- und schichtweise Werkstoff abzutragen oder abzutrennen, um Werkzeuge bestimmter geometrischer Form zu erzeugen, die auf mechanischem Wege nicht hergestellt werden könnten. Das Abbildungsprinzip beruht auf der anodischen Auflösung von Metallen. In einer Elektrolytlösung befinden sich zwei metallische Elektroden. Nach Anlegen einer Gleichspannung fließt Strom, wobei aufgrund von Ladungsaustauschvorgängen das als Anode geschaltete Metall in die Lösung geht. An der Kathode (Werkzeugelektrode) findet kein Abtrag statt. Zwischen beiden Elektroden bildet sich ein Arbeitsspalt aus. Berechnet werden soll die Spaltausbildung bei Bearbeitungsende. Dazu wird eine exakte, formelmäßige Wiedergabe der Werkstückkontur in gewissen Zeitintervallen benötigt. Man muss nämlich in einzelnen Punkten in Richtung der Normalen zur Werkstückkontur den örtlichen Abtrag pro Zeiteinheit antragen können, wodurch sich eine neue Stufe der Spaltausbildung ergibt.

So fortfahrend lässt sich der Arbeitsspalt bei Bearbeitungsende berechnen. Man arbeitet hier am sinnvollsten mit Splines. Untersuchungen dazu wurden im Werkzeugmaschinenlabor der RWTH Aachen durchgeführt. Maschinenschema, Arbeitsprinzip und eine Skizze zur ersten Werkstückkontur nach einer Minute Abtrag sind den folgenden Bildern zu entnehmen (vgl. W. König: Fertigungsverfahren, Band 3, Düsseldorf 1979).

a)



Arbeitsmedium:
wässrige Elektrolytlösung,
z. B. NaCl, $NaNO_3$

Werkzeug:
abbildendes Formwerkzeug,
kein Verschleiß

b)

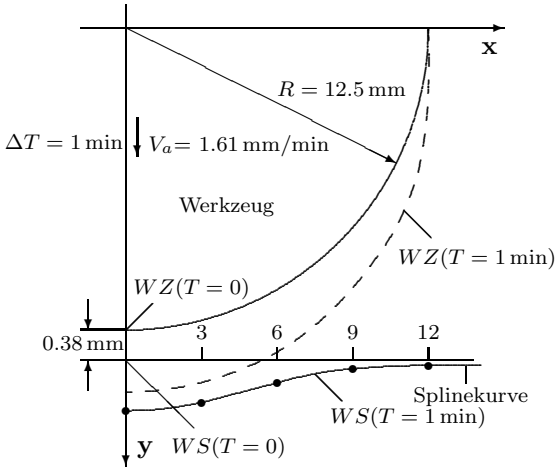


Abb. 10.12a,b.

Abbildung 10.12 b zeigt eine Skizze zur ersten Werkstückkontur unter den Versuchsbedingungen:

V_{sp}	=	2.170 mm^3/A min	=	spezifisches Abtragsvolumen
U	=	17.5 V	=	Arbeitsspannung
κ	=	0.167 s/cm	=	spezifische Leitfähigkeit
V_a	=	1.61 mm/min	=	Abtragsgeschwindigkeit
A_{sp}	=	0.38 mm	=	Anfangsstirnspalt
R	=	12.5 mm	=	Werkzeugradius.

□

2. *Fahrzeugrückleuchte*

Für die Konstruktion der Dichtfläche einer Fahrzeugrückleuchte stehen dem Konstrukteur die Formlinienpläne (Horizontal- und Vertikalschnitte des Prototyps in bestimmten Abständen) der entsprechenden Fahrzeugpartie zur Verfügung. Um die Kontur der Dichtfläche zu entwerfen, benötigt man weitere Horizontal- und Vertikalschnitte, die sich mit Hilfe mathematischer Modelle ermitteln lassen. Dazu werden zunächst die graphisch bekannten Formlinien digitalisiert, um sie analytisch z. B. durch kubische Splines in Ebenen parallel zur xz- bzw. yz-Ebene beschreiben zu können. □

3. *Spritzgießen*

Zur optimalen Steuerung des Spritzgießprozesses ist die Kenntnis des Werkstoffverhaltens der zu verarbeitenden Thermoplaste während der Abkühlung eine der Grundvoraussetzungen. Zu diesem Zweck werden bei jeweils konstantem Druck p und konstanter Abkühlgeschwindigkeit c Messreihen für das spezifische Volumen v in Abhängigkeit von der Temperatur T aufgenommen. Um das vollständige pvT- Diagramm $\big(\text{d. h. } v = f(p,T)\big)$ erstellen zu können, wird im ersten Schritt des Rechenganges jede einzelne Messreihe für $p_j = \text{const.}$, $j = 0(1)m$, durch Ersatzfunktionen dargestellt. Dabei hat es sich als zweckmäßig erwiesen, den Feststoff- und Übergangsbereich durch einen Ausgleichsspline (s. Abschnitt 10.3), den Schmelzbereich durch eine Ausgleichsgerade zu beschreiben. Da die ebenfalls zu ermittelnde Schmelztemperatur in der Regel nicht mit einem Messwert zusammenfällt, wird der Spline über den letzten Messwert hinaus linear verlängert und mit der Ausgleichsgerade zum Schnitt gebracht.

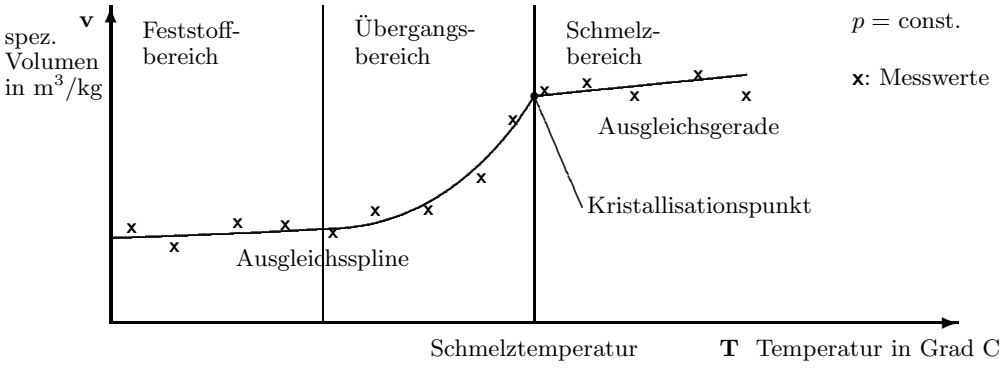


Abb. 10.13. Qualitative Darstellung des Sachverhaltes

4. *Radialventilator*

Für einen Radialventilator mit rückwärts gekrümmten Schaufeln wurden durch Messungen die dimensionslosen Kennzahlen ϕ (Lieferzahl) und ψ (Druckzahl) ermittelt, mit deren Hilfe sich die vom Ventilator erzeugte Druckdifferenz ΔP_t in Abhängigkeit von der Laufdrehzahl n und dem Volumenstrom $\dot{V}$ errechnen lässt. Um für einen definierten Drehzahl- und Volumenstrombereich die Druckdifferenzen in tabellarischer Form zu ermitteln, muss zunächst eine Ersatzfunktion $\psi(\phi)$ mit Hilfe einer interpolierenden Splinefunktion aufgestellt werden. □

5. *Prandtl-Rohr*

Im unten dargestellten Versuchsaufbau werden hinter dem Axialventilator im angeschlossenen Rohr (DIN 400) mit Hilfe eines Prandtl-Rohrs die dynamischen Drücke $\Delta P_{dyn,i}$ in Abhängigkeit von dem Radius r_i an 9 Stellen gemessen. Durch einfache Umrechnung ergeben sich daraus die lokalen Strömungsgeschwindigkeiten v_i. Um die für die anschließende Messauswertung erforderlichen Größen „Volumenstrom“ sowie „mittlere Strömungsgeschwindigkeit“ zu ermitteln, soll die Ersatzfunktion $\Phi = v(r)$ durch Splineinterpolation aufgestellt werden.

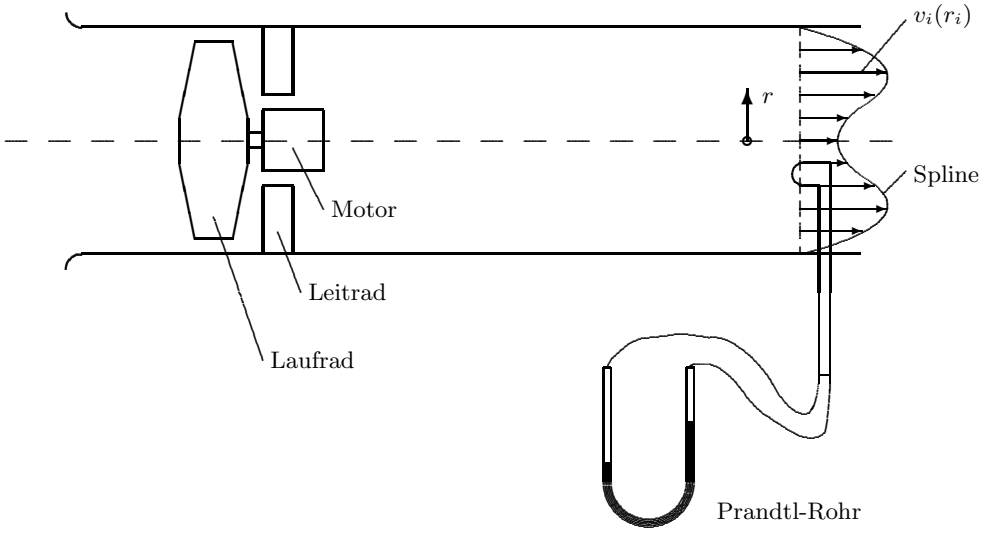


Abb. 10.14.

□

6. *Walzenkörper*

Wenn bei Walzenkörpern mit beliebig komplexer Profilform die analytische Darstellung der Kontur, die Querschnittsfläche, das Volumen, das Gewicht, die Mantelfläche, das Trägheitsmoment und die Schwerpunktskoordinaten zu ermitteln sind, wird man Splineverfahren für die Kontur und Quadraturverfahren einsetzen oder die Splinefunktion direkt integrieren. Beispiele für Profile von Walzenkörpern sind in den folgenden Abbildungen angegeben.

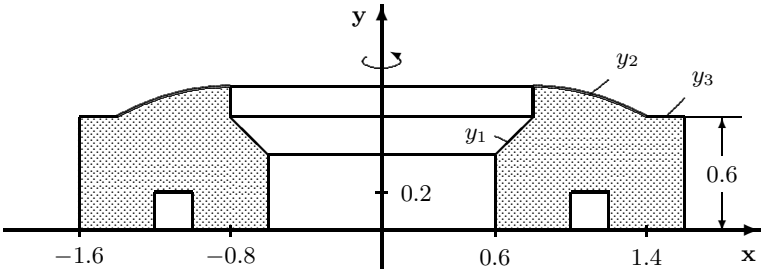


Abb. 10.15.

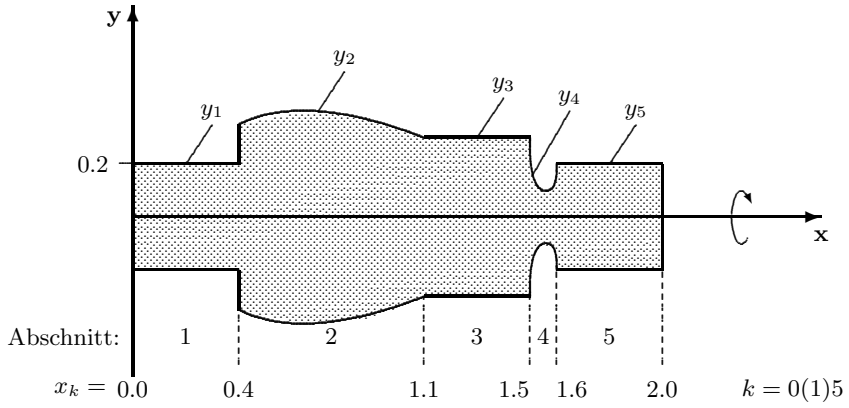


Abb. 10.16.

□

10.1.4 Definition verschiedener Arten nichtparametrischer kubischer Splinefunktionen

Definition 10.4.
Gegeben seien $n+1$ Punkte $P_i = (x_i, y_i)$, $i = 0(1)n$, $n \geq 2$, mit monoton angeordneten Knoten $x_i : a = x_0 < x_1 < \ldots < x_n = b$. Gesucht ist eine nichtparametrische Splinefunktion S, die wie folgt definiert wird:

(1) S ist in jedem Intervall $[x_i, x_{i+1}]$ für $i = 0(1)n-1$ durch ein kubisches Polynom S_i gegeben

$$S(x) \equiv S_i(x) := a_i + b_i\,(x - x_i) + c_i\,(x - x_i)^2 + d_i\,(x - x_i)^3\,, \quad x \in [x_i, x_{i+1}]\,. \tag{10.3}$$

(2) S erfüllt die $n+1$ Interpolationsbedingungen $S(x_i) = y_i$ für $i = 0(1)n$.

(3) S ist in $I = [a, b]$ zweimal stetig differenzierbar, d. h. $S \in C^2[a, b]$.

(4) S erfüllt eine der folgenden sechs zusätzlichen Randbedingungen

(i) für die *natürliche kubische Splinefunktion*

$$S''(x_0) = 0\,, \qquad S''(x_n) = 0$$

(ii) für die *kubische Splinefunktion mit vorgegebenen zweiten Randableitungen* (auch verallgemeinerte natürliche kubische Splinefunktion genannt)

$$S''(x_0) = y_0''\,, \qquad S''(x_n) = y_n''$$

(iii) für die *kubische Splinefunktion mit not-a-knot-Randbedingungen*

$$S_0'''(x_1) = S_1'''(x_1)\,, \qquad S_{n-2}'''(x_{n-1}) = S_{n-1}'''(x_{n-1})$$

Diese Bedingungen besagen, dass die 3. Ableitung der Splinefunktion S in den Knoten x_1 und x_{n-1} stetig ist. Damit sind x_1 und x_{n-1} keine „echten“ Knoten der Splinefunktion („not a knot“). Hier muss $n \geq 3$ sein.

(iv) für die *kubische Splinefunktion mit vorgegebenen ersten Randableitungen*

$$S'(x_0) = y_0'\,, \qquad S'(x_n) = y_n'$$

(v) für die *kubische Splinefunktion mit vorgegebenen dritten Randableitungen*[1]

$$S'''(x_0) = y_0'''\,, \qquad S'''(x_n) = y_n'''$$

(vi) für die *periodische kubische Splinefunktion* mit der Periode $p = x_n - x_0$ unter der Voraussetzung, dass für die vorgegebenen Punkte $P_0 = (x_0, y_0)$ und $P_n = (x_n, y_n)$ gilt $y_0 = y_n$ und damit $S(x_0) = S(x_n)$,

$$S'(x_0) = S'(x_n)\,, \qquad S''(x_0) = S''(x_n)$$

[1] Für die praktische Anwendung hat die Vorgabe der dritten Ableitungen für die nur zweimal stetig differenzierbare Funktion S wenig Bedeutung.

Bemerkung. Der 4-parametrige Ansatz in Formel (10.3) ergibt sich aus der Forderung, dass S zweimal stetig differenzierbar sein soll. Polynom-Splines dritten Grades (zweimal stetig differenzierbar, 4-parametrig) gehören zur Klasse der Splinefunktionen von ungeradem Grad $2k-1$ (k-mal stetig differenzierbar, $2k$-parametrig). In besonderen Fällen benutzt man auch Splinefunktionen von geradem Grad $2k$, z. B. zum flächentreuen Ausgleich von Histogrammen oder empirischen Häufigkeitsverteilungen (siehe [SPAT1986]).

Die Eigenschaften (2), (3) und (4) aus Definition 10.4 führen auf die Bedingungen zur Berechnung der Koeffizienten der n kubischen Polynome S_i von S mit je vier Koeffizienten a_i, b_i, c_i, d_i; man benötigt insgesamt $4n$ Bedingungen. Diese ergeben sich aus den $n+1$ *Interpolationsbedingungen* in den Stützpunkten P_i (Eigenschaft (2))

$$S(x_i) = y_i\,, \qquad i = 0(1)n\,, \tag{10.4}$$

den je drei *Anschlussbedingungen* in den $n-1$ inneren Stützpunkten P_1 bis P_{n-1}, in denen je zwei kubische Polynome zweimal stetig differenzierbar aneinander schließen (Eigenschaft (3))

$$\left.\begin{aligned} S_{i-1}(x_i) &= S_i(x_i) && (10.5)\\ S'_{i-1}(x_i) &= S'_i(x_i) && (10.6)\\ S''_{i-1}(x_i) &= S''_i(x_i) && (10.7) \end{aligned}\right\}\,, \quad i = 1(1)n{-}1$$

und den beiden *Randbedingungen*, je nach dem Typ der Splinefunktion (Eigenschaft (4)). Insgesamt sind dies

$$n+1 \quad + \quad 3\,(n-1) \quad + \quad 2 \quad = \quad 4n$$

Bedingungen für die $4n$ Koeffizienten.

Die Eigenschaft (3), aus der sich die Bedingungen (10.6) und (10.7) ergeben, stellt die stärkste Forderung an die Splinefunktion S dar. Sie bewirkt den glatten Anschluss der Polynome S_{i-1} und S_i an dem Knoten x_i; dort haben die Graphen der benachbarten Polynome S_{i-1} und S_i dieselbe Tangente und darüber hinaus die gleiche Krümmung. Diese Eigenschaft macht die Splinefunktion S besonders geeignet zur angenäherten Darstellung einer Funktion f, über deren Verlauf man empirisch (z. B. durch Messungen) Informationen besitzt und von der bekannt ist, dass ihr Graph sich gut mit Hilfe eines biegsamen Kurvenlineals (Spline) zeichnen lässt. Die Krümmung κ_i von S_i ist gegeben durch

$$\kappa_i(x) = \frac{S''_i(x)}{\left(1 + {S'_i}^2(x)\right)^{3/2}}\,.$$

Wegen (10.6) und (10.7) gilt an dem Knoten x_i

$$\kappa_{i-1}(x_i) = \frac{S''_{i-1}(x_i)}{\left(1 + {S'_{i-1}}^2(x_i)\right)^{3/2}} = \frac{S''_i(x_i)}{\left(1 + {S'_i}^2(x_i)\right)^{3/2}} = \kappa_i(x_i)\,,$$

d. h. die Polynome S_{i-1} und S_i haben an dem Knoten x_i, $i = 1(1)n{-}1$, die gleiche Krümmung. S''' ist an den Knoten $x_1, ..., x_{n-1}$ im Allgemeinen unstetig.

Beispiel 10.5.

Gegeben: Die Wertetabelle $\big(i,\, x_i,\, y_i\big)$, $i = 0, 1, 2$

i	0	1	2
x_i	0	1	3
y_i	1	2	1

Gesucht: Zu der gegebenen Wertetabelle drei interpolierende kubische Splinefunktionen S der Darstellung

$$S(x) \equiv S_i(x) = a_i + b_i\,(x - x_i) + c_i\,(x - x_i)^2 + d_i\,(x - x_i)^3$$

für $x \in [x_i, x_{i+1}]$, $i = 0, 1$, mit verschiedenen Randbedingungen.

Fall 1: Natürliche kubische Splinefunktion mit
$S''(0) = S''(3) = 0$

Fall 2: Kubische Splinefunktion mit vorgegebenen 1. Randableitungen
$S'(0) = 1, \quad S'(3) = 1$

Fall 3: Periodische kubische Splinefunktion
$S'(0) = S'(3), \quad S''(0) = S''(3),$
wobei $S(0) = S(3) = 1$ durch die gegebene Wertetabelle erfüllt ist.

Die 8 Splinekoeffizienten sollen in allen 3 Fällen mit Hilfe der Interpolationsbedingungen (10.4), der Anschlussbedingungen (10.5), (10.6), (10.7) und der jeweiligen Randbedingungen berechnet werden. Die 3 Splinekurven sollen skizziert werden.

Lösung:

Polynome und Ableitungen:

$$\begin{aligned}
S_0(x) &= a_0 + b_0\,x + c_0\,x^2 + d_0\,x^3\,, && x \in [0,1]\\
S_1(x) &= a_1 + b_1\,(x-1) + c_1\,(x-1)^2 + d_1\,(x-1)^3\,, && x \in [1,3]\\
S_0'(x) &= b_0 + 2\,c_0\,x + 3\,d_0\,x^2\\
S_1'(x) &= b_1 + 2\,c_1\,(x-1) + 3\,d_1\,(x-1)^2\\
S_0''(x) &= 2\,c_0 + 6\,d_0\,x\\
S_1''(x) &= 2\,c_1 + 6\,d_1\,(x-1)
\end{aligned}$$

Interpolationsbedingungen:

① $S_0(0) \stackrel{!}{=} 1: \quad a_0 = 1$

② $S_1(1) \stackrel{!}{=} 2: \quad a_1 = 2$

③ $S_1(3) \stackrel{!}{=} 1: \quad a_1 + 2b_1 + 4c_1 + 8d_1 = 1 \quad \Rightarrow \quad 2b_1 + 4c_1 + 8d_1 = 1 - 2 = -1$

Anschlussbedingungen in $x_1 = 1$:

$$\textcircled{4} \quad S_0(1) \stackrel{!}{=} S_1(1): \quad a_0 + b_0 + c_0 + d_0 = 2 \quad \Rightarrow \quad b_0 + c_0 + d_0 = 2 - 1 = 1$$
$$\textcircled{5} \quad S_0'(1) \stackrel{!}{=} S_1'(1): \quad b_0 + 2c_0 + 3d_0 = b_1$$
$$\textcircled{6} \quad S_0''(1) \stackrel{!}{=} S_1''(1): \quad 2c_0 + 6d_0 = 2c_1$$

Die beiden fehlenden Bedingungen ergeben sich aus den Randbedingungen für die Fälle 1, 2 und 3:

Fall 1:
$$\textcircled{7} \quad S_0''(0) \stackrel{!}{=} 0: \quad 2c_0 = 0 \quad \Rightarrow \quad c_0 = 0$$
$$\textcircled{8} \quad S_1''(3) \stackrel{!}{=} 0: \quad 2c_1 + 12d_1 = 0 \quad \Rightarrow \quad c_1 = -6d_1$$

Fall 2:
$$\textcircled{7} \quad S_0'(0) \stackrel{!}{=} 1: \quad b_0 = 1$$
$$\textcircled{8} \quad S_1'(3) \stackrel{!}{=} 1: \quad b_1 + 4c_1 + 12d_1 = 1$$

Fall 3:
$$\textcircled{7} \quad S_0'(0) \stackrel{!}{=} S_1'(3): \quad b_0 = b_1 + 4c_1 + 12d_1$$
$$\textcircled{8} \quad S_0''(0) \stackrel{!}{=} S_1''(3): \quad 2c_0 = 2c_1 + 12d_1$$

Lösung für Fall 1:

i	a_i	b_i	c_i	d_i
0	1	$\frac{5}{4}$	0	$-\frac{1}{4}$
1	2	$\frac{1}{2}$	$-\frac{3}{4}$	$\frac{1}{8}$

$$S_0(x) = 1 + \tfrac{5}{4}x - \tfrac{1}{4}x^3\,, \qquad x \in [0,1]$$
$$S_1(x) = 2 + \tfrac{1}{2}(x-1) - \tfrac{3}{4}(x-1)^2 + \tfrac{1}{8}(x-1)^3\,, \qquad x \in [1,3]$$

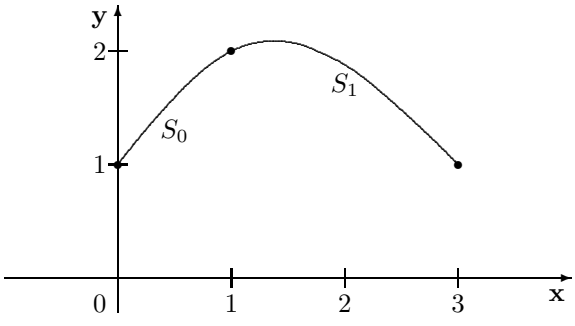


Abb. 10.17. Die natürliche Splinekurve mit $S_0''(0) = 0$ und $S_1''(3) = 0$

Lösung für Fall 2:

i	a_i	b_i	c_i	d_i
0	1	1	$\frac{3}{4}$	$-\frac{3}{4}$
1	2	$\frac{1}{4}$	$-\frac{3}{2}$	$\frac{9}{16}$

$$
\begin{aligned}
S_0(x) &= 1 + x + \tfrac{3}{4}x^2 - \tfrac{3}{4}x^3\,, && x \in [0,1] \\
S_1(x) &= 2 + \tfrac{1}{4}(x-1) - \tfrac{3}{2}(x-1)^2 + \tfrac{9}{16}(x-1)^3\,, && x \in [1,3]
\end{aligned}
$$

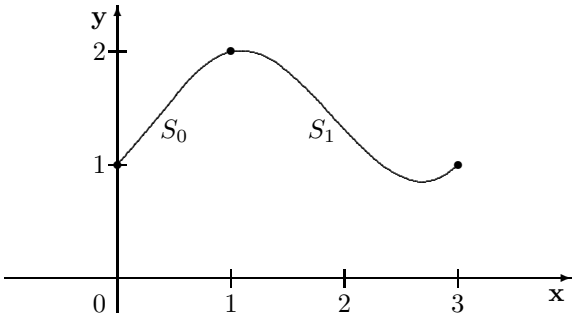


Abb. 10.18. Die Splinekurve mit den Randableitungen $S_0'(0) = 1$ und $S_1'(3) = 1$

Lösung für Fall 3:

i	a_i	b_i	c_i	d_i
0	1	$\frac{1}{2}$	$\frac{3}{2}$	-1
1	2	$\frac{1}{2}$	$-\frac{3}{2}$	$\frac{1}{2}$

$$
\begin{aligned}
S_0(x) &= 1 + \tfrac{1}{2}x + \tfrac{3}{2}x^2 - x^3\,, && x \in [0,1] \\
S_1(x) &= 2 + \tfrac{1}{2}(x-1) - \tfrac{3}{2}(x-1)^2 + \tfrac{1}{2}(x-1)^3\,, && x \in [1,3]
\end{aligned}
$$

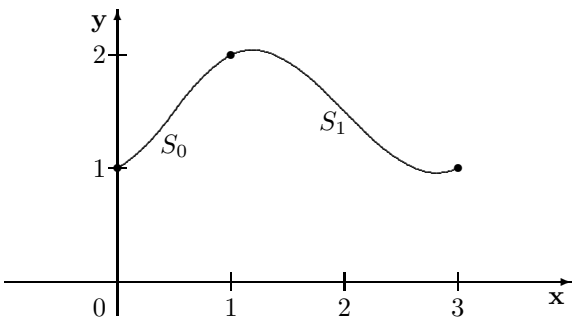


Abb. 10.19. Die periodische Splinekurve mit $S_0'(0) = S_1'(3)$ und $S_0''(0) = S_1''(3)$ □

Eigenschaft der natürlichen kubischen Splinefunktion S

Unter allen in $[a,b]$ zweimal stetig differenzierbaren Funktionen f mit $f(x_i)=y_i$ ist die natürliche Splinefunktion S mit den Knoten x_i und $S(x_i)=y_i$ diejenige, die das Integral $I(f''^2;a,b)$ minimiert; es gilt

$$\int_{x_0}^{x_n} f''^2(x)\,\mathrm{d}x \geq \int_{x_0}^{x_n} S''^2(x)\,\mathrm{d}x = \sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} S_i''^2(x)\,\mathrm{d}x\,.$$

Gilt $f'^2 \ll 1$, so minimiert die natürliche Splinefunktion näherungsweise die Biegungsenergie der durch die (x_i,y_i) verlaufenden Interpolationskurve.

Die Bezeichnung „natürlicher Spline“ ergibt sich aus der Vorgabe der Randbedingungen $S''(x_0)=S''(x_n)=0$, denn sie beschreiben genau den Fall einer Straklatte, die in den Punkten (x_i,y_i) fixiert wird und sich außerhalb von $[x_0,x_n]$ als Gerade mit der Krümmung Null fortsetzen lässt. Links von x_0 handelt es sich um eine Gerade mit der Steigung $S_0'(x_0)$, rechts von x_n um eine mit der Steigung $S_{n-1}'(x_n)$.

Die Verwendung einer *periodischen Splinefunktion* ist sinnvoll bei der Annäherung einer solchen periodischen Funktion, bei der am Anfang und Ende eines Periodenintervalls der Länge $p=x_n-x_0$ zusätzlich zu den Funktionswerten auch die Ableitungen übereinstimmen (vgl. Abb. 10.20). Voraussetzung ist, dass für die Stützpunkte $P_0=(x_0,y_0)$ und $P_n=(x_n,y_n)$ gilt $y_0=y_n$.

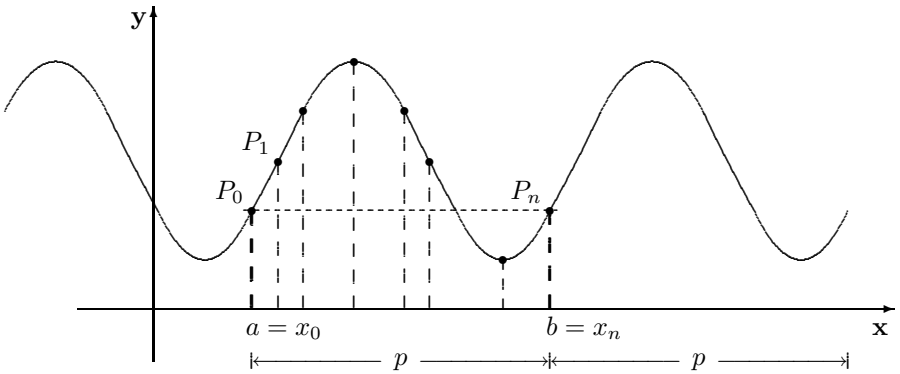


Abb. 10.20. Periodische kubische Splinefunktion

Man konstruiert dann die periodische Splinefunktion S für $[x_0,x_n]=[a,b]$ mit $b-a=p$, $S(x_0)=S(x_n)$, $S'(x_0)=S'(x_n)$, $S''(x_0)=S''(x_n)$ und erhält die gesamte Darstellung für $x\in(-\infty,\infty)$ aus der Beziehung $S(x)=S\big(a+(x-a) \bmod p\big)$.

10.1.5 Berechnung der nichtparametrischen kubischen Splines

Zur Berechnung der $4n$ Koeffizienten der n kubischen Polynome S_i, $i = 0(1)n-1$, mit der Darstellung (10.3) (vgl. Definition 10.4) werden die Bedingungen (10.4) bis (10.7) verwendet. Aus rechentechnischen Gründen verwendet man die Bedingungen (10.5) und (10.7) auch für $i = n$, so dass man mit $S_n(x_n) = a_n$ und $S_n''(x_n) = 2c_n$ zwei zusätzliche Bedingungen erhält, aber auch zwei zusätzliche Koeffizienten a_n und c_n berechnen muss. Man hat also formal für die $4n$ gesuchten Koeffizienten der $S_0, S_1, \ldots, S_{n-1}$ und die beiden zusätzlichen Koeffizienten a_n und c_n die folgenden $4n$ Bedingungen (10.8) bis (10.11) und zwei Randbedingungen.

$$\begin{aligned} S_i(x_i) &= y_i\,, & i = 0(1)n & \qquad (10.8)\\ S_i(x_i) &= S_{i-1}(x_i)\,, & i = 1(1)n & \qquad (10.9)\\ S_i'(x_i) &= S_{i-1}'(x_i)\,, & i = 1(1)n-1 & \qquad (10.10)\\ S_i''(x_i) &= S_{i-1}''(x_i)\,, & i = 1(1)n & \qquad (10.11) \end{aligned}$$

Mit (10.3) erhält man die Ableitungen

$$\begin{aligned} S_i'(x) &= b_i + 2c_i\,(x - x_i) + 3d_i\,(x - x_i)^2\,, \qquad (10.12)\\ S_i''(x) &= 2c_i + 6d_i\,(x - x_i)\,. \end{aligned}$$

Damit ergibt sich für (10.11)

$$S_i''(x_i) = 2c_i = S_{i-1}''(x_i) = 2c_{i-1} + 6d_{i-1}\,(x_i - x_{i-1})\,.$$

Mit

$$h_i = x_{i+1} - x_i \quad \text{für} \quad i = 0(1)n-1 \qquad (10.13)$$

folgt daraus

$$d_{i-1} = \frac{1}{3h_{i-1}}\,(c_i - c_{i-1}) \quad \text{für} \quad i = 1(1)n\,, \qquad (10.14)$$

und aus (10.14) erhält man durch Umindizierung

$$d_i = \frac{1}{3h_i}(c_{i+1} - c_i) \quad \text{für} \quad i = 0(1)n-1\,. \qquad (10.15)$$

Aus (10.8) ergibt sich mit (10.3)

$$S_i(x_i) = a_i = y_i \quad \text{für} \quad i = 0(1)n\,.$$

Damit und mit (10.3) erhält man für (10.9)

$$\begin{aligned} S_i(x_i) = y_i = S_{i-1}(x_i) = y_{i-1} + b_{i-1}(x_i - x_{i-1}) + c_{i-1}(x_i - x_{i-1})^2 \\ + d_{i-1}(x_i - x_{i-1})^3 \quad \text{für} \quad i = 1(1)n \end{aligned}$$

und daraus mit (10.13)

$$b_{i-1}h_{i-1} = y_i - y_{i-1} - c_{i-1}h_{i-1}^2 - d_{i-1}h_{i-1}^3 \quad \text{für} \quad i = 1(1)n$$

und nach Umindizierung

$$b_i = \frac{y_{i+1} - y_i}{h_i} - h_i(c_i + d_i h_i) \quad \text{für} \quad i = 0(1)n-1\,.$$

Für d_i wird (10.15) eingesetzt, so dass folgt

$$b_i = \frac{y_{i+1} - y_i}{h_i} - \frac{h_i}{3}(c_{i+1} + 2c_i) \quad \text{für} \quad i = 0(1)n{-}1\,. \tag{10.16}$$

Aus (10.10) folgt mit (10.12) und mit (10.13)

$$S_i'(x_i) = b_i = S_{i-1}'(x_i) = b_{i-1} + 2c_{i-1}h_{i-1} + 3d_{i-1}h_{i-1}^2 \quad \text{für} \quad i = 1(1)n{-}1$$

und daraus mit (10.14)

$$b_i - b_{i-1} = h_{i-1}(c_i + c_{i-1})\,. \tag{10.17}$$

Setzt man nun mit (10.16) b_i und b_{i-1} in (10.17) ein, erhält man für $i = 1(1)n{-}1$

$$\frac{y_{i+1} - y_i}{h_i} - \frac{h_i}{3}(c_{i+1} + 2c_i) - \frac{y_i - y_{i-1}}{h_{i-1}} + \frac{h_{i-1}}{3}(c_i + 2c_{i-1}) = h_{i-1}(c_i + c_{i-1})\,.$$

Daraus folgt

$$\begin{aligned} &h_{i-1}c_{i-1} + 2\,(h_{i-1} + h_i)c_i + h_i c_{i+1} = 3\,\frac{y_{i+1} - y_i}{h_i} - 3\,\frac{y_i - y_{i-1}}{h_{i-1}} \\ &\text{für} \quad i = 1(1)n{-}1 \end{aligned}$$

oder ausgeschrieben für $n \geq 3$

$$\begin{cases} \underline{i = 1:} \\ \quad 2\,(h_0 + h_1)\,c_1 + h_1 c_2 = 3\,\dfrac{y_2 - y_1}{h_1} - 3\,\dfrac{y_1 - y_0}{h_0} - h_0 c_0 \\ \underline{i = 2(1)n-2\,,\ n \geq 4:} \\ \quad h_{i-1}c_{i-1} + 2\,(h_{i-1} + h_i)\,c_i + h_i c_{i+1} = 3\,\dfrac{y_{i+1} - y_i}{h_i} - 3\dfrac{y_i - y_{i-1}}{h_{i-1}} \\ \underline{i = n{-}1\,,\ n \geq 3:} \\ \quad h_{n-2}c_{n-2} + 2\,(h_{n-2}{+}h_{n-1})c_{n-1} = 3\,\dfrac{y_n - y_{n-1}}{h_{n-1}} - 3\,\dfrac{y_{n-1} - y_{n-2}}{h_{n-2}} - h_{n-1}c_n \end{cases} \tag{10.18}$$

(10.18) ist also ein lineares Gleichungssystem, bestehend aus $n-1$ Gleichungen für $c_1, c_2, \ldots, c_{n-1}$. Die beiden auf der rechten Seite auftretenden Koeffizienten c_0 und c_n werden mit Hilfe von zwei Randbedingungen bei x_0 und x_n festgelegt oder durch zwei andere Bedingungen in Definition 10.4.

Im Fall $n = 2$ gibt es nur eine Gleichung

$$2\,(h_0 + h_1)\,c_1 = 3\,\frac{y_2 - y_1}{h_1} - 3\,\frac{y_1 - y_0}{h_0} - h_0\,c_0 - h_1\,c_2\,,$$

aus der c_1 sofort berechnet werden kann.

Algorithmus 10.6. (*Kubische Splinefunktion*)

Gegeben: (x_i, y_i), $i = 0(1)n$, $n \geq 3$, $x_0 < x_1 < \ldots < x_n$, eine der Randbedingungen (4)(i) bis (v) in Definition 10.4 (siehe die Algorithmen 10.7, 10.9, 10.11, 10.13 und 10.15)

Gesucht: Die Koeffizienten a_i, b_i, c_i, d_i, $i = 0(1)n-1$, der kubischen Polynome S_i (siehe (10.3) in Definition 10.4)

1. $a_i = y_i, \qquad i = 0(1)n-1$
2. $h_i = x_{i+1} - x_i, \quad i = 0(1)n-1$
3. c_0 und c_n sind durch die gewählten Randbedingungen festzulegen
4. Berechnung der Koeffizienten $c_1, c_2, \ldots, c_{n-1}$ aus dem Gleichungssystem:

   4.1 Erste Gleichung für $i = 1$:
   $$2\,(h_0 + h_1)\,c_1 + h_1 c_2 = 3\,\frac{y_2 - y_1}{h_1} - 3\,\frac{y_1 - y_0}{h_0} - h_0 c_0$$

   4.2 Gleichungen für $i = 2(1)n-2$ und $n \geq 4$:
   $$h_{i-1}c_{i-1} + 2\,(h_{i-1} + h_i)\,c_i + h_i c_{i+1} = 3\,\frac{y_{i+1} - y_i}{h_i} - 3\frac{y_i - y_{i-1}}{h_{i-1}}$$

   4.3 Letzte Gleichung für $i = n-1$:
   $$h_{n-2}c_{n-2} + 2\,(h_{n-2} + h_{n-1})c_{n-1} = 3\,\frac{y_n - y_{n-1}}{h_{n-1}} - 3\,\frac{y_{n-1} - y_{n-2}}{h_{n-2}} - h_{n-1}c_n$$

   Mit den unter 4. berechneten $c_1, \ldots, c_{n-1}$ und den unter 3. gegebenen c_0 und c_n sind zu berechnen:

5. $b_i = \dfrac{y_{i+1} - y_i}{h_i} - \dfrac{h_i}{3}\,(c_{i+1} + 2c_i)\,, \quad i = 0(1)n-1$
6. $d_i = \dfrac{1}{3h_i}\,(c_{i+1} - c_i)\,, \qquad i = 0(1)n-1.$

Bemerkung. In dem Gleichungssystem für $c_1, \ldots, c_{n-1}$ und in den Formeln für b_i und d_i treten nur Koordinatendifferenzen $h_i = x_{i+1} - x_i$ und $y_{i+1} - y_i$ usw. auf.
$(y_{i+1} - y_i)/h_i$ ist die Steigung der Sehne zwischen den Stützpunkten (x_i, y_i) und (x_{i+1}, y_{i+1}). Daher wirkt sich eine Verschiebung aller Stützpunkte in x-Richtung nicht auf die Koeffizienten aus, eine Verschiebung in y-Richtung nur auf die Koeffizienten a_i.

Das Gleichungssystem 4. im Algorithmus 10.6 besitzt die Form

$$\boldsymbol{A}\,\boldsymbol{c} = \boldsymbol{a}$$

mit

$$\boldsymbol{A} = \begin{pmatrix} 2(h_0+h_1) & h_1 & & & & \\ h_1 & 2(h_1+h_2) & h_2 & & & \\ & h_2 & 2(h_2+h_3) & h_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & h_{n-3} & 2(h_{n-3}+h_{n-2}) & h_{n-2} & \\ & & & h_{n-2} & 2(h_{n-2}+h_{n-1}) & \end{pmatrix},$$

$$\boldsymbol{c} = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_{n-1} \end{pmatrix}, \quad \boldsymbol{a} = \begin{pmatrix} 3\,\dfrac{y_2-y_1}{h_1} - 3\,\dfrac{y_1-y_0}{h_0} - h_0 c_0 \\ 3\,\dfrac{y_3-y_2}{h_2} - 3\,\dfrac{y_2-y_1}{h_1} \\ \vdots \\ 3\,\dfrac{y_n-y_{n-1}}{h_{n-1}} - 3\,\dfrac{y_{n-1}-y_{n-2}}{h_{n-2}} - h_{n-1} c_n \end{pmatrix}.$$

Eigenschaften der Matrix A

Die Matrix $\boldsymbol{A}$ ist tridiagonal, symmetrisch, stark diagonal dominant und besitzt nur positive Elemente, d. h. sie ist streng regulär und positiv definit. Da eine tridiagonale, diagonal dominante Matrix stets invertierbar ist (det $\boldsymbol{A} \neq 0$), sind Gleichungssysteme mit solchen Matrizen stets eindeutig lösbar. Bei der numerischen Lösung sollte man den Gaußschen Algorithmus bzw. das Cholesky-Verfahren für tridiagonale Matrizen (s. Abschnitt 4.9) verwenden. Das System ist gut konditioniert; Pivotsuche und Nachiteration sind nicht erforderlich.

Mit den **Randbedingungen (i)** für die natürliche kubische Splinefunktion (siehe Definition 10.4) ergeben sich

$$S''(x_0) = 2c_0 = 0\,, \quad S''(x_n) = 2c_n = 0\,, \quad \text{also } c_0 = c_n = 0\,.$$

Damit erhält man den

Algorithmus 10.7. (*Natürliche kubische Splinefunktion*)
Im Algorithmus 10.6 sind zu setzen:

3. $c_0 = 0,\ c_n = 0$

4.1 $2\,(h_0+h_1)\,c_1 + h_1 c_2 = 3\,\dfrac{y_2-y_1}{h_1} - 3\,\dfrac{y_1-y_0}{h_0}$

4.3 $h_{n-2}c_{n-2} + 2\,(h_{n-2}+h_{n-1})\,c_{n-1} = 3\,\dfrac{y_n-y_{n-1}}{h_{n-1}} - 3\,\dfrac{y_{n-1}-y_{n-2}}{h_{n-2}}$

Beispiel 10.8.

Zu den vier Stützpunkten (x_i, y_i), $i = 0(1)3$,

i	0	1	2	3
x_i	0	1	2	3
y_i	2	1	2	2

soll die natürliche kubische Splinefunktion S erzeugt werden, die also den Randbedingungen

$$S''(0) = y_0'' = 0 \quad \text{und} \quad S''(3) = y_3'' = 0$$

genügt.
Mit den Algorithmen 10.6 und 10.7 erhält man die Segmente

$$\begin{aligned} S_0(x) &= 2 - \tfrac{8}{5}\,x + \tfrac{3}{5}\,x^3\,, && x \in [0,1]\,, \\ S_1(x) &= 1 + \tfrac{1}{5}\,(x-1) + \tfrac{9}{5}\,(x-1)^2 - (x-1)^3\,, && x \in [1,2]\,, \\ S_2(x) &= 2 + \tfrac{4}{5}\,(x-2) - \tfrac{6}{5}\,(x-2)^2 + \tfrac{2}{5}\,(x-2)^3\,, && x \in [2,3]\,. \end{aligned}$$

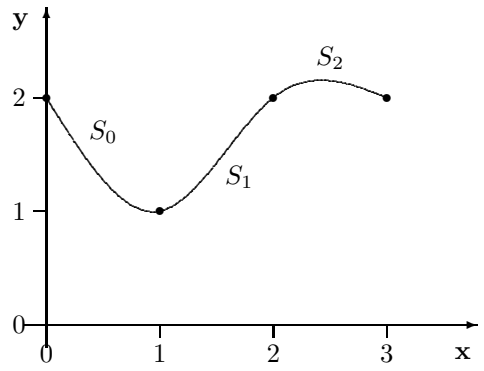


Abb. 10.21. Die natürliche Splinekurve zu den gegebenen vier Stützpunkten

□

Mit den **Randbedingungen (ii)** für die verallgemeinerte natürliche kubische Splinefunktion erhält man wegen

$$\begin{aligned} S''(x_0) = 2c_0 = y_0'' \quad &\text{und} \quad S''(x_n) = 2c_n = y_n'' \\ c_0 = \frac{y_0''}{2} \quad &\text{und} \quad c_n = \frac{y_n''}{2}\,. \end{aligned}$$

Damit folgt der

Algorithmus 10.9. (*Kubische Splinefunktion mit vorgegebenen zweiten Randableitungen*)
Gegeben:

$$S''(x_0) = y_0'', \quad S''(x_n) = y_n''$$

Im Algorithmus 10.6 sind zu setzen:

3. $c_0 = \dfrac{y_0''}{2}$, $c_n = \dfrac{y_n''}{2}$

4.1 $2\,(h_0 + h_1)\,c_1 + h_1 c_2 = 3\,\dfrac{y_2 - y_1}{h_1} - 3\,\dfrac{y_1 - y_0}{h_0} - h_0\,\dfrac{y_0''}{2}$

4.3 $h_{n-2}c_{n-2} + 2\,(h_{n-2} + h_{n-1})\,c_{n-1} = 3\,\dfrac{y_n - y_{n-1}}{h_{n-1}} - 3\,\dfrac{y_{n-1} - y_{n-2}}{h_{n-2}} - h_{n-1}\,\dfrac{y_n''}{2}$

Beispiel 10.10.

Zu denselben vier Stützpunkten wie im Beispiel 10.8 soll die kubische Splinefunktion S erzeugt werden, die den Randbedingungen

$$S''(0) = y_0'' = 3 \quad \text{und} \quad S''(3) = y_3'' = -1$$

genügt.
Die Algorithmen 10.6 und 10.9 liefern die Segmente

$$\begin{aligned}
S_0(x) &= 2 - \tfrac{221}{90}\,x + \tfrac{3}{2}\,x^2 - \tfrac{2}{45}\,x^3\,, && x \in [0,1]\,,\\
S_1(x) &= 1 + \tfrac{37}{90}\,(x-1) + \tfrac{41}{30}\,(x-1)^2 - \tfrac{7}{9}\,(x-1)^3\,, && x \in [1,2]\,,\\
S_2(x) &= 2 + \tfrac{73}{90}\,(x-2) - \tfrac{29}{30}\,(x-2)^2 + \tfrac{7}{45}\,(x-2)^3\,, && x \in [2,3]\,.
\end{aligned}$$

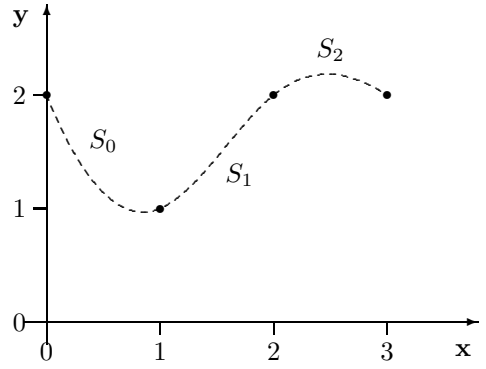


Abb. 10.22. Die Splinekurve mit den zweiten Randableitungen $S''(0) = y_0'' = 3$ und $S''(3) = y_3'' = -1$ □

Mit den **Bedingungen (iii)**

$$S_0'''(x_1) = S_1'''(x_1) \quad \text{und} \quad S_{n-2}'''(x_{n-1}) = S_{n-1}'''(x_{n-1})\,, \quad n \geq 3\,,$$

erhält man für c_0 und c_n die Beziehungen

$$\begin{cases} c_0 = c_1 \quad + \dfrac{h_0}{h_1} \quad (c_1 - c_2) \\ c_n = c_{n-1} + \dfrac{h_{n-1}}{h_{n-2}} \, (c_{n-1} - c_{n-2}) \, , \end{cases} \tag{10.19}$$

so dass sich der folgende Algorithmus ergibt.

Algorithmus 10.11. (*Kubische Splinefunktion mit not-a-knot-Bedingungen*)
Im Algorithmus 10.6 sind zu setzen:

3. c_0 und c_n gemäß (10.19)

4.1 $(h_0 + 2h_1)\, c_1 + (h_1 - h_0)\, c_2 = \dfrac{3\, h_1}{h_1 + h_0} \left(\dfrac{y_2 - y_1}{h_1} - \dfrac{y_1 - y_0}{h_0} \right)$

4.3 $(h_{n-2} - h_{n-1})\, c_{n-2} + (2h_{n-2} + h_{n-1})\, c_{n-1}$
$= \dfrac{3\, h_{n-2}}{h_{n-1} + h_{n-2}} \left(\dfrac{y_n - y_{n-1}}{h_{n-1}} - \dfrac{y_{n-1} - y_{n-2}}{h_{n-2}} \right)$

Beispiel 10.12.

Zu denselben vier Stützpunkten wie im Beispiel 10.8 soll die kubische Splinefunktion S mit not-a-knot-Bedingungen erzeugt werden.
Mit den Algorithmen 10.6 und 10.11 erhält man die Segmente

$$\begin{aligned} S_0(x) &= 2 - 3\, x + \tfrac{5}{2}\, x^2 - \tfrac{1}{2}\, x^3 \, , && x \in [0,1] \, , \\ S_1(x) &= 1 + \tfrac{1}{2}\, (x-1) + (x-1)^2 - \tfrac{1}{2}\, (x-1)^3 \, , && x \in [1,2] \, , \\ S_2(x) &= 2 + (x-2) - \tfrac{1}{2}\, (x-2)^2 - \tfrac{1}{2}\, (x-2)^3 \, , && x \in [2,3] \, . \end{aligned}$$

Wegen der not-a-knot-Bedingungen sind S_0 und S_1 in $[0,2]$ identisch und S_1 und S_2 in $[1,3]$. Also gilt in diesem Fall $S_0 = S_1 = S_2$, und S ist daher das kubische Interpolationspolynom zu den vier Stützpunkten. Wenn S_1 und S_2 nach Potenzen von x geordnet werden, ergibt sich S_0.

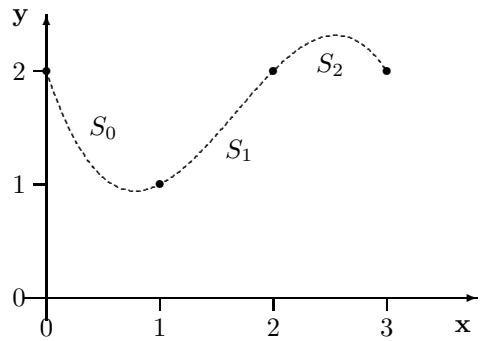


Abb. 10.23. Die not-a-knot-Splinekurve zu den gegebenen vier Stützpunkten

□

Mit den **Randbedingungen (iv)**

$$S'(x_0) = y'_0 \quad \text{und} \quad S'(x_n) = y'_n$$

erhält man für c_0 und c_n

$$\begin{cases} c_0 = \dfrac{1}{2h_0} \left(3\,\dfrac{y_1 - y_0}{h_0} - 3y'_0 - h_0 c_1 \right) \\ c_n = -\dfrac{1}{2h_{n-1}} \left(3\,\dfrac{y_n - y_{n-1}}{h_{n-1}} - 3y'_n + h_{n-1} c_{n-1} \right) \end{cases} \tag{10.20}$$

Algorithmus 10.13. (*Kubische Splinefunktion mit vorgegebenen ersten Randableitungen*)
Gegeben:

$$S'(x_0) = y'_0\,, \quad S'(x_n) = y'_n$$

Im Algorithmus 10.6 sind zu setzen:

3. c_0 und c_n gemäß (10.20)

4.1 $$\left(\frac{3}{2}\,h_0 + 2h_1 \right) c_1 + h_1 c_2 = 3 \left(\frac{y_2 - y_1}{h_1} - \frac{1}{2} \left(3\,\frac{y_1 - y_0}{h_0} - y'_0 \right) \right)$$

4.3 $$h_{n-2} c_{n-2} + \left(2h_{n-2} + \frac{3}{2} h_{n-1} \right) c_{n-1} = 3 \left(\frac{1}{2} \left(3\,\frac{y_n - y_{n-1}}{h_{n-1}} - y'_n \right) - \frac{y_{n-1} - y_{n-2}}{h_{n-2}} \right)$$

Beispiel 10.14.

Zu denselben vier Stützpunkten wie im Beispiel 10.8 soll die kubische Splinefunktion S erzeugt werden, die den Randbedingungen

$$S'(0) = y'_0 = -2 \quad \text{und} \quad S'(3) = y'_3 = -1$$

genügt.
Mit den Algorithmen 10.6 und 10.13 ergeben sich die Segmente

$$\begin{aligned} S_0(x) &= 2 - 2\,x + \tfrac{11}{15}\,x^2 + \tfrac{4}{15}\,x^3\,, && x \in [0,1]\,, \\ S_1(x) &= 1 + \tfrac{4}{15}\,(x-1) + \tfrac{23}{15}\,(x-1)^2 - \tfrac{12}{15}\,(x-1)^3\,, && x \in [1,2]\,, \\ S_2(x) &= 2 + \tfrac{14}{15}\,(x-2) - \tfrac{13}{15}\,(x-2)^2 - \tfrac{1}{15}\,(x-2)^3\,, && x \in [2,3]\,. \end{aligned}$$

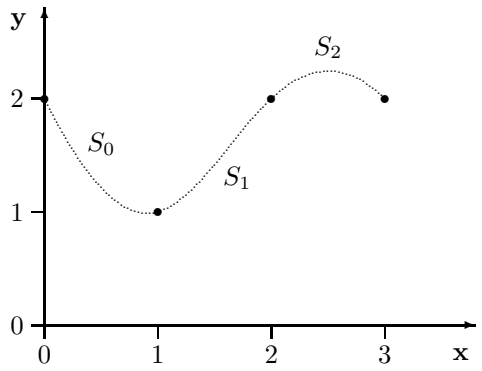


Abb. 10.24. Die Splinekurve mit den ersten Randableitungen $S'(0) = y'_0 = -2$ und $S'(3) = y'_3 = -1$ □

Die Splinekurven der 4 vorangegangenen Beispiele werden hier zum Vergleich gemeinsam dargestellt.

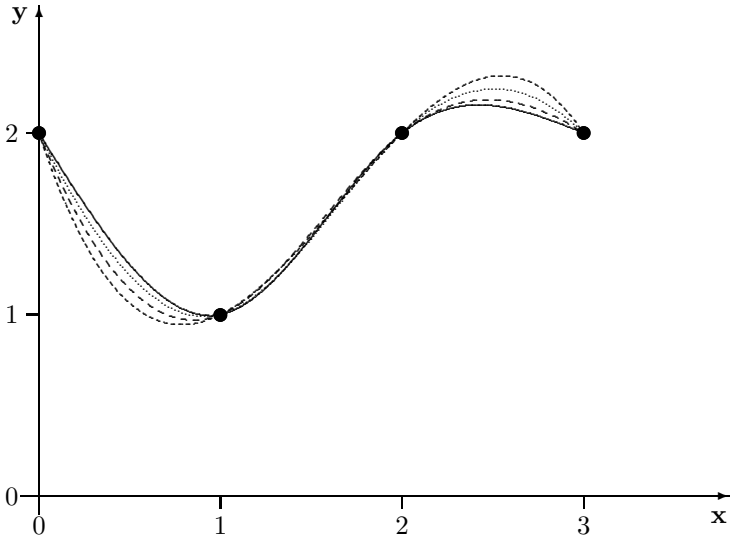


Abb. 10.25.

Mit den **Randbedingungen (v)**

$$S'''(x_0) = y'''_0 \quad \text{und} \quad S'''(x_n) = y'''_n$$

erhält man für c_0 und c_n

$$\begin{cases} c_0 = \quad c_1 \quad - \quad \dfrac{y'''_0\, h_0}{2} \\ c_n = c_{n-1} + \dfrac{y'''_n\, h_{n-1}}{2} \end{cases} \tag{10.21}$$

Algorithmus 10.15. (*Kubische Splinefunktion mit vorgegebenen dritten Randableitungen*)
Gegeben:

$$S'''(x_0) = y_0''', \quad S'''(x_n) = y_n'''$$

Im Algorithmus 10.6 sind zu setzen:

3. c_0 und c_n gemäß (10.21)

4.1 $(3h_0 + 2h_1)\, c_1 + h_1 c_2 = 3\,\dfrac{y_2 - y_1}{h_1} - 3\,\dfrac{y_1 - y_0}{h_0} + \dfrac{y_0''' \, h_0^2}{2}$

4.3 $h_{n-2} c_{n-2} + (2h_{n-2} + 3h_{n-1})\, c_{n-1} = 3\,\dfrac{y_n - y_{n-1}}{h_{n-1}} - 3\,\dfrac{y_{n-1} - y_{n-2}}{h_{n-2}} - \dfrac{y_n''' \, h_{n-1}^2}{2}$

Beispiel 10.16.

Gegeben: Zwei Wertetabellen der Funktion $f(x) = \dfrac{1}{x^2}$ mit unterschiedlich verteilten Knoten x_i, $i = 0(1)4$

a) Äquidistante Verteilung der Knoten

i	0	1	2	3	4
x_i	0.1	0.3	0.5	0.7	0.9
y_i	100	$11.\overline{1}$	4	2.041	1.235

b) Angepasste Verteilung der Knoten

i	0	1	2	3	4
x_i	0.1	0.15	0.25	0.5	0.9
y_i	100	$44.\overline{4}$	16	4	1.235

Gesucht: Die kubischen Splinefunktionen zu a) und b) mit den folgenden Randbedingungen

(i) Natürliche kubische Splinefunktion

(ii) Kubische Splinefunktion mit vorgegebenen zweiten Randableitungen
$S''(x_0) = f''(x_0) = 60\,000$, $S''(x_4) = f''(x_4) = 9.145$.

(iii) Kubische Splinefunktion mit not-a-knot Bedingungen

(iv) Kubische Splinefunktion mit vorgegebenen ersten Randableitungen
$S'(x_0) = f'(x_0) = -2000$, $S'(x_4) = f'(x_4) = -2.743$.

Zu (i): Natürliche kubische Splinefunktion

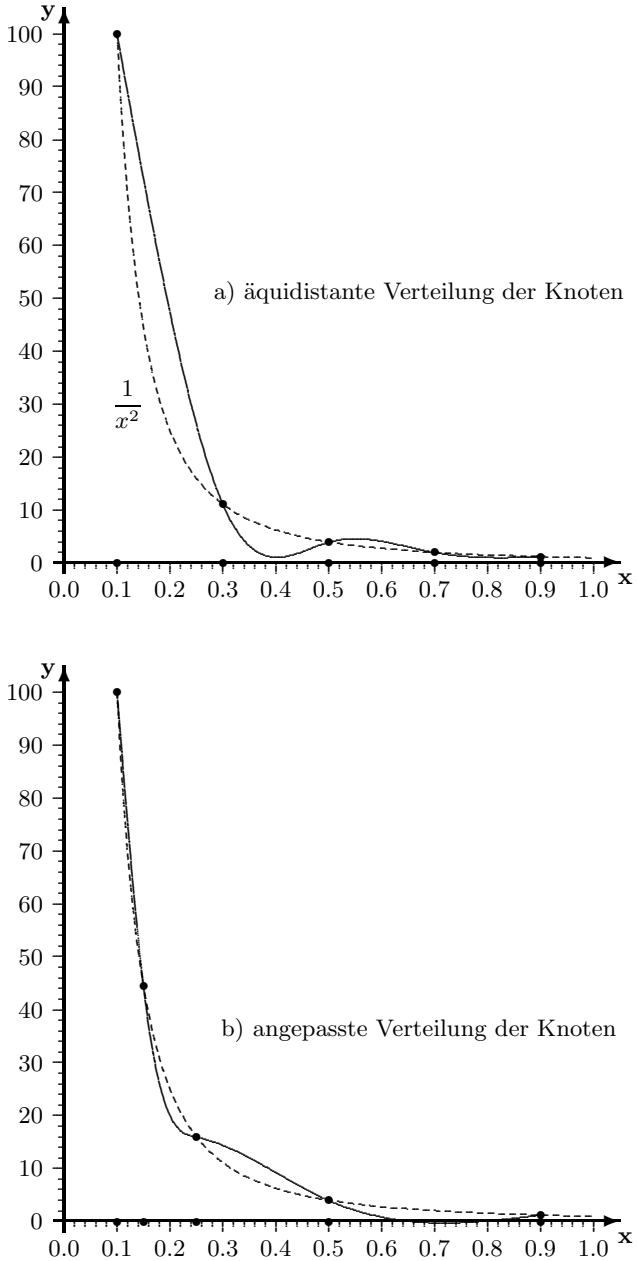


Abb. 10.26. Natürliche Splinefunktionen zu den Knotenverteilungen a) und b)

Zu (ii): Kubische Splinefunktion mit vorgegebenen zweiten Randableitungen

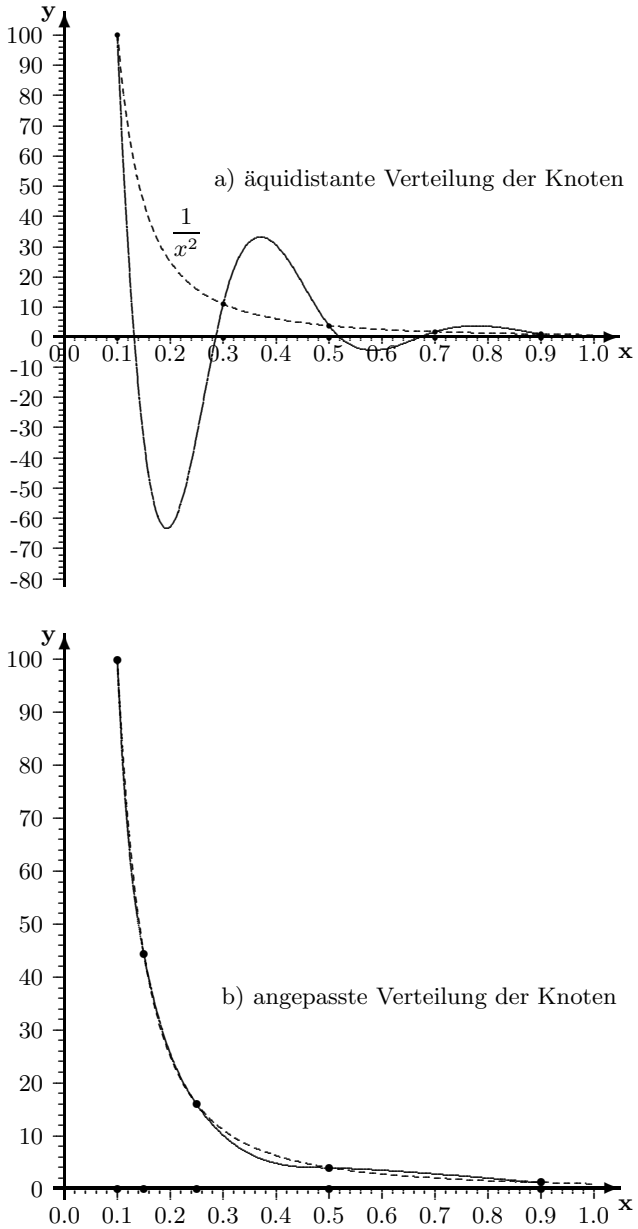


Abb. 10.27. Splinefunktionen mit vorgegebenen zweiten Randableitungen zu den Knotenverteilungen a) und b)

Zu (iii): Kubische Splinefunktion mit not-a-knot Bedingungen

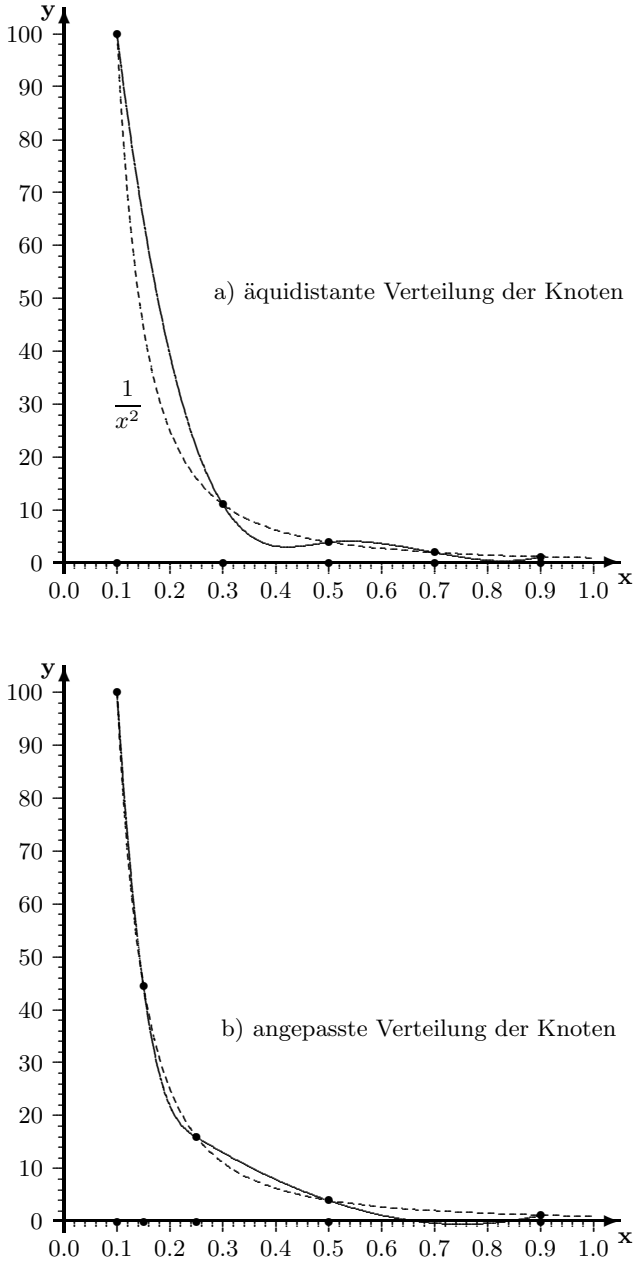


Abb. 10.28. Splinefunktionen mit not-a-knot Bedingungen zu den Knotenverteilungen a) und b)

Zu (iv): Kubische Splinefunktion mit vorgegebenen ersten Randableitungen

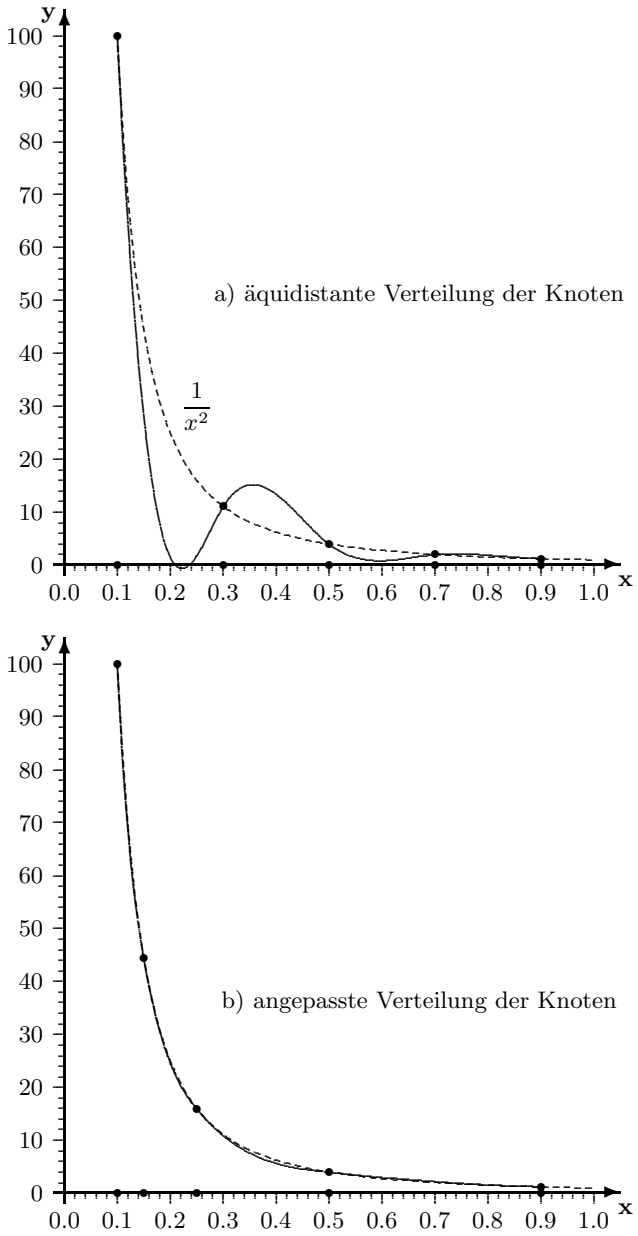


Abb. 10.29. Splinefunktionen mit vorgegebenen ersten Randableitungen zu den Knotenverteilungen a) und b)

Zusammenfassung. Wichtig sind die geeignete Methode *und* die geeignete Wahl der Knoten, nämlich dichter in Bereichen starker Steigung und weniger dicht sonst. □

Periodische Funktionen

Wenn es zu einer Funktion $f : \mathbb{R} \to \mathbb{R}$ eine positive Zahl p gibt, so dass für alle $x \in \mathbb{R}$ gilt

$$f(x \pm p) = f(x)\,,$$

dann heißt f periodisch mit der Periode p. Bekannte Beispiele sind die Sinus- und die Kosinusfunktion mit $p = 2\,\pi$. Eine periodische Funktion nimmt alle Werte bereits in irgendeinem Intervall $[a, b]$ der Länge $p = b - a$ an, einem Periodenintervall.

Die Funktion f habe das Periodenintervall $[a, b]$ mit der Länge $p = b - a$. Wenn x nicht im Intervall $[a, b]$ liegt ($x < a$ oder $x > b$), berechnet man den Funktionswert $f(x)$ an einer Stelle $\overline{x}$ aus dem Periodenintervall, für die gilt

$$\overline{x} = x \pm n\,p\,, \quad n \in \mathbb{N}\,.$$

Wegen der Periodizität von f ist dann $f(x) = f(x \pm n\,p) = f(\overline{x})$. Mit der Modulo-Funktion ergibt sich für die Stelle $\overline{x}$ im Periodenintervall

$$\overline{x} = a + (x - a) \bmod p$$

und damit für die Auswertung der periodischen Funktion f

$$f(x) = f\big(a + (x - a) \bmod p\big)\,.$$

Periodische kubische Splinefunktion

Die Randbedingungen (vi) in Definition 10.4 für die periodische kubische Splinefunktion S sind

$$S'(x_0) = S'(x_n) \quad \text{und} \quad S''(x_0) = S''(x_n)\,.$$

Aus ihnen folgen, wenn auch hier wie bei den bisher behandelten Splinearten aus rechentechnischen Gründen das Polynom S_n verwendet wird, mit

$$\begin{aligned} S'(x_0) = S'_0(x_0) &= b_0\,, & S'(x_n) = S'_n(x_n) &= b_n\,,\\ S''(x_0) = S''_0(x_0) &= 2\,c_0\,, & S''(x_n) = S''_n(x_n) &= 2\,c_n \end{aligned}$$

die Bedingungen

$$b_0 = b_n \quad \text{und} \quad c_0 = c_n\,.$$

Für die Berechnung der n Koeffizienten $c_1, c_2, \ldots, c_{n-1}, c_n = c_0$ werden n Gleichungen benötigt. Darum wird gefordert, dass die Anschlussbedingungen (10.10) auch für $i = n$ gelten. Zusätzlich sei also $S'_n(x_n) = S'_{n-1}(x_n)$. Damit erhält man außer (10.16) für $i = n$ und mit $b_n = b_0$ die Gleichung

$$b_0 - b_{n-1} = h_{n-1}\,(c_n + c_{n-1})\,.$$

Mit (10.15) für $i = 0$ und $i = n-1$ und mit $c_0 = c_n$ folgt daraus

$$\frac{y_1 - y_0}{h_0} - \frac{h_0}{3}\,(c_1 + 2\,c_n) - \frac{y_n - y_{n-1}}{h_{n-1}} + \frac{h_{n-1}}{3}\,(c_n + 2\,c_{n-1}) = h_{n-1}\,(c_n + c_{n-1})\,.$$

Das liefert die letzte Gleichung

$$h_0\,c_1 + h_{n-1}\,c_{n-1} + 2\,(h_{n-1}+h_0)\,c_n = 3\,\frac{y_1-y_0}{h_0} - 3\,\frac{y_n-y_{n-1}}{h_{n-1}}\,.$$

In der ersten Gleichung (10.17) muss $c_0 = c_n$ gesetzt werden. Der folgende Algorithmus gilt für $n \geq 3$. Der Fall $n = 2$ wird anschließend behandelt.

Algorithmus 10.17. (*Periodische kubische Splinefunktion*)

Gegeben: (x_i, y_i), $i = 0(1)n$, $n \geq 3$, mit $x_0 < x_1 < \ldots < x_n$ und $y_0 = y_n$.

Gesucht: Die Koeffizienten a_i, b_i, c_i, d_i, $i = 0(1)n{-}1$, der kubischen Polynome S_i (siehe (10.3) in Definition 10.4) der periodischen kubischen Splinefunktion S mit der Periode $p = x_n - x_0$.

1. $a_i = y_i, \qquad i = 0(1)n{-}1$

2. $h_i = x_{i+1} - x_i, \quad i = 0(1)n{-}1$

3. Berechnung der Koeffizienten $c_1, c_2, \ldots, c_n$ aus dem folgenden Gleichungssystem

   Erste Gleichung:

   $$2\,(h_0+h_1)\,c_1 + h_1\,c_2 + h_0\,c_n = 3\,\frac{y_2-y_1}{h_1} - 3\,\frac{y_1-y_0}{h_0}$$

   Gleichungen für $i = 2(1)n{-}1$:

   $$h_{i-1}\,c_{i-1} + 2\,(h_{i-1}+h_i)\,c_i + h_i\,c_{i+1} = 3\,\frac{y_{i+1}-y_i}{h_i} - 3\,\frac{y_i-y_{i-1}}{h_{i-1}}$$

   Letzte Gleichung:

   $$h_0\,c_1 + h_{n-1}\,c_{n-1} + 2\,(h_{n-1}+h_0)\,c_n = 3\,\frac{y_1-y_0}{h_0} - 3\,\frac{y_n-y_{n-1}}{h_{n-1}}$$

4. Setze $c_0 = c_n$

5. $b_i = \dfrac{y_{i+1}-y_i}{h_i} - \dfrac{h_i}{3}(c_{i+1}+2c_i), \quad i = 0(1)n{-}1$

6. $d_i = \dfrac{1}{3h_i}(c_{i+1}-c_i), \qquad i = 0(1)n{-}1$

Das unter 3. angegebene Gleichungssystem hat die Matrix

$$\boldsymbol{A} = \begin{pmatrix} 2(h_0+h_1) & h_1 & & & & h_0 \\ h_1 & 2(h_1+h_2) & h_2 & & & \\ & h_2 & 2(h_2+h_3) & h_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & h_{n-2} & 2(h_{n-2}+h_{n-1}) & h_{n-1} \\ h_0 & & & & h_{n-1} & 2(h_{n-1}+h_0) \end{pmatrix}$$

Eigenschaften der Matrix A

Die Matrix $\boldsymbol{A}$ ist zyklisch tridiagonal, symmetrisch, diagonal dominant und besitzt nur positive Elemente; $\boldsymbol{A}$ ist damit streng regulär, positiv definit und gut konditioniert. Ein Algorithmus zur Lösung von Gleichungssystemen mit zyklisch tridiagonalen Matrizen ist im Abschnitt 4.10 angegeben.

Im Fall $n = 2$ lauten die Gleichungen für c_1 und c_2

$$\begin{aligned} 2\,(h_0 + h_1)\,c_1 \;+\; (h_0 + h_1)\,c_2 &= 3\left(\frac{y_2 - y_1}{h_1} - \frac{y_1 - y_0}{h_0}\right) \\ (h_0 + h_1)\,c_1 \;+\; 2\,(h_0 + h_1)\,c_2 &= 3\left(\frac{y_1 - y_0}{h_0} - \frac{y_2 - y_1}{h_1}\right) \end{aligned}$$

Sie haben die Lösungen

$$c_1 = \frac{3}{h_0 + h_1}\left(\frac{y_2 - y_1}{h_1} - \frac{y_1 - y_0}{h_0}\right), \quad c_2 = -c_1 = c_0\,.$$

Mit $m_0 = \dfrac{y_1 - y_0}{h_0}$, $m_1 = \dfrac{y_2 - y_1}{h_1}$ sind die Koeffizienten der Polynome S_0 und S_1:

$$\begin{aligned} a_0 &= y_0 & a_1 &= y_1 \\ b_0 &= \frac{h_0\,m_1 + h_1\,m_0}{h_0 + h_1} & b_1 &= b_0 \\ c_0 &= 3\,\frac{m_0 - m_1}{h_0 + h_1} & c_1 &= -c_0 \\ d_0 &= 2\,\frac{m_1 - m_0}{h_0\,(h_0 + h_1)} & d_1 &= -\frac{h_0}{h_1}\,d_0 \end{aligned}$$

Die Auswertung der periodischen Splinefunktion S mit der Periode $p = x_n - x_0$ an einer Stelle $x \in \mathbf{R}$ erfolgt nach der Vorschrift $S(x) = S\big(x_0 + (x - x_0) \bmod p\big)$; dabei ist $x_0 \;\leq\; x_0 + (x - x_0) \bmod p \;<\; x_n$.

Beispiel 10.18.

Zu denselben vier Stützpunkten wie im Beispiel 10.8 soll die periodische kubische Splinefunktion S erzeugt werden. Mit $y_0 = y_3 = 2$ erfüllt sie die Voraussetzung, ihre Periode ist $p = x_3 - x_0 = 3$.
Mit dem Algorithmus 10.17 ergeben sich die Segmente

$$\begin{aligned} S_0(x) &= 2 - x - x^2 + x^3\,, & x \in [0,1]\,,\\ S_1(x) &= 1 + 2\,(x-1)^2 - (x-1)^3\,, & x \in [1,2]\,,\\ S_2(x) &= 2 + (x-2) - (x-2)^2\,, & x \in [2,3]\,. \end{aligned}$$

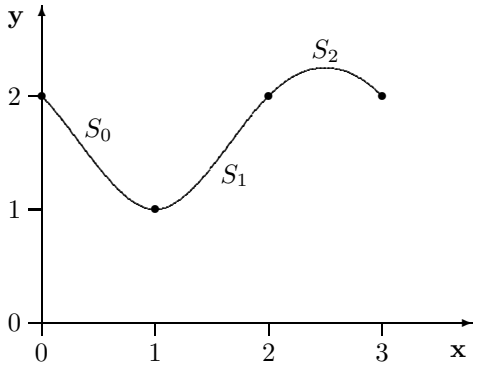


Abb. 10.30. Die periodische Splinekurve zu den gegebenen vier Stützpunkten.

□

10.1.6 Berechnung der parametrischen kubischen Splines

Im Fall von Stützpunkten P_i allgemeiner Lage ($P_i \in \mathbf{R}^m$, $m = 2$ oder $m = 3$, $i = 0(1)n$, $n \geq 2$) muss parametrisch gerechnet werden (vgl. Abschnitt 10.1.1). Der interpolierende kubische Spline hat dann für $m = 2$ die Darstellung (10.1):

$$(1) \qquad \boldsymbol{x}(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \boldsymbol{S}(t) = \begin{pmatrix} S_x(t) \\ S_y(t) \end{pmatrix} \equiv \begin{pmatrix} S_{ix}(t) \\ S_{iy}(t) \end{pmatrix} = \boldsymbol{S}_i(t)$$

und für $m = 3$ die Darstellung (10.2):

$$(2) \qquad \boldsymbol{x}(t) = \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix} = \boldsymbol{S}(t) = \begin{pmatrix} S_x(t) \\ S_y(t) \\ S_z(t) \end{pmatrix} \equiv \begin{pmatrix} S_{ix}(t) \\ S_{iy}(t) \\ S_{iz}(t) \end{pmatrix} = \boldsymbol{S}_i(t)$$

mit den Segmenten

$$\begin{aligned} \boldsymbol{S}_i(t) &= \boldsymbol{a}_i + \boldsymbol{b}_i(t-t_i) + \boldsymbol{c}_i(t-t_i)^2 + \boldsymbol{d}_i(t-t_i)^3\,, \quad t \in [t_i, t_{i+1}]\,,\\ &\quad \boldsymbol{a}_i, \boldsymbol{b}_i, \boldsymbol{c}_i, \boldsymbol{d}_i \in \mathbb{R}^m\,, \quad i = 0(1)n{-}1\,. \end{aligned}$$

Bevor die Komponenten von $\boldsymbol{S}_i(t)$, also $S_{ix}(t)$, $S_{iy}(t)$ in (1) bzw. $S_{ix}(t)$, $S_{iy}(t)$, $S_{iz}(t)$ in (2), berechnet werden können, werden im Folgenden Parameterwerte (Knoten) t_i, $i = 0(1)n$, mit

$$t_0 < t_1 < \ldots < t_n$$

bereitgestellt (*Parametrisierung*).

Die den Knoten t_i zugeordneten Splinepunkte sollen mit den Stützpunkten übereinstimmen (Interpolationsbedingungen):

$$\boldsymbol{S}(t_i) = \boldsymbol{P}_i\,, \quad i = 0(1)n\,.$$

Zum Splinesegment $\boldsymbol{S}_i(t)$ gehört das Parameterintervall $[t_i, t_{i+1}]$ mit der Länge $h_i = t_{i+1} - t_i > 0$.

Für eine Parametrisierung kann man die positiven Intervall-Längen $h_0,\ h_1,\ \ldots,\ h_{n-1}$ vorgeben. Nach Wahl von t_0 (z. B. $t_0 = 0$) ergeben sich dann die Knoten

$$t_{i+1} = t_i + h_i\,, \quad i = 0(1)n{-}1\,.$$

Die Formeln in den Algorithmen 10.6, 10.7, 10.9, 10.11, 10.13, 10.15 und 10.17 enthalten die Intervall-Längen, nicht aber die Knoten; daher kann t_0 beliebig gewählt werden.

Somit ist die Aufgabe gestellt, jedem Segment $\boldsymbol{S}_i(t)$, $i = 0(1)n{-}1$, ein Parameterintervall der Länge $h_i > 0$ zuzuordnen. Die Intervall-Längen h_i können in verschiedener Weise gewählt werden.

(i) Am einfachsten ist die *äquidistante Parametrisierung* mit $h_i = h = \text{const} > 0$, z. B. $h = 1$. Sie liefert jedoch im Allgemeinen keine zufriedenstellenden Ergebnisse.

(ii) Günstiger ist eine Parametrisierung, die die Lage der Stützpunkte berücksichtigt. Die häufig benutzte *chordale Parametrisierung* verwendet als Länge des i-ten Parameterintervalls die Länge der Sehne zwischen den benachbarten und als verschieden vorausgesetzten Stützpunkten P_i und P_{i+1}

$$h_i = |\boldsymbol{P}_{i+1} - \boldsymbol{P}_i|, \quad i = 0(1)n{-}1,$$

und damit eine grobe Annäherung der Bogenlänge des betreffenden Segmentes.

Im Fall einer ebenen Kurve ($m = 2$) ist

$$h_i = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}, \quad i = 0(1)n{-}1\,,$$

und im Fall einer Raumkurve ($m = 3$)

$$h_i = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2}, \quad i = 0(1)n{-}1\,.$$

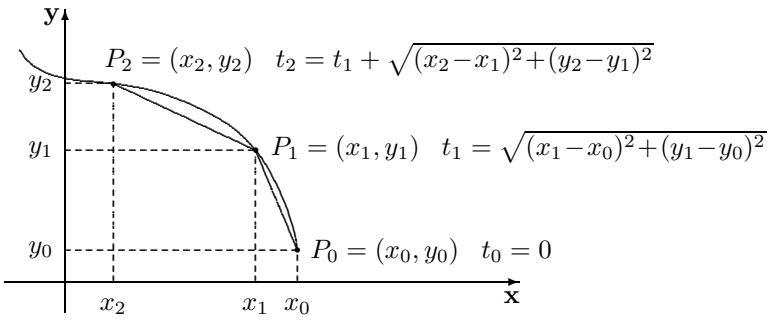


Abb. 10.31. Ermittlung von Parameterwerten mit chordaler Parametrisierung

Beispiel 10.19.

Zu den 9 gegebenen Punkten P_0 bis P_8 werden interpolierende kubische Splinekurven mit äquidistanter und chordaler Parametrisierung erzeugt.

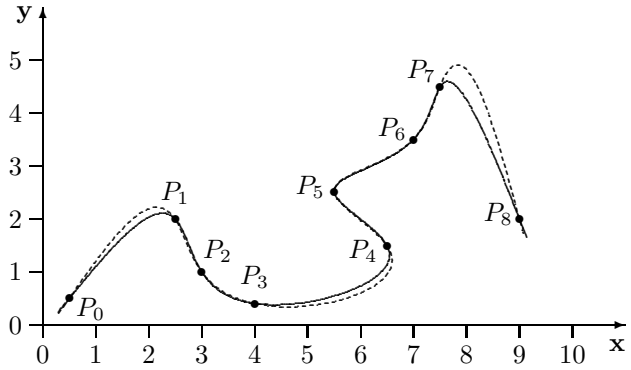


Abb. 10.32. Äquidistante Parametrisierung (gestrichelte Linie), chordale Parametrisierung (durchgezogene Linie) □

(iii) Wenn drei aufeinander folgende Stützpunkte annähernd kollinear sind, liefert die chordale Parametrisierung Intervalle, deren Längen von den Bogenlängen der Segmente nur wenig abweichen werden. Wenn dagegen die Stützpunkte stärker gekrümmte Segmente erwarten lassen, kann es günstig sein, diesen Segmenten Intervalle zuzuordnen, die länger sind als die Sehnen. Die folgende *Variante der chordalen Parametrisierung* berücksichtigt diese Aspekte.

Durch drei aufeinander folgende nicht kollineare Stützpunkte wird der durch sie bestimmte Kreis gelegt. Die Länge des Kreisbogens zwischen zwei Stützpunkten wird zur Erzeugung der Länge des zugehörigen Parameterintervalls benutzt.

Sind P, Q, R drei aufeinander folgende Punkte eines Kreises, ist $|\boldsymbol{R}-\boldsymbol{Q}|$ die Länge der Sehne zwischen Q und R und ist γ der Winkel zwischen PQ und PR, also der Umfangswinkel dieser Sehne bei P, dann ist die Länge des Kreisbogens zwischen Q und R

$$B = |\boldsymbol{R}-\boldsymbol{Q}|\frac{\gamma}{\sin\gamma}, \quad 0 \le \gamma < \pi \ .$$

Wegen $\lim\limits_{\gamma\to 0}(\gamma/\sin\gamma) = 1$ gilt die Formel auch im Grenzfall kollinearer Punkte P, Q, R.

Für die Punkte $P = P_{i-1}$, $Q = P_i$, $R = P_{i+1}$ ergibt sich mit γ_i als Winkel zwischen $P_{i-1}P_i$ und $P_{i-1}P_{i+1}$ für die Länge B_i des Kreisbogens zwischen P_i und P_{i+1}

$$B_i = |\boldsymbol{P}_{i+1}-\boldsymbol{P}_i|\frac{\gamma_i}{\sin\gamma_i} \ .$$

Für die Punkte $P = P_{i+2}$, $Q = P_{i+1}$, $R = P_i$ und γ_{i+1} als Winkel zwischen $P_{i+2}P_{i+1}$ und $P_{i+2}P_i$ folgt analog für die Länge B_{i+1} des Kreisbogens zwischen P_i und P_{i+1}

$$B_{i+1} = |\boldsymbol{P}_{i+1}-\boldsymbol{P}_i|\frac{\gamma_{i+1}}{\sin\gamma_{i+1}} \ .$$

Aus der Länge $|\boldsymbol{P}_{i+1}-\boldsymbol{P}_i|$ der Sehne und dem Mittel der Längen B_i und B_{i+1} wird mit dem Gewicht σ, $\sigma \ge 0$, die Intervall-Länge

$$\begin{aligned} h_i &= (1-\sigma)|\boldsymbol{P}_{i+1}-\boldsymbol{P}_i| + \sigma\tfrac{1}{2}\Big(B_i + B_{i+1}\Big) \\ &= |\boldsymbol{P}_{i+1}-\boldsymbol{P}_i|\left(1+\sigma\left(\tfrac{1}{2}\left(\frac{\gamma_i}{\sin\gamma_i}+\frac{\gamma_{i+1}}{\sin\gamma_{i+1}}\right)-1\right)\right) \end{aligned}$$

erzeugt. Es ist $h_i \ge |\boldsymbol{P}_{i+1}-\boldsymbol{P}_i|$.

Für $\sigma = 0$ entsteht die chordale Parametrisierung. Aufgrund der Konstruktion ist für $\sigma > 0$ der Beitrag der Kreisbogenlängen dort größer, wo die Stützpunkte stärker gekrümmte Kurvensegmente erwarten lassen und längere Intervalle deshalb zweckmäßig sind. Mit der Änderung des Gewichtes σ kann die Gestalt der Splinekurve (bei unveränderten Rand- oder Zusatzbedingungen) beeinflusst werden. Erfahrungsgemäß ergeben sich geeignete Splinekurven für $\sigma = 0$ (chordal) bis etwa $\sigma = 2$, evtl. auch $\sigma > 2$.

Beispiel 10.20.

Dieses Beispiel zeigt, wie sich der Verlauf einer natürlichen Splinekurve mit der Verallgemeinerung der chordalen Parametrisierung durch Wahl des Parameters σ beeinflussen lässt.

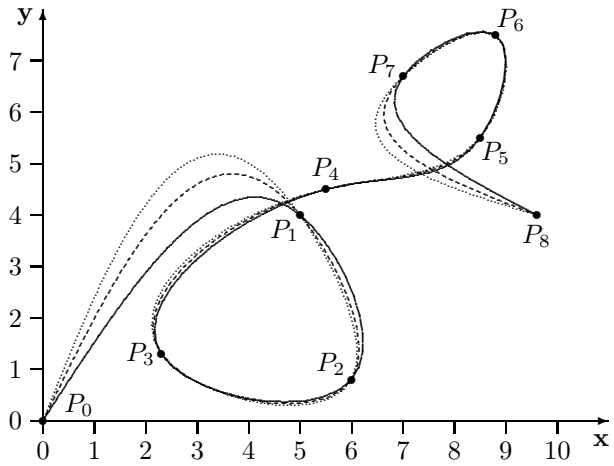


Abb. 10.33. Verallgemeinerte chordale Parametrisierung mit $\sigma = 0$ (durchgezogene Linie), $\sigma = 1.5$ (gestrichelte Linie) und $\sigma = 3$ (gepunktete Linie) □

Beispiel 10.21.

Dieses Beispiel zeigt, wie sich der Verlauf einer geschlossenen Splinekurve, die auch im Punkt $P_7 = P_0$ glatt ist, mit der Verallgemeinerung der chordalen Parametrisierung durch Wahl des Parameters σ beeinflussen lässt.

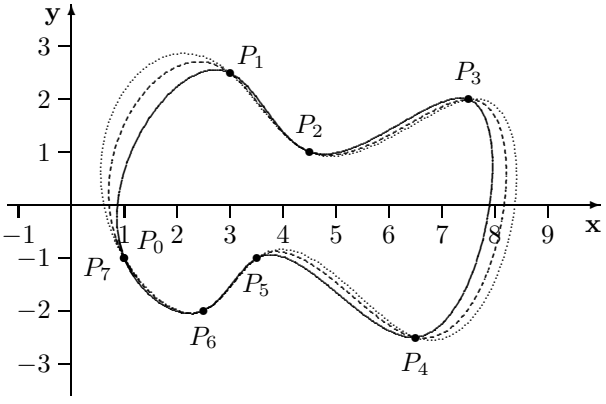


Abb. 10.34. Verallgemeinerte chordale Parametrisierung bei einer geschlossenen Kurve mit $\sigma = 0$ (durchgezogene Linie), $\sigma = 1.5$ (gestrichelte Linie) und $\sigma = 3$ (gepunktete Linie) □

Um die Formel für h_i auch für $i = 0$ und $i = n-1$ anwenden zu können, werden zwei zusätzliche Punkte, P_{-1} und P_{n+1}, benötigt, nämlich für die Winkel γ_0 und γ_n. Wenn die Kurve geschlossen ist mit $P_0 = P_n$ und wenn sie auch in diesem Punkt glatt ist, also nur eine Tangente besitzt, setzt man $P_{-1} := P_{n-1}$ und $P_{n+1} := P_1$. In allen anderen Fällen (also auch bei einer geschlossenen Kurve, die in $P_0 = P_n$ nicht glatt ist) ergibt sich P_{-1} durch Spiegelung von P_2 an der Gerade bzw. Ebene, die durch den Mittelpunkt von P_0 und P_1 geht und zu P_0P_1 senkrecht ist. Analog wird P_{n-2} an der Gerade bzw. Ebene, die durch den Mittelpunkt von P_{n-1} und P_n geht und zu $P_{n-1}P_n$ senkrecht ist, gespiegelt, um P_{n+1} zu erhalten. Im nachfolgenden Algorithmus werden sogleich die Sehnenvektoren $\boldsymbol{s}_{-1} = \boldsymbol{P}_0 - \boldsymbol{P}_{-1}$ und $\boldsymbol{s}_n = \boldsymbol{P}_{n+1} - \boldsymbol{P}_n$ angegeben.

Mit den Vektoren $\boldsymbol{s} := \boldsymbol{Q} - \boldsymbol{P}$ und $\boldsymbol{t} := \boldsymbol{R} - \boldsymbol{P}$ kann der im Ausdruck für B enthaltene Sinus des von $\boldsymbol{s}$ und $\boldsymbol{t}$ eingeschlossenen Winkels γ mit Hilfe von Skalarprodukten wie folgt berechnet werden:

$$\sin\gamma = \sqrt{1-\cos^2\gamma} = \sqrt{1 - \frac{(\boldsymbol{s}^\mathsf{T}\boldsymbol{t})^2}{(\boldsymbol{s}^\mathsf{T}\boldsymbol{s})(\boldsymbol{t}^\mathsf{T}\boldsymbol{t})}}\,.$$

Im Fall einer ebenen Kurve ($m = 2$) gilt außerdem

$$\sin\gamma = \frac{|\det(\boldsymbol{s},\boldsymbol{t})|}{\sqrt{(\boldsymbol{s}^\mathsf{T}\boldsymbol{s})(\boldsymbol{t}^\mathsf{T}\boldsymbol{t})}}\,.$$

Algorithmus 10.22. (*Berechnung der Intervall-Längen und Parameterwerte*)

Gegeben: Stützpunkte $P_i = (x_i, y_i)$ bzw. $P_i = (x_i, y_i, z_i)$, $i = 0(1)n$, $n \geq 2$, $P_i \neq P_{i+1} \neq P_{i+2} \neq P_i$ für $i = 0(1)n-2$, Gewicht $\sigma \geq 0$.

Gesucht: Intervall-Längen $h_i > 0$, $i = 0(1)n-1$, und Parameterwerte t_i, $i = 0(1)n$.

1. Erzeugung der Sehnenvektoren für $i = 0(1)n-1$: $\boldsymbol{s}_i := \boldsymbol{P}_{i+1} - \boldsymbol{P}_i \neq \boldsymbol{0}$.
2. Wenn $\sigma = 0$: $h_i := |\boldsymbol{s}_i|$, $i = 0(1)n-1$, chordale Parametrisierung, weiter bei 3.
   Wenn $\sigma > 0$:

   2.1 Bereitstellung weiterer Sehnenvektoren.

   (a) Kurve geschlossen und auch in $P_n = P_0$ glatt
   $\boldsymbol{s}_{-1} := \boldsymbol{s}_{n-1}$,
   $\boldsymbol{s}_n := \boldsymbol{s}_0$.

   (b) In allen anderen Fällen

   $$\boldsymbol{s}_{-1} := 2\frac{\boldsymbol{s}_0^\mathsf{T}\boldsymbol{s}_1}{|\boldsymbol{s}_0|^2}\boldsymbol{s}_0 - \boldsymbol{s}_1,$$

   $$\boldsymbol{s}_n := 2\frac{\boldsymbol{s}_{n-1}^\mathsf{T}\boldsymbol{s}_{n-2}}{|\boldsymbol{s}_{n-1}|^2}\boldsymbol{s}_{n-1} - \boldsymbol{s}_{n-2}\,.$$

2.2 Für $i = 0(1)n{-}1$:

Berechne C mit $(*)$ für
$\boldsymbol{s} := \boldsymbol{s}_{i-1}$, $\boldsymbol{t} := \boldsymbol{s}_{i-1} + \boldsymbol{s}_i$ und setze $C_i := C$.

Berechne C mit $(*)$ für
$\boldsymbol{s} := \boldsymbol{s}_i + \boldsymbol{s}_{i+1}$, $\boldsymbol{t} := \boldsymbol{s}_{i+1}$ und setze $C_{i+1} := C$.

$h_i := |\boldsymbol{s}_i| \left(1 + \sigma\big(0.5(C_i + C_{i+1}) - 1\big)\right)$.

3. $t_0 := 0$; für $i = 0(1)n{-}1$ werden berechnet: $t_{i+1} := t_i + h_i$.

$(*)$ Berechne mit $\boldsymbol{s}$ und $\boldsymbol{t}$ $(\boldsymbol{s} \neq 0,\ \boldsymbol{t} \neq 0)$
$Z := \boldsymbol{s}^{\mathsf{T}}\boldsymbol{t}$
$N := (\boldsymbol{s}^{\mathsf{T}}\boldsymbol{s})(\boldsymbol{t}^{\mathsf{T}}\boldsymbol{t})$
$S := \sqrt{1 - Z^2/N}$
Wenn $S = 0$: $C := 1$
Wenn $S > 0$: $\gamma := \arcsin(S)$
Wenn $Z < 0$: $\gamma := \pi - \gamma$
$C := \gamma/S$.

Nachdem mit einer Parametrisierung die Knoten t_i, $i = 0(1)n$, bereitgestellt worden sind, wird wie folgt vorgegangen.

Ebene Splinekurve durch die Punkte $\boldsymbol{P_i = (x_i, y_i)}$, $\boldsymbol{i = 0(1)n}$, $\boldsymbol{n \geq 2}$

Um den kubischen Spline (1) zu erhalten, wird zu den Wertepaaren (t_i, x_i), $i = 0(1)n$, eine Splinefunktion S_x erzeugt mit der Darstellung

$$\left\{\begin{array}{lcl} S_x(t) & \equiv & S_{ix}(t) = a_{ix} + b_{ix}(t - t_i) + c_{ix}(t - t_i)^2 + d_{ix}(t - t_i)^3 \\ & & \text{für } t \in [t_i, t_{i+1}], \quad i = 0(1)n{-}1\,, \end{array}\right. \tag{10.22}$$

und zu den Wertepaaren (t_i, y_i), $i = 0(1)n$, eine Splinefunktion S_y mit der Darstellung

$$\left\{\begin{array}{lcl} S_y(t) & \equiv & S_{iy}(t) = a_{iy} + b_{iy}(t - t_i) + c_{iy}(t - t_i)^2 + d_{iy}(t - t_i)^3 \\ & & \text{für } t \in [t_i, t_{i+1}]\,, \quad i = 0(1)n{-}1\,. \end{array}\right. \tag{10.23}$$

Je nach Vorgabe der Randbedingungen (s. Definition 10.4) werden S_x und S_y mit dem Algorithmus 10.6 oder 10.17 berechnet. Diese Algorithmen werden also anstelle der Punkte (x_i, y_i) je einmal mit den Wertepaaren (t_i, x_i) und (t_i, y_i) durchgeführt.

Räumliche Splinekurve durch die Punkte $P_i = (x_i, y_i, z_i)$, $i = 0(1)n$, $n \geq 2$

Hier müssen die Komponenten für den kubischen Spline (2) ermittelt werden. Wie bei der ebenen Splinekurve werden zu den Wertepaaren (t_i, x_i) und (t_i, y_i) die Splinefunktionen (10.22) und (10.23) erzeugt. Ferner wird zu den Wertepaaren (t_i, z_i), $i = 0(1)n$, eine Splinefunktion S_z berechnet mit der Darstellung

$$\begin{aligned} S_z(t) \quad \equiv \quad & S_{iz}(t) = a_{iz} + b_{iz}(t-t_i) + c_{iz}(t-t_i)^2 + d_{iz}(t-t_i)^3 \\ & \text{für } t \in [t_i, t_{i+1}]\,, \quad i = 0(1)n{-}1\,. \end{aligned}$$

Dafür wird je nach Vorgabe der Randbedingungen der Algorithmus 10.6 oder 10.17 eingesetzt.

Bei einer ebenen Kurve muss der Algorithmus 10.6 oder 10.17 zweimal ausgeführt werden, bei einer Raumkurve dreimal. Dabei ist zu beachten, dass die linearen Gleichungssysteme zur Berechnung der c_{ix}, c_{iy} bzw. c_{ix}, c_{iy}, c_{iz} die gleiche Matrix haben, aber verschiedene rechte Seiten. Deshalb ist die Dreieckzerlegung der Matrix nur einmal durchzuführen, die Vorwärts- und Rückwärtselimination zweimal im ebenen Fall, dreimal im räumlichen Fall. Dies bringt eine erhebliche Einsparung an Rechenaufwand.

Für eine geschlossene, ebene oder räumliche Splinekurve zu Stützpunkten P_0, P_1, ..., $P_n = P_0$, die auch im Punkt $P_0 = P_n$ glatt ist, also genau eine Tangente besitzt, müssen die Komponenten S_x, S_y und gegebenenfalls S_z periodisch sein. Sie werden also mit dem Algorithmus 10.17 berechnet.
In allen anderen Fällen werden die Komponenten mit dem Algorithmus 10.6 und mit einem der Algorithmen 10.7, 10.9, 10.11, 10.13 oder 10.15 ermittelt. Dabei kann $P_n \neq P_0$ oder $P_n = P_0$ sein.

Berechnung der Krümmung einer ebenen Splinekurve

Eine ebene Splinekurve $t \mapsto \boldsymbol{S}(t)$, $t \in [t_0, t_n]$ mit den Segmenten

$$t \mapsto \boldsymbol{S}_i(t) = \begin{pmatrix} S_{ix}(t) \\ S_{iy}(t) \end{pmatrix}, \quad t \in [t_i, t_{i+1}]\,, \quad i = 0(1)n{-}1\,,$$

besitzt die Krümmung

$$\begin{aligned} \kappa_i(t) \quad &= \quad \frac{\det\left(\boldsymbol{S}_i'(t), \boldsymbol{S}_i''(t)\right)}{|\,\boldsymbol{S}_i'(t)\,|^3} \\ &= \quad \frac{S_{ix}'(t)\,S_{iy}''(t) - S_{ix}''(t)\,S_{iy}'(t)}{\left(S_{ix}'^2(t) + S_{iy}'^2(t)\right)^{3/2}}\,, \quad t \in [t_i, t_{i+1}]\,. \end{aligned}$$

Dabei sind

$$\begin{aligned} S_{ix}'(t) &= b_{ix} + 2\,c_{ix}\,(t-t_i) + 3\,d_{ix}\,(t-t_i)^2\,, \\ S_{ix}''(t) &= 2\,c_{ix} + 6\,d_{ix}\,(t-t_i)\,, \\ S_{iy}'(t) &= b_{iy} + 2\,c_{iy}\,(t-t_i) + 3\,d_{iy}\,(t-t_i)^2\,, \\ S_{iy}''(t) &= 2\,c_{iy} + 6\,d_{iy}\,(t-t_i)\,. \end{aligned}$$

Wegen der Stetigkeit der ersten und zweiten Ableitungen gilt für die Krümmung an den inneren Knoten

$$\kappa_{i-1}(t_i) = \kappa_i(t_i)\,, \quad i = 1(1)n{-}1\,.$$

Die Krümmung κ der Splinekurve ist also eine auf dem Intervall $[t_0, t_n]$ stetige Funktion.

10.1.7 Kombinierte interpolierende Polynom-Splines

Bei der Herstellung faserverstärkter Kunststoff-Hohlkörper werden vorwiegend rotationssymmetrische Wickelkerne verwendet. Die Meridiankurve eines Wickelkerns, die sich häufig aus geradlinigen und aus punktweise gegebenen krummlinigen Abschnitten zusammensetzt, soll insgesamt zweimal stetig differenzierbar sein. Eine Splinefunktion S zur Darstellung eines krummlinigen Abschnittes im Intervall $[x_0, x_n]$ muss tangential und mit verschwindender Krümmung an benachbarte geradlinige Abschnitte anschließen. Daher müssen die Steigungen $S'(x_0)$ und $S'(x_n)$ vorgeschrieben werden und es müssen $S''(x_0) = 0$ und $S''(x_n) = 0$ sein. Da mit einer kubischen Splinefunktion nur zwei Randbedingungen erfüllt werden können, kommt sie für die Lösung dieser Aufgabe nicht in Frage (siehe Beispiel 10.2 und Bemerkung 10.3).

Etwas allgemeiner ist die folgende Aufgabe. Gegeben seien zwei Funktionen $f \in C^2[a, x_0]$ und $g \in C^2[x_n, b]$ mit $f(x_0) = y_0$ und $g(x_n) = y_n$ sowie die Stützpunkte (x_0, y_0), (x_1, y_1), $\ldots$, (x_{n-1}, y_{n-1}), (x_n, y_n), $n \geq 3$, mit $x_0 < x_1 < \ldots < x_n$. Zu diesen Stützpunkten soll eine interpolierende Splinefunktion S so erzeugt werden, dass die aus f, S und g zusammengesetzte Funktion auf dem gesamten Intervall $[a, b]$ zweimal stetig differenzierbar ist.

Dafür muss S die folgenden Randbedingungen erfüllen:

$$S'(x_0) = S'_0(x_0) = f'(x_0) = y'_0\,, \quad S'(x_n) = S'_{n-1}(x_n) = g'(x_n) = y'_n \tag{10.24}$$

$$S''(x_0) = S''_0(x_0) = f''(x_0) = y''_0\,, \quad S''(x_n) = S''_{n-1}(x_n) = g''(x_n) = y''_n \tag{10.25}$$

Zur Erfüllung dieser vier Randbedingungen fehlen einer kubischen Splinefunktion zwei Koeffizienten. Darum werden für die Segmente S_0 und S_{n-1} Polynome vierten Grades angesetzt

$$S(x) \equiv S_i(x)\,, \quad i = 0(1)n{-}1\,, \quad \text{mit} \tag{10.26}$$

$$\begin{aligned}
S_0(x) &= a_0 + b_0\,(x - x_0) + c_0\,(x - x_0)^2 + d_0\,(x - x_0)^3 + e_0\,(x - x_0)^4,\\
&\qquad x \in [x_0, x_1]\\
S_i(x) &= a_i + b_i\,(x - x_i) + c_i\,(x - x_i)^2 + d_i\,(x - x_i)^3,\\
&\qquad x \in [x_i, x_{i+1}]\,, \quad i = 1(1)n{-}2\\
S_{n-1}(x) &= a_{n-1} + b_{n-1}(x - x_{n-1}) + c_{n-1}(x - x_{n-1})^2 + d_{n-1}(x - x_{n-1})^3 + e_{n-1}(x - x_{n-1})^4,\\
&\qquad x \in [x_{n-1}, x_n]
\end{aligned}$$

Analog zum Vorgehen im Abschnitt 10.1.5 führt die Auswertung der Interpolationsbedingungen (10.4), der Anschlussbedingungen (10.5), (10.6) und (10.7) sowie der Randbedingungen (10.24) und (10.25) zur Berechnung aller Koeffizienten der Splinefunktion (10.26).

Der folgende Algorithmus enthält auch die Fälle, in denen bei x_0 oder x_n nur eine Randbedingung gestellt wird.

Algorithmus 10.23. (*Kombinierte Splinefunktion*)

Gegeben: (x_i, y_i), $i = 0(1)n$, $n \geq 3$, $x_0 < x_1 < \ldots < x_n$,
eine der unten angegebenen Randbedingungen (i) bis (v)

Gesucht: Die Koeffizienten der Polynome (10.26) $S_0(x), S_1(x), \ldots, S_{n-2}(x), S_{n-1}(x)$

1. $a_i = y_i, \qquad i = 0(1)n{-}1$
2. $h_i = x_{i+1} - x_i, \quad i = 0(1)n{-}1$
3. c_0 und c_n sind mittels einer der unten angegebenen Randbedingungen (i) bis (v) festzulegen
4. Berechnung der Koeffizienten $c_1, \ldots, c_{n-1}$ aus dem folgenden Gleichungssystem

   4.1 Erste Gleichung:

$$(h_0 + 2\,h_1)\,c_1 + h_1\,c_2 = 3\,\frac{y_2 - y_1}{h_1} - 6\,\frac{y_1 - y_0}{h_0} + 3\,y_0' + h_0\,c_0$$

   4.2 Gleichungen für $i = 2(1)n{-}2$, $n \geq 4$:

$$h_{i-1}\,c_{i-1} + 2\,(h_{i-1} + h_i)\,c_i + h_i\,c_{i+1} = 3\,\frac{y_{i+1} - y_i}{h_i} - 3\,\frac{y_i - y_{i-1}}{h_{i-1}}$$

   4.3 Letzte Gleichung:

$$h_{n-2}\,c_{n-2} + (2\,h_{n-2} + h_{n-1})\,c_{n-1} = 6\,\frac{y_n - y_{n-1}}{h_{n-1}} - 3\,\frac{y_{n-1} - y_{n-2}}{h_{n-2}} - 3\,y_n' + h_{n-1}\,c_n$$

5. Berechnung der Koeffizienten e_0 und e_{n-1}

   5.1 $e_0 = \dfrac{1}{h_0^3}\left(y_0' - \dfrac{y_1 - y_0}{h_0} + \dfrac{h_0}{3}\,(2\,c_0 + c_1)\right)$

   5.2 $e_{n-1} = \dfrac{1}{h_{n-1}^3}\left(\dfrac{y_n - y_{n-1}}{h_{n-1}} - y_n' + \dfrac{h_{n-1}}{3}\,(c_{n-1} + 2\,c_n)\right)$

6. $d_0 = \dfrac{1}{3\,h_0}\,(c_1 - c_0) - 2\,h_0\,e_0$

   $d_i = \dfrac{1}{3\,h_i}\,(c_{i+1} - c_i)\,, \quad i = 1(1)n{-}2$

   $d_{n-1} = \dfrac{1}{3\,h_{n-1}}\,(c_n - c_{n-1}) - 2\,h_{n-1}\,e_{n-1}$

7. $b_0 = y'_0$

$$b_i = \frac{y_{i+1} - y_i}{h_i} - \frac{h_i}{3}\,(c_{i+1} + 2\,c_i)\,, \quad i = 1(1)n{-}2$$

$$b_{n-1} = \frac{y_n - y_{n-1}}{h_{n-1}} - \frac{h_{n-1}}{3}\,(c_n + 2\,c_{n-1}) + h_{n-1}^3\,e_{n-1}$$

Randbedingungen

(i) $S'(x_0) = y'_0,\ S''(x_0) = y''_0,\ S'(x_n) = y'_n,\ S''(x_n) = y''_n$

3. $c_0 = \frac{1}{2}\,y''_0,\ c_n = \frac{1}{2}\,y''_n$

Randbedingungen

(ii) $S'(x_0) = y'_0,\ S'(x_n) = y'_n,\ S''(x_n) = y''_n$

3. $c_0 = \dfrac{1}{2}\left(\dfrac{3}{h_0}\left(\dfrac{y_1 - y_0}{h_0} - y'_0\right) - c_1\right),\ c_n = \dfrac{1}{2}\,y''_n$

4.1 Erste Gleichung:

$$\left(\frac{3}{2}\,h_0 + 2\,h_1\right)c_1 + h_1\,c_2 = 3\left(\frac{y_2 - y_1}{h_1} - \frac{1}{2}\left(3\,\frac{y_1 - y_0}{h_0} - y'_0\right)\right)$$

5.1 $e_0 = 0$

Randbedingungen

(iii) $S''(x_0) = y''_0,\ S'(x_n) = y'_n,\ S''(x_n) = y''_n$

3. $c_0 = \frac{1}{2}\,y''_0,\ c_n = \frac{1}{2}\,y''_n$

4.1 Erste Gleichung:

$$2\,(h_0 + h_1)\,c_1 + h_1\,c_2 = 3\,\frac{y_2 - y_1}{h_1} - 3\,\frac{y_1 - y_0}{h_0} - h_0\,c_0$$

5.1 $e_0 = 0$

7. $b_0 = \dfrac{y_1 - y_0}{h_0} - \dfrac{h_0}{3}\,(c_1 + 2\,c_0)$

Randbedingungen

(iv) $S'(x_0) = y'_0,\ S''(x_0) = y''_0,\ S'(x_n) = y'_n$

3. $c_0 = \frac{1}{2}\,y''_0,\ c_n = \dfrac{1}{2}\left(\dfrac{3}{h_{n-1}}\left(y'_n - \dfrac{y_n - y_{n-1}}{h_{n-1}}\right) - c_{n-1}\right)$

4.3 Letzte Gleichung:

$$h_{n-2}\,c_{n-2} + \left(2\,h_{n-2} + \frac{3}{2}\,h_{n-1}\right)c_{n-1} = 3\left(\frac{1}{2}\left(3\,\frac{y_n - y_{n-1}}{h_{n-1}} - y'_n\right) - \frac{y_{n-1} - y_{n-2}}{h_{n-2}}\right)$$

5.2 $e_{n-1} = 0$

Randbedingungen

(v) $S'(x_0) = y'_0,\ S''(x_0) = y''_0,\ S''(x_n) = y''_n$

3. $c_0 = \frac{1}{2}\, y''_0,\ c_n = \frac{1}{2}\, y''_n$

4.3 Letzte Gleichung:

$$h_{n-2}\, c_{n-2} + 2\,(h_{n-2} + h_{n-1})\, c_{n-1} = 3\,\frac{y_n - y_{n-1}}{h_{n-1}} - 3\,\frac{y_{n-1} - y_{n-2}}{h_{n-2}} - h_{n-1}\, c_n$$

5.2 $e_{n-1} = 0$

Bemerkung. Wenn bei x_0 die Steigung y'_0 und die Krümmung κ_0 der Splinekurve vorgegeben werden, folgt aus der Formel für die Krümmung für die zweite Randableitung

$$y''_0 = \kappa_0\,(1 + {y'_0}^2)^{3/2}.$$

Bei x_n gilt analog

$$y''_n = \kappa_n\,(1 + {y'_n}^2)^{3/2}.$$

Es folgt ein einfaches Beispiel für den Fall der Randbedingungen (i).

Beispiel 10.24.

Gegeben: Die Funktionen

$$\begin{aligned} f(x) &= 2 + \sqrt{4 - (x-2)^2}\,, && x \in [0.1, 2]\,, \\ g(x) &= 2 + 2\,(x-7)\,, && x \in [7, 8]\,, \end{aligned}$$

mit $f(2) = 4$ und $g(7) = 2$ sowie die Wertetabelle ($n = 3$)

i	0	1	2	3
x_i	2	3.5	5	7
y_i	4	3	1	2

Gesucht: Die kombinierte Splinefunktion (10.26) $S(x) \equiv S_i(x)$, $i = 0(1)2$, zu den Randbedingungen (i)

$$\begin{array}{lllllll} S'(x_0) & = & f'(2) & = & 0 & = & y'_0\,, \\ S''(x_0) & = & f''(2) & = & -\frac{1}{2} & = & y''_0\,, \\ S'(x_3) & = & g'(7) & = & 2 & = & y'_3\,, \\ S''(x_3) & = & g''(7) & = & 0 & = & y''_3\,. \end{array}$$

Lösung: Die Berechnung der Koeffizienten der Polynome $S_0(x)$, $S_1(x)$, $S_2(x)$ erfolgt nach dem Algorithmus 10.23.

1. $a_0 = 4,\ a_1 = 3,\ a_2 = 1.$
2. $h_0 = 1.5,\ h_1 = 1.5,\ h_2 = 2.$
3. $c_0 = -0.25,\ c_3 = 0.$
4. Die benötigten Steigungen sind
$$\frac{y_1 - y_0}{h_0} = -\frac{2}{3},\ \frac{y_2 - y_1}{h_1} = -\frac{4}{3},\ \frac{y_3 - y_2}{h_2} = \frac{1}{2}.$$
Wegen $n = 3$ besteht das Gleichungssystem nur aus der ersten und letzten Gleichung.

4.1 $4.5\,c_1 + 1.5\,c_2 = -4 + 4 + 0 - \frac{3}{8} = -\frac{3}{8}$

4.3 $1.5\,c_1 + 5\,c_2 = 3 + 4 - 6 + 0 = 1$

Die Lösungen sind
$$c_1 = -\tfrac{1}{6}, \quad c_2 = \tfrac{1}{4}.$$

5.1 $e_0 = \frac{8}{81}$

5.2 $e_2 = -\frac{1}{6}$

6. $d_0 = -\frac{5}{18},\ d_1 = \frac{5}{54},\ d_2 = \frac{5}{8}$

7. $b_0 = 0,\ b_1 = -\frac{31}{24},\ b_2 = -\frac{7}{6}$

Damit sind die Polynome

$$\begin{aligned}
S_0(x) &= 4 - \tfrac{1}{4}(x-2)^2 - \tfrac{5}{18}(x-2)^3 + \tfrac{8}{81}(x-2)^4, && x \in [2, 3.5],\\
S_1(x) &= 3 - \tfrac{31}{24}(x-3.5) - \tfrac{1}{6}(x-3.5)^2 + \tfrac{5}{54}(x-3.5)^3, && x \in [3.5, 5],\\
S_2(x) &= 1 - \tfrac{7}{6}(x-5) - \tfrac{1}{4}(x-5)^2 + \tfrac{5}{8}(x-5)^3 - \tfrac{1}{6}(x-5)^4, && x \in [5, 7].
\end{aligned}$$

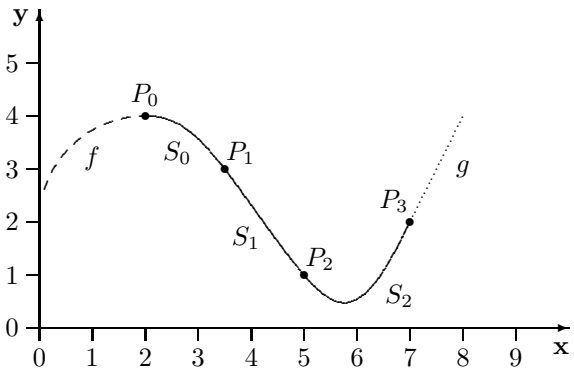


Abb. 10.35. Kombinierter Spline mit ersten und zweiten Randableitungen

□

Die kombinierten Polynom-Splines können auch als Komponenten parametrischer Splines zur Darstellung von ebenen Kurven oder Raumkurven Verwendung finden, wenn zusätzliche Randbedingungen erfüllt werden sollen.

10.1.8 Näherungsweise Ermittlung von Randableitungen durch Interpolation

Wenn die vorzugebenden Randableitungen y'_0, y'_n oder y''_0, y''_n für die Splinefunktion S nicht bekannt sind, können für sie geeignete Werte mit Hilfe der ersten bzw. zweiten Ableitungen von Interpolationspolynomen bei x_0 und x_n bereitgestellt werden (zu Interpolationspolynomen siehe Abschnitt 9.5).

Erste Randableitungen

Das Newtonsche Interpolationspolynom 2. Grades zu den Stützpunkten (x_0, y_0), (x_1, y_1), (x_2, y_2) ist mit den Steigungen

$$m_0 = \frac{y_1 - y_0}{x_1 - x_0}\,, \quad m_1 = \frac{y_2 - y_1}{x_2 - x_1} \tag{10.27}$$

$$N(x) = y_0 + m_0\,(x - x_0) + \frac{m_1 - m_0}{x_2 - x_0}\,(x - x_0)\,(x - x_1)\,. \tag{10.28}$$

Als Wert für die erste Randableitung bei x_0 eignet sich

$$y'_0 = N'(x_0) = m_0 - \frac{m_1 - m_0}{x_2 - x_0}\,(x_1 - x_0)\,.$$

Am rechten Rand wird das quadratische Interpolationspolynom zu den Stützpunkten (x_n, y_n), (x_{n-1}, y_{n-1}), (x_{n-2}, y_{n-2}) verwendet. Mit den Steigungen

$$m_{n-1} = \frac{y_n - y_{n-1}}{x_n - x_{n-1}}\,, \quad m_{n-2} = \frac{y_{n-1} - y_{n-2}}{x_{n-1} - x_{n-2}} \tag{10.29}$$

lautet es

$$N(x) = y_n + m_{n-1}\,(x - x_n) + \frac{m_{n-1} - m_{n-2}}{x_n - x_{n-2}}\,(x - x_n)\,(x - x_{n-1})\,. \tag{10.30}$$

Als Wert für die erste Randableitung bei x_n nimmt man

$$y'_n = N'(x_n) = m_{n-1} + \frac{m_{n-1} - m_{n-2}}{x_n - x_{n-2}}\,(x_n - x_{n-1})\,.$$

Bemerkung. Die lineare Interpolation liefert noch einfacher

$$y'_0 = m_0 \quad \text{und} \quad y'_n = m_{n-1}\,.$$

Ihre Verwendung hat zur Folge, dass die Segmente S_0 und S_{n-1} im Allgemeinen einen Wendepunkt besitzen. Daher ist die lineare Interpolation nicht zu empfehlen.

Zweite Randableitungen

Bei quadratischer Interpolation ergeben sich mit den zweiten Ableitungen der Polynome (10.28) und (10.30) bei x_0 bzw. x_n

$$y_0'' = N''(x_0) = 2\,\frac{m_1 - m_0}{x_2 - x_0}\,, \quad y_n'' = N''(x_n) = 2\,\frac{m_{n-1} - m_{n-2}}{x_n - x_{n-2}}\,.$$

Wenn mindestens vier Stützpunkte zur Verfügung stehen ($n \geq 3$), kann kubisch interpoliert werden. Das Newtonsche Interpolationspolynom 3. Grades zu den Stützpunkten (x_0, y_0), (x_1, y_1), (x_2, y_2), (x_3, y_3) lautet

$$N(x) = y_0 + b_1\,(x - x_0) + b_2\,(x - x_0)\,(x - x_1) + b_3\,(x - x_0)\,(x - x_1)\,(x - x_2)\,.$$

Seine zweite Ableitung ist

$$N''(x) = 2\,b_2 + 2\,b_3\,(x - x_0 + x - x_1 + x - x_3)\,.$$

Als Wert für die zweite Randableitung bei x_0 eignet sich

$$y_0'' = N''(x_0) = 2\,b_2 + 2\,b_3\,(2\,x_0 - x_1 - x_2)\,.$$

Die Koeffizienten ergeben sich wie folgt. Mit (10.27) ist

$$b_2 = \frac{m_1 - m_0}{x_2 - x_0}\,.$$

Mit $m_2 = \dfrac{y_3 - y_2}{x_3 - x_2}$ und $c_2 = \dfrac{m_2 - m_1}{x_3 - x_1}$ ergibt sich

$$b_3 = \frac{c_2 - b_2}{x_3 - x_0}\,.$$

Analog lautet das kubische Interpolationspolynom zu den Stützpunkten (x_n, y_n), (x_{n-1}, y_{n-1}), (x_{n-2}, y_{n-2}), (x_{n-3}, y_{n-3})

$$N(x) = y_n + b_1\,(x - x_n) + b_2\,(x - x_n)\,(x - x_{n-1}) + b_3\,(x - x_n)\,(x - x_{n-1})\,(x - x_{n-2})\,.$$

Für die zweite Randableitung bei x_n wird

$$y_n'' = N''(x_n) = 2\,b_2 + 2\,b_3\,(2\,x_n - x_{n-1} - x_{n-2})\,.$$

verwendet. Mit (10.29) ist

$$b_2 = \frac{m_{n-1} - m_{n-2}}{x_n - x_{n-2}}\,.$$

Weiter ist mit $m_{n-3} = \dfrac{y_{n-2} - y_{n-3}}{x_{n-2} - x_{n-3}}$ und $c_2 = \dfrac{m_{n-2} - m_{n-3}}{x_{n-1} - x_{n-3}}$

$$b_3 = \frac{b_2 - c_2}{x_n - x_{n-3}}\,.$$

10.1.9 Konvergenz und Fehlerabschätzungen interpolierender kubischer Splines

Anders als bei der Polynom-Interpolation ist die Konvergenz interpolierender Splines gegen die anzunähernde Funktion immer gewährleistet (siehe hierzu auch das in Abschnitt 9.6 angegebene Beispiel 18). Grundlegend ist der folgende Satz (s. [BOOR2001], [HALL1968]).

Satz 10.25.
Die Funktion $f : [a,b] \to \mathbb{R}$ sei viermal stetig differenzierbar, und S sei eine interpolierende Splinefunktion zu den Stützstellen $a = x_0 < x_1 < \ldots < x_n = b$, $n \geq 2$, und den Stützpunkten $\big(x_i, f(x_i)\big)$, $i = 0(1)n$.
Ferner seien

$$\begin{aligned} M &= \max |\, f^{(4)}(x)\,| \qquad \text{für} \quad x \in [a,b]\,, \\ h &= \max (x_{i+1} - x_i) \qquad \text{für} \quad i = 0(1)n{-}1\,. \end{aligned}$$

1. Wenn S ein Spline mit vorgegebenen 1. Randableitungen $S'(a) = f'(a)$, $S'(b) = f'(b)$ oder mit vorgegebenen 2. Randableitungen $S''(a) = f''(a)$, $S''(b) = f''(b)$ ist, dann gelten für $x \in [a,b]$ die Abschätzungen

$$\begin{aligned} |\, f(x) - S(x)\,| &\leq \tfrac{5}{384}\, M\, h^4, \\ |\, f'(x) - S'(x)\,| &\leq \tfrac{1}{24}\, M\, h^3, \\ |\, f''(x) - S''(x)\,| &\leq \tfrac{3}{8}\, M\, h^2. \end{aligned}$$

2. Wenn f periodisch ist mit der Periode $b - a$ und S der zugehörige periodische Spline, dann gelten für $x \in [a,b]$

$$|\, f^{(k)}(x) - S^{(k)}(x)\,| \leq \frac{3}{8}\, M\, h^{4-k}, \quad k = 0,1,2.$$

Wenn mit wachsender Anzahl der Teilintervalle deren maximale Länge h gegen Null strebt, konvergieren demnach S gegen f, S' gegen f' und S'' gegen f''.

Während für die im Satz unter 1. genannten Splinetypen $|\, f(x) - S(x)\,| = O(h^4)$ gilt, ist für natürliche interpolierende Splines mit $S''(a) = S''(b) = 0$ die globale Approximationsgüte nur $|\, f(x) - S(x)\,| = O(h^2)$.

Die im Satz 10.25 unter 1. für $|\, f(x) - S(x)\,|$ angegebene Abschätzung kann man verwenden, um eine äquidistante Anordnung der Stützstellen $a = x_0, x_1, \ldots, x_n = b$ derart zu bestimmen, dass für die Splinefunktion S zu den Stützpunkten $\big(x_i, f(x_i)\big)$, $i = 0(1)n$, und den Randbedingungen $S'(x_0) = f'(x_0)$, $S'(x_n) = f'(x_n)$ und für die Funktion f für alle $x \in [x_0, x_n]$ gilt

$$|\, f(x) - S(x)\,| \leq 0.5 \cdot 10^{-m}, \quad m \geq 1\,.$$

Mit der genannten Abschätzung

$$|\, f(x) - S(x)\,| \leq \frac{5}{384}\, M\, h^4 \leq 0.5 \cdot 10^{-m}$$

folgt für den maximalen Abstand h benachbarter Stützstellen

$$h \le \left(\frac{384}{5M}\, 0.5 \cdot 10^{-m} \right)^{1/4} .$$

Damit ergeben sich für die Anzahl n der mindestens erforderlichen Teilintervalle

$$n = \mathrm{INT}\left(\frac{x_n - x_0}{h} + 1 \right)$$

und für die feste Schrittweite

$$\Delta x = \frac{x_n - x_0}{n} .$$

Die äquidistanten Stützstellen sind also x_0 und

$$x_i = x_{i-1} + \Delta x\,, \quad i = 1(1)n\,.$$

Im folgenden Beispiel wird dieses Vorgehen angewendet.

Beispiel 10.26.

Gegeben: Die Funktion

$$f(x) = \int_0^x \frac{\sin t}{t}\, \mathrm{d}t\,, \quad x \in [0, 2]\,.$$

Gesucht: Eine Splinefunktion S mit den 1. Randableitungen $S'(0) = f'(0)$ und $S'(2) = f'(2)$, für die

$$|\, f(x) - S(x)\,| \le 0.000\,005$$

gilt für alle $x \in [0, 2]$.

Lösung: Benötigt werden die Ableitungen

$$\begin{aligned} f'(x) &= \frac{\sin x}{x}\,, \\ f^{(4)}(x) &= \left(\frac{3}{x^2} - \frac{6}{x^4} \right) \sin x + \left(\frac{6}{x^3} - \frac{1}{x} \right) \cos x\,. \end{aligned}$$

Durch Berechnung von $f^{(4)}(x)$ für $x = 1,\ 2,\ 1.5,\ 1.75,\ 1.875$ findet man

$$|\, f^{(4)}(x)\,| \le |\, f^{(4)}(1.88)\,| = 0.238 = M\,.$$

Damit ergeben sich

$$\begin{aligned} h &\le \left(\frac{384}{5 \cdot 0.238}\, 0.000\,005 \right)^{1/4} = 0.200\,419\,, \\ n &= \mathrm{INT}\left(\frac{2}{0.200\,419} + 1 \right) = 10\,, \\ \Delta x &= \frac{2}{10} = 0.2\,. \end{aligned}$$

Für die gesuchte Splinefunktion S müssen also die Stützpunkte

$$\big(0, f(0)\big),\ \big(0.2, f(0.2)\big),\ \ldots,\ \big(2, f(2)\big)$$

verwendet werden. Die 10 Funktionswerte $f(0.2), \ldots, f(2)$ werden mit dem Romberg-Quadraturverfahren mit 9 gültigen Dezimalen berechnet. Die Randableitungen sind

$$\begin{array}{lcll} S'(0) & = & f'(0) & = & 1\,, \\ S'(2) & = & f'(2) & = & 0.454\,648\,713\,. \end{array}$$

An einigen Stellen werden die exakten Funktionswerte mit denen der Splinefunktion verglichen.

x	$f(x)$	$S(x)$	$\lvert f(x) - S(x) \rvert$
$\pi/7$	0.443 807 121	0.443 806 876	0.000 000 245
$\pi/6$	0.515 689 196	0.515 688 851	0.000 000 345
$\pi/5$	0.614 700 083	0.614 699 927	0.000 000 156
$\pi/4$	0.758 975 881	0.758 975 855	0.000 000 026
$\pi/3$	0.985 458 844	0.985 458 411	0.000 000 433
$\pi/2$	1.370 762 168	1.370 761 939	0.000 000 229

An diesen Stellen beträgt die Abweichung höchsten 1/10 der geforderten maximalen Abweichung. □

10.2 Hermite-Splines fünften Grades

10.2.1 Definition der nichtparametrischen und parametrischen Hermite-Splines

Von der Funktion f seien an $n+1$ Knoten x_i neben den Funktionswerten $y_i = f(x_i)$ auch die Steigungen $y_i' = f'(x_i)$ gegeben, d. h. es liegen $n+1$ Wertetripel (x_i, y_i, y_i') für $i = 0(1)n$ vor. Hier lässt sich durch Hermite-Splines eine gute Anpassung erreichen, denn das Ziel ist jetzt die Konstruktion einer möglichst „glatten“ Kurve durch die vorgegebenen Punkte (x_i, y_i) mit den Steigungen y_i' mit Hilfe von Polynom-Splines fünften Grades (Hermite-Splines). Unter der Voraussetzung monotoner Anordnung der x_i

$$a = x_0 < x_1 < \ldots < x_n = b$$

kann die gesuchte Kurve durch eine nichtparametrische Splinefunktion S mit $S(x) \approx f(x)$ dargestellt werden, die sich stückweise aus Polynomen S_i fünften Grades für $x \in [x_i, x_{i+1}]$, $i = 0(1)n-1$, zusammensetzt.

Die S_i müssen dann gewissen Randbedingungen genügen, und es ergeben sich je nach Vorgabe der Randbedingungen die unten in Fall 1 aufgeführten verschiedenen Arten von Hermite-Splinefunktionen S. Lässt sich die Bedingung der strengen Monotonie der Knoten x_i nicht erfüllen, so müssen auch hier parametrische Hermite-Splines verwendet werden, s. dazu auch Abschnitt 10.1.

Fall 1: Arten von nichtparametrischen Hermite-Splinefunktionen und ihre Berechnung

Gesucht ist auf $[a, b] = [x_0, x_n]$ eine Splinefunktion S mit den Eigenschaften:

(1) S ist in $[a, b]$ dreimal stetig differenzierbar.

(2) S ist in jedem Intervall $[x_i, x_{i+1}]$ für $i = 0(1)n-1$ durch ein Polynom S_i fünften Grades gegeben.

(3) S erfüllt die Interpolationsbedingungen $S(x_i) = y_i,\ S'(x_i) = y'_i,\ i = 0(1)n$.

(4) Es sei eine der folgenden Randbedingungen (i) bis (v) vorgegeben:

(i) $S(x_0) = S(x_n),\ S'(x_0) = S'(x_n),\ S''(x_0) = S''(x_n),\ S'''(x_0) = S'''(x_n)$, dann heißt S eine *periodische Hermite-Splinefunktion.*

(ii) $S''(x_0) = S''(x_n) = 0$, dann heißt S eine *natürliche Hermite-Splinefunktion.*

(iii) $S''(x_0) = y''_0,\ S''(x_n) = y''_n$.

(iv) Krümmungsradien r_0 und r_n an den Stellen x_0 bzw. x_n.

(v) $S'''(x_0) = y'''_0,\ S'''(x_n) = y'''_n$.

Fall 2: Parametrische Hermite-Splines

Sind die x_i nicht monoton angeordnet, so muss analog zu den in Abschnitt 10.1.1 beschriebenen parametrischen kubischen Splines verfahren werden.

10.2.2 Berechnung der nichtparametrischen Hermite-Splines

Es werden die unter Fall 1 in Abschnitt 10.2.1 beschriebenen Hermite-Splines berechnet. Zur Konstruktion von S gemäß Eigenschaft (2) in Abschnitt 10.2.1 wird angesetzt

$$\left\{\begin{array}{lll} S(x) & \equiv & S_i(x) := a_i + b_i(x - x_i) + c_i(x - x_i)^2 \\ & & + d_i(x - x_i)^3 + e_i(x - x_i)^4 + f_i(x - x_i)^5, \\ & & x \in [x_i, x_{i+1}], \quad i = 0(1)n-1. \end{array}\right. \tag{10.31}$$

Der 6-parametrige Ansatz ergibt sich aus der Forderung $S \in C^3[a, b]$. Für die $6\,n$ Koeffizienten der n Polynome S_i müssen $6\,n$ Bedingungen aufgestellt werden, die aus den Eigenschaften (1) bis (4) in Abschnitt 10.2.1 folgen. Da hier (ebenfalls aus formalen Gründen) vier Koeffizienten (a_n, b_n, c_n, d_n) mehr berechnet werden, werden insgesamt $6\,n + 4$ Interpolations-, Anschluss- und Randbedingungen benötigt. Die Eigenschaften (1) und (3) von S führen zu folgenden Bedingungen:

$$\left.\begin{array}{lll} (a) & S_i(x_i) = y_i, & i = 0(1)n, \\ (b) & S'_i(x_i) = y'_i, & i = 0(1)n, \end{array}\right\} \; 2\,n + 2 \text{ Interpolationsbedingungen}$$

$$\left.\begin{array}{lll} (c) & S_i(x_i) = S_{i-1}(x_i), & i = 1(1)n, \\ (d) & S_i'(x_i) = S_{i-1}'(x_i), & i = 1(1)n, \\ (e) & S_i''(x_i) = S_{i-1}''(x_i), & i = 1(1)n, \\ (f) & S_i'''(x_i) = S_{i-1}'''(x_i), & i = 1(1)n, \end{array}\right\} 4\,n \text{ Anschlussbedingungen}$$

wobei formal $S_n(x_n) = a_n$, $S_n'(x_n) = b_n$, $S_n''(x_n) = 2\,c_n$, $S_n'''(x_n) = 3\,d_n$ gesetzt wird; zusätzlich gibt es zwei Randbedingungen gemäß Eigenschaft (4). Daraus folgt der

Algorithmus 10.27. (*Nichtperiodische Hermite-Splines*)

Gegeben: (x_i, y_i, y_i'), $i = 0(1)n$, x_i streng monoton angeordnet

Gesucht: Koeffizienten a_i, b_i, c_i, d_i, e_i, f_i der Polynome (10.31)

1. $a_i = y_i$, $b_i = y_i'$ für $i = 0(1)n$.

2. (ii) $c_0 = c_n = 0$

   (iii) $c_0 = \frac{1}{2}y_0''$, $c_n = \frac{1}{2}y_n''$

   (iv) $c_0 = (1 + b_0^2)^{3/2}/2r_0$, $c_n = (1 + b_n^2)^{3/2}/2r_n$

   (v) $c_0 = \frac{1}{3}\left[\frac{10(a_1-a_0)}{h_0^2} - \frac{2(2b_1+3b_0)}{h_0} - \frac{y_0'''h_0}{6} + c_1\right]$

   $c_n = \frac{1}{3}\left[\frac{10(a_{n-1}-a_n)}{h_{n-1}^2} + \frac{2(2b_{n-1}+3b_n)}{h_{n-1}} + \frac{y_n'''h_{n-1}}{6} + c_{n-1}\right]$

3. Gleichungssysteme für $c_1, c_2, \ldots, c_{n-1}$ mit $h_i = x_{i+1} - x_i$:

$$3\left(\frac{\alpha}{h_0} + \frac{1}{h_1}\right)c_1 - \frac{1}{h_1}c_2 = 10\left[\frac{a_2-a_1}{h_1^3} - \frac{a_1-a_0}{h_0^3}\right]$$

$$+4\left[\frac{b_0}{h_0^2} - \frac{3}{2}\left(\frac{1}{h_1^2} - \frac{1}{h_0^2}\right)b_1 - \frac{b_2}{h_1^2}\right] + \beta_1,$$

$$-\frac{c_{i-1}}{h_{i-1}} + 3\left(\frac{1}{h_{i-1}} + \frac{1}{h_i}\right)c_i - \frac{1}{h_i}c_{i+1} = 10\left[\frac{a_{i+1}-a_i}{h_i^3} - \frac{a_i-a_{i-1}}{h_{i-1}^3}\right]$$

$$+4\left[\frac{b_{i-1}}{h_{i-1}^2} - \frac{3}{2}\left(\frac{1}{h_i^2} - \frac{1}{h_{i-1}^2}\right)b_i - \frac{b_{i+1}}{h_i^2}\right] \text{ für } i = 2(1)n-2\,,$$

$$-\frac{c_{n-2}}{h_{n-2}} + 3\left(\frac{1}{h_{n-2}} + \frac{\alpha}{h_{n-1}}\right)c_{n-1} = 10\left[\frac{a_n-a_{n-1}}{h_{n-1}^3} - \frac{a_{n-1}-a_{n-2}}{h_{n-2}^3}\right]$$

$$+4\left[\frac{b_{n-2}}{h_{n-2}^2} - \frac{3}{2}\left(\frac{1}{h_{n-1}^2} - \frac{1}{h_{n-2}^2}\right)b_{n-1} - \frac{b_n}{h_{n-1}^2}\right] + \beta_2$$

Dabei sind die Größen α, β_1 und β_2 je nach Wahl der Randbedingungen (RB) (ii), (iii), (iv), (v) wie folgt zu setzen:

$$\alpha = \begin{cases} 1 & \text{für } RB(ii),(iii),(iv) \\ 8/9 & \text{für } RB(v) \end{cases}$$

$$\beta_1 = \begin{cases} 0 & \text{für } RB(ii) \\ y_0''/2h_0 & \text{für } RB(iii) \\ (1+b_0^2)^{3/2}/2h_0 r_0 & \text{für } RB(iv) \\ \frac{10}{3h_0^3}(a_1-a_0) - \frac{2}{3h_0^2}(2b_1+3b_0) - \frac{y_0'''}{18} & \text{für } RB(v) \end{cases}$$

$$\beta_2 = \begin{cases} 0 & \text{für } RB(ii) \\ y_n''/2h_{n-1} & \text{für } RB(iii) \\ (1+b_n^2)^{3/2}/2h_{n-1} r_n & \text{für } RB(iv) \\ -\frac{10}{3h_{n-1}^3}(a_n-a_{n-1}) + \frac{2}{3h_{n-1}^2}(3b_n+2b_{n-1}) + \frac{y_n'''}{18} & \text{für } RB(v) \end{cases}$$

4. $d_i = \frac{10}{h_i^3}(a_{i+1}-a_i) - \frac{2}{h_i^2}(2b_{i+1}+3b_i) + \frac{1}{h_i}(c_{i+1}-3c_i),\ i = 0(1)n{-}1$

$d_n = d_{n-1} - \frac{2}{h_{n-1}^2}(b_n - b_{n-1}) + \frac{2}{h_{n-1}}(c_n + c_{n-1})$

5. $e_i = \frac{1}{2h_i^3}(b_{i+1}-b_i) - \frac{1}{h_i^2}c_i - \frac{1}{4h_i}(d_{i+1}+5d_i),\ i = 0(1)n{-}1$

6. $f_i = \frac{1}{10h_i^3}(c_{i+1} - c_i - 3d_i h_i - 6e_i h_i^2),\ i = 0(1)n{-}1$

Das System 3. in Algorithmus 10.25 ist ein lineares Gleichungssystem für $n-1$ Koeffizienten c_1, c_2,..., c_{n-1}; es hat die Form $\boldsymbol{A}\boldsymbol{c} = \boldsymbol{a}$, mit

$$\boldsymbol{A} = \begin{pmatrix} 3\left(\frac{\alpha}{h_0}+\frac{1}{h_1}\right) & -\frac{1}{h_1} & & & \\ -\frac{1}{h_1} & 3\left(\frac{1}{h_1}+\frac{1}{h_2}\right) & -\frac{1}{h_2} & & \\ \ddots & \ddots & \ddots & & \\ & -\frac{1}{h_{n-3}} & 3\left(\frac{1}{h_{n-3}}+\frac{1}{h_{n-2}}\right) & -\frac{1}{h_{n-2}} & \\ & & -\frac{1}{h_{n-2}} & 3\left(\frac{1}{h_{n-2}}+\frac{\alpha}{h_{n-1}}\right) & \end{pmatrix}$$

$$\boldsymbol{a} = \begin{pmatrix} 10\left[\frac{a_2-a_1}{h_1^3} - \frac{a_1-a_0}{h_0^3}\right] + 4\left[\frac{b_0}{h_0^2} - \frac{3}{2}\left(\frac{1}{h_1^2}-\frac{1}{h_0^2}\right)b_1 - \frac{b_2}{h_1^2}\right] + \beta_1 \\ 10\left[\frac{a_3-a_2}{h_2^3} - \frac{a_2-a_1}{h_1^3}\right] + 4\left[\frac{b_1}{h_1^2} - \frac{3}{2}\left(\frac{1}{h_2^2}-\frac{1}{h_1^2}\right)b_2 - \frac{b_3}{h_2^2}\right] \\ \vdots \\ 10\left[\frac{a_n-a_{n-1}}{h_{n-1}^3} - \frac{a_{n-1}-a_{n-2}}{h_{n-2}^3}\right] + 4\left[\frac{b_{n-2}}{h_{n-2}^2} - \frac{3}{2}\left(\frac{1}{h_{n-1}^2}-\frac{1}{h_{n-2}^2}\right)b_{n-1} - \frac{b_n}{h_{n-1}^2}\right] + \beta_2 \end{pmatrix}$$

$\boldsymbol{c} = (c_1, c_2, \ldots, c_{n-1})^{\mathsf{T}}$.

Eigenschaften der Matrix A

Die Matrix $\boldsymbol{A}$ ist tridiagonal, symmetrisch, stark diagonal dominant, besitzt positive Hauptdiagonalelemente und negative, von Null verschiedene Nebendiagonalelemente; sie ist also positiv definit und damit streng regulär. Das Gleichungssystem ist folglich eindeutig lösbar nach der in Abschnitt 4.10 beschriebenen Methode. Pivotisierung und Nachiteration sind überflüssig.

Algorithmus 10.28. (*Periodische Hermite-Splines*)

Gegeben: (x_i, y_i, y_i'), $i = 0(1)n$, streng monotone Anordnung der x_i

Gesucht: Koeffizienten a_i, b_i, c_i, d_i, e_i, f_i der Polynome (10.31) mit den Randbedingungen $S^{(k)}(x_0) = S^{(k)}(x_n)$ für k = 0(1)3

1. $a_i = y_i$, $b_i = y_i'$ für $i = 0(1)n$

2. $c_0 = c_n$, $c_1 = c_{n+1}$, $a_1 = a_{n+1}$, $b_1 = b_{n+1}$, $h_0 = h_n$

3. Gleichungssystem für $c_1, c_2, \ldots, c_n$ mit $h_i = x_{i+1} - x_i$:

$$3\left(\frac{1}{h_0} + \frac{1}{h_1}\right) c_1 - \frac{1}{h_1} c_2 - \frac{1}{h_0} c_n = 10\left[\frac{a_2 - a_1}{h_1^3} - \frac{a_1 - a_0}{h_0^3}\right]$$

$$+4\left[\frac{b_0}{h_0^3} - \frac{3}{2}\left(\frac{1}{h_1^2} - \frac{1}{h_0^2}\right) b_1 - \frac{b_2}{h_1^2}\right],$$

$$-\frac{c_{i-1}}{h_{i-1}} + 3\left(\frac{1}{h_{i-1}} + \frac{1}{h_i}\right) c_i - \frac{1}{h_i} c_{i+1} = 10\left[\frac{a_{i+1} - a_i}{h_i^3} - \frac{a_i - a_{i-1}}{h_{i-1}^3}\right]$$

$$+4\left[\frac{b_{i-1}}{h_{i-1}^3} - \frac{3}{2}\left(\frac{1}{h_i^2} - \frac{1}{h_{i-1}^2}\right) b_i - \frac{b_{i+1}}{h_i^2}\right],\ i = 2(1)n{-}1\,,$$

$$-\frac{c_1}{h_0} - \frac{1}{h_{n-1}} c_{n-1} + 3\left(\frac{1}{h_{n-1}} + \frac{1}{h_0}\right) c_n = 10\left[\frac{a_1 - a_n}{h_0^3} - \frac{a_n - a_{n-1}}{h_{n-1}^3}\right]$$

$$+4\left[\frac{b_{n-1}}{h_{n-1}^3} - \frac{3}{2}\left(\frac{1}{h_0^2} - \frac{1}{h_{n-1}^2}\right) b_n - \frac{b_1}{h_0^2}\right]$$

4. $$d_i = \frac{10}{h_i^3}(a_{i+1} - a_i) - \frac{2}{h_i^2}(2b_{i+1} + 3b_i) + \frac{1}{h_i}(c_{i+1} - 3c_i),\ i = 0(1)n{-}1$$

$$d_n = d_{n-1} - \frac{2}{h_{n-1}^2}(b_n - b_{n-1}) + \frac{2}{h_{n-1}}(c_n + c_{n-1})$$

5. $e_i = \frac{1}{2h_i^3}(b_{i+1} - b_i) - \frac{1}{h_i^2}c_i - \frac{1}{4h_i}(d_{i+1} + 5d_i),\ i = 0(1)n{-}1$

6. $f_i = \frac{1}{10h_i^3}(c_{i+1} - c_i - 3d_ih_i - 6e_ih_i^2),\ i = 0(1)n{-}1$

Das System 3. in Algorithmus 10.27 ist ein lineares Gleichungssystem von n Gleichungen für die n Unbekannten c_1, c_2,..., c_n mit einer zyklisch tridiagonalen, symmetrischen, stark diagonal dominanten Matrix mit positiven Hauptdiagonalelementen und negativen, von Null verschiedenen Elementen außerhalb der Hauptdiagonale; die Matrix ist also positiv definit und damit streng regulär. Das System sollte nach dem Gaußschen Algorithmus für zyklisch tridiagonale Matrizen gemäß Abschnitt 4.10 gelöst werden.

10.2.3 Berechnung der parametrischen Hermite-Splines

Sind Wertetripel (x_i, y_i, y_i'), $i = 0(1)n$ einer Kurve C gegeben, die x_i aber nicht streng monoton angeordnet, so wird näherungsweise eine Parameterdarstellung $\big(x(t), y(t)\big) \approx \big(S_x(t), S_y(t)\big)$ von C ermittelt, indem die Hermite-Splinefunktion S_x und S_y zu den Wertetripeln $(t_i, x_i, \dot{x}_i)$ bzw. $(t_i, y_i, \dot{y}_i)$ gemäß 10.2.1, Fall 1, berechnet werden mit monoton angeordneten Parameterwerten t_i; ihre Berechnung erfolgt analog zu Abschnitt 10.1.6. Die $\dot{x}_i, \dot{y}_i$ sind nur aus den vorgegebenen y_i', $i = 0(1)n$, wegen $\dot{x}_i^2 + \dot{y}_i^2 = 1$ und $y' = \dot{y}_i/\dot{x}_i$ wie folgt zu ermitteln:

$$\dot{x}_i = \frac{\sigma_i}{+\sqrt{1 + {y_i'}^2}} \quad \text{mit} \quad \begin{cases} \sigma_i = \operatorname{sgn}(\boldsymbol{x}_{i+1} - \boldsymbol{x}_i)^{\mathsf{T}}\dot{\boldsymbol{x}}_{i+} & \text{für } i = 0(1)n{-}1, \\ \sigma_i = \operatorname{sgn}(\boldsymbol{x}_n - \boldsymbol{x}_{n-1})^{\mathsf{T}}\dot{\boldsymbol{x}}_{n+} & \text{für } i = n, \end{cases}$$
$$\dot{y}_i = \dot{x}_i y_i' \quad \text{für} \quad i = 0(1)n,$$

mit den Bezeichnungen

$$\boldsymbol{x}_i := \begin{pmatrix} x_i \\ y_i \end{pmatrix}, \quad \dot{\boldsymbol{x}}_i := \begin{pmatrix} \dot{x}_i \\ \dot{y}_i \end{pmatrix}, \quad \dot{\boldsymbol{x}}_{i+} := \begin{pmatrix} |\dot{x}_i| \\ |\dot{x}_i| y_i' \end{pmatrix}.$$

Das Vorzeichen σ_i von $\dot{x}_i$ wurde so bestimmt, dass der von $(\boldsymbol{x}_{i+1} - \boldsymbol{x}_i)$ und $\dot{\boldsymbol{x}}_i$ eingeschlossene Winkel immer $< \pi/2$ ist, d. h. für das Skalarprodukt $(\boldsymbol{x}_{i+1} - \boldsymbol{x}_i)^{\mathsf{T}}\dot{\boldsymbol{x}}_i > 0$ gilt. Falls für ein festes i das Skalarprodukt $(\boldsymbol{x}_{i+1} - \boldsymbol{x}_i)^{\mathsf{T}}\dot{\boldsymbol{x}}_{i+}$ verschwindet, wird $\sigma_i = sgn(\boldsymbol{x}_i - \boldsymbol{x}_{i-1})^{\mathsf{T}}\dot{\boldsymbol{x}}_{i+}$ gewählt, sofern $(\boldsymbol{x}_i - \boldsymbol{x}_{i-1})^{\mathsf{T}}\dot{\boldsymbol{x}}_{i+} \neq 0$ gilt, andernfalls ist das Problem nicht eindeutig lösbar. Ebenfalls nicht eindeutig ist die Vorgabe von $\boldsymbol{x}_0$, $\boldsymbol{x}_1$ und y_0' , wenn $(\boldsymbol{x}_1 - \boldsymbol{x}_0)^{\mathsf{T}}\dot{\boldsymbol{x}}_{0+} = 0$ gilt.

Im Falle einer vertikalen Tangente wird gesetzt:

$$\dot{\boldsymbol{x}}_{i+} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \text{mit} \quad \dot{\boldsymbol{x}}_i = \sigma_i \dot{\boldsymbol{x}}_{i+}.$$

Die Berechnung der Splinefunktionen S_x bzw. S_y erfolgt nun analog zu I. in Abschnitt 10.2.1. Dabei wird S_x zu den Wertetripeln $(t_i, x_i, \dot{x}_i)$ und S_y zu den Wertetripeln $(t_i, y_i, \dot{y}_i)$ berechnet. Die $\dot{x}_i$, $\dot{y}_i$ sind aus den y_i' wie zuvor beschrieben zu ermitteln. Als Randbedingungen können hier vorgegeben werden:

(1) natürliche Randbedingungen ,

(2) periodische Randbedingungen ,

(3) y_0'', y_n'' ,

(4) $(\ddot{x}_0, \ddot{y}_0)$, $(\ddot{x}_n, \ddot{y}_n)$,

(5) Krümmungsradien r_0, r_n ,

(6) $(\dddot{x}_0, \dddot{y}_0)$, $(\dddot{x}_n, \dddot{y}_n)$.

Die Splinefunktionen S_x und S_y mit den Randbedingungen (1) bis (6) werden wie folgt berechnet:

Zu (1): S_x, S_y sind natürlich. Die Berechnung von S_x zu den Wertetripeln $(t_i, x_i, \dot{x}_i)$ erfolgt nach Algorithmus 10.23 mit (i), indem in den Formeln x_i durch t_i, y_i durch x_i, y_i' durch $\dot{x}_i$ ersetzt wird. Die Berechnung von S_y zu den Wertetripeln $(t_i, y_i, \dot{y}_i)$ erfolgt nach Algorithmus 10.23 mit (i), indem man in den Formeln x_i durch t_i, y_i' durch $\dot{y}_i$ ersetzt wird, y_i bleibt.

Zu (2): S_x, S_y sind periodisch. Die Berechnung von S_x erfolgt nach Algorithmus 10.23, in den Formeln ist zunächst x_i durch t_i, y_i durch x_i und y_i' durch $\dot{x}_i$ zu ersetzen. Die Berechnung von S_y erfolgt nach Algorithmus 10.23 mit t_i statt x_i, y_i bleibt, $\dot{y}_i$ statt y_i' .

Zu (3): Berechnung von S_x gemäß Algorithmus 10.23 (iii), dort ist t_i statt x_i, x_i statt y_i, $\dot{x}_i$ statt y_i' zu setzen, und es sind die Randbedingungen $\ddot{x}_0 = 1$, $\ddot{x}_n = 1$ statt y_0'', y_n'' zu verwenden, wobei $\dot{x}_0$, $\dot{x}_n \neq 0$ sei. Berechnung von S_y gemäß Algorithmus 10.23 (iii) mit t_i statt x_i, y_i bleibt, $\dot{y}_i$ statt y_i' und den Randbedingungen $\ddot{y}_0$, $\ddot{y}_n$ statt y_0'', y_n''. Dabei werden $\ddot{y}_0$, $\ddot{y}_n$ wie folgt berechnet:

$$\ddot{y}_0 = \frac{1}{\dot{x}_0}(\dot{x}_0^3 y_0'' + \dot{y}_0), \quad \ddot{y}_n = \frac{1}{\dot{x}_n}(\dot{x}_n^3 y_n'' + \dot{y}_n)$$

Zu (4): Berechnung von S_x gemäß Algorithmus 10.23 (iii) mit t_i statt x_i, x_i statt y_i, $\dot{x}_i$ statt y_i', $\ddot{x}_0$ statt y_0'' und $\ddot{x}_n$ statt y_n'' .
Berechnung von S_y gemäß Algorithmus 10.23 (iii), indem dort t_i statt x_i, $\dot{y}_i$ statt y_i', $\ddot{y}_0$ statt y_0'' und $\ddot{y}_n$ statt y_n'' gesetzt wird.

Zu (5): Die Berechnung von S_x erfolgt gemäß Algorithmus 10.23 (iii), dort ist t_i statt x_i, x_i statt y_i, $\dot{x}_i$ statt y_i', $\ddot{x}_0$ statt y_0'' , $\ddot{x}_n$ statt y_n'' zu setzen. Dabei werden $\ddot{x}_0$, $\ddot{x}_n$ wie folgt ermittelt

$$\ddot{x}_0 = \begin{cases} -\frac{1}{r_0 \dot{y}_0} & \text{für} \quad \dot{x}_0 = 0, \\ 1 & \text{sonst.} \end{cases}$$

$$\ddot{x}_n = \begin{cases} -\frac{1}{r_n \dot{y}_n} & \text{für} \quad \dot{x}_n = 0, \\ 1 & \text{sonst.} \end{cases}$$

Die Berechnung von S_y erfolgt gemäß Algorithmus 10.23 (iii) mit t_i statt x_i, y_i bleibt, $\dot{y}_i$ statt y_i', $\ddot{y}_0$ statt y_0'', $\ddot{y}_n$ statt y_n'' . Dabei sind $\ddot{y}_0$ und $\ddot{y}_n$ aus den folgenden Formeln zu

berechnen:

$$\ddot{y}_0 = \begin{cases} 1 & \text{für} \quad \dot{x}_0 = 0, \\ \frac{1}{\dot{x}_0}\left(\frac{1}{r_0} + \dot{y}_0\right) & \text{sonst.} \end{cases}$$

$$\ddot{y}_n = \begin{cases} 1 & \text{für} \quad \dot{x}_n = 0, \\ \frac{1}{\dot{x}_n}\left(\frac{1}{r_n} + \dot{y}_n\right) & \text{sonst.} \end{cases}$$

Zu (6): Die Berechnung von S_x erfolgt nach Algorithmus 10.23 (v) mit t_i statt x_i, x_i statt y_i, $\dot{x}_i$ statt y_i', $\ddot{x}_0$ statt y_0''' und $\ddot{x}_n$ statt y_n'''. Die Berechnung von S_y wird nach dem Algorithmus 10.23 (v) mit t_i statt x_i, $\dot{y}_i$ bleibt, $\dot{y}_i$ statt y_i', $\ddot{y}_0$ statt y_0''', $\ddot{y}_n$ statt y_n''' vorgenommen.

Bei Vorgabe anderer Randbedingungen müssen die Formeln entsprechend umgerechnet werden. Die Formeln für den Fall der Vorgabe von Wertequadrupeln (x_i, y_i, y_i', y_i'') sind in [SPAT1986], S.55 ff. zu finden.

Beispiel 10.29.

Gegeben: Die folgende Tabelle von Messwerten:

i	0	1	2
x_i	2	4	8
y_i	3	4	3
y_i'	1	0	-0.5

Gesucht: a) Die natürliche kubische Splinefunktion S zu den Wertepaaren (x_i, y_i), $i = 0(1)2$. Sie besitzt die Segmente

$$S_0(x) = 3 + \tfrac{5}{8}(x-2) - \tfrac{1}{32}(x-2)^3, \qquad x \in [2,4]$$

$$S_1(x) = 4 + \tfrac{1}{4}(x-4) - \tfrac{3}{16}(x-4)^2 + \tfrac{1}{64}(x-4)^3, \quad x \in [4,8]$$

b) Die natürliche Hermite-Spline-Funktion

$$S : S_i(x) = a_i + b_i(x-x_i) + c_i(x-x_i)^2 + d_i(x-x_i)^3 + e_i(x-x_i)^4 + f_i(x-x_i)^5, \quad x \in [x_i, x_{i+1}], \qquad i = 0, 1$$

Lösung: Man erhält gemäß Algorithmus 10.25 für nichtperiodische Hermite-Splines mit natürlichen Randbedingungen für die Koeffizienten der S_i:

i	a_i	b_i	c_i	d_i	e_i	f_i
0	3	1	0	$-5/16$	$1/8$	$-1/64$
1	4	0	$-1/8$	$1/16$	$-5/256$	$1/512$
2	3	$-1/2$	0	$1/16$	—	—

Damit lauten die Polynome S_0, S_1:

$$S_0(x) = 3 + 1(x-2) - \tfrac{5}{16}(x-2)^3 + \tfrac{1}{8}(x-2)^4 - \tfrac{1}{64}(x-2)^5 \qquad \text{für } x \in [2,4],$$
$$S_1(x) = 4 - \tfrac{1}{8}(x-2)^2 + \tfrac{1}{16}(x-4)^3 - \tfrac{5}{256}(x-4)^4 + \tfrac{1}{512}(x-4)^5 \qquad \text{für } x \in [4,8].$$

Was passieren kann, wenn man die Möglichkeiten, die die Hermite-Splines bieten, überschätzt, zeigt die nächste Skizze. Hier wurde lediglich die Steigung im Punkt (2,3) einmal auf $\tan(63.43)^\circ = 2$ und einmal auf $\tan(89.43)^\circ = 100$ verändert.

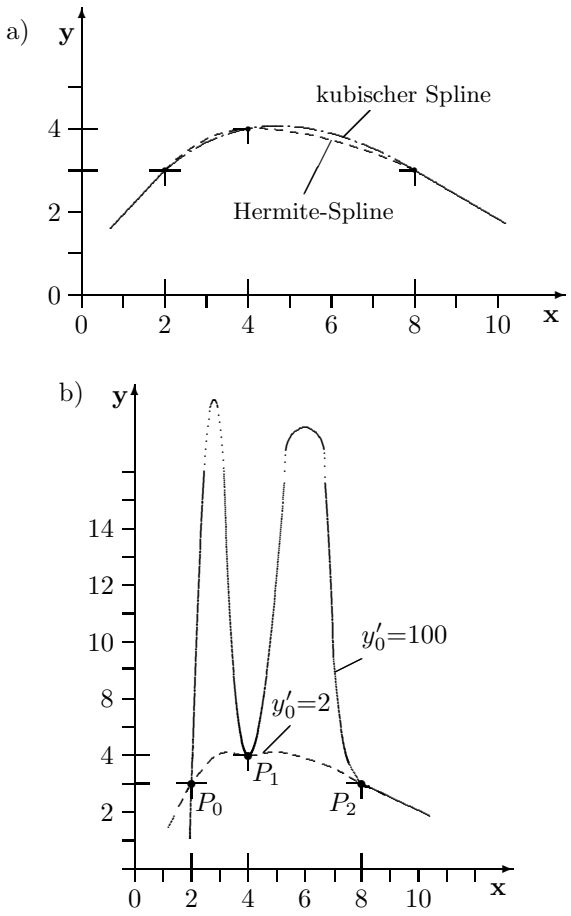


Abb. 10.36 a,b. Vergleich: Hermite Spline – Kubischer Spline

Die Abbildungen 10.36 a und b machen den unterschiedlichen Verlauf der beiden Splinefunktionen deutlich. Sie zeigen zugleich die Einsatzmöglichkeiten und die Grenzen der Hermite-Splines. Man kann mit ihrer Hilfe z. B. im Rohrleitungsbau, Flugzeug- , Schiff- und Karosseriebau Strakpläne bei Vorgabe relativ weniger Punkte und der zugehörigen Ableitungen zeichnen lassen. Dabei ist es auch ein Vorteil, dass die natürlichen Hermite-Splines es ermöglichen, eine Kurve aus geraden und gekrümmten Kurvenstücken glatt zusammenzusetzen, da an jeder Anschlussstelle die Steigung vorgegeben werden kann

und die Krümmung Null ist. Ein Nachteil der Hermite-Splines gegenüber den kubischen Splines ist ihre größere Welligkeit, da sie aus Polynomen fünften Grades zusammengesetzt sind. Bei sehr hohen Genauigkeitsanforderungen, wenn etwa die Toleranzgrenze im μm-Bereich liegt, kann man die Hermite-Splines nicht anwenden. □

Beispiel 10.30. (Parametrische Hermite-Splines)

Ausgegangen wird von der folgenden Wertetabelle:

I	$T(I)$	$X(I)$	$Y(I)$	$XT(I)$	$YT(I)$
1	0.00000E+00	1.00000E+00	1.00000E+00	3.16200E−01	9.48700E−01
2	1.11803E+00	1.50000E+00	2.00000E+00	7.07100E−01	7.07100E−01
3	1.82514E+00	2.00000E+00	2.50000E+00	1.00000E+00	0.00000E−00
4	2.53225E+00	2.50000E+00	2.00000E+00	7.07100E−01	−7.07100E−01
5	3.53225E+00	2.50000E+00	1.00000E+00	−8.94400E−01	−4.47200E−01
6	4.23935E+00	2.00000E+00	1.50000E+00	0.00000E+00	1.00000E+00
7	4.94646E+00	2.50000E+00	2.00000E+00	7.07100E−01	7.07100E−01
8	6.06450E+00	3.00000E+00	3.00000E+00	5.54700E−01	8.32100E−01
9	7.06450E+00	4.00000E+00	3.00000E+00	7.07100E−01	−7.07100E−01

Es ergeben sich die Koeffizienten für S_x, S_y wie folgt:

Spline für X(T)

I	$AX(I)$	$BX(I)$	$CX(I)$	$DX(I)$	$EX(I)$	$FX(I)$
1	1.0000E+00	3.1620E−01	0.0000E+00	−4.0013E−01	9.0480E−01	−4.0532E−01
2	1.5000E+00	7.0710E−01	−2.2065E−01	−1.4203E+00	4.5128E+00	−2.9173E+00
3	2.0000E+00	1.0000E+00	−9.4618E−03	−3.2428E+00	5.9150E+00	−3.0243E+00
4	2.5000E+00	7.0710E−01	1.6388E−01	−1.6344E+00	8.4312E−01	−7.9716E−02
5	2.5000E+00	−8.9440E−01	−4.7773E−01	9.4092E−01	3.2406E−01	−2.3972E−01
6	2.0000E+00	0.0000E+00	1.6429E+00	6.5890E−01	−3.7210E+00	2.1261E+00
7	2.5000E+00	7.0710E−01	−6.0544E−01	7.6487E−01	−7.3595E−01	3.1324E−01
8	3.0000E+00	5.5470E−01	8.1809E−01	1.3891E+00	−3.1584E+00	1.3965E+00

Spline für Y(T)

I	$AY(I)$	$BY(I)$	$CY(I)$	$DY(I)$	$EY(I)$	$FY(I)$
1	1.0000E+00	9.4870E−01	0.0000E+00	6.2472E−01	−1.1388E+00	4.8409E−01
2	2.0000E+00	7.0710E−01	3.1951E−01	1.5828E+00	−4.3940E+00	2.1446E+00
3	2.5000E+00	0.0000E+00	−1.9223E+00	−1.2205E−01	3.8789E+00	−2.6328E+00
4	2.0000E+00	−7.0710E−01	1.4691E−01	−2.3152E+00	2.4652E+00	−5.8984E−01
5	1.0000E+00	−4.4720E−01	2.0942E+00	1.6472E+00	−4.9928E+00	2.4606E+00
6	1.5000E+00	1.0000E+00	−6.9055E−01	−1.7160E−01	1.3149E+00	−7.3482E−01
7	2.0000E+00	7.0710E−01	2.9227E−01	−1.2650E−01	1.0561E−01	−8.2498E−02
8	3.0000E+00	8.3210E−01	−4.9292E−01	−6.8544E−01	2.2834E−01	1.1792E−01

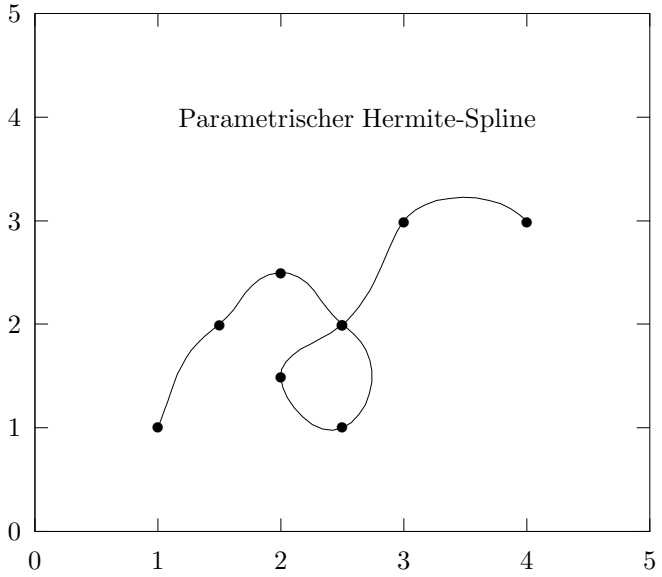


Abb. 10.37.

□

10.3 Polynomiale kubische Ausgleichssplines

10.3.1 Aufgabenstellung und Motivation

Im Abschnitt 10.1 wurde zu $n+1$ Stützpunkten (x_0, y_0), (x_1, y_1), ..., (x_n, y_n) mit monoton angeordneten Stützstellen $x_0 < x_1 < \ldots < x_n$ nach Vorgabe zweier zusätzlicher Bedingungen (z. B. Randbedingungen) eine auf dem Intervall $[x_0, x_n]$ zweimal stetig differenzierbare, interpolierende, kubische Splinefunktion S erzeugt.

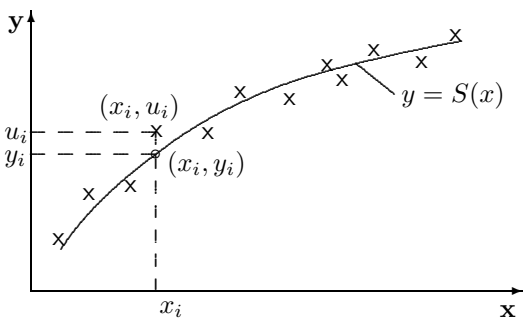


Abb. 10.38. Kubischer Ausgleichsspline

Bei praktischen Anwendungen werden an den Stützstellen x_i die zugehörigen Stützwerte durch Messungen ermittelt, so dass anstelle exakter Stützwerte y_i mit Fehlern behaftete Messwerte u_i zur Verfügung stehen. Es ist nicht zu erwarten, dass eine Splineinterpolation mit diesen Messwerten wegen deren Streuung zu einem brauchbaren Ergebnis führt. Daher stellt sich die Aufgabe, die mit Fehlern behafteten Messwerte u_i durch solche Stützwerte y_i zu ersetzen, die sich für eine Interpolation besser eignen (Abschnitt 10.3.2). Mit diesen Stützwerten y_i kann dann die Splineinterpolation wie im Abschnitt 10.1 erfolgen.

Anwendungsbeispiel

Zur Beherrschung der aerodynamischen Aufheizung eines Flugkörpers beim Wiedereintritt in die Erdatmosphäre werden im Stoßwellenlabor des Instituts für Luft- und Raumfahrt der RWTH Aachen in verschiedenen Stoßwellenrohren und Stoßwellenkanälen häufig instationäre Wandtemperatur- bzw. Wärmeübergangsmessungen an Wiedereintrittskörpern durchgeführt (s. Abbildungen 10.39a u. b). Dies geschieht mit Hilfe von Dünnschichtwiderstandsthermometern. Dabei handelt es sich um dünne Platinschichten, die auf Pyrex-Glas aufgetragen sind. Die Filmthermometer werden mit konstantem Strom betrieben. Sie werden beim Experiment vom hocherhitzten Versuchsgas des Stoßwellenrohres bzw. -kanals umströmt. Die hierdurch eintretende Änderung ihrer Oberflächentemperatur bewirkt eine entsprechende Zunahme des Thermometerwiderstandes, die als Spannungsänderung auf einem Oszillographen sichtbar gemacht wird. Die hier demonstrierten Ergebnisse wurden in der von G. Engeln-Müllges ausgegebenen Studienarbeit [SCHU1977] ermittelt, die Bilder wurden vom Institut für Hochtemperatur-Gasdynamik der RWTH Aachen zur Verfügung gestellt.

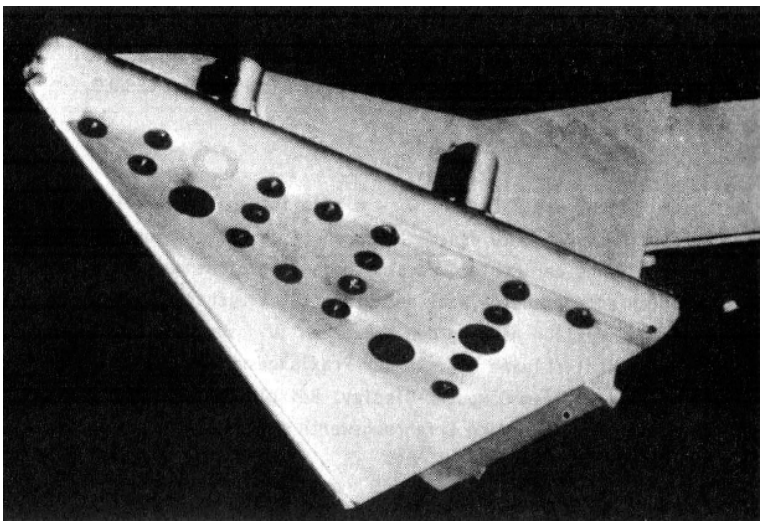


Abb. 10.39a. Deltaflügel mit Modellhalter, Drucksonden und Temperaturfühler

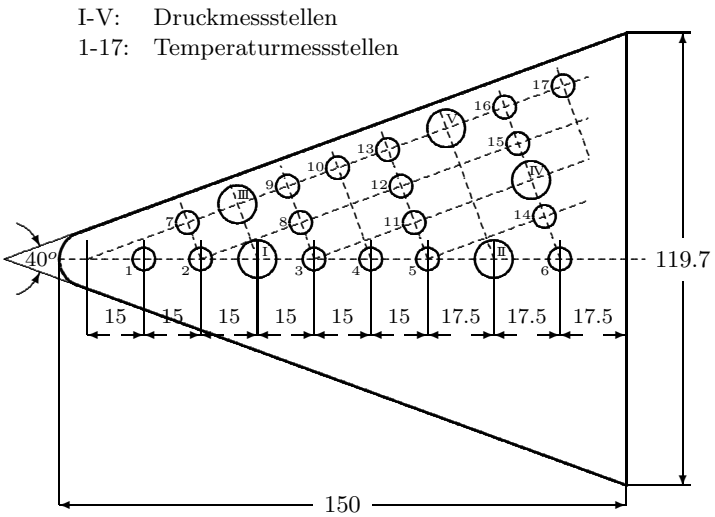


Abb. 10.39b. Grundriss des Deltaflügels mit der Lage der Filmthermometer und Drucksonden

Aus der zeitlichen Oberflächentemperaturänderung $\delta T(t)$ lässt sich dann der Wärmefluss auf das Filmthermometer bestimmen. Die Oszillogramme erlauben eine schnelle qualitative Beurteilung der Messung. Um jedoch von der qualitativen Auswertung der Oszillogramme loszukommen, werden die Messsignale analog auf Magnetband aufgezeichnet und anschließend mit Hilfe eines Analog-Digital-Wandlers digitalisiert und aufgezeichnet. Das folgende Bild zeigt ein mit 500 Messwerten digitalisiertes Oszillogramm des zeitlichen Spannungsverlaufes.

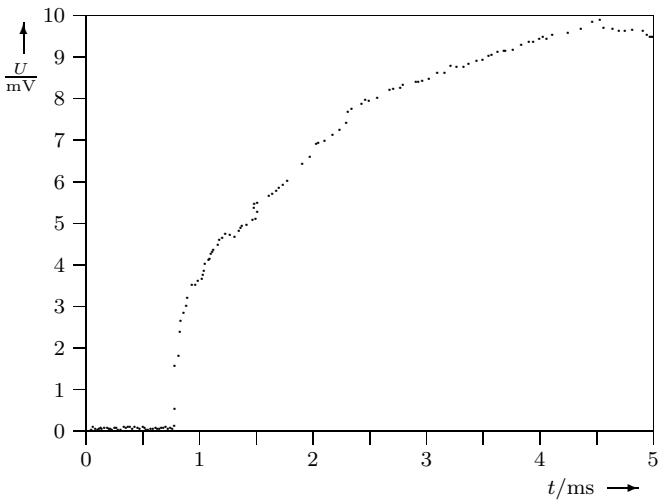


Abb. 10.40. Digitalisiertes Oszillogramm

Die Abbildung 10.40 veranschaulicht sehr deutlich die Notwendigkeit eines Ausgleichs der hier durch elektrische Störungen (Signalrauschen, Netzstörungen usw.) hervorgeru-

fenen Streuung der Messwerte. Im Folgenden werden zur Beschreibung des gemessenen Zusammenhangs Ausgleichssplines mit verschiedenen Gewichten für ein Beispiel mit 179 Messpunkten angegeben (Abbildung 10.41). An den Abbildungen ist zu erkennen, dass man sich für kleine Gewichte w_i der Ausgleichsgerade nähert, und dass für große w_i nahezu Interpolation vorliegt.

Um nun dem Nullpunkt nahe zu kommen, wurde speziell in diesem Beispiel das Gewicht $w_0 = 10^6$ gewählt, alle übrigen $w_i = 10^2$. Man kann aber auch für den Nullpunkt direkt die Interpolationsbedingung $\Phi(0) = 0$ ansetzen.

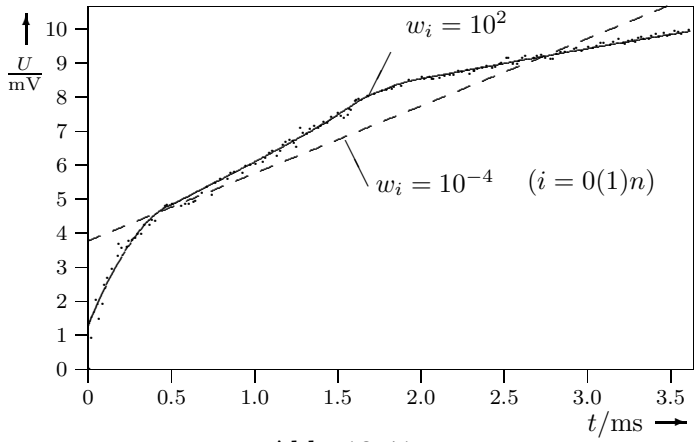


Abb. 10.41.

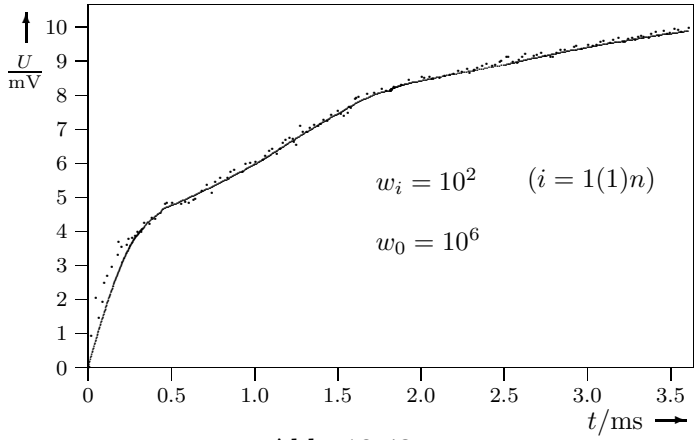


Abb. 10.42.

Aus der Abbildung erkennt man, dass die Kurve zwar nun im Nullpunkt beginnt, aber der Anstieg den Messwerten nicht gerecht wird. Im Folgenden wird noch ein aus mehreren getesteten Fällen sehr gutes Ergebnis herausgegriffen mit $w_0 = 10^6,\ w_1 = 10^4,\ w_i = 10^2$ für $i \geq 2$, (s. Abb. 10.43)

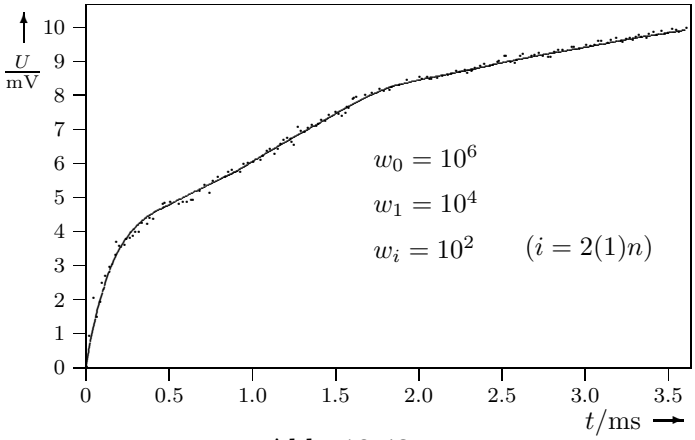


Abb. 10.43.

10.3.2 Konstruktion der nichtparametrischen Ausgleichssplines

Im Folgenden wird gezeigt, wie zu den gegebenen Messpunkten (x_i, u_i), $i = 0(1)n$, mit

$$x_0 < x_1 < \ldots < x_n$$

Ersatzpunkte (x_i, y_i) konstruiert werden können, zu denen dann wie im Abschnitt 10.1 die interpolierende kubische Splinefunktion

$$\begin{aligned} S(x) \equiv S_i(x) &= a_i + b_i\,(x - x_i) + c_i\,(x - x_i)^2 + d_i\,(x - x_i)^3\,, \\ x &\in [x_i, x_{i+1}]\,, i = 0(1)n{-}1\,, \end{aligned} \tag{10.32}$$

erzeugt wird. Dabei sind die Fälle

(a) S nicht periodisch

(b) S periodisch

zu unterscheiden.

Für die Stützwerte der Ersatzpunkte verwendet man den Ansatz

$$y_i = u_i - \frac{1}{w_i}\,r_i\,, \quad i = 0(1)n\,, \tag{10.33}$$

mit $r_i \neq 0$ und den vorzugebenden Gewichten w_i, $w_i > 0$. Der Abstand eines Messpunktes von seinem Ersatzpunkt ist

$$|\,u_i - y_i\,| = \frac{1}{w_i}\,|r_i|\,.$$

Bei festem r_i ist dieser Abstand umso kleiner, je größer das Gewicht w_i ist; w_i kann daher als ein Maß für die Qualität des Messpunktes (x_i, u_i) angesehen werden.

Zur Bestimmung der Zahlen r_i benutzt man die Eigenschaft der Splinefunktion (10.32), nur zweimal stetig differenzierbar zu sein. Die dritten Ableitungen benachbarter Segmente von S am gemeinsamen Knoten sind verschieden:

$$S'''_{i-1}(x_i) \neq S'''_i(x_i)\,, \quad i = 1(1)n{-}1\,.$$

Daher setzt man

$$r_i = S'''_i(x_i) - S'''_{i-1}(x_i)\,;$$

mit $S'''_i(x) = 6\,d_i$ ergibt sich

$$r_i = 6\,(d_i - d_{i-1})\,, \quad i = 1(1)n{-}1\,,$$

und durch Einsetzen in (10.33)

$$y_i = u_i - \frac{6}{w_i}\,(d_i - d_{i-1})\,, \quad i = 1(1)n{-}1\,. \tag{10.34}$$

Für jede kubische Splinefunktion gilt $\big($siehe (10.15)$\big)$ mit $h_i = x_{i+1} - x_i$

$$d_i = \frac{1}{3\,h_i}\,(c_{i+1} - c_i)\,, \quad i = 0(1)n{-}1\,. \tag{10.35}$$

Damit folgt für (10.34)

$$y_i = u_i - \frac{2}{w_i}\left(\frac{1}{h_i}\,c_{i+1} - \left(\frac{1}{h_i} + \frac{1}{h_{i-1}}\right)c_i + \frac{1}{h_{i-1}}\,c_{i-1}\right), \quad i = 1(1)n{-}1\,. \tag{10.36}$$

(a) S nicht periodisch

Da im Allgemeinen $S'''_0(x_0) \neq 0$ und $S'''_{n-1}(x_n) \neq 0$ sind, setzt man $r_0 = S'''_0(x_0)$ und $r_n = -S'''_{n-1}(x_n)$. Mit $S'''_i(x) = 6\,d_i$ folgen aus (10.33) für $i = 0$ und $i = n$

$$y_0 = u_0 - \frac{6}{w_0}\,d_0\,, \quad y_n = u_n + \frac{6}{w_n}\,d_{n-1}\,. \tag{10.37}$$

Mit (10.35) ergeben sich weiter

$$\left\{\begin{array}{lcl} y_0 & = & u_0 - \dfrac{2}{w_0\,h_0}\,(c_1 - c_0)\,, \\[2ex] y_n & = & u_n + \dfrac{2}{w_n\,h_{n-1}}\,(c_n - c_{n-1})\,. \end{array}\right. \tag{10.38}$$

(b) S periodisch

Das Periodenintervall ist $[x_0, x_n]$. In diesem Fall müssen

$$u_0 = u_n \qquad \text{und} \qquad w_0 = w_n$$

sein. Die Segmente S_{n-1} und S_0 sind bei x_n bzw. x_0 als benachbart anzusehen. Daher setzt man

$$r_0 = S_0'''(x_0) - S_{n-1}'''(x_n) = r_n \,.$$

Mit $S_i'''(x) = 6\,d_i$ folgt aus (10.33)

$$y_0 = u_0 - \frac{6}{w_0}\,(d_0 - d_{n-1}) = y_n \,. \tag{10.39}$$

Mit (10.35) ergibt sich weiter

$$y_0 = u_0 - \frac{2}{w_0}\Big(\frac{1}{h_0}\,(c_1 - c_0) - \frac{1}{h_{n-1}}\,(c_n - c_{n-1})\Big) = y_n \,. \tag{10.40}$$

Nach (10.36), (10.38) und (10.40) können in beiden Fällen die Ersatzstützwerte $y_0, y_1, \ldots, y_n$ bestimmt werden, wenn die Gewichte $w_0, w_1, \ldots, w_n$ vorgegeben und die Koeffizienten $c_0, c_1, \ldots, c_n$ bekannt sind. Es stellt sich also die Aufgabe, diese Koeffizienten zu ermitteln.

Die Lösung dieser Aufgabe wird für den nicht periodischen Fall (a) skizziert. In diesem Fall sind die Koeffizienten $c_1, \ldots, c_{n-1}$ der kubischen Splinefunktion (10.32) die Lösungen des linearen Gleichungssystems (vgl. (10.18) und Algorithmus 10.6)

$$h_{i-1}\,c_{i-1} + 2\,(h_{i-1} + h_i)\,c_i + h_i\,c_{i+1} = 3\left(\frac{y_{i+1} - y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}}\right), i = 1(1)n\text{–}1 \,. \tag{10.41}$$

In die Gleichungen dieses Systems werden rechts die Stützwerte (10.36) und (10.38) eingesetzt. Auf den rechten Seiten stehen dann die Messwerte $u_0, u_1, \ldots, u_n$. Alle c-Koeffizienten gehören auf die linken Seiten der Gleichungen. Die Koeffizienten c_0 und c_n werden mit Hilfe der Randbedingungen

(i) $S'(x_0) = y_0' \,, \quad S'(x_n) = y_n'$ oder

(ii) $S''(x_0) = y_0'' \,, \quad S''(x_n) = y_n''$

eliminiert. Aus (10.36) geht hervor, dass die in (10.41) rechts vorkommenden Stützwerte y_{i-1}, y_i, y_{i+1} die Koeffizienten c_{i-2}, c_{i-1}, c_i, c_{i+1}, c_{i+2} enthalten. Darum entsteht aus (10.41) für $n \geq 6$ ein lineares Gleichungssystem für $c_1, \ldots, c_{n-1}$ mit fünfdiagonaler Matrix. Mit der Lösung dieses Gleichungssystems und den Randbedingungen (i) oder (ii) stehen dann alle Koeffizienten $c_0, c_1, \ldots, c_n$ zur Verfügung.

Um die Diskussion von Sonderfällen, die für die Praxis kaum Bedeutung haben, zu vermeiden, wird $n \geq 6$ vorausgesetzt.

Mit der obigen Herleitung und unter Verwendung von (10.35), (10.36), (10.37), (10.39) und (10.16) ergibt sich der

Algorithmus 10.31.

Gegeben: Messpunkte (x_i, u_i), $i = 0(1)n$, $n \geq 6$, $x_0 < x_1 < \ldots x_n$; Gewichte w_i, $i = 0(1)n$, $w_i > 0$; Typ der Ausgleichssplinefunktion:

(i) mit vorgegebenen 1. Randableitungen
Algorithmus 10.32

(ii) mit vorgegebenen 2. Randableitungen
Algorithmus 10.33

(iii) periodisch mit Periodenintervall $[x_0, x_n]$, $u_0 = u_n$, $w_0 = w_n$
Algorithmus 10.34

Gesucht: Die Koeffizienten a_i, b_i, c_i, d_i, $i = 0(1)n{-}1$, der ausgleichenden kubischen Splinefunktion (10.32)

1. Berechnung der $h_i = x_{i+1} - x_i$, $i = 0(1)n{-}1$.
2. Berechnung der Koeffizienten c_i:
   Je nach Typ der Ausgleichssplinefunktion mit dem angegebenen Algorithmus.
3. Berechnung der Koeffizienten d_i:

$$d_i = \frac{1}{3\,h_i}\,(c_{i+1} - c_i), \quad i = 0(1)n{-}1$$

4. Berechnung der Koeffizienten a_i:
   (i) und (ii) (S nicht periodisch)

$$\begin{aligned} a_0 &= y_0 = u_0 - \tfrac{6}{w_0}\,d_0\,, \\ a_i &= y_i = u_i - \tfrac{6}{w_i}\,(d_i - d_{i-1})\,, \quad i = 1(1)n{-}1\,, \\ a_n &= y_n = u_n + \tfrac{6}{w_n}\,d_{n-1} \end{aligned}$$

   (iii) (S periodisch)

$$\begin{aligned} a_0 &= y_0 = u_0 - \tfrac{6}{w_0}\,(d_0 - d_{n-1}) = y_n = a_n\,, \\ a_i &= y_i = u_i - \tfrac{6}{w_i}\,(d_i - d_{i-1})\,, \quad i = 1(1)n{-}1 \end{aligned}$$

5. Berechnung der Koeffizienten b_i:

$$b_i = \frac{1}{h_i}\,(a_{i+1} - a_i) - \frac{h_i}{3}\,(c_{i+1} + 2\,c_i)\,, \quad i = 0(1)n{-}1$$

Bemerkung. Die konstruierten Ersatzstützpunkte sind $(x_i, y_i) = (x_i, a_i)$ für $i = 0(1)n$.

In den folgenden Algorithmen werden die Abkürzungen

$$h_i = x_{i+1} - x_i\,, \qquad i = 0(1)n{-}1\,,$$
$$W_i = \frac{6}{w_i}\,, \qquad i = 0(1)n\,,$$
$$H_i = \frac{1}{h_i} + \frac{1}{h_{i+1}}\,, \quad i = 0(1)n{-}2\,,$$

verwendet.

Algorithmus 10.32.
(zu (i): *Ausgleichsspline mit vorgegebenen 1. Randableitungen*).

Gegeben:

$$S'(x_0) = y'_0\,, \quad S'(x_n) = y'_n$$

1. Gleichungssystem für $c_1, c_2, \ldots, c_{n-1}$:
Mit

$$F_1 = \frac{h_0 - \frac{W_0}{h_0^2} - \frac{W_1}{h_0}H_0}{2h_0^2 + \frac{1}{h_0}(W_0 + W_1)}, \qquad F_2 = \frac{\frac{W_1}{h_0 h_1}}{2h_0^2 + \frac{1}{h_0}(W_0 + W_1)},$$

$$F_3 = \frac{\frac{W_{n-1}}{h_{n-2}h_{n-1}}}{2h_{n-1}^2 + \frac{1}{h_{n-1}}(W_{n-1} + W_n)}, \qquad F_4 = \frac{h_{n-1} - \frac{W_{n-1}}{h_{n-1}}H_{n-2} - \frac{W_n}{h_{n-1}^2}}{2h_{n-1}^2 + \frac{1}{h_{n-1}}(W_{n-1} + W_n)},$$

$$G_1 = W_1 H_0 + \frac{W_0}{h_0} - h_0^2, \quad G_2 = \frac{W_1}{h_1}, \quad G_3 = 3\big((u_1 - u_0) - y'_0 h_0\big), \quad G_4 = \frac{W_{n-1}}{h_{n-2}},$$

$$G_5 = \frac{W_n}{h_{n-1}} + W_{n-1}H_{n-2} - h_{n-1}^2, \qquad G_6 = 3\big(y'_n h_{n-1} - (u_n - u_{n-1})\big)$$

ergibt sich das folgende fünfdiagonale lineare Gleichungssystem:

$$\left(F_1 G_1 + 2(h_0 + h_1) + \frac{W_0}{h_0^2} + W_1 H_0^2 + \frac{W_2}{h_1^2}\right) c_1$$
$$+ \left(h_1 - \frac{W_1}{h_1}H_0 - \frac{W_2}{h_1}H_1 - F_1 G_2\right) c_2 + \left(\frac{W_2}{h_1 h_2}\right) c_3$$
$$= 3\left(\frac{u_2 - u_1}{h_1} - \frac{u_1 - u_0}{h_0}\right) - F_1 G_3,$$

$$\left(h_1 - \frac{W_1}{h_1}H_0 - \frac{W_2}{h_1}H_1 - F_2 G_1\right) c_1$$
$$+ \left(2(h_1 + h_2) + \frac{W_1}{h_1^2} + W_2 H_1^2 + \frac{W_3}{h_2^2} - F_2 G_2\right) c_2$$

$$+\left(h_2-\frac{W_2}{h_2}H_1-\frac{W_3}{h_2}H_2\right)c_3+\left(\frac{W_3}{h_2h_3}\right)c_4=3\left(\frac{u_3-u_2}{h_2}-\frac{u_2-u_1}{h_1}\right)-F_2G_3,$$

$$\left(\frac{W_{i-1}}{h_{i-2}h_{i-1}}\right)c_{i-2}+\left(h_{i-1}-\frac{W_{i-1}}{h_{i-1}}H_{i-2}-\frac{W_i}{h_{i-1}}H_{i-1}\right)c_{i-1}$$
$$+\left(2(h_{i-1}+h_i)+\frac{W_{i-1}}{h_{i-1}^2}+W_iH_{i-1}^2+\frac{W_{i+1}}{h_i^2}\right)c_i$$
$$+\left(h_i-\frac{W_i}{h_i}H_{i-1}-\frac{W_{i+1}}{h_i}H_i\right)c_{i+1}$$
$$+\left(\frac{W_{i+1}}{h_ih_{i+1}}\right)c_{i+2}=3\left(\frac{u_{i+1}-u_i}{h_i}-\frac{u_i-u_{i-1}}{h_{i-1}}\right),\quad i=3(1)n{-}3\ ,$$

$$\left(\frac{W_{n-3}}{h_{n-4}h_{n-3}}\right)c_{n-4}+\left(h_{n-3}-\frac{W_{n-3}}{h_{n-3}}H_{n-4}-\frac{W_{n-2}}{h_{n-3}}H_{n-3}\right)c_{n-3}$$
$$+\left(2(h_{n-3}+h_{n-2})+\frac{W_{n-3}}{h_{n-3}^2}+W_{n-2}H_{n-3}^2+\frac{W_{n-1}}{h_{n-2}^2}-F_3G_4\right)c_{n-2}$$
$$+\left(h_{n-2}-\frac{W_{n-2}}{h_{n-2}}H_{n-3}-\frac{W_{n-1}}{h_{n-2}}H_{n-2}+F_3G_5\right)c_{n-1}$$
$$=3\left(\frac{u_{n-1}-u_{n-2}}{h_{n-2}}-\frac{u_{n-2}-u_{n-3}}{h_{n-3}}\right)-F_3G_6\ ,$$

$$\left(\frac{W_{n-2}}{h_{n-3}h_{n-2}}\right)c_{n-3}+\left(h_{n-2}-\frac{W_{n-2}}{h_{n-2}}H_{n-3}-\frac{W_{n-1}}{h_{n-2}}H_{n-2}-F_4G_4\right)c_{n-2}$$
$$+\left(2(h_{n-2}+h_{n-1})+\frac{W_{n-2}}{h_{n-2}^2}+W_{n-1}H_{n-2}^2+\frac{W_n}{h_{n-1}^2}+F_4G_5\right)c_{n-1}$$
$$=3\left(\frac{u_n-u_{n-1}}{h_{n-1}}-\frac{u_{n-1}-u_{n-2}}{h_{n-2}}\right)-F_4G_6\ .$$

2. Berechnung von c_0 und c_n:

$$c_0 = \frac{3\big((u_1-u_0)-y_0'h_0\big)+\left(W_1H_0+\frac{W_0}{h_0}-h_0^2\right)c_1-\frac{W_1}{h_1}c_2}{2h_0^2+\frac{1}{h_0}(W_0+W_1)},$$

$$c_n = \frac{3\big(y_n'h_{n-1}-(u_n-u_{n-1})\big)+\left(W_{n-1}H_{n-2}+\frac{W_n}{h_{n-1}}-h_{n-1}^2\right)c_{n-1}-\frac{W_{n-1}}{h_{n-2}}c_{n-2}}{2h_{n-1}^2+\frac{1}{h_{n-1}}(W_{n-1}+W_n)}$$

Algorithmus 10.33.
(zu (ii): *Ausgleichsspline mit vorgegebenen 2. Randableitungen*).

Gegeben:

$$S''(x_0) = y_0'' \,, \quad S''(x_n) = y_n''$$

1. Fünfdiagonales Gleichungssystem für $c_1, c_2, \ldots, c_{n-1}$:

$$\left(2(h_0+h_1) + \frac{W_0}{h_0^2} + W_1 H_0^2 + \frac{W_2}{h_1^2}\right) c_1 + \left(h_1 - \frac{W_1}{h_1} H_0 - \frac{W_2}{h_1} H_1\right) c_2$$
$$+ \left(\frac{W_2}{h_1 h_2}\right) c_3 = 3\left(\frac{u_2-u_1}{h_1} - \frac{u_1-u_0}{h_0}\right) - \frac{y_0''}{2}\left(h_0 - \frac{W_0}{h_0^2} - \frac{W_1}{h_0} H_0\right),$$

$$\left(h_1 - \frac{W_1}{h_1} H_0 - \frac{W_2}{h_1} H_1\right) c_1 + \left(2(h_1+h_2) + \frac{W_1}{h_1^2} + W_2 H_1^2 + \frac{W_3}{h_2^2}\right) c_2$$
$$+ \left(h_2 - \frac{W_2}{h_2} H_1 - \frac{W_3}{h_2} H_2\right) c_3 + \left(\frac{W_3}{h_2 h_3}\right) c_4$$
$$= 3\left(\frac{u_3-u_2}{h_2} - \frac{u_2-u_1}{h_1}\right) - \frac{y_0''}{2}\left(\frac{W_1}{h_0 h_1}\right),$$

$$\left(\frac{W_{i-1}}{h_{i-2} h_{i-1}}\right) c_{i-2} + \left(h_{i-1} - \frac{W_{i-1}}{h_{i-1}} H_{i-2} - \frac{W_i}{h_{i-1}} H_{i-1}\right) c_{i-1}$$
$$+ \left(2(h_{i-1}+h_i) + \frac{W_{i-1}}{h_{i-1}^2} + W_i H_{i-1}^2 + \frac{W_{i+1}}{h_i^2}\right) c_i$$
$$+ \left(h_i - \frac{W_i}{h_i} H_{i-1} - \frac{W_{i+1}}{h_i} H_i\right) c_{i+1} + \left(\frac{W_{i+1}}{h_i h_{i+1}}\right) c_{i+2}$$
$$= 3\left(\frac{u_{i+1}-u_i}{h_i} - \frac{u_i - u_{i-1}}{h_{i-1}}\right), \quad i = 3(1)n{-}3 \,,$$

$$\left(\frac{W_{n-3}}{h_{n-4} h_{n-3}}\right) c_{n-4} + \left(h_{n-3} - \frac{W_{n-3}}{h_{n-3}} H_{n-4} - \frac{W_{n-2}}{h_{n-3}} H_{n-3}\right) c_{n-3}$$
$$+ \left(2(h_{n-3}+h_{n-2}) + \frac{W_{n-3}}{h_{n-3}^2} + W_{n-2} H_{n-3}^2 + \frac{W_{n-1}}{h_{n-2}^2}\right) c_{n-2}$$
$$+ \left(h_{n-2} - \frac{W_{n-2}}{h_{n-2}} H_{n-3} - \frac{W_{n-1}}{h_{n-2}} H_{n-2}\right) c_{n-1}$$
$$= 3\left(\frac{u_{n-1}-u_{n-2}}{h_{n-2}} - \frac{u_{n-2}-u_{n-3}}{h_{n-3}}\right) - \frac{y_n''}{2}\left(\frac{W_{n-1}}{h_{n-2} h_{n-1}}\right),$$

$$\left(\frac{W_{n-2}}{h_{n-3} h_{n-2}}\right) c_{n-3} + \left(h_{n-2} - \frac{W_{n-2}}{h_{n-2}} H_{n-3} - \frac{W_{n-1}}{h_{n-2}} H_{n-2}\right) c_{n-2}$$
$$+ \left(2(h_{n-2}+h_{n-1}) + \frac{W_{n-2}}{h_{n-2}^2} + W_{n-1} H_{n-2}^2 + \frac{W_n}{h_{n-1}^2}\right) c_{n-1}$$

$$= 3\left(\frac{u_n - u_{n-1}}{h_{n-1}} - \frac{u_{n-1} - u_{n-2}}{h_{n-2}}\right) - \frac{y_n''}{2}\left(h_{n-1} - \frac{W_{n-1}}{h_{n-1}}H_{n-2} - \frac{W_n}{h_{n-1}^2}\right).$$

2. $c_0 = \frac{y_0''}{2}, \qquad c_n = \frac{y_n''}{2}.$

Bemerkung. Mit $y_0'' = y_n'' = 0$ entsteht der natürliche Ausgleichsspline.

Algorithmus 10.34. (zu (iii): *Periodischer Ausgleichsspline*)

Voraussetzungen: $u_0 = u_n$, $w_0 = w_n$

1. Fünfdiagonales Gleichungssystem für $c_1, c_2, \ldots, c_{n-1}, c_n$:

$$\left(\frac{W_{i-1}}{h_{i-2}h_{i-1}}\right)c_{i-2} + \left(h_{i-1} - \frac{W_{i-1}}{h_{i-1}}H_{i-2} - \frac{W_i}{h_{i-1}}H_{i-1}\right)c_{i-1}$$
$$+ \left(2(h_{i-1}+h_i) + \frac{W_{i-1}}{h_{i-1}^2} + W_i H_{i-1}^2 + \frac{W_{i+1}}{h_i^2}\right)c_i$$
$$+ \left(h_i - \frac{W_i}{h_i}H_{i-1} - \frac{W_{i+1}}{h_i}H_i\right)c_{i+1} + \left(\frac{W_{i+1}}{h_i h_{i+1}}\right)c_{i+2}$$
$$= 3\left(\frac{u_{i+1}-u_i}{h_i} - \frac{u_i - u_{i-1}}{h_{i-1}}\right), \qquad i = 1(1)n\ .$$

Wegen der Periodizität sind hierin sämtliche negativen Indizes $(-k)$ zu ersetzen durch $(n-k)$ und sämtliche Indizes $(n+k)$ durch (k), $k \in \mathbb{N}$, ferner c_0 durch c_n.

2. $c_0 = c_n$.

Bemerkung (zur Wahl der Gewichte).
Für $w_i \to \infty$ folgt aus (10.33) $y_i = u_i$; der Ausgleichsspline ist dann der interpolierende Spline zu den Messpunkten (x_i, u_i). Es lässt sich zeigen, dass für $w_i \to 0$ die im Sinne der Fehlerquadratmethode ausgleichende Gerade entsteht. Man kann also durch die Wahl der Gewichte erreichen, dass die sich ergebende ausgleichende Splinefunktion nahe an den Messwerten u_i verläuft (große w_i) oder mehr ausgleicht (kleine w_i). In der Praxis geschieht dies interaktiv. Bei gleich bleibenden Versuchsbedingungen kann man dann Erfahrungswerte für die Gewichte verwenden.

10.3.3 Berechnung der parametrischen kubischen Ausgleichssplines

Wie im Abschnitt 10.1.6 seien $n+1$ Punkte P_i, $i = 0(1)n$, in der Ebene oder im Raum gegeben. Jetzt werde aber angenommen, dass die Koordinaten dieser Punkte durch Messungen ermittelt wurden und deshalb mit Fehlern behaftet sind.

Hier stellt sich die Aufgabe, diese Messpunkte durch solche Punkte zu ersetzen, deren Koordinaten die Fehler ausgleichen und sich für eine Interpolation mit parametrischen kubischen Splines besser eignen.

In der Ebene seien die Messpunkte $(\overline{x}_i, \overline{y}_i)$, $i = 0(1)n$. Die benötigten Knoten t_0, t_1, …, t_n werden mit der chordalen Parametrisierung zu den Messpunkten bereitgestellt:

$$t_0 = 0,\ t_{i+1} = t_i + \sqrt{(\overline{x}_{i+1} - \overline{x_i})^2 + (\overline{y}_{i+1} - \overline{y_i})^2}, \quad i = 0(1)n\,.$$

Nun werden mit dem im vorangehenden Abschnitt angegebenen Algorithmus 10.31 zu den Wertepaaren $(t_i, \overline{x}_i)$ und $(t_i, \overline{y}_i)$ (anstelle der Wertepaare (x_i, u_i)) die ausgleichenden kubischen Splinefunktionen mit den Segmenten

$$\left.\begin{aligned} S_{ix}(t) &= a_{ix} + b_{ix}\,(t-t_i) + c_{ix}\,(t-t_i)^2 + d_{ix}\,(t-t_i)^3, \\ S_{iy}(t) &= a_{iy} + b_{iy}\,(t-t_i) + c_{iy}\,(t-t_i)^2 + d_{iy}\,(t-t_i)^3, \end{aligned}\right\} \ t \in [t_i, t_{i+1}]\,, i = 0(1)n{-}1\,,$$

erzeugt. Mit ihnen ist dann der parametrische kubische Ausgleichsspline

$$\boldsymbol{x}(t) = \boldsymbol{S}(t) = \boldsymbol{S}_i(t) = \begin{pmatrix} S_{ix}(t) \\ S_{iy}(t) \end{pmatrix}.$$

Für beide Koordinaten können dieselben Gewichte w_0, w_1, …, w_n verwendet werden oder auch verschiedene Gewichte, also w_{x0}, w_{x1}, …, w_{xn} für die x-Koordinaten $\overline{x}_0$, $\overline{x}_1$, …, $\overline{x}_n$, und w_{y0}, w_{y1}, …, w_{yn} für die y-Koordinaten $\overline{y}_0$, $\overline{y}_1$, …, $\overline{y}_n$.

Zu Messpunkten $(\overline{x}_i, \overline{y}_i, \overline{z}_i)$, $i = 0(1)n$, im Raum werden die Knoten

$$t_0 = 0,\ t_{i+1} = t_i + \sqrt{(\overline{x}_{i+1} - \overline{x}_i)^2 + (\overline{y}_{i+1} - \overline{y}_i)^2 + (\overline{z}_{i+1} - \overline{z}_i)^2}, \quad i = 0(1)n{-}1$$

berechnet. Mit dem Algorithmus 10.31 wird zu den Wertepaaren $(t_i, \overline{z}_i)$ außerdem die ausgleichende kubische Splinefunktion mit den Segmenten

$$S_{iz}(t) = a_{iz} + b_{iz}\,(t-t_i) + c_{iz}\,(t-t_i)^2 + d_{iz}\,(t-t_i)^3, \quad t \in [t_i, t_{i+1}]\,, i = 0(1)n{-}1\,,$$

ermittelt. Der parametrische kubische Ausgleichsspline ist dann

$$\boldsymbol{x}(t) = \boldsymbol{S}(t) = \boldsymbol{S}_i(t) = \begin{pmatrix} S_{ix}(t) \\ S_{iy}(t) \\ S_{iz}(t) \end{pmatrix}.$$

10.4 Entscheidungshilfen für die Auswahl einer geeigneten Splinemethode

Wenn in der Ebene Stützpunkte (x_i, y_i), $i = 0(1)n$, gegeben sind, für deren x-Koordinaten $x_0 < x_1 < \ldots < x_n$ gilt, können nichtparametrische Splinefunktionen eingesetzt werden. Die monoton angeordneten Stützstellen (Knoten) x_i sollen in Bereichen starker Steigung enger beieinander liegen als in Bereichen geringer Steigung. Darum sind äquidistant angeordnete Stützstellen im Allgemeinen weniger gut geeignet als solche, die dem zu erwartenden Kurvenverlauf angepasst sind.

Bei beliebiger Anordnung der Stützpunkte in der Ebene oder im Raum werden parametrische Splines verwendet. Dann müssen den Stützpunkten Parameterwerte zugeordnet werden, in erster Linie mittels der chordalen Parametrisierung.

Am besten eignen sich zweimal stetig differenzierbare, kubische Splinefunktionen und parametrische kubische Splines mit vorgegebenen 1. Randableitungen. Wenn diese Randableitungen nicht zur Verfügung stehen, können geeignete Werte für sie bestimmt werden, indem durch die ersten und letzten drei/vier Stützpunkte das quadratische/kubische Interpolationspolynom gelegt und dessen Ableitung im betreffenden Randpunkt berechnet wird. Bei Änderungen der 1. Randableitungen kann deren Einfluss auf den Kurvenverlauf gut eingeschätzt werden.

Bei den natürlichen Splines verschwinden die 2. Randableitungen und somit auch die Krümmung in den Randstützpunkten. Sie können deshalb zweimal stetig differenzierbar geradlinig über die Randpunkte hinaus fortgesetzt werden.

Splines mit vorgegebenen 2. Randableitungen und not-a-knot-Splines sind von geringerer Bedeutung.

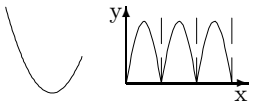
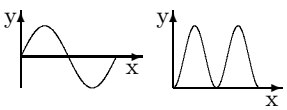
Wenn bei monoton angeordneten Stützstellen x_i die zugehörigen Stützwerte y_i fehlerbehaftet sind, liefert eine interpolierende Splinefunktion oft kein brauchbares Ergebnis. Dann kann mit einer Ausgleichssplinefunktion zu geeignet gewählten Gewichten ein zufriedenstellendes Ergebnis erzielt werden. Entsprechendes gilt für parametrische Ausgleichssplines, wenn alle Koordinaten der Stützpunkte mit Fehlern behaftet sein können.

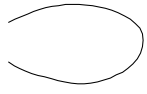
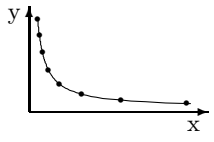
Wenn mit (x_i, y_i, y_i') an den monoton angeordneten Stützstellen x_i sowohl die Stützwerte y_i als auch die Steigungen y_i' vorgegeben werden sollen, wird eine dreimal stetig differenzierbare Hermite-Splinefunktion eingesetzt, deren Segmente Polynome fünften Grades sind. Weil solche Polynome stärker schwingen, zeigen die Hermite-Splines oft nicht so günstige Ergebnisse wie die kubischen Splinefunktionen.

Zu vorgegebenen (x_i, y_i, y_i') kann auch eine einmal stetig differenzierbare Subsplinefunktion erzeugt werden (Kapitel 11).

Die folgende (sehr grobe) Orientierungstabelle soll einen Eindruck vermitteln, welche der interpolierenden kubischen Splinearten sich zur Darstellung eines bestimmten Kurventyps eignet. Dabei werden die kubischen Bézier-Splines, die in Abschnitt 12.3 behandelt werden, sowie die Akima- und Renner-Subsplines in Kapitel 11 mit berücksichtigt. Zusätzlich zu den Informationen in der Tabelle sollte man unbedingt die entsprechenden Hinweise in den zugehörigen Abschnitten beachten.
In der Orientierungstabelle werden die folgenden Abkürzungen benutzt:

Abkürzung	Bedeutung
++	gut geeignet
+	geeignet
-	Anwendung möglich, Ergebnisse nicht gut
- -	ungeeignet
gl.	glatt
m. A. d. St.	monotone Anordnung der Stützstellen
k. m. A. d. St.	keine monotone Anordnung der Stützstellen
n. k. S.	natürlicher kubischer Spline
p. k. S.	periodischer kubischer Spline
k. S. 1. R.	kubischer Spline mit vorgegebenen 1. Randableitungen
par. n. k. S.	parametrischer natürlicher kubischer Spline
par. p. k. S.	parametrischer periodischer kubischer Spline
par. S. 1. R.	parametrischer kubischer Spline mit vorgeg. 1. Randabl.
kub. Bez. S.	kubischer Bézier-Spline
A. Ss.	Akima-Subspline
R. Ss.	Renner-Subspline

Nr.	Verfahren Kurventyp (z. B. von der Form...)	n. k. S.	p. k. S.	k. S. 1.R.	par. n. k.S.	par. p. k.S.	par. k. 1.R.	kub. Bez. S.	A. Ss.	R. Ss.
1	konvex, offen, m.A.d.St.	+	- -	++	+	- -	+	+	+	+
										
2	konvex, offen, k.m.A.d.St.	- -	- -	- -	+	- -	++	+	- -	+
										
3	konkav–konvex, offen, m.A.d.St.	+	- -	++	+	- -	+	++	++	+
										
4	konkav–konvex, offen, k.m.A.d.St.	- -	- -	- -	+	- -	+	+ +	- -	++
										
5	konkav–konvex, periodisch in f(x) und f'(x), m.A.d.St.	- -	+	- -	- -	- -	- -	- -	++	+
										
6	konvex bzw. konkav-konvex, gl. geschl.	- -	- -	- -	- -	++	- -	+	- -	++
										

Nr.	Verfahren Kurventyp (z. B. von der Form...)	n. k. S.	p. k. S.	k. S. 1.R.	par. n. k.S.	par. p. k.S.	par. k. 1.R.	kub. Bez. S.	A. Ss.	R. Ss.
7	konvex bzw. konkav-konvex, n. glatt geschlossen	- -	- -	- -	+	- -	++	+	- -	- -
										
8	konvex, offen, k.m.A.d.St.	- -	- -	- -	+	- -	++	+	- -	++
										
9	konvex, offen, m.A.d.St., starke Steigung	-	- -	++	-	- -	+	++	++	+
										
10		- -	- -	- -	- -	- -	- -	-	++	+
										
11		- -	- -	- -	- -	- -	- -	-	- -	++
										

Ergänzende Literatur zu Kapitel 10

[BOOR2001]; [ENGE1996], 7; [HAMM1994], 6; [HERM2001], Kap.6.6, [HOSC1989]; [MAES1988], 6.2; [OPFE2002], Kap.4; [PALM1988]; [PREU2001]; [RALS1979] IV, 8; [REIN1971]; [SCHW1997], 3.7; [SHAM1973], II. 1.3; [SPAT1973]; [SPAT1974]; [STOE1989], 2.4; [STUM1982], III; [UBER1995], Kap. 9; [WERN1993], III.

Kapitel 11

Akima- und Renner-Subsplines

Wie die interpolierenden kubischen Splines (Kapitel 10) setzen sich die interpolierenden Akima-Subsplines und Renner-Subsplines intervallweise aus kubischen Polynomen zusammen. Während die kubischen Splines zweimal stetig differenzierbar sind, wird von den Subsplines nur die einmalige stetige Differenzierbarkeit gefordert. Je zwei benachbarte Segmente eines Subspline haben im gemeinsamen Stützpunkt dieselbe Tangente, aber nicht dieselbe Krümmung. Daher kann sich an ein krummliniges Segment eines Subspline ein geradliniges mit der Krümmung Null tangential anschließen. Zwei benachbarte geradlinige Segmente, die zu verschiedenen Geraden gehören, erzeugen eine Ecke (Abb. 11.3). Abweichend von den Originalarbeiten von Akima [AKIM1970] und Renner [RENN1981], [RENN1982] werden Ecken hier zugelassen, so dass der Subspline dann nur stückweise stetig differenzierbar ist. Falls Ecken nicht erwünscht sind, können sie durch Einfügen weiterer Punkte vermieden werden (Abb. 11.7). Ein Vorteil der Akima- und Renner-Subsplines gegenüber den anderen Splines ist, dass für die Berechnung ihrer Koeffizienten kein lineares Gleichungssystem gelöst werden muss.

Beim Akima-Subspline wird die monotone Anordnung der Stützstellen vorausgesetzt; er eignet sich daher zur Darstellung einer Funktion. Wenn durch Punkte beliebiger Lage in einer Ebene oder im Raum eine glatte Kurve gelegt werden soll, wird ein Renner-Subspline verwendet.

11.1 Akima-Subsplines

Von einer (unbekannten oder bekannten) Funktion $f \in C^1[a,b]$ seien an den $n+1$ monoton angeordneten Stützstellen (Knoten) $x_i,\ i = 0(1)n$,

$$a = x_0 < x_1 < \ldots < x_n = b$$

die Stützwerte $y_i = f(x_i)$ gegeben; es sei $n \geq 2$. Durch die $n{+}1$ Punkte $P_i = (x_i, y_i)$, auch Stützpunkte genannt, werde eine glatte Kurve gelegt. Diese soll durch die stückweise definierte Subsplinefunktion S mit

$$\begin{aligned} S(x) \equiv S_i(x) \quad &= \quad a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3, \qquad (11.1) \\ &\qquad x \in [x_i, x_{i+1}], \;\; i = 0(1)n{-}1 \end{aligned}$$

dargestellt werden. S sei also in jedem Teilintervall durch ein Polynom höchstens dritten Grades gegeben.

Das Segment S_i soll durch die Punkte (x_i, y_i) und (x_{i+1}, y_{i+1}) gehen. Daraus folgen mit der Intervall-Länge $h_i = x_{i+1} - x_i > 0$ die Bedingungen

$$\begin{aligned} &S_i(x_i) = a_i = y_i, \\ \text{(i)} \qquad &S_i(x_{i+1}) = y_i + b_i\, h_i + c_i\, h_i^2 + d_i\, h_i^3 = y_{i+1}. \end{aligned}$$

An den Stützstellen x_i und x_{i+1} soll S_i die Steigungen t_i bzw. t_{i+1} besitzen. Daraus folgen mit

$$S_i'(x) = b_i + 2\, c_i\, (x - x_i) + 3\, d_i (x - x_i)^2$$

die Bedingungen

$$\begin{aligned} &\text{(ii)} \qquad S_i'(x_i) = b_i = t_i, \\ &\text{(iii)} \qquad S_i'(x_{i+1}) = t_i + 2\, c_i\, h_i + 3\, d_i\, h_i^2 = t_{i+1}. \end{aligned}$$

Mit der Steigung

$$m_i = \frac{y_{i+1} - y_i}{x_{i+1} - x_i} = \frac{y_{i+1} - y_i}{h_i}$$

der Sehne zwischen den Punkten P_i und P_{i+1} folgen aus (i), (ii) und (iii)

$$\begin{aligned} c_i\, h_i \quad &+ \quad d_i\, h_i^2 \quad = \quad m_i \quad - \quad t_i\,, \\ 2\, c_i\, h_i \quad &+ \quad 3\, d_i\, h_i^2 \quad = \quad t_{i+1} \quad - \quad t_i\,. \end{aligned}$$

Aus diesen Gleichungen ergeben sich

$$\begin{aligned} c_i \quad &= \quad \tfrac{1}{h_i} \;\; (3\, m_i - 2\, t_i - t_{i+1}) \;\;, \\ d_i \quad &= \quad \tfrac{1}{h_i^2} \;\; (t_i + t_{i+1} - 2\, m_i) \quad . \end{aligned}$$

Somit können alle Koeffizienten a_i, b_i, c_i, d_i berechnet werden, wenn die Steigungen t_i und t_{i+1} an den Stützstellen x_i bzw. x_{i+1} bekannt sind. Andernfalls müssen geeignete Werte für die Steigungen zunächst ermittelt werden.

Akima zieht zur Bestimmung der Steigung t_i im Punkt P_i nur die diesem Punkt benachbarten Punkte P_{i-2}, P_{i-1}, P_{i+1}, P_{i+2} heran, ähnlich wie der Zeichner einer punktweise gegebenen Kurve nur wenige Punkte in der Nachbarschaft eines Punktes berücksichtigt, um den Kurvenverlauf dort festzulegen. Da ein Zeichner drei (oder mehr) kollineare Punkte geradlinig verbinden wird, soll auch die Funktion S diese Eigenschaft besitzen.

Die Steigungsformel von Akima lautet

$$t_i = \frac{|m_i - m_{i+1}|m_{i-1} + |m_{i-2} - m_{i-1}|m_i}{|m_i - m_{i+1}| + |m_{i-2} - m_{i-1}|} \,. \tag{11.2}$$

Falls der Nenner nicht verschwindet, liefert sie

- im Sonderfall P_{i-2}, P_{i-1}, P_i kollinear, $m_{i-2} = m_{i-1}$, die Steigung $t_i = m_{i-1}$,
- im Sonderfall P_i, P_{i+1}, P_{i+2} kollinear, $m_i = m_{i+1}$, die Steigung $t_i = m_i$;

ein geradliniges Segment ist also die Tangente eines sich anschließenden krummlinigen Segmentes (Abb. 11.3).

Für t_0, t_1, t_{n-1}, t_n ist die Steigungsformel (11.2) nur dann anwendbar, wenn weitere Sehnensteigungen $m_{-2}, m_{-1}, m_n, m_{n+1}$ bereitgestellt werden. Dabei ist zu unterscheiden, ob S nicht periodisch oder, falls $y_n = y_0$ ist, periodisch mit der Periode $p = b - a = x_n - x_0$ sein soll. Im periodischen Fall liefert die Steigungsformel mit den zusätzlichen Sehnensteigungen $t_0 = t_n$.

Wenn der Nenner der Steigungsformel verschwindet und $m_{i-1} \neq m_i$ ist, liegen P_{i-2}, P_{i-1}, P_i und P_i, P_{i+1}, P_{i+2} auf Geraden verschiedener Steigung, und in P_i entsteht zeichnerisch eine Ecke (Abb. 11.3); das ist mit der Forderung nach stetiger Differenzierbarkeit nicht verträglich. Während Akima auf die Wiedergabe einer solchen Ecke zugunsten der stetigen Differenzierbarkeit verzichtet, sollen hier auch Ecken dargestellt werden. Dafür ist es erforderlich, an jedem Knoten eine links- und rechtsseitige Steigung zu notieren. Ecken können nur dann auftreten, wenn $n \geq 4$ ist.

Algorithmus 11.1. (*Akima-Subspline*)

Gegeben: (x_i, y_i), $i = 0(1)n$, $n \geq 2$, mit $x_i < x_{i+1}$ für $i = 0(1)n-1$,
nicht periodisch bzw. (mit $y_n = y_0$) periodisch.

Gesucht: Der Akima-Subspline (11.1).

1. Für $i = 0(1)n-1$ werden die Intervall-Längen
$$h_i := x_{i+1} - x_i > 0$$
und die Sehnensteigungen
$$m_i := (y_{i+1} - y_i)/h_i$$
berechnet.

2. Bereitstellung weiterer Sehnensteigungen

   2.1 nicht periodisch
   $$\begin{aligned} m_{-2} &:= 3m_0 - 2m_1 \\ m_{-1} &:= 2m_0 - m_1 \\ m_n &:= 2m_{n-1} - m_{n-2} \\ m_{n+1} &:= 3m_{n-1} - 2m_{n-2} \end{aligned}$$

   2.2 periodisch
   $$\begin{aligned} m_{-2} &:= m_{n-2} \\ m_{-1} &:= m_{n-1} \\ m_n &:= m_0 \\ m_{n+1} &:= m_1 \,. \end{aligned}$$

3. Für $i = 0(1)n$ werden die linksseitigen und rechtsseitigen Steigungen t_i^L bzw. t_i^R berechnet (t_0^L, t_n^R werden berechnet, aber unter 4. nicht benutzt).

   $L_i := |m_{i-2} - m_{i-1}|$

   $NE_i := L_i + |m_i - m_{i+1}|$

   $NE_i > 0:$
   $$\begin{aligned} \alpha_i &:= L_i / NE_i \\ t_i^L &:= (1-\alpha_i)\, m_{i-1} + \alpha_i m_i \\ t_i^R &:= t_i^L \end{aligned}$$

   $NE_i = 0:$
   $$\begin{aligned} t_i^L &:= m_{i-1} \\ t_i^R &:= m_i \end{aligned}$$

4. Berechnung der Koeffizienten für $i = 0(1)n-1$:
   $$\begin{aligned} a_i &:= y_i \\ b_i &:= t_i^R \\ c_i &:= \tfrac{1}{h_i}\left(3m_i - 2t_i^R - t_{i+1}^L\right) \\ d_i &:= \tfrac{1}{h_i^2}\left(t_i^R + t_{i+1}^L - 2m_i\right) \end{aligned}$$

Beispiel 11.2.

Gegeben: Die Wertetabelle

i	0	1	2	3
x_i	0	0.5	1.5	3.5
y_i	0	0.5	−0.5	1.5

Gesucht: Der nicht periodische Akima-Subspline.

Lösung: Nach Algorithmus 11.1 werden mit $n = 3$ berechnet

$$
\begin{array}{llcll}
1. & h_0 & = & x_1 - x_0 & = & 0.5 \\
 & h_1 & = & x_2 - x_1 & = & 1 \\
 & h_2 & = & x_3 - x_2 & = & 2 \\
 & m_0 & = & (y_1 - y_0) \,/\, h_0 & = & 1 \\
 & m_1 & = & (y_2 - y_1) \,/\, h_1 & = & -1 \\
 & m_2 & = & (y_3 - y_2) \,/\, h_2 & = & 1 \\
2. & m_{-2} & = & 3\, m_0 - 2\, m_1 & = & 5 \\
 & m_{-1} & = & 2\, m_0 - m_1 & = & 3 \\
 & m_3 & = & 2\, m_2 - m_1 & = & 3 \\
 & m_4 & = & 3\, m_2 - 2\, m_1 & = & 5 \\
3. & L_0 & = & |m_{-2} - m_{-1}| & = & 2 \\
 & NE_0 & = & L_0 + |m_0 - m_1| = 2 + 2 & = & 4 \\
 & \alpha_0 & = & L_0 \,/\, NE_0 & = & 0.5 \\
 & t_0^L & = & (1 - \alpha_0)\, m_{-1} + \alpha_0\, m_0 & = & 2 \quad = \quad t_0^R \\
 & L_1 & = & |m_{-1} - m_0| & = & 2 \\
 & NE_1 & = & L_1 + |m_1 - m_2| = 2 + 2 & = & 4 \\
 & \alpha_1 & = & L_1 \,/\, NE_1 & = & 0.5 \\
 & t_1^L & = & (1 - \alpha_1)\, m_0 + \alpha_1\, m_1 & = & 0 \quad = \quad t_1^R \\
 & L_2 & = & |m_0 - m_1| & = & 2 \\
 & NE_2 & = & L_2 + |m_2 - m_3| = 2 + 2 & = & 4 \\
 & \alpha_2 & = & L_2 \,/\, NE_2 & = & 0.5 \\
 & t_2^L & = & (1 - \alpha_2)\, m_1 + \alpha_2\, m_2 & = & 0 \quad = \quad t_2^R \\
 & L_3 & = & |m_1 - m_2| & = & 2 \\
 & NE_3 & = & L_3 + |m_3 - m_4| = 2 + 2 & = & 4 \\
 & \alpha_3 & = & L_3 \,/\, NE_3 & = & 0.5 \\
 & t_3^L & = & (1 - \alpha_3)\, m_2 + \alpha_3\, m_3 & = & 2 \quad = \quad t_3^R \\
4. & a_0 & = & y_0 & = & 0 \\
 & b_0 & = & t_0^R & = & 2 \\
 & c_0 & = & (3\, m_0 - 2\, t_0^R - t_1^L) \,/\, h_0 & = & -2 \\
 & d_0 & = & (t_0^R + t_1^L - 2\, m_0) \,/\, h_0^2 & = & 0 \\
 & a_1 & = & y_1 & = & 0.5 \\
 & b_1 & = & t_1^R & = & 0 \\
 & c_1 & = & (3\, m_1 - 2\, t_1^R - t_2^L) \,/\, h_1 & = & -3 \\
 & d_1 & = & (t_1^R + t_2^L - 2\, m_1) \,/\, h_1^2 & = & 2 \\
 & a_2 & = & y_2 & = & -0.5 \\
 & b_2 & = & t_2^R & = & 0 \\
 & c_2 & = & (3\, m_2 - 2\, t_2^R - t_3^L) \,/\, h_2 & = & 0.5 \\
 & d_2 & = & (t_2^R + t_3^L - 2\, m_2) \,/\, h_2^2 & = & 0
\end{array}
$$

Damit lautet die Darstellung des Akima-Subspline:

$$
\begin{array}{lcll}
S_0(x) & = & 2\, x - 2\, x^2\,, & x \in [0, 0.5] \\
S_1(x) & = & 0.5 - 3\, (x - 0.5)^2 + 2\, (x - 0.5)^3\,, & x \in [0.5, 1.5] \\
S_2(x) & = & -0.5 + 0.5\, (x - 1.5)^2\,, & x \in [1.5, 3.5]\,.
\end{array}
$$

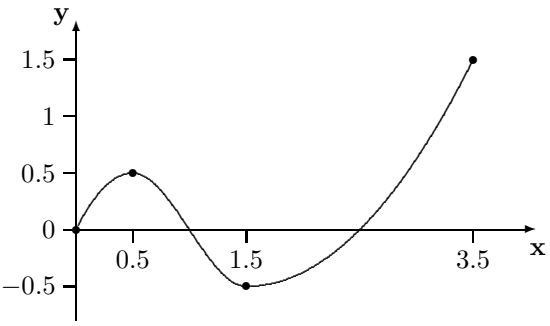


Abb. 11.1. Akima-Subspline mit der Steigungsformel (11.2)

□

In der Steigungsformel (11.2) von Akima kommen die Sehnensteigungen $m_{i-2}, \ldots, m_{i+1}$ vor, jedoch nicht die Intervall-Längen, d. h. die Abstände benachbarter Stützstellen. Daher liefert die Formel von Akima z. B. zu **jedem** System mit vier Stützpunkten (x_0, y_0), (x_1, y_1), (x_2, y_2), (x_3, y_3), das die Sehnensteigungen $m_0 = 1$, $m_1 = -1$, $m_2 = 1$ besitzt, dieselben Steigungen $t_0 = 2$, $t_1 = 0$, $t_2 = 0$, $t_3 = 2$ wie im Beispiel 11.2.

Eine Steigungsformel, die Intervall-Längen enthält und damit die Lage der Stützstellen berücksichtigt, ist (siehe [WODI1991])

$$t_i = \frac{h_i\,|m_i - m_{i+1}|\,m_{i-1} + h_{i-1}\,|m_{i-2} - m_{i-1}|\,m_i}{h_i\,|m_i - m_{i+1}| + h_{i-1}\,|m_{i-2} - m_{i-1}|}\,; \tag{11.3}$$

sie stimmt im Sonderfall $h_{i-1} = h_i$ mit der Formel (11.2) von Akima überein.

Für die Anwendung der Steigungsformel (11.3) muss der Algorithmus 11.1 wie folgt ergänzt und geändert werden:

Unter 2. Bereitstellung weiterer Intervall-Längen

2.1 nicht periodisch $\quad h_{-1} := h_1\,, \quad h_n := h_{n-2}$
2.2 periodisch $\quad h_{-1} := h_{n-1}\,, \quad h_n := h_0$

Unter 3. $NE_i > 0$: $\quad t_i^L := \dfrac{h_i\,(1-\alpha_i)\,m_{i-1} + h_{i-1}\,\alpha_i\,m_i}{h_i\,(1-\alpha_i) + h_{i-1}\,\alpha_i}\,.$

Beispiel 11.3.

Für die Wertetabelle des Beispiels 11.2 werden im Folgenden die Steigungen mit der Formel (11.3) und die mit ihnen sich ergebenden Koeffizienten berechnet.

2. $h_{-1} = h_1 = 1,$
   $h_3 = h_1 = 1$

3. $$t_0^L = \frac{h_0\,(1-\alpha_0)\,m_{-1} + h_{-1}\,\alpha_0\,m_0}{h_0\,(1-\alpha_0) + h_{-1}\,\alpha_0} = \frac{5}{3} = t_0^R$$
   $$t_1^L = \frac{h_1\,(1-\alpha_1)\,m_0 + h_0\,\alpha_1\,m_1}{h_1\,(1-\alpha_1) + h_0\,\alpha_1} = \frac{1}{3} = t_1^R$$
   $$t_2^L = \frac{h_2\,(1-\alpha_2)\,m_1 + h_1\,\alpha_2\,m_2}{h_2\,(1-\alpha_2) + h_1\,\alpha_2} = -\frac{1}{3} = t_2^R$$
   $$t_3^L = \frac{h_3\,(1-\alpha_3)\,m_2 + h_2\,\alpha_3\,m_3}{h_3\,(1-\alpha_3) + h_2\,\alpha_3} = \frac{7}{3} = t_3^R$$

4. $$b_0 = t_0^R = \frac{5}{3}$$
   $$c_0 = (3\,m_0 - 2\,t_0^R - t_1^L)\,/\,h_0 = -\frac{4}{3}$$
   $$d_0 = (t_0^R + t_1^L - 2\,m_0)\,/\,h_0^2 = 0$$
   $$b_1 = t_1^R = \frac{1}{3}$$
   $$c_1 = (3\,m_1 - 2\,t_1^R - t_2^L)\,/\,h_1 = -\frac{10}{3}$$
   $$d_1 = (t_1^R + t_2^L - 2\,m_1)\,/\,h_1^2 = 2$$
   $$b_2 = t_2^R = -\frac{1}{3}$$
   $$c_2 = (3\,m_2 - 2\,t_2^R - t_3^L)\,/\,h_2 = \frac{2}{3}$$
   $$d_2 = (t_2^R + t_3^L - 2\,m_2)\,/\,h_2^2 = 0$$

Der mit der Formel (11.3) erzeugte Subspline hat die Darstellung:

$$S_0(x) = \tfrac{5}{3}\,x - \tfrac{4}{3}\,x^2\,, \qquad x \in [0, 0.5]$$
$$S_1(x) = 0.5 + \tfrac{1}{3}\,(x-0.5) - \tfrac{10}{3}\,(x-0.5)^2 + 2\,(x-0.5)^3\,, \qquad x \in [0.5, 1.5]$$
$$S_2(x) = -0.5 - \tfrac{1}{3}\,(x-1.5) + \tfrac{2}{3}\,(x-1.5)^2\,, \qquad x \in [1.5, 3.5]\,.$$

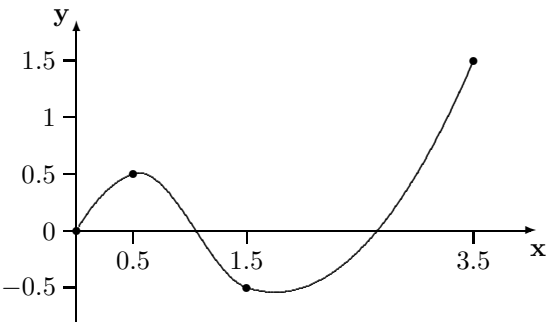


Abb. 11.2. Subspline mit Steigungsformel (11.3)

□

Beispiel 11.4.

Gegeben: Die Wertetabelle

i	0	1	2	3	4	5	6	7	8	9
x_i	0	1	2	3	4	5	6	7	9	10
y_i	0	1	1	1	−1	−3	−2	−1	−1	0

Gesucht: Ein nicht periodischer und ein periodischer Akima-Subspline.

Lösung: Ecken sind die Punkte (x_3, y_3) und (x_5, y_5) sowie im periodischen Fall außerdem (x_1, y_1).

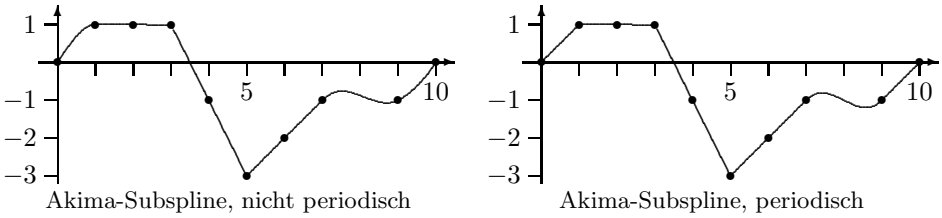


Abb. 11.3. Akima-Subsplines

□

11.2 Renner-Subsplines

Gegeben seien $n+1$ Punkte $P_i = (x_i, y_i)$ in der Ebene ($m = 2$) oder $P_i = (x_i, y_i, z_i)$ im Raum ($m = 3$), $i = 0(1)n$, $n \geq 3$, mit $P_i \neq P_{i+1}$ für $i = 0(1)n-1$. Wenn P_{i-1}, P_i und P_{i+1} kollinear sind, muss P_i zwischen P_{i-1} und P_{i+1} liegen. Durch diese Punkte werde in der Reihenfolge der Nummerierung eine glatte Kurve gelegt. Die Kurve wird mit Hilfe eines Parameters t stückweise definiert durch

$$\begin{aligned} &\boldsymbol{S}(t) \equiv \boldsymbol{S}_i(t) = \boldsymbol{a}_i + \boldsymbol{b}_i\, t + \boldsymbol{c}_i\, t^2 + \boldsymbol{d}_i\, t^3\,, \qquad \boldsymbol{a}_i,\ \boldsymbol{b}_i,\ \boldsymbol{c}_i,\ \boldsymbol{d}_i \in \mathbb{R}^m \\ &\text{mit } m = 2 \text{ oder } m = 3\,, \quad t \in [0, h_i], \quad h_i > 0\,, \quad i = 0(1)n-1\,. \end{aligned} \tag{11.4}$$

Jedes der n Kurvensegmente ist also durch ein Polynom höchstens dritten Grades gegeben. Zu berechnen sind für $i = 0(1)n-1$ die Koeffizienten $\boldsymbol{a}_i$, $\boldsymbol{b}_i$, $\boldsymbol{c}_i$, $\boldsymbol{d}_i$ und die Länge h_i des Parameterintervalls.

Das Kurvensegment $\boldsymbol{S}_i$ soll die Stützpunkte P_i und P_{i+1} verbinden. Also müssen gelten

$$\boldsymbol{S}_i(0) = \boldsymbol{a}_i = \boldsymbol{P}_i\,,$$

(i)
$$\boldsymbol{S}_i(h_i) = \boldsymbol{P}_i + \boldsymbol{b}_i\, h_i + \boldsymbol{c}_i\, h_i^2 + \boldsymbol{d}_i\, h_i^3 = \boldsymbol{P}_{i+1}\,.$$

Ferner sollen die Tangentenvektoren von $\boldsymbol{S}_i$ in P_i und P_{i+1} mit den vorgegebenen Einheitsvektoren $\boldsymbol{v}_i$ bzw. $\boldsymbol{v}_{i+1}$, $|\boldsymbol{v}_i| = |\boldsymbol{v}_{i+1}| = 1$, übereinstimmen. Daraus folgen mit

$$\boldsymbol{S}_i'(t) = \boldsymbol{b}_i + 2\,\boldsymbol{c}_i\,t + 3\,\boldsymbol{d}_i\,t^2$$

(ii) $$\boldsymbol{S}_i'(0) = \boldsymbol{b}_i = \boldsymbol{v}_i\,,$$

(iii) $$\boldsymbol{S}_i'(h_i) = \boldsymbol{v}_i + 2\,\boldsymbol{c}_i\,h_i + 3\,\boldsymbol{d}_i\,h_i^2 = \boldsymbol{v}_{i+1}\,.$$

Mit dem Sehnenvektor

$$\boldsymbol{s}_i = \boldsymbol{P}_{i+1} - \boldsymbol{P}_i \neq \boldsymbol{0}$$

folgen aus (i), (ii) und (iii)

$$\begin{array}{rcrcrcl} \boldsymbol{c}_i\,h_i & + & \boldsymbol{d}_i\,h_i^2 & = & \frac{1}{h_i}\,\boldsymbol{s}_i & - & \boldsymbol{v}_i \\ 2\,\boldsymbol{c}_i\,h_i & + & 3\,\boldsymbol{d}_i\,h_i^2 & = & \boldsymbol{v}_{i+1} & - & \boldsymbol{v}_i \end{array}\,,$$

und aus diesen Gleichungen

$$\begin{aligned} \boldsymbol{c}_i &= \tfrac{1}{h_i}\,\left(\tfrac{3}{h_i}\,\boldsymbol{s}_i - 2\,\boldsymbol{v}_i - \boldsymbol{v}_{i+1}\right)\,, \\ \boldsymbol{d}_i &= \tfrac{1}{h_i^2}\,\left(\boldsymbol{v}_i + \boldsymbol{v}_{i+1} - \tfrac{2}{h_i}\,\boldsymbol{s}_i\right)\,. \end{aligned}$$

Wenn man fordert, dass $|\boldsymbol{S}_i'(h_i/2)| = 1$ ist, kann h_i berechnet werden. Mit $\boldsymbol{b}_i$, $\boldsymbol{c}_i$, $\boldsymbol{d}_i$ ergibt sich

$$\begin{aligned} 4\,h_i\,\boldsymbol{S}_i'\left(\tfrac{h_i}{2}\right) &= 4\,h_i\left(\boldsymbol{b}_i + 2\,\boldsymbol{c}_i\,\tfrac{h_i}{2} + 3\,\boldsymbol{d}_i\,\tfrac{h_i^2}{4}\right) \\ &= 6\,\boldsymbol{s}_i - h_i\,(\boldsymbol{v}_i + \boldsymbol{v}_{i+1})\,. \end{aligned}$$

Das Quadrat des Betrages dieses Vektors ($\boldsymbol{s}^{\mathsf{T}}\,\boldsymbol{t}$ ist das Skalarprodukt der Vektoren $\boldsymbol{s}$ und $\boldsymbol{t}$) führt mit

$$A = 16 - |\boldsymbol{v}_i + \boldsymbol{v}_{i+1}|^2\,, \quad B = 6\,\boldsymbol{s}_i^{\mathsf{T}}(\boldsymbol{v}_i + \boldsymbol{v}_{i+1})\,, \quad C = 36\,|\boldsymbol{s}_i|^2$$

zu der quadratischen Gleichung

$$A\,h_i^2 + 2\,B\,h_i - C = 0$$

mit der positiven Lösung

$$h_i = (-B + \sqrt{B^2 + A\,C})\,/\,A\,.$$

Für die Länge des i-ten Kurvensegmentes erhält man mit der Simpsonschen Formel (siehe Abschnitt 14.3.2)

$$\int_0^{h_i} |\boldsymbol{S}_i'(t)|\,\mathrm{d}\,t \approx \frac{h_i}{6}\,\left(\,|\boldsymbol{S}_i'(0)| + 4\,\left|\boldsymbol{S}_i'\left(\tfrac{h_i}{2}\right)\right| + |\boldsymbol{S}_i'(h_i)|\,\right) = h_i\,.$$

Darum ist t angenähert Bogenlängenparameter.

Falls die Kurventangenten in den Punkten P_i nicht bekannt sind, müssen die Tangenteneinheitsvektoren $\boldsymbol{v}_i$ für $i = 0(1)n$ geeignet bestimmt werden. Dabei werden die zu den Sehnenvektoren $\boldsymbol{s}_i$ gehörigen Einheitsvektoren $\boldsymbol{s}_i^0 = \boldsymbol{s}_i/|\boldsymbol{s}_i|$ benötigt. Die Formel von Renner für den (zunächst nicht normierten) Tangentenvektor $\boldsymbol{v}_i$ im Punkt P_i lautet:

$$\begin{aligned} \boldsymbol{v}_i &= (1-\alpha_i)\,\boldsymbol{s}_{i-1} + \alpha_i\,\boldsymbol{s}_i \qquad \text{mit} \\ \alpha_i &= \frac{F(\boldsymbol{s}_{i-2}^0, \boldsymbol{s}_{i-1}^0)}{F(\boldsymbol{s}_{i-2}^0, \boldsymbol{s}_{i-1}^0) + F(\boldsymbol{s}_i^0, \boldsymbol{s}_{i+1}^0)} \quad . \end{aligned}$$

Hier ist $F(\boldsymbol{s}, \boldsymbol{t})$ der (nicht negative) Flächeninhalt des von den Vektoren $\boldsymbol{s}, \boldsymbol{t} \in \mathbb{R}^m$ aufgespannten Parallelogramms. Genau dann, wenn $\boldsymbol{s}$ und $\boldsymbol{t}$ linear abhängig sind, ist $F(\boldsymbol{s}, \boldsymbol{t}) = 0$.

Der Nenner von α_i sei positiv. Wenn P_{i-2}, P_{i-1}, P_i kollinear sind, sind $\boldsymbol{s}_{i-2}, \boldsymbol{s}_{i-1}$ linear abhängig, also $F(\boldsymbol{s}_{i-2}^0, \boldsymbol{s}_{i-1}^0) = 0$. Wegen $\alpha_i = 0$ ist dann $\boldsymbol{v}_i = \boldsymbol{s}_{i-1}$. Sind P_i, P_{i+1}, P_{i+2} kollinear, so sind $\boldsymbol{s}_i, \boldsymbol{s}_{i+1}$ linear abhängig und $F(\boldsymbol{s}_i^0, \boldsymbol{s}_{i+1}^0) = 0$. Wegen $\alpha_i = 1$ ist dann $\boldsymbol{v}_i = \boldsymbol{s}_i$. Ein geradliniger Abschnitt der Kurve ist also die Tangente eines in P_i beginnenden oder endenden krummlinigen Abschnitts.
Wenn der Nenner von α_i verschwindet und P_{i-1}, P_i, P_{i+1} nicht kollinear sind ($F(\boldsymbol{s}_{i-1}, \boldsymbol{s}_i) > 0$), ist P_i eine Ecke. In diesem Fall werden der Ecke P_i der „linksseitige“ Tangentenvektor $\boldsymbol{v}_i^L = \boldsymbol{s}_{i-1}$ und der „rechtsseitige“ $\boldsymbol{v}_i^R = \boldsymbol{s}_i$ zugeordnet.

Für die Berechnung der Vektoren $\boldsymbol{v}_0, \boldsymbol{v}_1, \boldsymbol{v}_{n-1}, \boldsymbol{v}_n$ müssen zusätzlich Sehnenvektoren $\boldsymbol{s}_{-2}, \boldsymbol{s}_{-1}, \boldsymbol{s}_n, \boldsymbol{s}_{n+1}$ und somit Punkte $P_{-2}, P_{-1}, P_{n+1}, P_{n+2}$ bereitgestellt werden. Die Konstruktion dieser Punkte hängt vom Typ der zu erzeugenden Kurve ab.
Falls $P_n = P_0$ ist, ist die Kurve geschlossen. Wenn in diesem Fall die Tangentenvektoren $\boldsymbol{v}_0$ und $\boldsymbol{v}_n$ linear unabhängig sind, sind die Tangenten der Kurve in P_0 und P_n verschieden; die Kurve heiße dann *geschlossen mit Eckpunkt*. Sind die Tangentenvektoren $\boldsymbol{v}_0$ und $\boldsymbol{v}_n$ gleich, dann ist wegen $\boldsymbol{S}_0'(0) = \boldsymbol{v}_0 = \boldsymbol{v}_n = \boldsymbol{S}_{n-1}'(h_{n-1})$ die Kurve auch in $P_0 = P_n$ einmal stetig differenzierbar und hat dort genau eine Tangente; die Kurve heiße dann *geschlossen ohne Eckpunkt* (Abbildung 11.5). Der Fall $\boldsymbol{v}_n = -\boldsymbol{v}_0$, in dem $P_0 = P_n$ ein Rückkehrpunkt ist, bleibt außer Betracht.
Bei einer geschlossenen Kurve ohne Eckpunkt werden zusätzlich die Punkte
$P_{-2} = P_{n-2}$, $P_{-1} = P_{n-1}$, $P_{n+1} = P_1$, $P_{n+2} = P_2$ benutzt. Damit ergeben sich $\boldsymbol{s}_{-2} = \boldsymbol{s}_{n-2}$, $\boldsymbol{s}_{-1} = \boldsymbol{s}_{n-1}$, $\boldsymbol{s}_n = \boldsymbol{s}_0$, $\boldsymbol{s}_{n+1} = \boldsymbol{s}_1$ sowie $\alpha_0 = \alpha_n$ und $\boldsymbol{v}_0 = \boldsymbol{v}_n$.
Bei einer nicht geschlossenen oder einer geschlossenen Kurve mit Eckpunkt werden (anders als in [RENN1981]) die zusätzlichen Punkte wie folgt erzeugt. P_{-2} und P_{-1} ergeben sich durch Spiegelung von P_3 bzw. P_2 an der Gerade / Ebene, die durch den Mittelpunkt von P_0 und P_1 geht und zur Gerade $P_0\,P_1$ senkrecht ist. Ebenso werden P_{n-2} und P_{n-3} an der Gerade / Ebene, die durch den Mittelpunkt von P_{n-1} und P_n geht und zur Gerade $P_{n-1}\,P_n$ senkrecht ist, gespiegelt, um P_{n+1} bzw. P_{n+2} zu erhalten.

Im Algorithmus werden sogleich die Sehnenvektoren $\boldsymbol{s}_{-2}, \boldsymbol{s}_{-1}, \boldsymbol{s}_n, \boldsymbol{s}_{n+1}$ angegeben.

Für $\boldsymbol{s}, \boldsymbol{t} \in \mathbb{R}^3$ ist

$$F^2(\boldsymbol{s}, \boldsymbol{t}) = |\boldsymbol{s} \times \boldsymbol{t}|^2 = |\,\boldsymbol{s}\,|^2\,|\,\boldsymbol{t}\,|^2 - (\boldsymbol{s}^\mathsf{T}\,\boldsymbol{t})^2$$

und für Einheitsvektoren, $|\,\boldsymbol{s}\,| = |\,\boldsymbol{t}\,| = 1$, also

$$F(\boldsymbol{s}, \boldsymbol{t}) = \sqrt{1 - (\boldsymbol{s}^\mathsf{T}\,\boldsymbol{t})^2}\,.$$

Für $\boldsymbol{s}, \boldsymbol{t} \in \mathbb{R}^2$ gilt

$$F(\boldsymbol{s}, \boldsymbol{t}) = \sqrt{1 - (\boldsymbol{s}^\mathsf{T}\,\boldsymbol{t})^2} = |\det(\boldsymbol{s}, \boldsymbol{t})|\,.$$

Algorithmus 11.5. (*Renner-Subspline*)

Gegeben: $n+1$ Punkte $P_i = (x_i, y_i)$ $(m = 2)$ oder $P_i = (x_i, y_i, z_i)$ $(m = 3)$, $i = 0(1)n$, $n \geq 3$, mit $P_i \neq P_{i+1}$ für $i = 0(1)n-1$. Bei kollinearen Punkten P_{i-1}, P_i, P_{i+1} muss P_i zwischen P_{i-1} und P_{i+1} liegen, $i = 1(1)n-1$.

Gesucht: Der Renner-Subspline (11.4).

1. Für $i = 0(1)n-1$ werden die Sehnenvektoren $\boldsymbol{s}_i := \boldsymbol{P}_{i+1} - \boldsymbol{P}_i \neq \boldsymbol{0}$ berechnet.

2. Bereitstellung weiterer Sehnenvektoren

   2.1 Kurve geschlossen ($P_n = P_0$) und ohne Eckpunkt

   $$\begin{aligned} \boldsymbol{s}_{-2} &:= \boldsymbol{s}_{n-2} \\ \boldsymbol{s}_{-1} &:= \boldsymbol{s}_{n-1} \\ \boldsymbol{s}_{n} &:= \boldsymbol{s}_{0} \\ \boldsymbol{s}_{n+1} &:= \boldsymbol{s}_{1} \end{aligned}$$

   2.2 In allen anderen Fällen

   $$\begin{aligned} \boldsymbol{s} &:= \boldsymbol{s}_0/|\,\boldsymbol{s}_0\,| \\ \boldsymbol{s}_{-2} &:= 2(\boldsymbol{s}^\mathsf{T}\,\boldsymbol{s}_2)\,\boldsymbol{s} - \boldsymbol{s}_2 \\ \boldsymbol{s}_{-1} &:= 2(\boldsymbol{s}^\mathsf{T}\,\boldsymbol{s}_1)\,\boldsymbol{s} - \boldsymbol{s}_1 \\ \boldsymbol{s} &:= \boldsymbol{s}_{n-1}/|\,\boldsymbol{s}_{n-1}\,| \\ \boldsymbol{s}_{n} &:= 2(\boldsymbol{s}^\mathsf{T}\,\boldsymbol{s}_{n-2})\,\boldsymbol{s} - \boldsymbol{s}_{n-2} \\ \boldsymbol{s}_{n+1} &:= 2(\boldsymbol{s}^\mathsf{T}\,\boldsymbol{s}_{n-3})\,\boldsymbol{s} - \boldsymbol{s}_{n-3} \end{aligned}$$

3. Für $i = -2(1)n+1$ werden zu den Sehnenvektoren die Einheitsvektoren

   $$\boldsymbol{s}_i^0 := \boldsymbol{s}_i/|\,\boldsymbol{s}_i\,|$$

   berechnet.

4. Ermittlung der links- und rechtsseitigen Tangenteneinheitsvektoren für $i = 0(1)n$:

$m = 2$	$m = 3$
$L_i := \lvert\det(\boldsymbol{s}_{i-2}^0, \boldsymbol{s}_{i-1}^0)\rvert$	$L_i := \sqrt{1 - (\boldsymbol{s}_{i-2}^{0\,\mathsf{T}} \boldsymbol{s}_{i-1}^0)^2}$
$NE_i := L_i + \lvert\det(\boldsymbol{s}_i^0, \boldsymbol{s}_{i+1}^0)\rvert$	$NE_i := L_i + \sqrt{1 - (\boldsymbol{s}_i^{0\,\mathsf{T}} \boldsymbol{s}_{i+1}^0)^2}$

$$
\begin{array}{lrcl}
NE_i > 0: & \alpha_i & := & L_i / NE_i \\
 & \boldsymbol{v}_i^L & := & (1-\alpha_i)\,\boldsymbol{s}_{i-1} + \alpha_i\,\boldsymbol{s}_i \\
 & \boldsymbol{v}_i^L & := & \boldsymbol{v}_i^L / \lvert\boldsymbol{v}_i^L\rvert \\
 & \boldsymbol{v}_i^R & := & \boldsymbol{v}_i^L \\
NE_i = 0: & \boldsymbol{v}_i^L & := & \boldsymbol{s}_{i-1}^0 \\
 & \boldsymbol{v}_i^R & := & \boldsymbol{s}_i^0
\end{array}
$$

($\boldsymbol{v}_0^L$ und $\boldsymbol{v}_n^R$ werden im Folgenden nicht benutzt)

5. Für $i = 0(1)n-1$ werden die Längen h_i der Parameterintervalle berechnet:

$$
\begin{aligned}
A &:= 16 - \lvert\boldsymbol{v}_i^R + \boldsymbol{v}_{i+1}^L\rvert^2 \\
B &:= 6\,\boldsymbol{s}_i^{\mathsf{T}}(\boldsymbol{v}_i^R + \boldsymbol{v}_{i+1}^L) \\
C &:= 36\,\lvert\boldsymbol{s}_i\rvert^2 \\
h_i &:= \frac{-B + \sqrt{B^2 + AC}}{A}
\end{aligned}
$$

6. Berechnung der Koeffizienten für $i = 0(1)n-1$:

$$
\begin{aligned}
\boldsymbol{a}_i &:= \boldsymbol{P}_i \\
\boldsymbol{b}_i &:= \boldsymbol{v}_i^R \\
\boldsymbol{c}_i &:= \tfrac{1}{h_i}\left(\tfrac{3}{h_i}\,\boldsymbol{s}_i - 2\,\boldsymbol{v}_i^R - \boldsymbol{v}_{i+1}^L\right) \\
\boldsymbol{d}_i &:= \tfrac{1}{h_i^2}\left(\boldsymbol{v}_i^R + \boldsymbol{v}_{i+1}^L - \tfrac{2}{h_i}\,\boldsymbol{s}_i\right)
\end{aligned}
$$

Bemerkung. Wenn zu jedem Stützpunkt P_i ein Tangenteneinheitsvektor $\boldsymbol{v}_i$ gegeben ist, entfallen die Schritte 2. und 3., und unter 4. werden $\boldsymbol{v}_i^L = \boldsymbol{v}_i^R := \boldsymbol{v}_i$ gesetzt für $i = 0(1)n$.

Der Renner-Subspline in der Ebene ($m = 2$) ist

$$
\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \boldsymbol{S}(t) \equiv \boldsymbol{S}_i(t) = \begin{pmatrix} S_{ix}(t) \\ S_{iy}(t) \end{pmatrix} = \begin{pmatrix} a_{ix} + b_{ix}t + c_{ix}t^2 + d_{ix}t^3 \\ a_{iy} + b_{iy}t + c_{iy}t^2 + d_{iy}t^3 \end{pmatrix},
$$

$t \in [0, h_i]\,,\ i = 0(1)n{-}1$. Darin sind

$$\boldsymbol{a}_i = \begin{pmatrix} a_{ix} \\ a_{iy} \end{pmatrix}, \quad \boldsymbol{b}_i = \begin{pmatrix} b_{ix} \\ b_{iy} \end{pmatrix}, \quad \boldsymbol{c}_i = \begin{pmatrix} c_{ix} \\ c_{iy} \end{pmatrix}, \quad \boldsymbol{d}_i = \begin{pmatrix} d_{ix} \\ d_{iy} \end{pmatrix}.$$

Analog ist im Raum ($m = 3$)

$$\begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix} = \boldsymbol{S}(t) \equiv \boldsymbol{S}_i(t) = \begin{pmatrix} S_{ix}(t) \\ S_{iy}(t) \\ S_{iz}(t) \end{pmatrix} = \begin{pmatrix} a_{ix} + b_{ix}t + c_{ix}t^2 + d_{ix}t^3 \\ a_{iy} + b_{iy}t + c_{iy}t^2 + d_{iy}t^3 \\ a_{iz} + b_{iz}t + c_{iz}t^2 + d_{iz}t^3 \end{pmatrix}.$$

Die Koordinatenfunktionen eines Renner-Subspline sind kubische Polynome.

Beispiel 11.6.

Gegeben: 4 Punkte ($n = 3$)

i	0	1	2	3
x_i	-1	1	0	-1
y_i	$-\sqrt{3}/3$	$-\sqrt{3}/3$	$2\sqrt{3}/3$	$-\sqrt{3}/3$

Die Punkte liegen auf dem Kreis mit dem Mittelpunkt (0,0) und dem Radius $r = \frac{2}{3}\sqrt{3}$.

Gesucht: Der geschlossene Renner-Subspline ohne Eckpunkt.

Lösung: Nach Algorithmus 11.5 werden berechnet

1. Sehnenvektoren

$$\boldsymbol{s}_0 = \boldsymbol{P}_1 - \boldsymbol{P}_0 = \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \quad \boldsymbol{s}_1 = \boldsymbol{P}_2 - \boldsymbol{P}_1 = \begin{pmatrix} -1 \\ \sqrt{3} \end{pmatrix}, \quad \boldsymbol{s}_2 = \boldsymbol{P}_3 - \boldsymbol{P}_2 = \begin{pmatrix} -1 \\ -\sqrt{3} \end{pmatrix}$$

2. Weitere Sehnenvektoren mit 2.1

$$\boldsymbol{s}_{-2} = \boldsymbol{s}_1 = \begin{pmatrix} -1 \\ \sqrt{3} \end{pmatrix}, \qquad \boldsymbol{s}_{-1} = \boldsymbol{s}_2 = \begin{pmatrix} -1 \\ -\sqrt{3} \end{pmatrix},$$

$$\boldsymbol{s}_3 = \boldsymbol{s}_0 = \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \qquad \boldsymbol{s}_4 = \boldsymbol{s}_1 = \begin{pmatrix} -1 \\ \sqrt{3} \end{pmatrix}$$

3. Einheitsvektoren

Alle Sehnenvektoren haben die Länge 2:

$|\,\boldsymbol{s}_i\,| = 2$ für $i = -2(1)4$.

$$\begin{aligned}
\boldsymbol{s}^0_{-2} &= \boldsymbol{s}^0_1 = \tfrac{1}{2}\begin{pmatrix} -1 \\ \sqrt{3} \end{pmatrix} = \boldsymbol{s}^0_4 \\
\boldsymbol{s}^0_{-1} &= \boldsymbol{s}^0_2 = \tfrac{1}{2}\begin{pmatrix} -1 \\ -\sqrt{3} \end{pmatrix} \\
\boldsymbol{s}^0_0 &= \boldsymbol{s}^0_3 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}
\end{aligned}$$

4. Tangenteneinheitsvektoren

$$\begin{aligned}
L_0 &= |\det(\boldsymbol{s}^0_{-2}, \boldsymbol{s}^0_{-1})| = \tfrac{1}{4}\begin{vmatrix} -1 & -1 \\ \sqrt{3} & -\sqrt{3} \end{vmatrix} = \tfrac{\sqrt{3}}{2} \\
NE_0 &= L_0 + |\det(\boldsymbol{s}^0_0, \boldsymbol{s}^0_1)| = \tfrac{\sqrt{3}}{2} + \tfrac{1}{2}\begin{vmatrix} 1 & -1 \\ 0 & \sqrt{3} \end{vmatrix} = \sqrt{3} \\
\alpha_0 &= L_0/NE_0 = \tfrac{1}{2}
\end{aligned}$$

$$\boldsymbol{v}^L_0 = (1-\alpha_0)\,\boldsymbol{s}_{-1} + \alpha_0\,\boldsymbol{s}_0 = \tfrac{1}{2}\begin{pmatrix} -1 \\ -\sqrt{3} \end{pmatrix} + \tfrac{1}{2}\begin{pmatrix} 2 \\ 0 \end{pmatrix} = \tfrac{1}{2}\begin{pmatrix} 1 \\ -\sqrt{3} \end{pmatrix}$$

$\boldsymbol{v}^L_0$ ist bereits Einheitsvektor, $|\boldsymbol{v}^L_0| = 1$

$$\boldsymbol{v}^L_0 = \tfrac{1}{2}\begin{pmatrix} 1 \\ -\sqrt{3} \end{pmatrix} = \boldsymbol{v}^R_0$$

Es gelten $L_0 = L_1 = L_2 = L_3 = \sqrt{3}/2$,
$NE_0 = NE_1 = NE_2 = NE_3 = \sqrt{3}$, also
$\alpha_0 = \alpha_1 = \alpha_2 = \alpha_3 = \frac{1}{2}$.

$$\begin{aligned}
\boldsymbol{v}^L_1 &= (1-\alpha_1)\,\boldsymbol{s}_0 + \alpha_1\,\boldsymbol{s}_1 = \tfrac{1}{2}\begin{pmatrix} 2 \\ 0 \end{pmatrix} + \tfrac{1}{2}\begin{pmatrix} -1 \\ \sqrt{3} \end{pmatrix} = \tfrac{1}{2}\begin{pmatrix} 1 \\ \sqrt{3} \end{pmatrix} = \boldsymbol{v}^R_1 \\
\boldsymbol{v}^L_2 &= (1-\alpha_2)\,\boldsymbol{s}_1 + \alpha_2\,\boldsymbol{s}_2 = \tfrac{1}{2}\begin{pmatrix} -1 \\ \sqrt{3} \end{pmatrix} + \tfrac{1}{2}\begin{pmatrix} -1 \\ -\sqrt{3} \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \end{pmatrix} = \boldsymbol{v}^R_2 \\
\boldsymbol{v}^L_3 &= (1-\alpha_3)\,\boldsymbol{s}_2 + \alpha_3\,\boldsymbol{s}_3 = \tfrac{1}{2}\begin{pmatrix} -1 \\ -\sqrt{3} \end{pmatrix} + \tfrac{1}{2}\begin{pmatrix} 2 \\ 0 \end{pmatrix} = \tfrac{1}{2}\begin{pmatrix} 1 \\ -\sqrt{3} \end{pmatrix} = \boldsymbol{v}^R_3
\end{aligned}$$

5. Längen der Parameterintervalle

$$
\begin{aligned}
A &= 16 - |\boldsymbol{v}_0^R + \boldsymbol{v}_1^L|^2 = 16 - \left| \tfrac{1}{2} \begin{pmatrix} 1+1 \\ -\sqrt{3}+\sqrt{3} \end{pmatrix} \right|^2 = 16 - 1 = 15 \\
B &= 6\, \boldsymbol{s}_0^{\mathsf{T}} (\boldsymbol{v}_0^R + \boldsymbol{v}_1^L) = 6 \begin{pmatrix} 2 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = 12 \\
C &= 36\, |\, \boldsymbol{s}_0\, |^2 = 36 \cdot 2^2 = 144 \\
h_0 &= \frac{-B + \sqrt{B^2 + AC}}{A} = \frac{-12 + \sqrt{144 + 15 \cdot 144}}{15} = \frac{12}{5}
\end{aligned}
$$

Es gilt $h_0 = h_1 = h_2 = \dfrac{12}{5} = 2.4$.

6. Koeffizienten

$$
\begin{aligned}
\boldsymbol{a}_0 &= \boldsymbol{P}_0 = \begin{pmatrix} -1 \\ -\sqrt{3}/3 \end{pmatrix} \\
\boldsymbol{b}_0 &= \boldsymbol{v}_0^R = \tfrac{1}{2} \begin{pmatrix} 1 \\ -\sqrt{3} \end{pmatrix} \\
\boldsymbol{c}_0 &= \tfrac{1}{h_0} \left(\tfrac{3}{h_0}\, \boldsymbol{s}_0 - 2\, \boldsymbol{v}_0^R - \boldsymbol{v}_1^L \right) = \tfrac{5}{12} \begin{pmatrix} 1 \\ \sqrt{3}/2 \end{pmatrix} \\
\boldsymbol{d}_0 &= \tfrac{1}{h_0^2} \left(\boldsymbol{v}_0^R + \boldsymbol{v}_1^L - \tfrac{2}{h_0}\, \boldsymbol{s}_0 \right) = \tfrac{25}{216} \begin{pmatrix} -1 \\ 0 \end{pmatrix} \\
\boldsymbol{a}_1 &= \boldsymbol{P}_1 = \begin{pmatrix} 1 \\ -\sqrt{3}/3 \end{pmatrix} \\
\boldsymbol{b}_1 &= \boldsymbol{v}_1^R = \tfrac{1}{2} \begin{pmatrix} 1 \\ \sqrt{3} \end{pmatrix} \\
\boldsymbol{c}_1 &= \tfrac{1}{h_1} \left(\tfrac{3}{h_1}\, \boldsymbol{s}_1 - 2\, \boldsymbol{v}_1^R - \boldsymbol{v}_2^L \right) = \tfrac{5}{48} \begin{pmatrix} -5 \\ \sqrt{3} \end{pmatrix} \\
\boldsymbol{d}_1 &= \tfrac{1}{h_1^2} \left(\boldsymbol{v}_1^R + \boldsymbol{v}_2^L - \tfrac{2}{h_1}\, \boldsymbol{s}_1 \right) = \tfrac{25}{432} \begin{pmatrix} 1 \\ -\sqrt{3} \end{pmatrix} \\
\boldsymbol{a}_2 &= \boldsymbol{P}_2 = \begin{pmatrix} 0 \\ 2\sqrt{3}/3 \end{pmatrix} \\
\boldsymbol{b}_2 &= \boldsymbol{v}_2^R = \begin{pmatrix} -1 \\ 0 \end{pmatrix} \\
\boldsymbol{c}_2 &= \tfrac{1}{h_2} \left(\tfrac{3}{h_2}\, \boldsymbol{s}_2 - 2\, \boldsymbol{v}_2^R - \boldsymbol{v}_3^L \right) = \tfrac{5}{48} \begin{pmatrix} 1 \\ -3\sqrt{3} \end{pmatrix} \\
\boldsymbol{d}_2 &= \tfrac{1}{h_2^2} \left(\boldsymbol{v}_2^R + \boldsymbol{v}_3^L - \tfrac{2}{h_2}\, \boldsymbol{s}_2 \right) = \tfrac{25}{432} \begin{pmatrix} 1 \\ \sqrt{3} \end{pmatrix}
\end{aligned}
$$

Damit ergibt sich die Darstellung des Renner-Subspline:

$$\begin{cases} S_{0x}(t) &= -1 + \frac{1}{2}\,t + \frac{5}{12}\,t^2 - \frac{25}{216}\,t^3\,, \\ S_{0y}(t) &= -\frac{\sqrt{3}}{3} - \frac{\sqrt{3}}{2}\,t + \frac{5}{24}\,\sqrt{3}\,t^2\,, \qquad t \in [0,\, 2.4], \end{cases}$$

$$\begin{cases} S_{1x}(t) &= 1 + \frac{1}{2}\,t - \frac{25}{48}\,t^2 + \frac{25}{432}\,t^3\,, \\ S_{1y}(t) &= -\frac{\sqrt{3}}{3} + \frac{\sqrt{3}}{2}\,t + \frac{5}{48}\,\sqrt{3}\,t^2 - \frac{25}{432}\,\sqrt{3}\,t^3\,, \qquad t \in [0,\, 2.4], \end{cases}$$

$$\begin{cases} S_{2x}(t) &= -t + \frac{5}{48}\,t^2 + \frac{25}{432}\,t^3\,, \\ S_{2y}(t) &= \frac{2}{3}\,\sqrt{3} - \frac{5}{16}\,\sqrt{3}\,t^2 + \frac{25}{432}\,\sqrt{3}\,t^3\,, \qquad t \in [0,\, 2.4]. \end{cases}$$

Der Kreis hat den Radius r=1.15470, der minimale Abstand des Renner-Subspline vom Ursprung beträgt 1.09697, die maximale Abweichung vom Radius also 0.05773. Der Kreis hat den Umfang 7.25520, die gesamte Länge der Parameterintervalle ist $3 \cdot 2.4 = 7.2$. Mit nur 3 Stützpunkten auf dem Kreis nähert der Renner-Subspline diesen Kreis recht gut an. Wählt man 6 Stützpunkte auf dem Kreis, so ist die maximale Abweichung vom Radius 0.00362, und die gesamte Länge der Parameterintervalle ist 7.25207, also nur wenig verschieden vom Umfang 7.25520 des Kreises.

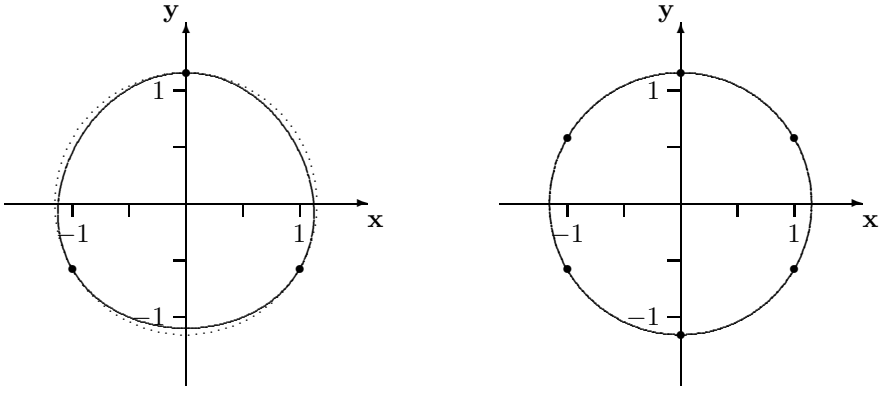


Abb. 11.4. Geschlossener Renner-Subspline ohne Eckpunkt

□

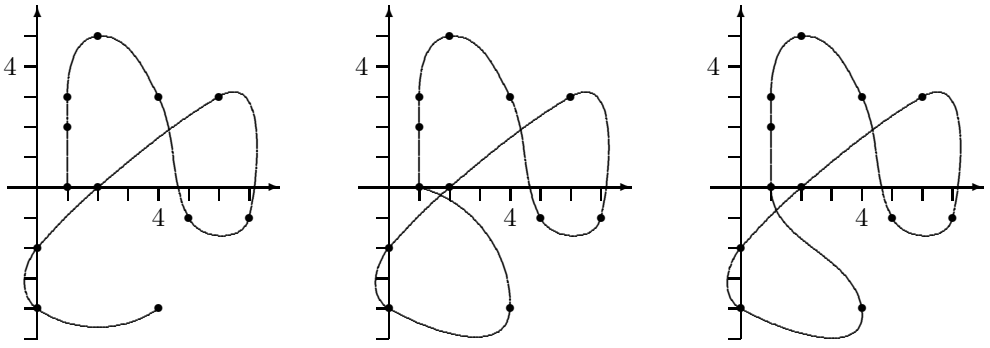
Beispiel 11.7.

Gegeben: Die 12 Punkte

i	0	1	2	3	4	5	6	7	8	9	10	11
x_i	1	1	1	2	4	5	7	6	2	0	0	4
y_i	0	2	3	5	3	-1	-1	3	0	-2	-4	-4

Gesucht:
a) Der nicht geschlossene Renner-Subspline zu den Punkten $P_0, \ldots, P_{11}$.
b) Der geschlossene Renner-Subspline mit Eckpunkt zu den Punkten $P_0, \ldots, P_{11}$, $P_{12} = P_0$.
c) Der geschlossene Renner-Subspline ohne Eckpunkt zu den Punkten $P_0, \ldots, P_{11}$, $P_{12} = P_0$.

Lösung:



a) Renner-Subspline, nicht geschlossen

b) Renner-Subspline, geschlossen mit Eckpunkt

c) Renner-Subspline, geschlossen ohne Eckpunkt

Abb. 11.5. Renner-Subsplines

□

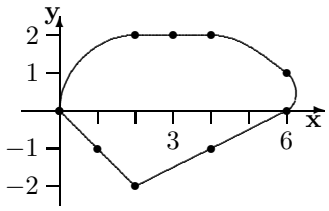
Beispiel 11.8.

Gegeben: Die 10 Punkte

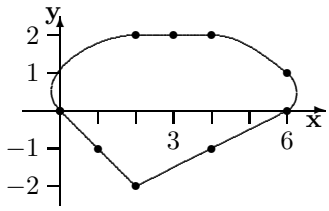
i	0	1	2	3	4	5	6	7	8	9
x_i	0	2	3	4	6	6	4	2	1	0
y_i	0	2	2	2	1	0	-1	-2	-1	0

Gesucht:
- der geschlossene Renner-Subspline mit Eckpunkt und
- der geschlossene Renner-Subspline ohne Eckpunkt

Lösung: Der Punkt $P_7 = (2, -2)$ ist eine Ecke, die nur mittels Abrundung beseitigt werden kann (siehe Beispiel 11.12).



Renner-Subspline, geschlossen, mit Eckpunkt



Renner-Subspline, geschlossen, ohne Eckpunkt

Abb. 11.6. Renner-Subsplines

□

11.3 Abrundung von Ecken bei Akima- und Renner-Kurven

Bei den Akima- und bei den Renner-Kurven wird genau dann bei P_i eine Ecke erzeugt, wenn sowohl die Punkte P_{i-2}, P_{i-1}, P_i als auch P_i, P_{i+1}, P_{i+2} kollinear sind, nicht jedoch die Punkte P_{i-1}, P_i, P_{i+1}. Falls solche Ecken unerwünscht sind, können sie durch Einfügen je eines Punktes beseitigt und die Kurve damit abgerundet werden.

Falls P_i eine Ecke ist, werden zwecks Abrundung ein Punkt P auf der Strecke zwischen P_i und P_{i-1} und ein Punkt Q auf der Strecke zwischen P_i und P_{i+1} erzeugt. Nach Umspeichern der Punkte P_{i+1} bis P_n nach P_{i+2} bis P_{n+1} werden $P_i = P$ und $P_{i+1} = Q$ gesetzt. Weil nun sowohl P_{i-1}, P_i, P_{i+1} als auch P_i, P_{i+1}, P_{i+2} nicht kollinear sind, ist die Ecke beseitigt. Der Bereich des Abrundungssegmentes von $P_i = P$ nach $P_{i+1} = Q$ wird mittels der Zahl β, $0 < \beta < 1$, festgelegt.

Der Algorithmus 11.9 ist nur für Akima-Subsplines anwendbar, der Algorithmus 11.10 dagegen für Akima- und Renner-Subsplines.

Algorithmus 11.9.

Gegeben: $n+1$ Punkte $P_i = (x_i, y_i)\,,\ i = 0(1)n,\ n \geq 4,\quad x_0 < x_1 < \ldots < x_n$ sowie $\beta,\ 0 < \beta < 1$.

Gesucht: Zu jeder Ecke ein geänderter und ein zusätzlicher Punkt, so dass der nicht periodische Subspline zu den nun vorliegenden Punkten anstelle jeder Ecke ein Abrundungssegment besitzt.

1. Setze $i := 2$.
2. Berechne die Sehnensteigungen
$$m_k = \frac{y_{k-1} - y_k}{x_{k-1} - x_k} \quad \text{für } k = i{-}2,\ i{-}1,\ i,\ i{+}1.$$
3. Falls $|\,m_{i-2} - m_{i-1}\,| = 0$ und $|\,m_i - m_{i+1}\,| = 0$ und $|\,m_{i-1} - m_i\,| > 0$ sind, ist P_i eine Ecke; weiter mit 4.
Andernfalls setze $i := i + 1$; weiter mit 5.
4. Beseitigung der Ecke P_i.
   4.1 Berechne $h = \min(x_i - x_{i-1}, x_{i+1} - x_i)$ und
   $P = (x_i - \beta h, y_i - \beta h m_{i-1})$, $Q = (x_i + \beta h, y_i + \beta h m_i)$.
   4.2 Umspeichern der Punkte P_{i+1} bis P_n:
   $P_{j+1} := P_j$ für $j = n(-1)i{+}1$.
   Setze $n := n + 1$.
   4.3 Setze $P_i := P$ und $P_{i+1} := Q$.
   4.4 Setze $i := i + 3$.
5. Falls $i \leq n{-}2$ ist, weiter mit 2.
Andernfalls Abbruch und Ausgabe der Punkte $P_0, P_1, \ldots, P_n$.

Algorithmus 11.10.

Gegeben: $n+1$ Punkte $P_i = (x_i, y_i)$ in der Ebene oder $P_i = (x_i, y_i, z_i)$ im Raum, $i = 0(1)n,\ n \geq 4,$ mit $P_i \neq P_{i+1}$ für $i = 0(1)n{-}1$.
Wenn P_{i-1}, P_i, P_{i+1} kollinear sind, muss P_i zwischen P_{i-1} und P_{i+1} liegen.
Es sei $P_n \neq P_0$. Ferner $\beta,\ 0 < \beta < 1$.

Gesucht: Zu jeder Ecke ein geänderter und ein zusätzlicher Punkt, so dass der Subspline zu den nun vorliegenden Punkten anstelle jeder Ecke ein Abrundungssegment besitzt.

1. Setze $i := 2$.
2. Berechne die Sehnenvektoren
$$\boldsymbol{p} = \boldsymbol{P}_{i-1} - \boldsymbol{P}_{i-2}, \quad \boldsymbol{q} = \boldsymbol{P}_i - \boldsymbol{P}_{i-1},$$
$$\boldsymbol{r} = \boldsymbol{P}_{i+1} - \boldsymbol{P}_i, \quad \boldsymbol{s} = \boldsymbol{P}_{i+2} - \boldsymbol{P}_{i+1}.$$

3. Berechne die Quadrate der Flächeninhalte der von je zwei Vektoren aufgespannten Parallelogramme
$$E = (\boldsymbol{p}^\mathsf{T}\boldsymbol{p})(\boldsymbol{q}^\mathsf{T}\boldsymbol{q}) - (\boldsymbol{p}^\mathsf{T}\boldsymbol{q})^2,$$
$$F = (\boldsymbol{q}^\mathsf{T}\boldsymbol{q})(\boldsymbol{r}^\mathsf{T}\boldsymbol{r}) - (\boldsymbol{q}^\mathsf{T}\boldsymbol{r})^2,$$
$$G = (\boldsymbol{r}^\mathsf{T}\boldsymbol{r})(\boldsymbol{s}^\mathsf{T}\boldsymbol{s}) - (\boldsymbol{r}^\mathsf{T}\boldsymbol{s})^2.$$
Falls $E = 0$ und $G = 0$ und $F > 0$ sind, ist P_i eine Ecke; weiter mit 4.
Andernfalls setze $i := i + 1$; weiter mit 5.
4. Beseitigung der Ecke P_i.
   4.1 Berechne $d = \min(|\boldsymbol{q}|, |\boldsymbol{r}|)$ und
   $\boldsymbol{P} = \boldsymbol{P}_i - \beta d\boldsymbol{q} / |\boldsymbol{q}|$, $\boldsymbol{Q} = \boldsymbol{P}_i + \beta d\boldsymbol{r} / |\boldsymbol{r}|$.
   4.2 Umspeichern der Punkte P_{i+1} bis P_n:
   $P_{j+1} := P_j$ für $j = n(-1)i{+}1$.
   Setze $n := n + 1$.
   4.3 Setze $\boldsymbol{P}_i := \boldsymbol{P}$ und $\boldsymbol{P}_{i+1} := \boldsymbol{Q}$.
   4.4 Setze $i := i + 3$.
5. Falls $i \le n{-}2$ ist, weiter mit 2.
Andernfalls Abbruch und Ausgabe der Punkte P_0, P_1, …, P_n.

Die Algorithmen 11.9 und 11.10, die für den Fall eines nicht periodischen bzw. eines nicht geschlossenen Subspline formuliert sind, können wie folgt auch bei einem periodischen Subspline (Periode $p = x_n - x_0$, $y_n = y_0$) bzw. einem geschlossenen Subspline ($P_n = P_0$) angewendet werden.

Wenn P_0 eine Ecke ist, werden die $n + 3$ Punkte
$$P_0, P_1, \ldots, P_n, P_{n+1}, P_{n+2}$$
bereitgestellt, und wenn P_0 keine Ecke ist, die $n + 3$ Punkte
$$P_{-1}, P_0, P_1, \ldots, P_n, P_{n+1}.$$
Dabei werden gesetzt im Fall

periodisch	geschlossen
$P_{-1} = (x_{n-1} - p, y_{n-1})$,	$P_{-1} = P_{n-1}$,
$P_{n+1} = (x_1 + p, y_1)$,	$P_{n+1} = P_1$,
$P_{n+2} = (x_2 + p, y_2)$,	$P_{n+2} = P_2$.

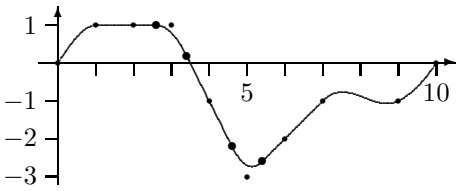
In beiden Fällen müssen von dem vom Algorithmus ausgegebenen Punkten der erste und der letzte Punkt gestrichen werden.
Falls P_0 eine Ecke ist, wird dieser Punkt beseitigt, und der nächste Punkt übernimmt dann die Rolle des ersten Punktes.

Beispiel 11.11. (Fortsetzung von Beispiel 11.4.)

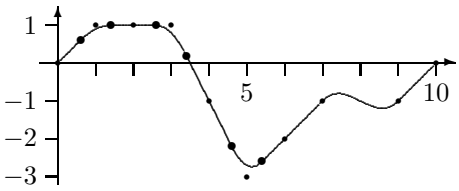
Gegeben: Dieselbe Wertetabelle wie in Beispiel 11.4.

Gesucht: Ein nicht periodischer und ein periodischer Akima-Subspline mit Abrundung.

Lösung:



Akima-Subspline, nicht periodisch, mit Abrundung



Akima-Subspline, periodisch, mit Abrundung

Abb. 11.7. Akima-Subsplines mit Abrundung ($\beta = 0.4$). Die neuen Punkte an den Abrundungsstellen sind hervorgehoben (vgl. Abb. 11.3). □

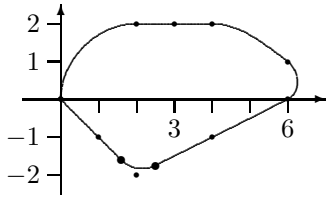
Beispiel 11.12. (Fortsetzung von Beispiel 11.8.)

Gegeben: Dieselben Punkte wie in Beispiel 11.8.

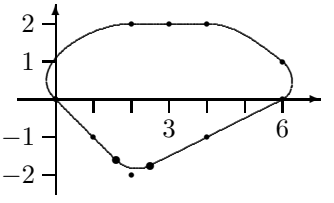
Gesucht: Jeweils mit Abrundung der Ecke $P_7 = (2, -2)$

- der geschlossene Renner-Subspline mit Eckpunkt und
- der geschlossene Renner-Subspline ohne Eckpunkt

Lösung:



Renner-Subspline, geschlossen, mit Eckpunkt, mit Abrundung



Renner-Subspline, geschlossen, ohne Eckpunkt, mit Abrundung

Abb. 11.8. Renner-Subsplines mit Abrundung ($\beta = 0.4$). Die neuen Punkte an den Abrundungsstellen sind hervorgehoben (vgl. Abb. 11.6). □

11.4 Berechnung der Länge einer Kurve

Bei der Herleitung der Formeln für den Renner-Subspline hatte sich ergeben, dass die Länge h_i des Parameterintervalls für das Segment $\boldsymbol{S}_i$ ungefähr gleich der Länge dieses Kurvensegmentes ist. Damit ergibt sich für die gesamte Länge L eines Renner-Subspline

$$L \approx \sum_{i=0}^{n-1} h_i \,. \tag{11.5}$$

Je kleiner die Abstände $|\boldsymbol{s}_i| = |\boldsymbol{P}_{i+1} - \boldsymbol{P}_i|$ benachbarter Stützpunkte sind, umso genauer liefert diese Formel die Länge L (siehe dazu auch Beispiel 11.14). Allgemein kann die Länge einer Kurve wie folgt ermittelt werden.
Die Länge L einer Kurve mit einer stetig differenzierbaren Parameterdarstellung

$$\boldsymbol{X} : [a,b] \to \mathbb{R}^m \,, \quad t \mapsto \boldsymbol{X}(t) \,, \quad t \in [a,b] \,,$$

$m = 2$ oder $m = 3$, ist gleich dem Integral über die Länge $|\boldsymbol{X}'(t)|$ ihres Tangentenvektors $\boldsymbol{X}'(t)$:

$$L = \int_a^b |\boldsymbol{X}'(t)| \,\mathrm{d}\, t \,. \tag{11.6}$$

Bei einer ebenen Kurve ($m = 2$) sind

$$\boldsymbol{X}(t) = \big(x(t), y(t)\big)^{\mathsf{T}} \,, \quad \boldsymbol{X}'(t) = \big(x'(t), y'(t)\big)^{\mathsf{T}} \,,$$
$$|\boldsymbol{X}'(t)| = \sqrt{x'^{\,2}(t) + y'^{\,2}(t)} \,.$$

und bei einer Raumkurve ($m = 3$)

$$\boldsymbol{X}(t) = \big(x(t), y(t), z(t)\big)^{\mathsf{T}} \,, \quad \boldsymbol{X}'(t) = \big(x'(t), y'(t), z'(t)\big)^{\mathsf{T}} \,,$$
$$|\boldsymbol{X}'(t)| = \sqrt{x'^{\,2}(t) + y'^{\,2}(t) + z'^{\,2}(t)} \,.$$

Wenn eine ebene Kurve durch die Polarkoordinaten t und $r(t)$ gegeben ist, ergibt sich mit

$$x(t) = r(t)\, \cos t \,, \quad y(t) = r(t)\, \sin t$$
$$|\, \boldsymbol{X}'(t)\,| = \sqrt{r^2(t) + r'^{\,2}(t)}. \tag{11.7}$$

Das Integral

$$\int_a^b f(t) \,\mathrm{d}\, t \quad \text{mit} \quad f(t) = |\boldsymbol{X}'(t)|$$

ist im Allgemeinen nicht elementar auswertbar und wird deshalb näherungsweise, z. B. mit dem Quadraturverfahren von Romberg (siehe Abschnitt 14.10), berechnet.
Angewendet auf parametrische kubische Splines (22) und (11.4) ergibt sich für die Länge des i-ten Segmentes

$$L_i = \int_{t_i}^{t_{i+1}} |\boldsymbol{S}_i'(t)|\,\mathrm{d}\,t$$

mit $$|\boldsymbol{S}_i'(t)| = |\boldsymbol{b}_i + 2\boldsymbol{c}_i\,(t-t_i) + 3\boldsymbol{d}_i\,(t-t_i)^2|$$

oder $$L_i = \int_{0}^{h_i} |\boldsymbol{S}_i'(t)|\,\mathrm{d}\,t$$

mit $$|\boldsymbol{S}_i'(t)| = |\boldsymbol{b}_i + 2\boldsymbol{c}_i\,t + 3\boldsymbol{d}_i\,t^2|\,.$$

Mit Verwendung der Koordinatenfunktionen ist für $m = 2$

$$|\boldsymbol{S}_i'(t)| = \sqrt{S_{ix}'^{\,2}(t) + S_{iy}'^{\,2}(t)}$$

und für $m = 3$

$$|\boldsymbol{S}_i'(t)| = \sqrt{S_{ix}'^{\,2}(t) + S_{iy}'^{\,2}(t) + S_{iz}'^{\,2}(t)}\,.$$

Für die Länge L einer parametrischen kubischen Splinekurve ergibt sich also

$$L = \sum_{i=0}^{n-1} L_i\,.$$

Bei einem Renner-Subspline mit Ecken gilt die stetige Differenzierbarkeit von $\boldsymbol{S}$ nicht im gesamten Intervall $[t_0, t_n]$, jedoch für jedes einzelne Segment $\boldsymbol{S}_i$, da die Ecken nur in Knoten t_i auftreten.

Der Graph einer stetig differenzierbaren Funktion

$$f : [a,b] \to \mathbb{R}\,, \quad x \mapsto f(x)\,, \quad x \in [a,b]$$

kann als Sonderfall einer ebenen Kurve (mit dem Parameter x anstelle von t) in der Form

$$\boldsymbol{X}(x) = \big(x, f(x)\big)^{\mathsf{T}}\,, \quad x \in [a,b]$$

dargestellt werden. Mit dem Tangentenvektor

$$\boldsymbol{X}'(x) = \big(1, f'(x)\big)^{\mathsf{T}}$$

ergibt sich für die Länge des Graphen

$$L = \int_a^b |\boldsymbol{X}'(x)|\,\mathrm{d}\,x = \int_a^b \sqrt{1 + f'^{\,2}(x)}\,\mathrm{d}\,x\,.$$

Der Graph des Segmentes S_i einer kubischen Splinefunktion (16) oder eines Akima-Subspline (11.1) hat somit die Länge

$$\begin{aligned} L_i &= \int_{x_i}^{x_{i+1}} \sqrt{1 + S_i'^{\,2}(x)}\,\mathrm{d}\,x \\ &= \int_{x_i}^{x_{i+1}} \sqrt{1 + \big(b_i + 2c_i\,(x-x_i) + 3d_i\,(x-x_i)^2\big)^2}\,\mathrm{d}\,x\,. \end{aligned} \tag{11.8}$$

Die gesamte Länge des Graphen einer Splinefunktion ist

$$L = \sum_{i=0}^{n-1} L_i \,.$$

Die Formel gilt auch für einen Akima-Subspline, der in einem oder in mehreren Knoten x_i Ecken besitzt.

Die obigen Ausführungen lassen sich sinngemäß auf Splines, die Polynome eines Grades $k > 3$ benutzen, übertragen.

Beispiel 11.13.

Gegeben: Der im Beispiel 11.2 ermittelte Akima-Subspline

$$\begin{array}{lcll} S_0(x) & = & 2\,x - 2\,x^2 = 2\,(x - x^2)\,, & x \in [0, 0.5]\,, \\ S_1(x) & = & 0.5 - 3\,(x - 0.5)^2 + 2\,(x - 0.5)^3\,, & x \in [0.5, 1.5]\,, \\ S_2(x) & = & -0.5 + 0.5\,(x - 1.5)^2\,, & x \in [1.5, 3.5]\,. \end{array}$$

Gesucht: Die Länge des Graphen dieses Akima-Subspline.

Lösung: Die Ableitungen der Koordinatenfunktionen sind

$$\begin{array}{lcl} S_0'(x) & = & 2\,(1 - 2x)\,, \\ S_1'(x) & = & -6\,(x - 0.5) + 6\,(x - 0.5)^2 = 6\,(x - 0.5)\,(x - 1.5)\,, \\ S_2'(x) & = & x - 1.5\,. \end{array}$$

Mit (11.8) ergeben sich für die Längen der Graphen der Segmente

$$\begin{array}{lcl} L_0 & = & \displaystyle\int_0^{0.5} \sqrt{1 + 4\,(1 - 2x)^2}\,\mathrm{d}\,x\,, \\ L_1 & = & \displaystyle\int_{0.5}^{1.5} \sqrt{1 + 36\,(x - 0.5)^2\,(x - 1.5)^2}\,\mathrm{d}\,x\,, \\ L_2 & = & \displaystyle\int_{1.5}^{3.5} \sqrt{1 + (x - 1.5)^2}\,\mathrm{d}\,x\,. \end{array}$$

Das Romberg-Verfahren liefert, auf 6 Dezimalen genau,

$$L_0 = 0.739\,471\,, \quad L_1 = 1.457\,240\,, \quad L_2 = 2.957\,886\,,$$

und damit ergibt sich für die gesamte Länge

$$L = L_0 + L_1 + L_2 = 5.154\,597.$$

□

11.5 Flächeninhalt einer geschlossenen ebenen Kurve

Gegeben sei eine geschlossene ebene Kurve mit einer stetig differenzierbaren Parameterdarstellung

$$\boldsymbol{X} : [a,b] \to \mathbb{R}^2,\ t \mapsto \boldsymbol{X}(t) = \big(x(t), y(t)\big)^{\mathsf{T}},\ t \in [a,b], \quad \text{mit } \boldsymbol{X}(a) = \boldsymbol{X}(b).$$

Die Kurve sei einfach geschlossen wie ein Kreis. Sie zerlegt die Ebene in genau zwei Gebiete, ein inneres und ein äußeres (beispielsweise sind die geschlossenen Kurven in Abb. 11.5 nicht einfach geschlossen).

Der Flächeninhalt des von der Kurve berandeten inneren Gebietes ist

$$F = \frac{1}{2} \int_a^b \det\big(\boldsymbol{X}(t), \boldsymbol{X}'(t)\big)\, \mathrm{d}t.$$

Wenn beim Durchlaufen der Kurve im Sinne wachsender Parameterwerte das innere Gebiet links liegt, ist der Flächeninhalt F positiv, andernfalls negativ.

Mit $\boldsymbol{X}'(t) = \big(x'(t), y'(t)\big)^{\mathsf{T}}$ ergibt sich

$$F = \frac{1}{2} \int_a^b \big(x(t)\, y'(t) - y(t)\, x'(t)\big)\, \mathrm{d}t.$$

Wenn die Kurve durch die Polarkoordinaten t und $r(t)$ gegeben ist mit

$$x(t) = r(t)\, \cos t\,, \quad y(t) = r(t)\, \sin t\,,$$

ergibt sich für den Flächeninhalt

$$F = \frac{1}{2} \int_a^b r^2(t)\, \mathrm{d}t. \tag{11.9}$$

Diese Integrale sind im Allgemeinen nicht elementar auswertbar und müssen deshalb mit einer Quadraturformel oder mit dem Romberg-Verfahren berechnet werden (siehe Kapitel 14).

Im Falle eines geschlossenen kubischen Spline oder Subspline

$$\boldsymbol{S}(t) \equiv \boldsymbol{S}_i(t) = \boldsymbol{a}_i + \boldsymbol{b}_i t + \boldsymbol{c}_i t^2 + \boldsymbol{d}_i t^3, \quad t \in [0, h_i]\,,\ i = 0(1)n{-}1, \tag{11.10}$$

mit $\boldsymbol{S}_0(0) = \boldsymbol{S}_{n-1}(h_{n-1})$ ist der Flächeninhalt des vom Spline umschlossenen Gebietes

$$F = \sum_{i=0}^{n-1} F_i \tag{11.11}$$

mit

$$F_i = \frac{1}{2} \int_0^{h_i} \det\big(\boldsymbol{S}_i(t), \boldsymbol{S}_i'(t)\big)\, \mathrm{d}t.$$

F_i ist der Flächeninhalt des Gebietes, das von den beiden Strecken vom Ursprung O zu den Punkten $\boldsymbol{S}_i(0)$ und $\boldsymbol{S}_i(h_i)$ und vom Splinesegment $\boldsymbol{S}_i$ zwischen diesen Punkten begrenzt wird.

Mit $\boldsymbol{S}_i(t)$ und mit

$$\boldsymbol{S}_i'(t) = \boldsymbol{b}_i + 2\,\boldsymbol{c}_i t + 3\,\boldsymbol{d}_i t^2$$

muss nun der Integrand von F_i berechnet werden.

$$\begin{aligned}
\det\big(\boldsymbol{S}_i(t), \boldsymbol{S}_i'(t)\big) &= \det(\boldsymbol{a}_i + \boldsymbol{b}_i t + \boldsymbol{c}_i t^2 + \boldsymbol{d}_i t^3, \boldsymbol{b}_i + 2\,\boldsymbol{c}_i t + 3\,\boldsymbol{d}_i t^2) \\
&= \det(\boldsymbol{a}_i, \boldsymbol{b}_i) + 2t\det(\boldsymbol{a}_i, \boldsymbol{c}_i) + 3t^2\det(\boldsymbol{a}_i, \boldsymbol{d}_i) \\
&\quad + 2t^2\det(\boldsymbol{b}_i, \boldsymbol{c}_i) + 3t^3\det(\boldsymbol{b}_i, \boldsymbol{d}_i) + t^2\det(\boldsymbol{c}_i, \boldsymbol{b}_i) \\
&\quad + 3t^4\det(\boldsymbol{c}_i, \boldsymbol{d}_i) + t^3\det(\boldsymbol{d}_i, \boldsymbol{b}_i) + 2t^4\det(\boldsymbol{d}_i, \boldsymbol{c}_i) \\
&= \det(\boldsymbol{a}_i, \boldsymbol{b}_i) + 2t\det(\boldsymbol{a}_i, \boldsymbol{c}_i) + 3t^2\det(\boldsymbol{a}_i, \boldsymbol{d}_i) \\
&\quad + t^2\det(\boldsymbol{b}_i, \boldsymbol{c}_i) + 2t^3\det(\boldsymbol{b}_i, \boldsymbol{d}_i) + t^4\det(\boldsymbol{c}_i, \boldsymbol{d}_i)
\end{aligned}$$

Der Integrand ist also ein Polynom 4. Grades. Damit ist

$$\begin{aligned}
\int_0^{h_i} \det\big(\boldsymbol{S}_i(t), \boldsymbol{S}_i'(t)\big)\,\mathrm{d}t &= h_i\det(\boldsymbol{a}_i, \boldsymbol{b}_i) + h_i^2\det(\boldsymbol{a}_i, \boldsymbol{c}_i) + h_i^3\det(\boldsymbol{a}_i, \boldsymbol{d}_i) \\
&\quad + \frac{h_i^3}{3}\det(\boldsymbol{b}_i, \boldsymbol{c}_i) + \frac{h_i^4}{2}\det(\boldsymbol{b}_i, \boldsymbol{d}_i) + \frac{h_i^5}{5}\det(\boldsymbol{c}_i, \boldsymbol{d}_i)
\end{aligned}$$

Mit den Koordinaten der Spline-Koeffizienten

$$\boldsymbol{a}_i = \begin{pmatrix} a_{ix} \\ a_{iy} \end{pmatrix},\ \boldsymbol{b}_i = \begin{pmatrix} b_{ix} \\ b_{iy} \end{pmatrix},\ \boldsymbol{c}_i = \begin{pmatrix} c_{ix} \\ c_{iy} \end{pmatrix},\ \boldsymbol{d}_i = \begin{pmatrix} d_{ix} \\ d_{iy} \end{pmatrix} \tag{11.12}$$

ergibt sich für den Flächeninhalt zum Segment $\boldsymbol{S}_i$

$$\begin{aligned}
F_i &= \frac{h_i}{2}\left\{ \begin{vmatrix} a_{ix} & b_{ix} \\ a_{iy} & b_{iy} \end{vmatrix} + h_i \left\{ \begin{vmatrix} a_{ix} & c_{ix} \\ a_{iy} & c_{iy} \end{vmatrix} \right.\right. \\
&\quad + h_i \left\{ \begin{vmatrix} a_{ix} & d_{ix} \\ a_{iy} & d_{iy} \end{vmatrix} + \frac{1}{3} \begin{vmatrix} b_{ix} & c_{ix} \\ b_{iy} & c_{iy} \end{vmatrix} \right. \\
&\quad \left.\left. + h_i \left\{ \frac{1}{2} \begin{vmatrix} b_{ix} & d_{ix} \\ b_{iy} & d_{iy} \end{vmatrix} + \frac{h_i}{5} \begin{vmatrix} c_{ix} & d_{ix} \\ c_{iy} & d_{iy} \end{vmatrix} \right\} \right\} \right\}.
\end{aligned} \tag{11.13}$$

Der Flächeninhalt eines Spline oder Subspline (11.10) kann also mit (11.11), (11.12) und (11.13) sehr einfach berechnet werden.
Im folgenden Beispiel wird eine geschlossene Kurve

$$t \mapsto \boldsymbol{X}(t), \quad t \in [a, b]$$

durch einen Subspline angenähert. Dazu werden mittels einer Zerlegung des Intervalls mit $\Delta t = (b-a)/n$ und den Knoten $t_i = a + i\Delta t$, $i = 0(1)n$, die Punkte

$$\boldsymbol{P}_i = \boldsymbol{X}(t_i)$$

und die Tangenteneinheitsvektoren

$$\boldsymbol{v}_i = \boldsymbol{X}'(t_i) \,/\, |\, \boldsymbol{X}'(t_i)\,|$$

berechnet. Zu $\boldsymbol{P}_i$ und $\boldsymbol{v}_i$ wird mit dem Algorithmus 11.5 der Subspline $\boldsymbol{S}(t)$ erzeugt. Dessen Flächeninhalt und Länge sind bei einer genügend feinen Zerlegung gute Näherungen für den Flächeninhalt und die Länge der Kurve $\boldsymbol{X}(t)$.

Beispiel 11.14.

Gegeben: Die einfach geschlossene Kurve

$$\boldsymbol{X}(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} r(t)\cos t \\ r(t)\sin t \end{pmatrix} \text{ mit } r(t) = \sin(2t) + 0.2\sin(8t),\ t \in \left[0, \tfrac{\pi}{2}\right]; \quad (11.14)$$

ihr Tangentenvektor ist

$$\boldsymbol{X}'(t) = \begin{pmatrix} r'(t)\cos t - r(t)\sin t \\ r'(t)\sin t + r(t)\cos t \end{pmatrix} \text{ mit } r'(t) = 2\,\cos(2t) + 1.6\cos(8t).$$

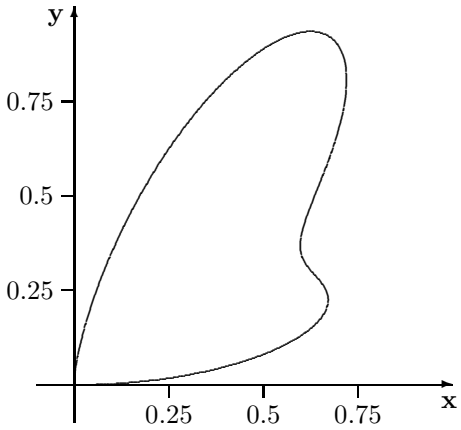


Abb. 11.9. Die einfach geschlossene Kurve (11.14)

Gesucht: Der Flächeninhalt F und die Länge L dieser geschlossenen Kurve.

Lösung: Die Integrale in den Formeln (11.9) für F sowie (11.6) mit (11.7) für L werden mit dem Romberg-Verfahren auf 6 Dezimalen genau berechnet.

Ergebnisse:

$$F = 0.408\,407, \quad L = 2.727\,158.$$

Im Folgenden wird die Kurve $\boldsymbol{X}$ durch einen Subspline $\boldsymbol{S}$ angenähert, der durch hinreichend viele Punkte der Kurve geht und in diesen Punkten dieselbe Tangente besitzt wie die Kurve.

Für drei Zerlegungen Z_n des Intervalls $[0, \pi/2]$ mit $\Delta t = \pi/2n$ und den Knoten $t_i = i\Delta t$, $i = 0(1)n$, und zwar für $n = 15$, $n = 20$ und $n = 25$, werden die Punkte $\boldsymbol{P}_i = \boldsymbol{X}(t_i)$ und die Tangenteneinheitsvektoren $\boldsymbol{v}_i = \boldsymbol{X}'(t_i) \,/\, |\, \boldsymbol{X}'(t_i)\,|$ berechnet. Zu $\boldsymbol{P}_i$ und $\boldsymbol{v}_i$ werden mit dem Algorithmus 11.5 (man beachte die Bemerkung) für die Zerlegungen Z_n die Subsplines $\boldsymbol{S}_n(t)$ erzeugt. Die Flächeninhalte F_n dieser Subsplines werden mit (11.11) und (11.13) berechnet und die Längen L_n angenähert mit (11.5).

Die Ergebnisse sind in der folgenden Tabelle zusammengestellt. Sie zeigen die gute Annäherung der Kurve $\boldsymbol{X}$ durch die Subsplines $\boldsymbol{S}_n$.

n	15	20	25
F_n	0.408 760	0.408 513	0.408 450
F	0.408 407	0.408 407	0.408 407
$\|\, F_n - F \,\|$	0.000 353	0.000 106	0.000 043
$\|\, F_n - F \,\| \,/\, F$	0.000 864	0.000 260	0.000 105
L_n	2.730 553	2.727 875	2.727 408
L	2.727 158	2.727 158	2.727 158
$\|\, L_n - L \,\|$	0.003 395	0.000 717	0.000 250
$\|\, L_n - L \,\| \,/\, L$	0.001 245	0.000 263	0.000 092

□

11.6 Entscheidungshilfen

Die Entscheidung für Akima- oder Renner-Subsplines hängt zunächst davon ab, ob die einmalige stetige Differenzierbarkeit genügt. Ein Akima- oder Renner-Subspline verbindet je drei aufeinander folgende kollineare Stützpunke geradlinig. Mit den Subsplines lassen sich Kurven darstellen, die sich zusammensetzen aus krummlinigen und geradlinigen Segmenten, die tangential aneinander schließen. Das ist mit den zweimal stetig differenzierbaren kubischen Splines nicht möglich. Wenn zwei benachbarte geradlinige Segmente eine Ecke erzeugen, kann diese durch Einfügen eines Abrundungssegmentes beseitigt werden. Bei beliebig vorgegebenen Stützpunkten $P_i = (x_i, y_i)$ in der Ebene oder $P_i = (x_i, y_i, z_i)$ im Raum müssen Renner-Subsplines benutzt werden. Wenn in der Ebene die Stützstellen x_i monoton angeordnet sind, verwendet man Akima-Subsplines. Vorteilhaft ist auch, dass die Berechnung der Polynomkoeffizienten nicht die Lösung eines linearen Gleichungssystems erfordert.
Die Orientierungstabelle in Abschnitt 10.4 enthält auch Hinweise für den Einsatz von Akima- und Renner-Subsplines.

Ergänzende Literatur zu Kapitel 11

[UBER1995]

Kapitel 12

Zweidimensionale Splines, Oberflächensplines, Bézier-Splines, B-Splines

12.1 Interpolierende zweidimensionale Polynomsplines dritten Grades zur Konstruktion glatter Flächen

Gegeben seien in der x, y-Ebene ein Rechteckgitter

$$a = x_0 < x_1 < \ldots < x_n = b\,, \quad c = y_0 < y_1 < \ldots < y_m = d$$

mit den Gitterpunkten (x_i, y_j), $i = 0(1)n$, $j = 0(1)m$, sowie eine auf dem Rechteck $R = \{\,(x,y) \;\big|\; a \le x \le b, c \le y \le d\,\}$ definierte Funktion

$$u : R \to \mathbb{R}\,, \quad (x,y) \mapsto u\,(x,y)$$

und deren Werte

$$u_{ij} := u(x_i, y_j), \quad i = 0(1)n, \quad j = 0(1)m,$$

in den Gitterpunkten.

Gesucht ist eine zweidimensionale interpolierende Splinefunktion

$$S : R \to \mathbb{R}\,, \quad (x,y) \mapsto S\,(x,y)\,,$$

deren Graph $\{\,(x,y,z)\,|\,(x,y) \in \mathbb{R},\, z = S(x,y)\,\}$ eine möglichst glatte Fläche sein soll.

Für S wird eine bikubische Splinefunktion gewählt, die durch die folgenden Eigenschaften definiert wird:

(1) S erfülle die Interpolationsbedingungen

$$S(x_i, y_j) = u_{ij}, \quad i = 0(1)n, \quad j = 0(1)m.$$

(2) S sei auf R einmal stetig differenzierbar, $\partial^2 S/\partial x \partial y$ sei stetig auf R.

(3) In jedem Teilrechteck R_{ij} mit

$$R_{ij} := \{\,(x,y)\,|\,x_i \le x \le x_{i+1},\, y_j \le y \le y_{j+1}\,\}\,, \quad i = 0(1)n{-}1,\, j = 0(1)m{-}1,$$

sei S identisch mit einem bikubischen Polynom

$$\begin{aligned} S(x,y) \equiv S_{ij}(x,y) &= \sum_{k=0}^{3}\sum_{s=0}^{3} a_{ijks}(x-x_i)^k(y-y_j)^s \\ \text{für} \quad (x,y) \in R_{ij}, &\quad i = 0(1)n{-}1, \quad j = 0(1)m{-}1. \end{aligned} \tag{12.1}$$

(4) S erfülle gewisse (noch vorzugebende) Randbedingungen.

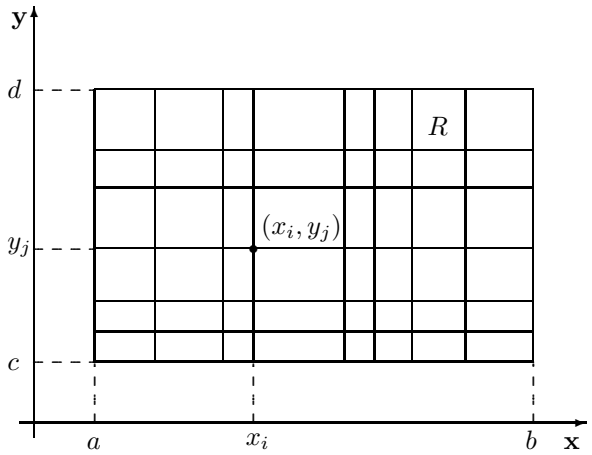


Abb. 12.1. Rechteckgitter

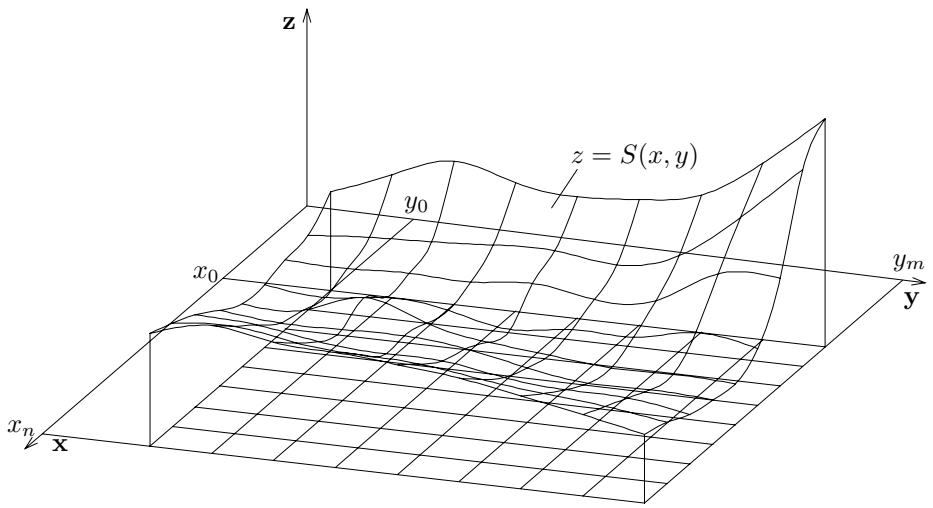


Abb. 12.2. Graph einer bikubischen Splinefunktion

Das bikubische Polynom in Gleichung (12.1) lautet ausführlich:

$$S_{ij}(x,y) = \sum_{k=0}^{3}\sum_{s=0}^{3} a_{ijks}(x-x_i)^k(y-y_j)^s =$$

$$\begin{array}{llll}
a_{ij00} & + a_{ij10}(x-x_i) & + a_{ij01}(y-y_j) & + \\
a_{ij20}(x-x_i)^2 & + a_{ij11}(x-x_i)(y-y_j) & + a_{ij02}(y-y_j)^2 & + \\
a_{ij30}(x-x_i)^3 & + a_{ij21}(x-x_i)^2(y-y_j) & + a_{ij12}(x-x_i)(y-y_j)^2 & + \\
a_{ij03}(y-y_j)^3 & + a_{ij31}(x-x_i)^3(y-y_j) & + a_{ij22}(x-x_i)^2(y-y_j)^2 & + \\
a_{ij13}(x-x_i)(y-y_j)^3 & + a_{ij32}(x-x_i)^3(y-y_j)^2 & + a_{ij23}(x-x_i)^2(y-y_j)^3 & + \\
a_{ij33}(x-x_i)^3(y-y_j)^3\,. & & &
\end{array}$$

Die 16 $m \cdot n$ Koeffizienten a_{ijks} von (12.1) müssen nun so bestimmt werden, dass S die Bedingungen (1) und (2) erfüllt. Die Interpolationsbedingungen (1) liefern $a_{ij00} = u_{ij}$ für $i = 0(1)n$–1, $j = 0(1)m$–1. Zur eindeutigen Bestimmung der a_{ijks} müssen dann noch (wie bei den eindimensionalen Splines) gewisse Randbedingungen auf R vorgegeben werden; eine Möglichkeit ist die Vorgabe der folgenden partiellen Ableitungen von S

$$\left\{\begin{array}{lllll}
\dfrac{\partial}{\partial x}S(x_i,y_j) & =: & p_{ij} = a_{ij10}, & i = 0,n, & j = 0(1)m, \\
\dfrac{\partial}{\partial y}S(x_i,y_j) & =: & q_{ij} = a_{ij01}, & i = 0(1)n, & j = 0,m, \\
\dfrac{\partial^2}{\partial x\partial y}S(x_i,y_j) & =: & s_{ij} = a_{ij11}, & i = 0,n, & j = 0,m.
\end{array}\right. \tag{12.2}$$

Falls die Ableitungen der Funktion u verfügbar sind, werden

$$\begin{aligned}
\frac{\partial}{\partial x}S(x_i,y_j) &= \frac{\partial}{\partial x}u(x_i,y_j)\,, \\
\frac{\partial}{\partial y}S(x_i,y_j) &= \frac{\partial}{\partial y}u(x_i,y_j)\,, \\
\frac{\partial^2}{\partial x\partial y}S(x_i,y_j) &= \frac{\partial^2}{\partial x\partial y}u(x_i,y_j)\,.
\end{aligned}$$

gesetzt. Andernfalls können die benötigten Ableitungen auch mit Hilfe eindimensionaler kubischer Splines oder anderer Interpolationsmethoden näherungsweise berechnet werden. Je nach Vorgabeart der Randbedingungen wird einer der folgenden Algorithmen eingesetzt. In [BOOR1962] wird nachgewiesen, dass zu gegebenen u_{ij} und gegebenen Randableitungen (12.2) genau eine bikubische Splinefunktion (12.1) existiert, welche die gegebenen u_{ij} interpoliert.

Berechnung der bikubischen Splinefunktion S

Im Folgenden werden drei Algorithmen zur Berechnung von S angegeben.

Algorithmus 12.1. (*Bikubische Splinefunktion*)

Gegeben: (i) $u_{ij} = u(x_i, y_j)$ für $i = 0(1)n$, $j = 0(1)m$.
(ii) die Randwerte der partiellen Ableitungen (12.2).

Gesucht: Die bikubische Splinefunktion (12.1).

1. Schritt: Berechnung der $a_{ij10} = p_{ij}$ für $i = 1(1)n{-}1$, $j = 0(1)m$ nach

$$\left\{\begin{array}{l} a_{i-1,j10}\frac{1}{h_{i-1}} + 2a_{ij10}\left(\frac{1}{h_{i-1}} + \frac{1}{h_i}\right) + a_{i+1,j10}\frac{1}{h_i} \\ = \frac{3}{h_{i-1}^2}(a_{ij00} - a_{i-1,j00}) + \frac{3}{h_i^2}(a_{i+1,j00} - a_{ij00}), \\ \text{für} \quad i = 1(1)n{-}1, \quad j = 0(1)m, \\ \\ \text{mit} \quad h_i = x_{i+1} - x_i \quad \text{für} \quad i = 0(1)n{-}1. \end{array}\right. \tag{12.3}$$

Dies sind $(m{+}1)$ lineare Gleichungssysteme mit je $(n{-}1)$ Gleichungen für $(n{+}1)$ Unbekannte. Durch Vorgabe der $2(m{+}1)$ Randwerte a_{ij10}, $i = 0, n$, $j = 0(1)m$, sind diese Systeme eindeutig lösbar.

2. Schritt: Bestimmung der $a_{ij01} = q_{ij}$ für $i = 0(1)n$, $j = 1(1)m{-}1$ mit

$$\left\{\begin{array}{l} a_{i,j-1,01}\frac{1}{k_{j-1}} + 2a_{ij01}\left(\frac{1}{k_{j-1}} + \frac{1}{k_j}\right) + a_{i,j+1,01}\frac{1}{k_j} \\ = \frac{3}{k_{j-1}^2}(a_{ij00} - a_{i,j-1,00}) + \frac{3}{k_j^2}(a_{i,j+1,00} - a_{ij00}), \\ \text{für} \quad i = 0(1)n, \quad j = 1(1)m{-}1, \\ \\ \text{mit} \quad k_j = y_{j+1} - y_j, \text{ für } j = 0(1)m{-}1. \end{array}\right. \tag{12.4}$$

Mit den vorgegebenen $2(n+1)$ Randwerten a_{ij01} , $i = 0(1)n$, $j = 0, m$ sind die Systeme eindeutig lösbar.

3. Schritt: Berechnung der $a_{ij11} = s_{ij}$ für $i = 1(1)n{-}1$, $j = 0, m$ aus den Gleichungssystemen

$$\left\{\begin{array}{l} \frac{1}{h_{i-1}}a_{i-1,j11} + 2a_{ij11}\left(\frac{1}{h_{i-1}} + \frac{1}{h_i}\right) + \frac{1}{h_i}a_{i+1,j11} \\ = \frac{3}{h_{i-1}^2}(a_{ij01} - a_{i-1,j01}) + \frac{3}{h_i^2}(a_{i+1,j01} - a_{ij01}), \\ \text{mit} \quad h_i = x_{i+1} - x_i \text{ für } i = 0(1)n{-}1. \end{array}\right. \tag{12.5}$$

Die vier Eckwerte a_{0011}, a_{n011}, a_{0m11} und a_{nm11} sind vorgegeben.

4. Schritt: Berechnung der Ableitungen $s_{ij} = a_{ij11}$, $i = 0, n$, $j = 1(1)m-1$ mit

$$\left\{\begin{array}{l} \frac{1}{k_{j-1}} a_{i,j-1,11} + 2a_{ij11}\left(\frac{1}{k_{j-1}} + \frac{1}{k_j}\right) + \frac{1}{k_j} a_{i,j+1,11} \\ = \frac{3}{k_{j-1}^2}(a_{ij10} - a_{i,j-1,10}) + \frac{3}{k_j^2}(a_{i,j+1,10} - a_{ij10}), \\ \text{mit} \quad k_j = y_{j+1} - y_j \quad \text{für} \quad j = 0(1)m-1. \end{array}\right. \tag{12.6}$$

Die erforderlichen Randwerte a_{ij11} für $i = 1(1)n-1$, $j = 0, m$ wurden mit dem 3. Schritt bestimmt, die a_{ij11} , $i = 0$, n , $j = 0, m$ sind vorgegeben.

5. Schritt: Bestimmung der Matrizen $\{\boldsymbol{G}(x_i)\}^{-1}$. Wegen

$$\left\{\begin{array}{l} \boldsymbol{G}(x_i) := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & h_i & h_i^2 & h_i^3 \\ 0 & 1 & 2h_i & 3h_i^2 \end{pmatrix} \\ \text{mit} \quad \det \boldsymbol{G}(x_i) = h_i^4 \neq 0, \quad h_i = x_{i+1} - x_i, \\ i = 0(1)n-1, \quad \text{existiert} \quad \{\boldsymbol{G}(x_i)\}^{-1}. \quad \text{Es gilt} \\ \{\boldsymbol{G}(x_i)\}^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -\frac{3}{h_i^2} & -\frac{2}{h_i} & \frac{3}{h_i^2} & -\frac{1}{h_i} \\ \frac{2}{h_i^3} & \frac{1}{h_i^2} & -\frac{2}{h_i^3} & \frac{1}{h_i^2} \end{pmatrix} \end{array}\right. \tag{12.7}$$

6. Schritt: Bestimmung der Matrizen $\{\boldsymbol{G}(y_j)^{\mathsf{T}}\}^{-1}$. Wegen

$$\left\{\begin{array}{l} \boldsymbol{G}(y_j) := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & k_j & k_j^2 & k_j^3 \\ 0 & 1 & 2k_j & 3k_j^2 \end{pmatrix} \\ \text{mit} \quad \det \boldsymbol{G}(y_j) = k_j^4 \neq 0, \quad k_j = y_{j+1} - y_j, \\ j = 0(1)m-1, \quad \text{existiert} \quad \{\boldsymbol{G}(y_j)^{\mathsf{T}}\}^{-1}. \quad \text{Es gilt} \\ \{[\boldsymbol{G}(y_j)]^{\mathsf{T}}\}^{-1} = \begin{pmatrix} 1 & 0 & -\frac{3}{k_j^2} & \frac{2}{k_j^3} \\ 0 & 1 & -\frac{2}{k_j} & \frac{1}{k_j^2} \\ 0 & 0 & \frac{3}{k_j^2} & -\frac{2}{k_j^3} \\ 0 & 0 & -\frac{1}{k_j} & \frac{1}{k_j^2} \end{pmatrix} \end{array}\right. \tag{12.8}$$

7. Schritt: Bestimmung der Matrizen $\boldsymbol{M}_{ij}$ nach

$$\left\{\begin{array}{l} \boldsymbol{M}_{ij} = \begin{pmatrix} a_{ij00} & a_{ij01} & a_{i,j+1,00} & a_{i,j+1,01} \\ a_{ij10} & a_{ij11} & a_{i,j+1,10} & a_{i,j+1,11} \\ a_{i+1,j00} & a_{i+1,j01} & a_{i+1,j+1,00} & a_{i+1,j+1,01} \\ a_{i+1,j10} & a_{i+1,j11} & a_{i+1,j+1,10} & a_{i+1,j+1,11} \end{pmatrix} \\ i = 0(1)n{-}1, \quad j = 0(1)m{-}1 \end{array}\right. \tag{12.9}$$

8. Schritt: Berechnung der Koeffizientenmatrizen $\boldsymbol{A}_{ij}$ für S_{ij} gemäß Gleichung

$$\begin{aligned} &\boldsymbol{A}_{ij} = \{\boldsymbol{G}(x_i)\}^{-1}\boldsymbol{M}_{ij}\{[\boldsymbol{G}(y_j)]^{\mathsf{T}}\}^{-1} = \{a_{ijks}\} \\ &k = 0(1)3, \quad s = 0(1)3, \quad i = 0(1)n{-}1, \quad j = 0(1)m{-}1 \end{aligned} \tag{12.10}$$

9. Schritt: Aufstellung der bikubischen Splinefunktion $S(x, y) \equiv S_{ij}(x, y)$ für jedes Rechteck R_{ij} gemäß (12.1).

Beispiel 12.2.

Von der Funktion $u : u(x, y) = x^2 \cos(y)$ sind in der folgenden Tabelle zu 20 Gitterpunkten (x_i, y_j), $i = 0(1)4$, $j = 0(1)3$, mit

$$\begin{array}{llllll} x_0 = 0, & & x_1 = 1, & x_2 = 2, & x_3 = 3.4, & x_4 = 5, \\ y_0 = -1.5, & y_1 = 0, & y_2 = 1, & y_3 = 1.5 & & \end{array}$$

die Funktionswerte $u_{ij} = x_i^2 \cos(y_j)$ angegeben.

i \ j	0	1	2	3
0	$0.000 \cdot 10^{+00}$	$0.000 \cdot 10^{+00}$	$0.000 \cdot 10^{+00}$	$0.000 \cdot 10^{+00}$
1	$7.074 \cdot 10^{-02}$	$1.000 \cdot 10^{+00}$	$5.403 \cdot 10^{-01}$	$7.074 \cdot 10^{-02}$
2	$2.829 \cdot 10^{-01}$	$4.000 \cdot 10^{+00}$	$2.161 \cdot 10^{+00}$	$2.829 \cdot 10^{-01}$
3	$8.177 \cdot 10^{-01}$	$1.156 \cdot 10^{+01}$	$6.246 \cdot 10^{+00}$	$8.177 \cdot 10^{-01}$
4	$1.768 \cdot 10^{+00}$	$2.500 \cdot 10^{+01}$	$1.351 \cdot 10^{+01}$	$1.768 \cdot 10^{+00}$

Ferner sind von den Ableitungen

$$\begin{array}{rcrcl} p_{ij} & = & \frac{\partial}{\partial x} u(x_i, y_j) & = & 2\, x_i \cos(y_j), \\ q_{ij} & = & \frac{\partial}{\partial y} u(x_i, y_j) & = & -x_i^2 \sin(y_j), \\ s_{ij} & = & \frac{\partial^2}{\partial x\, \partial y} u(x_i, y_j) & = & -2\, x_i \sin(y_j) \end{array}$$

die in (12.2) genannten Werte gegeben:

$$
\begin{aligned}
p_{00} &= 0.000 \cdot 10^{+00}, & p_{40} &= 7.074 \cdot 10^{-01} \\
p_{01} &= 0.000 \cdot 10^{+00}, & p_{41} &= 1.000 \cdot 10^{+01} \\
p_{02} &= 0.000 \cdot 10^{+00}, & p_{42} &= 5.403 \cdot 10^{+00} \\
p_{03} &= 0.000 \cdot 10^{+00}, & p_{43} &= 7.074 \cdot 10^{-01} \\
q_{00} &= 0.000 \cdot 10^{+00}, & q_{03} &= 0.000 \cdot 10^{+00} \\
q_{10} &= 9.975 \cdot 10^{-01}, & q_{13} &= -9.975 \cdot 10^{-01} \\
q_{20} &= 3.990 \cdot 10^{+00}, & q_{23} &= -3.990 \cdot 10^{+00} \\
q_{30} &= 1.153 \cdot 10^{+01}, & q_{33} &= -1.153 \cdot 10^{+01} \\
q_{40} &= 2.494 \cdot 10^{+01}, & q_{43} &= -2.494 \cdot 10^{+01} \\
s_{00} &= 0.000 \cdot 10^{+00}, & s_{03} &= 0.000 \cdot 10^{+00} \\
s_{40} &= 9.975 \cdot 10^{+00}, & s_{43} &= -9.975 \cdot 10^{+00}
\end{aligned}
$$

Zu diesen Daten wird mit Hilfe des Algorithmus 12.1 die interpolierende bikubische Splinefunktion S auf dem Rechteck $R = \{\, (x, y) \,|\, 0 \le x \le 5,\ -1.5 \le y \le 1.5 \,\}$ erzeugt.

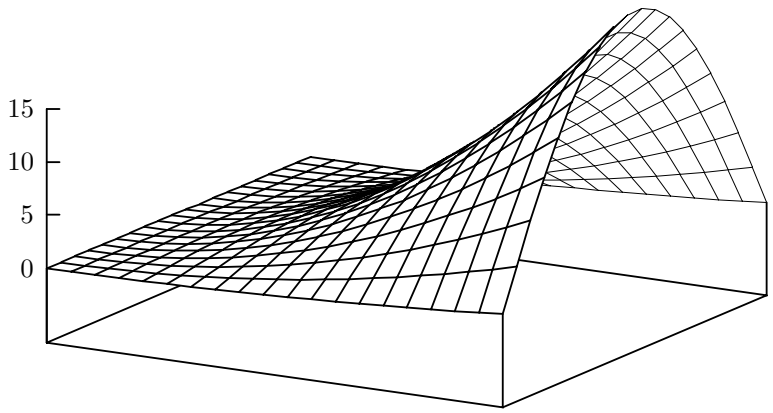


Abb. 12.3. Graph der Funktion $u(x, y) = x^2 \cos(y)$ über dem Rechteck R

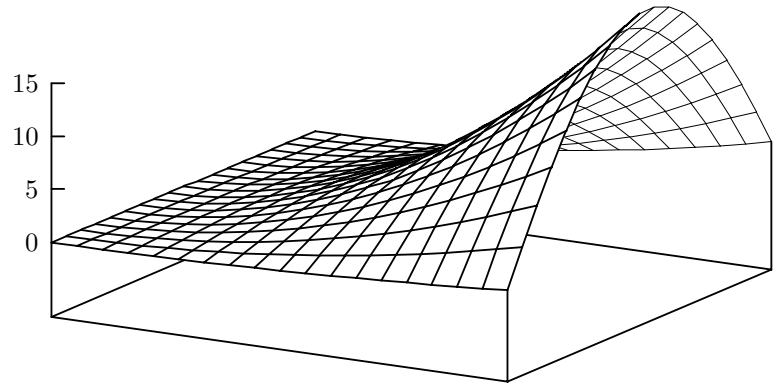


Abb. 12.4. Graph der bikubischen Splinefunktion S

□

Beim nächsten Algorithmus werden nur die Funktionswerte in den Gitterpunkten, nicht aber die Randwerte für die partiellen Ableitungen vorgegeben; diese werden mit Hilfe eindimensionaler Splines durch jeweils drei Punkte ermittelt.

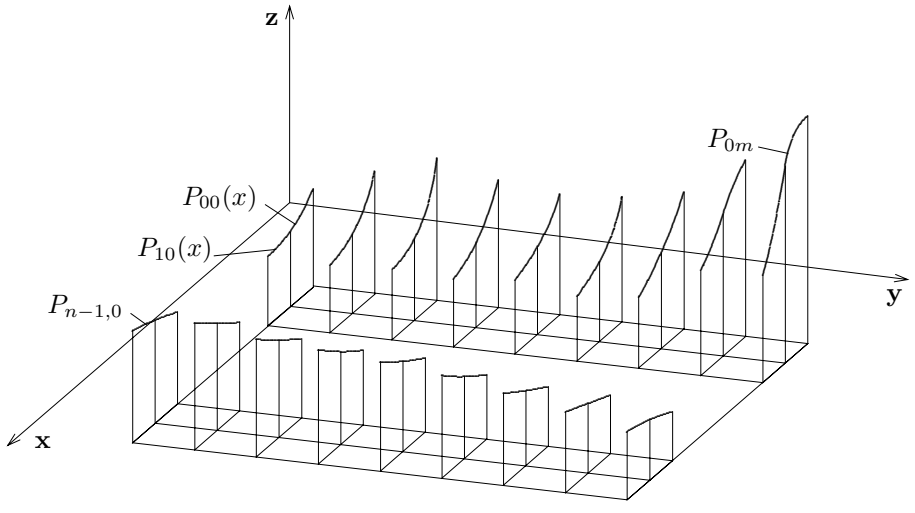


Abb. 12.5a.

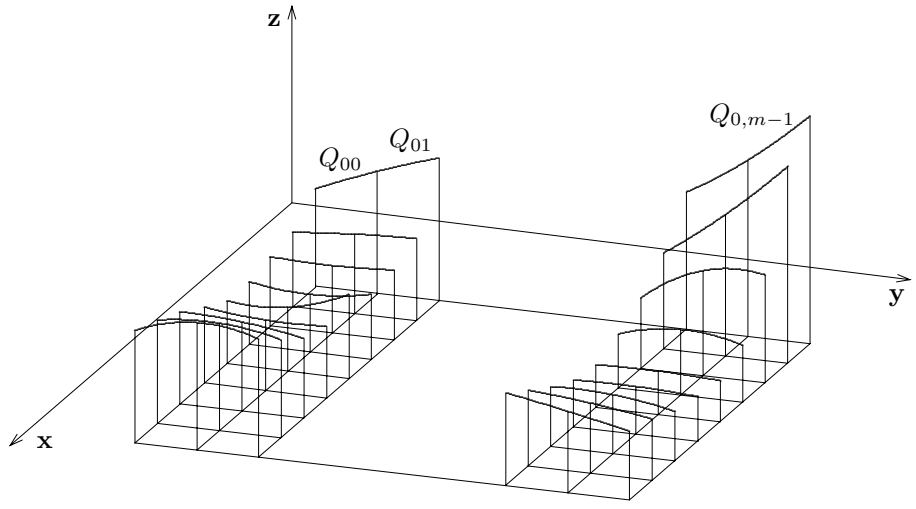


Abb. 12.5b.

Zur Berechnung der $s_{ij} = a_{ij11}$ für $i = 0, n$, $j = 0, m$ werden eindimensionale natürliche Splines durch die Punkte (x_i, q_{ij}) für $i = 0, 1, 2$ und $i = n{-}2, n{-}1, n$ und $j = 0, m$ gelegt und abgeleitet (vgl. Abbildung 12.5c).

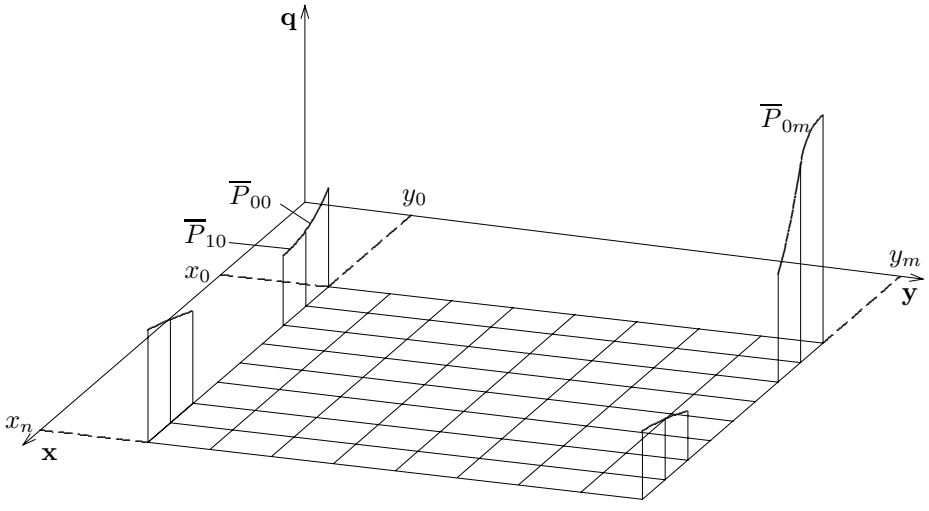


Abb. 12.5c.

Algorithmus 12.3. (*Bikubische Splinefunktion ohne Vorgabe von Randwerten*)

Gegeben: Funktionswerte $u_{ij} = u(x_i, y_j)$ für $i = 0(1)n$, $j = 0(1)m$ in den Gitterpunkten (x_i, y_j).

Gesucht: Die zugehörige bikubische Splinefunktion S (12.1).

Die zur Berechnung von S erforderlichen Randwerte für die partiellen Ableitungen $p_{ij} = a_{ij10}$, $q_{ij} = a_{ij01}$, $s_{ij} = a_{ij11}$ gemäß (12.2) werden hier mit Hilfe eindimensionaler (natürlicher) kubischer Splinefunktionen durch jeweils drei Punkte und mit deren Ableitungen ermittelt. Durch die Punkte (x_i, u_{ij}) werden für $i = 0, 1, 2$ und $i = n–2, n–1, n$ Splines für $j = 0(1)m$ gelegt und abgeleitet; sie liefern die p_{ij} am Rande. Durch die Punkte (y_j, u_{ij}) werden für $j = 0, 1, 2$ und $j = m–2, m–1, m$ Splines für $i = 0(1)n$ gelegt und abgeleitet; sie liefern die q_{ij} am Rande. Um diese Vorgehensweise im Algorithmus wiedererkennen zu können, wurden die folgenden Formeln (12.12) nicht in (12.11) eingearbeitet, (12.14) nicht in (12.13) und (12.16) nicht in (12.15), was zu Vereinfachungen geführt hätte.

Zur Berechnung der $s_{ij} = a_{ij11}$ für $i = 0, n$, $j = 0, m$ werden eindimensionale natürliche Splines durch die Punkte (x_i, q_{ij}) für $i = 0, 1, 2$ und $i = n–2, n–1, n$ und $j = 0, m$ gelegt und abgeleitet. Die a_{ij00} sind durch die u_{ij} vorgegeben. Dann wird die bikubische Splinefunktion S gemäß (12.1) wie folgt berechnet:

1. Schritt: Berechnung der Randwerte a_{ij10} , $i = 0, n$, $j = 0(1)m$ mit

$$\left\{\begin{array}{lcl} a_{0j10} & = & S_x(x_0, y_j) = b_{oj} = \\ & = & \frac{1}{h_0}(a_{1j00} - a_{0j00}) - \frac{h_0}{3}c_{1j} \\ & & \text{mit} \quad j = 0(1)m \quad \text{und} \\ a_{nj10} & = & S_x(x_n, y_j) = b_{n-1,j} + 2c_{n-1,j}h_{n-1} \\ & + & 3d_{n-1,j}h_{n-1}^2 \\ & & \text{mit} \quad j = 0(1)m, \quad h_i = x_{i+1} - x_i \end{array}\right. \tag{12.11}$$

mit den aus (12.12) zu ermittelnden Werten für die Koeffizienten b_{ij}, c_{ij} und d_{ij}

$$
\left\{\begin{array}{lcl}
1.\ u_{ij} & = & a_{ij00}, \quad i = 0,1,2,(n-2),(n-1),n, \\
 & & j = 0(1)m, \\
2.\ c_{0j} & = & c_{2j} = c_{n-2,j} = c_{nj} = 0, \quad j = 0(1)m, \\
 & & \text{(natürliche Splines)} \\
3.\ c_{ij} & = & \frac{3}{2(h_i + h_{i-1})} \left[\frac{1}{h_i}(a_{i+1,j00} - a_{ij00})\right. \\
 & - & \left.\frac{1}{h_{i-1}}(a_{ij00} - a_{i-1,j00})\right], \\
 & & i = 1,(n-1), j = 0(1)m, \\
4.\ b_{n-1,j} & = & \frac{1}{k_{n-1}}(a_{nj00} - a_{n-1,j00}) - \frac{2k_{n-1}}{3}c_{n-1,j}, \\
 & & j = 0(1)m, \\
5.\ d_{n-1,j} & = & \frac{1}{3k_{n-1}}c_{n-1,j}, \quad j = 0(1)m.
\end{array}\right. \tag{12.12}
$$

2. Schritt: Berechnung der Randwerte a_{ij01} , $i = 0(1)n$, $j = 0, m$ mit

$$
\left\{\begin{array}{lcl}
a_{i001} & = & S_y(x_i, y_0) = \beta_{i0} = \frac{1}{k_0}(a_{i100} - a_{i000}) \\
 & - & \frac{k_0}{3}\gamma_{i1}, \quad i = 0(1)n, \quad \text{und} \\
a_{im01} & = & S_y(x_i, y_m) = \beta_{i,m-1} + 2\gamma_{i,m-1}k_{m-1} \\
 & + & 3\delta_{i,m-1}k_{m-1}^2,\ i = 0(1)n, \\
 & & \text{mit} \quad k_j = y_{j+1} - y_j.
\end{array}\right. \tag{12.13}
$$

mit den gemäß (12.14) zu ermittelnden Koeffizienten

$$
\left\{
\begin{array}{lcl}
1.\ u_{ij} & = & \alpha_{ij} = a_{ij00}, \quad i = 0(1)n, \\
 & & j = 0, 1, 2, (m{-}2), (m{-}1), m \\
2.\ \gamma_{i0} & = & \gamma_{i2} = \gamma_{i,m-2} = \gamma_{im} = 0, \quad i = 0(1)n, \\
3.\ \gamma_{ij} & = & \frac{3}{2(k_{j-1}+k_j)} \left[\frac{1}{k_j}(a_{i,j+1,00} - a_{ij00}) \right. \\
 & - & \left. \frac{1}{k_{j-1}}(a_{ij00} - a_{i,j-1,00}) \right], \\
 & & i = 0(1)n, j = 1, (m{-}1). \\
4.\ \beta_{i,m-1} & = & \frac{1}{h_{m-1}}(a_{im00} - a_{i,m-1,00}) - \frac{2h_{m-1}}{3}\gamma_{i,m-1}, \\
 & & i = 0(1)n, \\
5.\ \delta_{i,m-1} & = & -\frac{1}{3k_{m-1}}\gamma_{i,m-1}, \quad i = 0(1)n.
\end{array}
\right. \tag{12.14}
$$

3. Schritt: Berechnung der Randwerte a_{ij11} , $i = 0, n$, $j = 0, m$ mit

$$
\left\{
\begin{array}{lcl}
a_{0j11} & = & S_{yx}(x_0, y_j) = s_{0j} = \widetilde{b}_{0j} = \\
 & = & \frac{1}{k_0}(a_{1j01} - a_{0j01}) - \frac{k_0}{3}\widetilde{c}_{1j}, \\
 & & j = 0, m, \quad \text{und} \\
a_{nj11} & = & S_{yx}(x_n, y_j) = s_{nj} = \\
 & = & \widetilde{b}_{n-1,j} + 2\widetilde{c}_{n-1,j}h_{n-1} + 3\widetilde{d}_{n-1,j}h_{n-1}^2, \\
 & & j = 0, m \quad \text{mit} \quad h_i = x_{i+1} - x_i.
\end{array}
\right. \tag{12.15}
$$

mit den gemäß (12.16) zu ermittelnden Koeffizienten

$$
\begin{cases}
1.\ q_{ij} & = & a_{ij01}, \quad i = 0,1,2,(n-2),(n-1),n, \\
 & & j = 0,m \\
2.\ \widetilde{c}_{0j} & = & \widetilde{c}_{2j} = \widetilde{c}_{n-2,j} = \widetilde{c}_{n,j} = 0, \quad j = 0,m \\
3.\ \widetilde{c}_{ij} & = & \frac{3}{2}\frac{1}{(h_{i-1}+h_i)}\left[\frac{1}{h_i}(a_{i+1,j01} - a_{ij01})\right. \\
 & - & \left.\frac{1}{h_{i-1}}(a_{ij01} - a_{i-1,j01})\right], \\
 & & i = 1,n-1, j = 0,m \\
4.\ \widetilde{b}_{n-1,j} & = & \frac{1}{h_{n-1}}(a_{nj01} - a_{n-1,j01}) - \frac{2h_{n-1}}{3}\widetilde{c}_{n-1,j}, \\
 & & j = 0,m \\
5.\ \widetilde{d}_{n-1,j} & = & -\frac{1}{3h_{n-1}}\widetilde{c}_{n-1,j}, \quad j = 0,m
\end{cases}
\tag{12.16}
$$

4. Schritt: Berechnung der partiellen Ableitungen a_{ij10} für $i = 1(1)n-1$, $j = 0(1)m$ mit (12.3).

5. Schritt: Lösung der Gleichungssysteme (12.4) zur Bestimmung der a_{ij01}, $i = 0(1)n$, $j = 1(1)m-1$.

6. Schritt: Bestimmung der Werte a_{ij11} , $i = 1(1)n-1$, $j = 0,m$ mit (12.5).

7. Schritt: Berechnung der partiellen Ableitungen a_{ij11}, $i = 0,n$, $j = 1(1)m-1$ mit (12.6).

8. Schritt: Bestimmung der Matrizen $\{\boldsymbol{G}(x_i)\}^{-1}$ mit (12.7).

9. Schritt: Bestimmung der Matrizen $\{[\boldsymbol{G}(y_j)]^{\mathsf{T}}\}^{-1}$ mit (12.8).

10. Schritt: Bestimmung der Matrizen $\boldsymbol{M}_{ij}$ gemäß (12.9).

11. Schritt: Berechnung der Koeffizientenmatrizen $\boldsymbol{A}_{ij}$ nach (12.10), $i = 0(1)n-1$, $j = 0(1)m-1$.

12. Schritt: Aufstellung der bikubischen Splinefunktion $S(x,y) \equiv S_{ij}(x,y), (x,y) \in R_{ij}$, gemäß (12.1).

In Algorithmus 12.3 werden eindimensionale Splines durch jeweils drei Punkte benutzt, man kann aber genauso jeweils eindimensionale Splines durch alle gegebenen Punkte (x_i,u_{ij}), $i = 0(1)n$, j fest bzw. (y_j,u_{ij}), $j = 0(1)m$, i fest, legen und ableiten.

Zu jedem Punkt (x_i, y_j, u_{ij}) des Graphen der Funktion $u(x, y)$ sei jetzt ein zur x, y-Ebene nicht paralleler Vektor

$$\boldsymbol{n}_{ij}^{\mathsf{T}} = (n_{ij1}, n_{ij2}, n_{ij3}), \quad n_{ij3} \neq 0,$$

gegeben, der die Flächennormale bestimmt. Damit können die Ableitungen

$$p_{ij} = u_x(x_i, y_j) \quad \text{und} \quad q_{ij} = u_y(x_i, y_j)$$

wie folgt ermittelt werden.

Der Graph der Funktion $u(x, y)$ besitzt den Ortsvektor

$$\boldsymbol{x}(x, y) = \begin{pmatrix} x \\ y \\ u(x, y) \end{pmatrix}.$$

Mit den Tangentenvektoren

$$\boldsymbol{x}_x(x, y) = \begin{pmatrix} 1 \\ 0 \\ u_x(x, y) \end{pmatrix}, \quad \boldsymbol{x}_y(x, y) = \begin{pmatrix} 0 \\ 1 \\ u_y(x, y) \end{pmatrix}$$

ergibt sich der Vektor der Flächennormale

$$\boldsymbol{n} = \boldsymbol{x}_x \times \boldsymbol{x}_y = \begin{pmatrix} -u_x \\ -u_y \\ 1 \end{pmatrix}.$$

Mit $n_3 \neq 0$ ist ein beliebiger Normalenvektor

$$n_3\, \boldsymbol{n} = \begin{pmatrix} -n_3\, u_x \\ -n3\, u_y \\ n_3 \end{pmatrix} = \begin{pmatrix} n_1 \\ n_2 \\ n_3 \end{pmatrix}.$$

Wenn ein solcher Vektor mit $n_3 \neq 0$ gegeben ist, sind demnach

$$u_x = -\frac{n_1}{n_3}, \quad u_y = -\frac{n_2}{n_3}.$$

Mit den gegebenen Normalenvektoren $\boldsymbol{n}_{ij}$ sind also für $i = 0(1)n$, $j = 0(1)m$

$$p_{ij} = a_{ij10} = -\frac{n_{ij1}}{n_{ij3}}, \quad q_{ij} = a_{ij01} = -\frac{n_{ij2}}{n_{ij3}}.$$

Dann sind nur noch die Ableitungen $s_{ij} = a_{ij11}$ über eindimensionale Splines zu berechnen wie im 3. Schritt des Algorithmus 12.3.

Algorithmus 12.4. (*Bikubische Splinefunktion mit Vorgabe der Normalen*)

Gegeben: (i) Funktionswerte $u_{ij} = u(x_i, y_j)$ für $i = 0(1)n$, $j = 0(1)m$ in den Gitterpunkten (x_i, y_j);
(ii) zu jedem Gitterpunkt (x_i, y_j) ein Normalenvektor $\boldsymbol{n}_{ij}^{\mathsf{T}} = (n_{ij1}, n_{ij2}, n_{ij3}), n_{ij3} \neq 0, i = 0(1)n, j = 0(1)m$. Alle n_{ij3} müssen entweder positiv oder negativ sein.

Gesucht: Die zugehörige bikubische Splinefunktion S (12.1), die in den Gitterpunkten die Ordinaten (i) und die Normalen (ii) besitzt.

1. Schritt: Berechnung der partiellen Ableitungen $p_{ij} = a_{ij10} = -n_{ij1}/n_{ij3}$ für $i = 0(1)n$, $j = 0(1)m$.
2. Schritt: Berechnung der partiellen Ableitungen $q_{ij} = a_{ij01} = -n_{ij2}/n_{ij3}$ für $i = 0(1)n$, $j = 0(1)m$.
3. Schritt: Berechnung der vier Randwerte für die gemischten partiellen Ableitungen $s_{ij} = a_{ij11}$ für $i = 0, n, j = 0, m$ gemäß (12.15) und (12.16).
4. Schritt: Bestimmung der a_{ij11} für $i = 1(1)n-1$, $j = 0, m$ gemäß (12.5).
5. Schritt: Bestimmung der a_{ij11} für $i = 0, n$, $j = 1(1)m-1$ gemäß (12.6).
6. Schritt: Bestimmung der Matrizen $\{\boldsymbol{G}(x_i)\}^{-1}$ gemäß (12.7).
7. Schritt: Bestimmung der Matrizen $\{[\boldsymbol{G}(y_j)]^{\mathsf{T}}\}^{-1}$ gemäß (12.8).
8. Schritt: Bestimmung der Matrizen $\boldsymbol{M}_{ij}$ gemäß (12.9).
9. Schritt: Bestimmung der Koeffizientenmatrizen $\boldsymbol{A}_{ij}$ für $i = 0(1)n-1$, $j = 0(1)m-1$ gemäß (12.10).
10. Schritt: Aufstellung der bikubischen Splinefunktion $S(x, y) \equiv S_{ij}(x, y)$ für jedes Rechteck R_{ij} gemäß (12.1).

12.2 Zweidimensionale interpolierende Oberflächensplines

Während im vorigen Abschnitt bei der bikubischen Splinefunktion die Stützstellen in einem Rechteckgitter liegen müssen, unterliegen sie jetzt keinen einschränkenden Bedingungen und können also in der x, y-Ebene beliebig vorgegeben werden.

Zu NX vorgegebenen Interpolationsstellen (x_i, y_i, z_i), $i = 1(1)NX$, mit verschiedenen Stützstellen (x_i, y_i) wird eine interpolierende Funktion Φ erzeugt, deren Graph eine glatte Fläche durch die Interpolationsstellen ist.

Die Oberflächensplines lassen sich physikalisch-technisch so interpretieren, dass sie die Verbiegung einer dünnen (ebenen) Platte unendlicher Ausdehnung beschreiben, die an mehreren, voneinander unabhängigen Punkten senkrecht zur Ruhelage abgelenkt wird unter der Forderung, die Biegungsenergie zu minimieren (analog zu den natürlichen Splines im eindimensionalen Fall).

Hier wird nur rezeptartig beschrieben, wie diese Oberflächensplines konstruiert werden. Um den Algorithmus verstehen zu können, muss zusätzlich die Arbeit von J. Meinguet [MEING1979] angesehen werden. Die auftretenden linearen Gleichungssysteme sind im Allgemeinen schlecht konditioniert.

Problemstellung

Gegeben seien in der x, y-Ebene NX paarweise verschiedene Stützstellen (x_i, y_i), $i = 1(1)NX$. Jeder Stützstelle sei genau ein Stützwert $z_i \in \mathbb{R}$ zugeordnet. Diese Stützwerte kann man auffassen als die Werte einer empirischen Funktion $f : z = f(x, y)$ an den gegebenen Stützstellen: $z_i = f(x_i, y_i)$. Gesucht ist eine Funktion $\Phi : z = \Phi(x, y)$, die den NX Interpolationsbedingungen $\Phi(x_i, y_i) = z_i$, $i = 1(1)NX$, genügt, so dass Φ die Funktion f annähert.

Damit der Graph der Funktion Φ eine genügend glatte Fläche über der x, y-Ebene ist, wird die Ableitungsordnung M, $M \geq 2$ vorgegeben.

Die Funktion Φ wird in der Gestalt

$$\Phi(x, y) = \sum_{j=1}^{n} c_j \varphi_j(x, y) \tag{12.17}$$

mit gegebenen Funktionen φ_j und zu bestimmenden Koeffizienten c_j angesetzt.

Für einige der Funktionen φ_j werden zweidimensionale Monome $p_j^{(M-1)}(x, y)$ bis zum Grad $M-1$ verwendet; deren Anzahl ist $MM := M\,(M+1)/2$.

Beispielsweise sind dies für $M = 2$ die 3 Monome $p_j^{(1)}$, $j = 1(1)3$,

$$p_1^{(1)} = 1, \quad p_2^{(1)} = x, \quad p_3^{(1)} = y$$

und für $M = 3$ die 6 Monome $p_j^{(2)}$, $j = 1(1)6$,

$$\begin{aligned} p_1^{(2)} &= 1, & p_2^{(2)} &= x, & p_3^{(2)} &= y \\ p_4^{(2)} &= x^2, & p_5^{(2)} &= xy, & p_6^{(2)} &= y^2. \end{aligned}$$

Die folgende Voraussetzung muss erfüllt sein:

Unter den gegebenen Stützstellen (x_i, y_i), $i = 1(1)NX$, gibt es MM Stützstellen (x_{i_j}, y_{i_j}) mit $i_j \in \{1\ 2 \ldots NX\}$ und $j = 1(1)MM$ derart, dass die Matrix

$$\boldsymbol{P}_j^{(M-1)} = \left(p_j^{(M-1)}(x_{i_j}, y_{i_j})\right)$$

nicht singulär ist $(NX \geq MM)$.

Zur Erläuterung dient das folgende

Beispiel 12.5.

Es sei $M = 2$, also $MM = 3$.

a) Unter den NX Stützstellen gebe es die drei Stützstellen

$$(x_{i_1}, y_{i_1}) = (0,0), \quad (x_{i_2}, y_{i_2}) = (0,1), \quad (x_{i_3}, y_{i_3}) = (1,0).$$

Dann ist die Determinante der Matrix $\boldsymbol{P}_j^{(1)}$ mit $j = 1,2,3$

$$\det\left(\boldsymbol{P}_j^{(1)}\right) = \left| p_j^{(1)}(x_{i_j}, y_{i_j}) \right| = \begin{vmatrix} 1 & x_{i_1} & y_{i_1} \\ 1 & x_{i_2} & y_{i_2} \\ 1 & x_{i_3} & y_{i_3} \end{vmatrix} = \begin{vmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{vmatrix} = -1 \neq 0,$$

d. h. die Matrix ist nicht singulär.

b) Aus den NX Stützstellen seien jetzt die drei Stützstellen

$$(x_{i_1}, y_{i_1}) = (0,0), \quad (x_{i_2}, y_{i_2}) = (0.5, 0.5), \quad (x_{i_3}, y_{i_3}) = (1,1)$$

ausgewählt. Damit ist

$$\det\left(\boldsymbol{P}_j^{(1)}\right) = \begin{vmatrix} 1 & 0 & 0 \\ 1 & 0.5 & 0.5 \\ 1 & 1 & 1 \end{vmatrix} = 0.$$

In diesem Fall erfüllen die ausgewählten Stützstellen nicht die an die Matrix $\boldsymbol{P}_j^{(1)}$ gestellte Bedingung bezüglich des Ranges. □

Im Ansatz (12.17) sei nun $n = NX + MM$. Die n Funktionen φ_j werden wie folgt gewählt.

1) $$\varphi_j(x,y) = \mathrm{E}\,(x - x_j, y - y_j), \quad j = 1(1)NX,$$

mit der sogenannten Kernfunktion

$$\mathrm{E}\,(x - x_j, y - y_j) = \left[\,(x - x_j)^2 + (y - y_j)^2\,\right]^{M-1} \ln\left[\,(x - x_j)^2 + (y - y_j)^2\,\right].$$

2) $$\varphi_{NX+j}(x,y) = p_j^{(M-1)}(x,y), \quad j = 1(1)MM$$

(Monome bis zum Grad $M-1$).

Die Funktion Φ soll die Interpolationsbedingungen

$$\Phi(x_i, y_i) = \sum_{j=1}^{NX+MM} c_j\, \varphi_j(x_i, y_i) = z_i, \quad i = 1(1)NX,$$

und zusätzlich gewisse Minimaleigenschaften (vgl. [MEING1979]) erfüllen. Diese Bedingungen führen auf ein lineares Gleichungssystem für die $NX + MM$ Koeffizienten c_j.

Dieses System lautet

$$\begin{pmatrix} \boldsymbol{G} & \boldsymbol{P} \\ \boldsymbol{P}^{\mathsf{T}} & \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{c}^{(1)} \\ \boldsymbol{c}^{(2)} \end{pmatrix} = \begin{pmatrix} \boldsymbol{z} \\ \boldsymbol{0} \end{pmatrix} \tag{12.18}$$

mit den Matrizen

$$\begin{aligned} \boldsymbol{G} &= \big(\mathrm{E}\,(x_i{-}x_j, y_i{-}y_j)\big), \quad i,j = 1(1)NX, \\ \boldsymbol{P} &= \big(p_j^{(M-1)}(x_i, y_i)\big), \quad i = 1(1)NX,\ j = 1(1)MM, \end{aligned}$$

und den Vektoren

$$\boldsymbol{c}^{(1)} = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_{NX} \end{pmatrix}, \quad \boldsymbol{c}^{(2)} = \begin{pmatrix} c_{NX+1} \\ c_{NX+2} \\ \vdots \\ c_{NX+MM} \end{pmatrix}, \quad \boldsymbol{z} = \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_{NX} \end{pmatrix}.$$

Alle Elemente der Hauptdiagonale $i = j$ der Matrix $\boldsymbol{G}$ verschwinden wegen $\mathrm{E}\,(x_i - x_i, y_i - y_i) = \mathrm{E}\,(0,0) = 0$ (dazu muss der unbestimmte Ausdruck $0 \cdot \infty$ behandelt werden).

$\boldsymbol{P}^{\mathsf{T}}$ ist die zu $\boldsymbol{P}$ transponierte Matrix. Mit wachsenden NX und M verschlechtert sich die Kondition des Systems (12.18) stark.

Es ist zu empfehlen, die Stützstellen (x_i, y_i), $i = 1(1)NX$, in einen Einheitskreis mit dem Mittelpunkt $(\overline{x}, \overline{y})$ zu transformieren:

$$(x_i, y_i) \mapsto \frac{1}{r}\,(x_i - \overline{x}, y_i - \overline{y})$$

$$\text{mit} \quad \overline{x} = \frac{1}{NX}\sum_{i=1}^{NX} x_i, \quad \overline{y} = \frac{1}{NX}\sum_{i=1}^{NX} y_i, \quad r = \max_{1 \le i \le NX} \sqrt{(x_i - \overline{x})^2 + (y_i - \overline{y})^2}.$$

12.3 Bézier-Splines

Hier werden nur eine knappe Darstellung der kubischen Bézier-Spline-Kurve und Bézier-Spline-Fläche sowie Algorithmen zu deren Berechnung angegeben. Analog zu den kubischen und bikubischen Splinefunktionen werden hier parametrische kubische bzw. bikubische Bézier-Polynome mit gleicher Krümmung an den Anschlussstellen aneinander gesetzt. Bei diesen parametrischen Kurven- und Flächen-Splines entfallen einschränkende Voraussetzungen wie die monotone Anordnung der Knoten bei den kubischen Splinefunktionen und die Verwendung eines Rechteckgitters bei der bikubischen Splinefunktion. Die Bézier-Splines sind nicht im engeren Sinne interpolierend wie die genannten Splinefunktionen.

Mit Bézier-Splines können auch Kurven und Flächen erzeugt werden, die einen „Knick“ haben, bei denen also an bestimmten Nahtstellen zwischen den kubischen Parabeln einer Bézier-Spline-Kurve bzw. den bikubischen „Pflastern“ einer Bézier-Spline-Fläche auf die stetige Differenzierbarkeit verzichtet und nur ein stetiger Übergang gefordert wird. Beispiele dafür treten bei der Konstruktion von Karosserieteilen auf, vergleiche folgende Abbildung aus [BEHR1975].

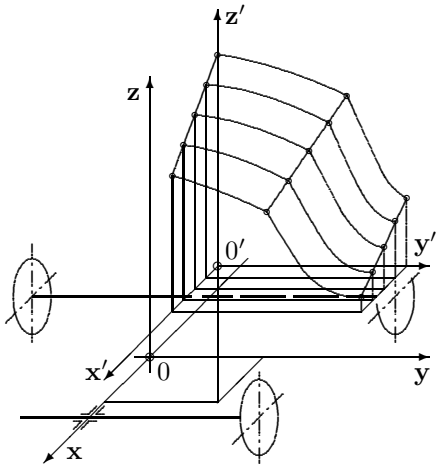


Abb. 12.6.

12.3.1 Bézier-Spline-Kurven

Eine Bézier-Spline-Kurve setzt sich stückweise aus Bézier-Kurven zusammen, die zweimal stetig differenzierbar aneinander schließen.
Eine Bézier-Kurve der Ordnung n, $n \geq 1$, lässt sich mit Hilfe der Bernstein-Polynome n-ten Grades

$$B_j^n(v) = \binom{n}{j}(1-v)^{n-j}v^j, \qquad j = 0(1)n, \tag{12.19}$$

und der $n{+}1$ Bézier-Punkte $\boldsymbol{b}_j, j = 0(1)n, \boldsymbol{b}_j \in \mathbb{R}^2$ oder $\boldsymbol{b}_j \in \mathbb{R}^3$, wie folgt darstellen:

$$\boldsymbol{P}(v) = \sum_{j=0}^{n} B_j^n(v)\boldsymbol{b}_j, \qquad v \in [0,1]. \tag{12.20}$$

Im Folgenden werden nur Bézier-Kurven der Ordnung $n{=}3$ betrachtet. Eine solche Kurve besitzt nach (12.19) und (12.20) die Darstellung

$$\boldsymbol{P}(v) = (1-v)^3\boldsymbol{b}_0 + 3(1-v)^2v\boldsymbol{b}_1 + 3(1-v)v^2\boldsymbol{b}_2 + v^3\boldsymbol{b}_3,\ v \in [0,1]. \tag{12.21}$$

$\boldsymbol{P}(0) = \boldsymbol{b}_0$ ist der Anfangspunkt, $\boldsymbol{P}(1) = \boldsymbol{b}_3$ ist der Endpunkt der Bézier-Kurve. Die Punkte $\boldsymbol{b}_1$ und $\boldsymbol{b}_2$ liegen nicht auf der Kurve.

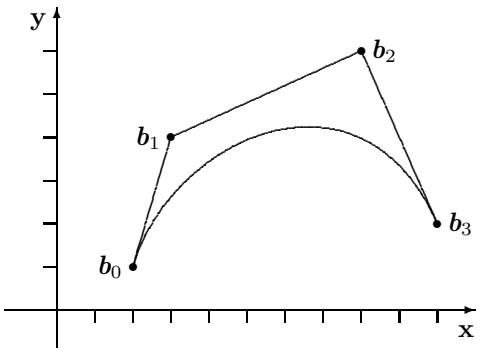


Abb. 12.7. Kubische Bézier-Kurve mit Bézier-Punkten

Die Ableitungen in den Randpunkten sind

$$\boldsymbol{P}'(0) = 3(\boldsymbol{b}_1 - \boldsymbol{b}_0), \quad \boldsymbol{P}'(1) = 3(\boldsymbol{b}_3 - \boldsymbol{b}_2). \tag{12.22}$$

Daher ist die Verbindungsgerade der Punkte $\boldsymbol{b}_0$ und $\boldsymbol{b}_1$ ($\boldsymbol{b}_0 \neq \boldsymbol{b}_1$) die Tangente der Kurve im Punkt $\boldsymbol{b}_0$, und im Endpunkt $\boldsymbol{b}_3$ berührt die Kurve die Verbindungsgerade von $\boldsymbol{b}_2$ und $\boldsymbol{b}_3$ ($\boldsymbol{b}_2 \neq \boldsymbol{b}_3$) (Abb. 12.7). Ferner gilt: Die Bézier-Kurve liegt ganz im kleinsten konvexen Bereich, der die Bézier-Punkte $\boldsymbol{b}_0, \boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3$ enthält. Mit diesen Eigenschaften lässt sich der ungefähre Verlauf einer Bézier-Kurve aufgrund der vorgegebenen Bézier-Punkte abschätzen.

Für die 2. Ableitung von (12.21) ergeben sich in den Randpunkten

$$\begin{cases} \boldsymbol{P}''(0) &= 6(\boldsymbol{b}_0 - 2\boldsymbol{b}_1 + \boldsymbol{b}_2), \\ \boldsymbol{P}''(1) &= 6(\boldsymbol{b}_1 - 2\boldsymbol{b}_2 + \boldsymbol{b}_3). \end{cases} \tag{12.23}$$

Eine kubische Bézier-Spline-Kurve setzt sich stückweise aus m Bézier-Kurven 3. Ordnung zusammen. Die Spline-Segmente besitzen die Darstellung

$$\begin{aligned} \boldsymbol{P}_k(v) \quad &= (1-v)^3 \boldsymbol{b}_{3k} + 3(1-v)^2 v \boldsymbol{b}_{3k+1} + 3(1-v)v^2 \boldsymbol{b}_{3k+2} \\ &\quad + v^3 \boldsymbol{b}_{3k+3},\ v \in [0,1],\ \ k = 0(1)m{-}1,\ \ m \geq 2. \end{aligned}$$

Die Bézier-Punkte sind so nummeriert, dass gilt

$$\boldsymbol{P}_{k-1}(1) = \boldsymbol{b}_{3(k-1)+3} = \boldsymbol{b}_{3k} = \boldsymbol{P}_k(0), \quad k = 1(0)m{-}1.$$

Da die Spline-Kurve durch alle Randpunkte $\boldsymbol{b}_0, \boldsymbol{b}_3, \ldots, \boldsymbol{b}_{3m}$ der m Kurvensegmente geht, werden diese im Folgenden Interpolationspunkte der kubischen Bézier-Spline-Kurve genannt.
In den Punkten $\boldsymbol{b}_3, \boldsymbol{b}_6, \ldots, \boldsymbol{b}_{3m-3}$ stoßen jeweils zwei Kurvensegmente aneinander. In diesen Punkten sollen die benachbarten Spline-Segmente bis zur 2. Ableitung übereinstimmen. Daher müssen die folgenden Bedingungen erfüllt sein:

$$\boldsymbol{P}'_{k-1}(1) = \boldsymbol{P}'_k(0), \quad \boldsymbol{P}''_{k-1}(1) = \boldsymbol{P}''_k(0), \quad k = 1(1)m{-}1. \tag{12.24}$$

Für die erste Bedingung ergibt sich (vgl. (12.22))

$$\boldsymbol{b}_{3k} - \boldsymbol{b}_{3k-1} = \boldsymbol{b}_{3k+1} - \boldsymbol{b}_{3k}. \tag{12.25}$$

Daraus folgt: Die Punkte $\boldsymbol{b}_{3k-1}, \boldsymbol{b}_{3k}, \boldsymbol{b}_{3k+1}$ sind kollinear, und der Interpolationspunkt $\boldsymbol{b}_{3k}$ ist deren Mittelpunkt (Abb. 12.8).

Aus der zweiten Bedingung (12.24) folgt (vgl. (12.23))

$$\boldsymbol{d}_k := \boldsymbol{b}_{3k-1} + (\boldsymbol{b}_{3k-1} - \boldsymbol{b}_{3k-2}) = \boldsymbol{b}_{3k+1} - (\boldsymbol{b}_{3k+2} - \boldsymbol{b}_{3k+1}). \tag{12.26}$$

Die Punkte $\boldsymbol{d}_k$, $k = 1(1)m-1$, heißen Gewichtspunkte (oder auch kurz Gewichte). Ihre aus (12.26) folgende Lage zeigt Abb. 12.8.

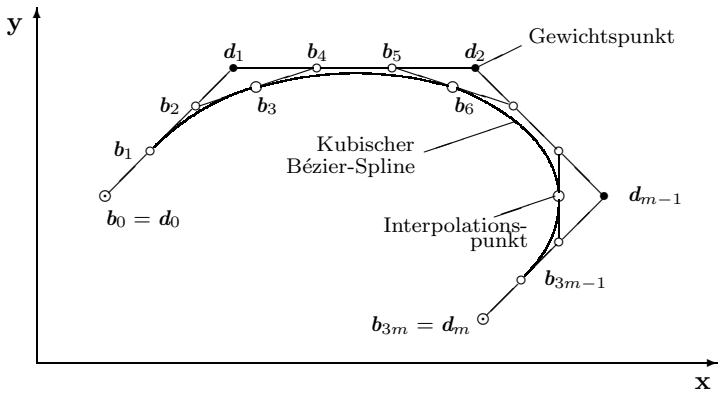


Abb. 12.8. • Vorgabepunkte, o Interpolationspunkte

Mittels (12.26) und (12.25) können die Bézier-Punkte $\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_{3m-2}, \boldsymbol{b}_{3m-1}$ mit Hilfe der $m+1$ Gewichtspunkte $\boldsymbol{d}_0, \boldsymbol{d}_1, \ldots, \boldsymbol{d}_{m-1}, \boldsymbol{d}_m$ wie folgt dargestellt werden:

$$\left\{\begin{array}{lcll} \boldsymbol{b}_{3k-2} & = & \frac{1}{3}(2\boldsymbol{d}_{k-1} + \boldsymbol{d}_k), & k = 1(1)m, \\ \boldsymbol{b}_{3k-1} & = & \frac{1}{3}(\boldsymbol{d}_{k-1} + 2\boldsymbol{d}_k), & k = 1(1)m, \\ \boldsymbol{b}_{3k} & = & \frac{1}{6}(\boldsymbol{d}_{k-1} + 4\boldsymbol{d}_k + \boldsymbol{d}_{k+1}), & k = 1(1)m-1. \end{array}\right. \tag{12.27}$$

Dies bedeutet, dass genau die $m+1$ Gewichtspunkte $\boldsymbol{d}_0, \ldots, \boldsymbol{d}_m$ vorgegeben werden müssen, um einen Bézier-Spline zu erzeugen; es werden also nicht die Interpolationspunkte vorgegeben.

Festzulegen sind noch die Randpunkte $\boldsymbol{b}_0$ und $\boldsymbol{b}_{3m}$. Mit der Wahl

$$\boldsymbol{b}_0 = \boldsymbol{d}_0, \quad \boldsymbol{b}_{3m} = \boldsymbol{d}_m \tag{12.28}$$

ergeben sich

$$\boldsymbol{P}_0''(0) = \boldsymbol{0}, \quad \boldsymbol{P}_{m-1}''(1) = \boldsymbol{0},$$

d. h. es liegt dann ein natürlicher kubischer Bézier-Spline vor.

Algorithmus 12.6. (*Kubische Bézier-Spline-Kurve*)

Gegeben: $m+1$ Gewichtspunkte $\boldsymbol{d}_k$, $k = 0(1)m$, $m \geq 2$, $\boldsymbol{d}_k \in \mathbb{R}^2$ oder $\boldsymbol{d}_k \in \mathbb{R}^3$.

Gesucht: m kubische Polynome
$$\boldsymbol{P}_k(v) = (1-v)^3\boldsymbol{b}_{3k} + 3(1-v)^2 v\boldsymbol{b}_{3k+1} + \\ +3(1-v)v^2\boldsymbol{b}_{3k+2} + v^3\boldsymbol{b}_{3k+3}, \quad v \in [0,1],$$
$k = 0(1)m-1$.

1. Schritt: Bestimmung der $3m-1$ Bézier-Punkte
$\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_{3m-1}$ mit (12.27).

2. Schritt: Wahl der Randpunkte mit (12.28).

Vorteile. Die Änderung eines Gewichtspunktes $\boldsymbol{d}_k$ wirkt sich nur auf die Bézier-Punkte $\boldsymbol{b}_{3k-3}, \boldsymbol{b}_{3k-2}, \ldots, \boldsymbol{b}_{3k+2}, \boldsymbol{b}_{3k+3}$ aus und ruft daher nur eine lokale Änderung des Kurvenverlaufs hervor. Ähnliches gilt beim Hinzufügen neuer Gewichtspunkte (Abb. 12.9). Deshalb können die kubischen Bézier-Splines in gut kontrollierter Weise zur Modellierung verwendet werden. Für die Erzeugung eines kubischen Bézier-Spline ist die Lösung eines linearen Gleichungssystems nicht erforderlich.

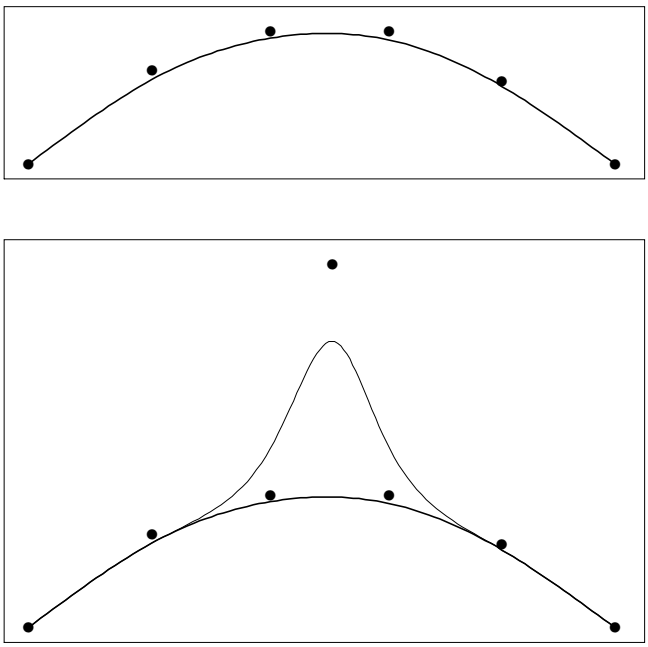


Abb. 12.9. Hinzufügen eines Gewichtspunktes

Nachteil. Es können nicht die Interpolationspunkte $\boldsymbol{b}_{3k}$, $k = 1(1)m-1$, vorgegeben werden, sondern nur die recht unanschaulichen Gewichtspunkte $\boldsymbol{d}_k$. Die Interpolationspunkte werden erst im Verlauf der Durchführung berechnet. Eine Modifizierung dazu ist in Abschnitt 12.3.3 zu finden; man erhält damit modifizierte (interpolierende) Bézier-Splines, siehe dazu auch andere Darstellungen in [HOSC1989].

Besonderheiten der kubischen Bézier-Splines

Will man mit Hilfe kubischer Bézier-Splines einen *Knick* erzeugen, so lässt man einfach drei aufeinander folgende Gewichtspunkte (z. B. $\boldsymbol{d}_{i-1}$, $\boldsymbol{d}_i$, $\boldsymbol{d}_{i+1}$) zusammenfallen. Dann ist nämlich die Interpolationsstelle $\boldsymbol{b}_{3i}$ mit diesen Gewichtspunkten identisch und damit die Bézier-Kurve an der Stelle $\boldsymbol{b}_{3i}$ nicht mehr differenzierbar.

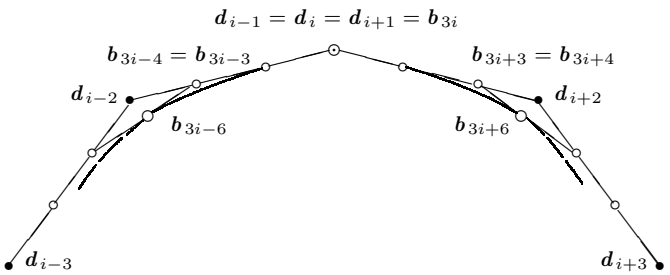


Abb. 12.10. Bézier-Spline mit Knick

Die Kurvensegmente zwischen $\boldsymbol{b}_{3i-4}$ und $\boldsymbol{d}_i$ bzw. $\boldsymbol{d}_i$ und $\boldsymbol{b}_{3i+4}$ sind Geradenstücke.

12.3.2 Bézier-Spline-Flächen

Eine Bézier-Spline-Fläche setzt sich stückweise aus Bézier-Flächen, den sogenannten Pflastern (patches) zusammen. Im Folgenden werden nur bikubische Bézier-Spline-Flächen und Pflaster betrachtet.
Eine bikubische Bézier-Fläche besitzt die Darstellung

$$\boldsymbol{P}(v,w) = \sum_{j=0}^{3}\sum_{s=0}^{3} B_j^3(v) B_s^3(w) \boldsymbol{b}_{js}, \quad v \in [0,1],\ w \in [0,1]. \tag{12.29}$$

Dabei sind $B_j^3(v), B_s^3(w)$ Bernstein-Polynome 3. Grades, vgl. (12.19), und $\boldsymbol{b}_{js} \in \mathbb{R}^3$, $j = 0(1)3, s = 0(1)3$, sind die 16 Bézier-Punkte der Fläche.
Diese Darstellung ergibt sich aus der folgenden Erzeugung der Bézier-Fläche. Im Raum sei eine kubische Bézier-Kurve

$$\boldsymbol{P}(v) = \sum_{j=0}^{3} B_j^3(v)\boldsymbol{b}_j, \quad v \in [0,1],$$

gegeben, vgl. (12.20), (12.21). Jeder Bézier-Punkt $\boldsymbol{b}_j$ durchlaufe nun selbst eine kubische Bézier-Kurve

$$\boldsymbol{b}_j(w) = \sum_{s=0}^{3} B_s^3(w)\boldsymbol{b}_{js}, \quad w \in [0,1],\ j = 0(1)3.$$

Die Kurve $\boldsymbol{P}(v)$ wird also, im Allgemeinen bei gleichzeitiger Gestaltänderung, im Raum bewegt und überstreicht dabei eine Fläche mit der Darstellung

$$\begin{aligned} \boldsymbol{P}(v,w) &= \sum_{j=0}^{3} B_j^3(v)\boldsymbol{b}_j(w) \\ &= \sum_{j=0}^{3} B_j^3(v) \sum_{s=0}^{3} B_s^3(w)\boldsymbol{b}_{js} \end{aligned}$$

in Übereinstimmung mit (12.29).

Für w = const. ergeben sich die v-Kurven, für v = const. die w-Kurven der Bézier-Fläche. Aus den Eigenschaften der Bernstein-Polynome folgt, dass die Bézier-Punkte (Randpunkte) $\boldsymbol{P}(0,0) = \boldsymbol{b}_{00}$, $\boldsymbol{P}(1,0) = \boldsymbol{b}_{30}$, $\boldsymbol{P}(0,1) = \boldsymbol{b}_{03}$, $\boldsymbol{P}(1,1) = \boldsymbol{b}_{33}$ auf der Bézier-Fläche liegen, vgl. Abb. 12.11.
Die Punkte $\boldsymbol{b}_{j0}$, $\boldsymbol{b}_{j3}$, $j = 0(1)3$, $\boldsymbol{b}_{0s}$, $\boldsymbol{b}_{3s}$, $s = 0(1)3$, sind die Bézier-Punkte der Randkurven der Bézier-Fläche (Abb. 12.11).

Wegen (12.22), angewendet auf die Randkurven, sind die Tangentialebenen der Bézier-Fläche in den Randpunkten durch die folgenden (nicht kollinearen) Punkttripel bestimmt: $(\boldsymbol{b}_{00}, \boldsymbol{b}_{10}, \boldsymbol{b}_{01})$, $(\boldsymbol{b}_{30}, \boldsymbol{b}_{31}, \boldsymbol{b}_{20})$, $(\boldsymbol{b}_{03}, \boldsymbol{b}_{02}, \boldsymbol{b}_{13})$, $(\boldsymbol{b}_{33}, \boldsymbol{b}_{23}, \boldsymbol{b}_{32})$.
Die 16 Bézier-Punkte sind im Raum so zu wählen, dass 9 (im Allgemeinen nicht ebene) Vierecke entstehen (Abb. 12.11).

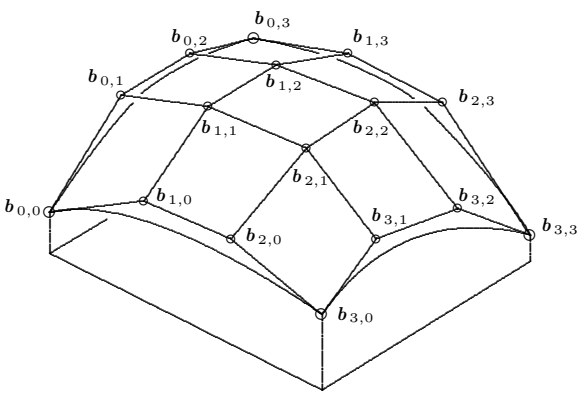


Abb. 12.11. Bikubische Bézier-Fläche, gleichzeitig Pflaster $i = 0$, $k = 0$ einer bikubischen Bézier-Spline-Fläche

Eine bikubische Bézier-Spline-Fläche setzt sich aus $m \cdot n$ bikubischen Pflastern zusammen (Abb. 12.11):

$$\begin{gathered} \boldsymbol{P}_{ik}(v,w) = \sum_{j=0}^{3} \sum_{s=0}^{3} B_j^3(v) B_s^3(w) \boldsymbol{b}_{3i+j,3k+s}, \\ i = 0(1)m-1,\ k = 0(1)n-1. \end{gathered} \tag{12.30}$$

Ausführlich lautet diese Darstellung eines Pflasters

$$\begin{aligned} \boldsymbol{P}_{ik}(v,w) &= \sum_{j=0}^{3} \left(\sum_{s=0}^{3} B_s^3(w) \boldsymbol{b}_{3i+j,3k+s} \right) B_j^3(v) \\ &= [\boldsymbol{b}_{3i,3k}(1-w)^3 \\ &+ 3\boldsymbol{b}_{3i,3k+1}(1-w)^2 w + 3\boldsymbol{b}_{3i,3k+2}(1-w)w^2 \\ &+ \boldsymbol{b}_{3i,3k+3} w^3](1-v)^3 + 3[\boldsymbol{b}_{3i+1,3k}(1-w)^3 \\ &+ 3\boldsymbol{b}_{3i+1,3k+1}(1-w)^2 w + 3\boldsymbol{b}_{3i+1,3k+2}(1-w)w^2 \\ &+ \boldsymbol{b}_{3i+1,3k+3} w^3](1-v)^2 v + 3[\boldsymbol{b}_{3i+2,3k}(1-w)^3 \\ &+ 3\boldsymbol{b}_{3i+2,3k+1}(1-w)^2 w + 3\boldsymbol{b}_{3i+2,3k+2}(1-w)w^2 \\ &+ \boldsymbol{b}_{3i+2,3k+3} w^3](1-v)v^2 + [\boldsymbol{b}_{3i+3,3k}(1-w)^3 \\ &+ 3\boldsymbol{b}_{3i+3,3k+1}(1-w)^2 w + 3\boldsymbol{b}_{3i+3,3k+2}(1-w)w^2 \\ &+ \boldsymbol{b}_{3i+3,3k+3} w^3]v^3 \end{aligned} \tag{12.31}$$

Die Nummerierung der Bézier-Punkte ist so gewählt, dass sich für benachbarte Pflaster Randkurven mit denselben Bézier-Punkten ergeben. Damit ist gesichert, dass benachbarte Pflaster stetig aneinander schließen.
Aus (12.31) folgt z. B., dass für die benachbarten Pflaster $(i-1,k)$ und (i,k) gilt:

$$\begin{aligned} \boldsymbol{P}_{i-1,k}(1,w) &= \sum_{s=0}^{3} B_s^3(w) \boldsymbol{b}_{3(i-1)+3,3k+s} \\ &= \sum_{s=0}^{3} B_s^3(w) \boldsymbol{b}_{3i,3k+s} = \boldsymbol{P}_{ik}(0,w). \end{aligned}$$

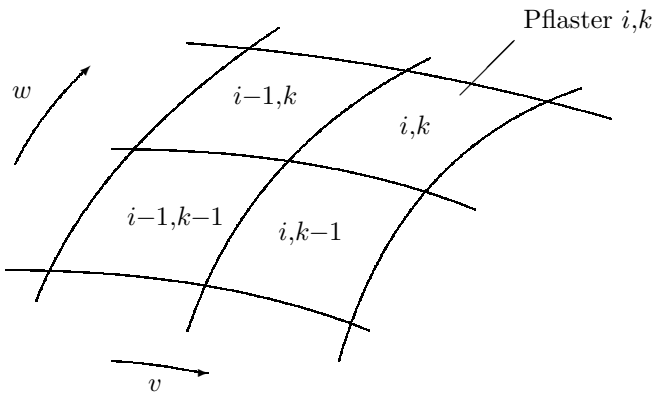


Abb. 12.12.

Berechnung der bikubischen Bézier-Splines

Für die hier zu konstruierende Bézier-Spline-Fläche wird vorausgesetzt, dass für die partiellen Ableitungen erster und zweiter Ordnung von (12.31) nach v bzw. w benachbarter Pflaster $(i-1,k-1)$, $(i-1,k)$, $(i,k-1)$ und (i,k) die folgenden Bedingungen erfüllt sind:

$$\left\{\begin{array}{lcl} \boldsymbol{P}_{i-1,k-1}(1,1)|_v &=& \boldsymbol{P}_{i,k-1}(0,1)|_v, \\ \boldsymbol{P}_{i-1,k}(1,0)|_v &=& \boldsymbol{P}_{i,k}(0,0)|_v, \\ \boldsymbol{P}_{i-1,k-1}(1,1)|_w &=& \boldsymbol{P}_{i-1,k}(1,0)|_w, \\ \boldsymbol{P}_{i,k-1}(0,1)|_w &=& \boldsymbol{P}_{i,k}(0,0)|_w, \end{array}\right. \tag{12.32}$$

wobei $|_v$, $|_w$ die partiellen Ableitungen nach v bzw. w bedeuten, und

$$\left\{\begin{array}{lcl} \boldsymbol{P}_{i-1,k-1}(1,1)|_{vv} &=& \boldsymbol{P}_{i,k-1}(0,1)|_{vv}, \\ \boldsymbol{P}_{i-1,k-1}(1,1)|_{vw} &=& \boldsymbol{P}_{i,k-1}(0,1)|_{vw}, \\ \boldsymbol{P}_{i-1,k}(1,0)|_{vv} &=& \boldsymbol{P}_{i,k}(0,0)|_{vv}, \\ \boldsymbol{P}_{i-1,k}(1,0)|_{vw} &=& \boldsymbol{P}_{i,k}(0,0)|_{vw}, \\ \boldsymbol{P}_{i-1,k-1}(1,1)|_{ww} &=& \boldsymbol{P}_{i-1,k}(1,0)|_{ww}, \\ \boldsymbol{P}_{i-1,k-1}(1,1)|_{wv} &=& \boldsymbol{P}_{i-1,k}(1,0)|_{wv}, \\ \boldsymbol{P}_{i,k-1}(0,1)|_{ww} &=& \boldsymbol{P}_{i,k}(0,0)|_{ww}, \\ \boldsymbol{P}_{i,k-1}(0,1)|_{wv} &=& \boldsymbol{P}_{i,k}(0,0)|_{wv}, \end{array}\right. \tag{12.33}$$

wobei $|_{vv}$, $|_{vw}$, $|_{ww}$, $|_{wv}$ die zweiten partiellen Ableitungen bezeichnen.

Aus den Bedingungen (12.32) und (12.33) lassen sich mit den sogenannten Gewichtspunkten

$$\begin{aligned} \boldsymbol{d}_{ik} := \;& \boldsymbol{b}_{3i-2,3k-2} - 2\boldsymbol{b}_{3i-2,3k-1} - 2\boldsymbol{b}_{3i-1,3k-2} + 4\boldsymbol{b}_{3i-1,3k-1} \\ =\;& 4\boldsymbol{b}_{3i+1,3k+1} - 2\boldsymbol{b}_{3i+1,3k+2} - 2\boldsymbol{b}_{3i+2,3k+1} + \boldsymbol{b}_{3i+2,3k+2} \\ =\;& -2\boldsymbol{b}_{3i-2,3k+1} + 4\boldsymbol{b}_{3i-1,3k+1} - 2\boldsymbol{b}_{3i-1,3k+2} + \boldsymbol{b}_{3i-2,3k+2} \\ =\;& -2\boldsymbol{b}_{3i+1,3k-2} + 4\boldsymbol{b}_{3i+1,3k-1} + \boldsymbol{b}_{3i+2,3k-2} - 2\boldsymbol{b}_{3i+2,3k-1} \end{aligned}$$

die folgenden Gleichungen ableiten:

$$
\left\{\begin{array}{lcl}
\boldsymbol{b}_{3i-2,3k-2} & = & \frac{4}{9}\boldsymbol{d}_{i-1,k-1}+\frac{2}{9}\boldsymbol{d}_{i-1,k}+\frac{2}{9}\boldsymbol{d}_{i,k-1}+\frac{1}{9}\boldsymbol{d}_{i,k} \\
 & & \text{für} \quad i=1(1)m,\ k=1(1)n, \\
\boldsymbol{b}_{3i-2,3k+2} & = & \frac{4}{9}\boldsymbol{d}_{i-1,k+1}+\frac{2}{9}\boldsymbol{d}_{i-1,k}+\frac{2}{9}\boldsymbol{d}_{i,k+1}+\frac{1}{9}\boldsymbol{d}_{i,k} \\
 & & \text{für} \quad i=1(1)m,\ k=0(1)n-1, \\
\boldsymbol{b}_{3i+2,3k-2} & = & \frac{4}{9}\boldsymbol{d}_{i+1,k-1}+\frac{2}{9}\boldsymbol{d}_{i,k-1}+\frac{2}{9}\boldsymbol{d}_{i+1,k}+\frac{1}{9}\boldsymbol{d}_{i,k} \\
 & & \text{für} \quad i=0(1)m-1,\ k=1(1)n, \\
\boldsymbol{b}_{3i+2,3k+2} & = & \frac{4}{9}\boldsymbol{d}_{i+1,k+1}+\frac{2}{9}\boldsymbol{d}_{i,k+1}+\frac{2}{9}\boldsymbol{d}_{i+1,k}+\frac{1}{9}\boldsymbol{d}_{i,k} \\
 & & \text{für} \quad i=0(1)m-1,\ k=0(1)n-1, \\
\boldsymbol{b}_{3i-2,3k} & = & \frac{1}{9}\boldsymbol{d}_{i-1,k-1}+\frac{4}{9}\boldsymbol{d}_{i-1,k}+\frac{1}{9}\boldsymbol{d}_{i-1,k+1}+ \\
 & & +\frac{1}{18}\boldsymbol{d}_{i,k-1}+\frac{2}{9}\boldsymbol{d}_{i,k}+\frac{1}{18}\boldsymbol{d}_{i,k+1} \\
 & & \text{für} \quad i=1(1)m,\ k=1(1)n-1, \\
\boldsymbol{b}_{3i,3k-2} & = & \frac{1}{9}\boldsymbol{d}_{i-1,k-1}+\frac{4}{9}\boldsymbol{d}_{i,k-1}+\frac{1}{9}\boldsymbol{d}_{i+1,k-1}+ \\
 & & +\frac{1}{18}\boldsymbol{d}_{i-1,k}+\frac{2}{9}\boldsymbol{d}_{i,k}+\frac{1}{18}\boldsymbol{d}_{i+1,k} \\
 & & \text{für} \quad i=1(1)m-1,\ k=1(1)n, \\
\boldsymbol{b}_{3i,3k+2} & = & \frac{1}{9}\boldsymbol{d}_{i-1,k+1}+\frac{4}{9}\boldsymbol{d}_{i,k+1}+\frac{1}{9}\boldsymbol{d}_{i+1,k+1}+ \\
 & & +\frac{1}{18}\boldsymbol{d}_{i-1,k}+\frac{2}{9}\boldsymbol{d}_{i,k}+\frac{1}{18}\boldsymbol{d}_{i+1,k} \\
 & & \text{für} \quad i=1(1)m-1,\ k=0(1)n-1, \\
\boldsymbol{b}_{3i+2,3k} & = & \frac{1}{9}\boldsymbol{d}_{i+1,k-1}+\frac{4}{9}\boldsymbol{d}_{i+1,k}+\frac{1}{9}\boldsymbol{d}_{i+1,k+1}+ \\
 & & +\frac{1}{18}\boldsymbol{d}_{i,k-1}+\frac{2}{9}\boldsymbol{d}_{i,k}+\frac{1}{18}\boldsymbol{d}_{i,k+1} \\
 & & \text{für} \quad i=0(1)m-1,\ k=1(1)n-1, \\
\boldsymbol{b}_{3i,3k} & = & \frac{1}{36}\boldsymbol{d}_{i-1,k-1}+\frac{1}{9}\boldsymbol{d}_{i,k-1}+\frac{1}{36}\boldsymbol{d}_{i+1,k-1}+ \\
 & & +\frac{1}{9}\boldsymbol{d}_{i-1,k}+\frac{4}{9}\boldsymbol{d}_{i,k}+\frac{1}{9}\boldsymbol{d}_{i+1,k}+ \\
 & & +\frac{1}{36}\boldsymbol{d}_{i-1,k+1}+\frac{1}{9}\boldsymbol{d}_{i,k+1}+\frac{1}{36}\boldsymbol{d}_{i+1,k+1} \\
 & & \text{für} \quad i=1(1)m-1,\ k=1(1)n-1.
\end{array}\right. \tag{12.34}
$$

Kennt man nun die $\boldsymbol{d}_{ik}$ in den Gleichungen (12.34), so lassen sich die nicht an den Rändern der Fläche gelegenen Bézier-Punkte mit Hilfe dieser Gleichungen bestimmen. Führt man also die $(m+1)(n+1)$ Gewichtspunkte $\boldsymbol{d}_{ik}$ als unabhängige Größen ein, so sind folglich die inneren Bézier-Punkte bekannt. Für die Berechnung der bikubischen Polynome (12.31) fehlen nur noch die an den Rändern liegenden $6(m+n)$ Bézier-Punkte $\boldsymbol{b}_{0,s}$, $\boldsymbol{b}_{3m,s}$, $\boldsymbol{b}_{j,0}$ und $\boldsymbol{b}_{j,3n}$, $s=0(1)3n$, $j=1(1)3m-1$, welche ebenfalls als unabhängige Größen vorgegeben werden.

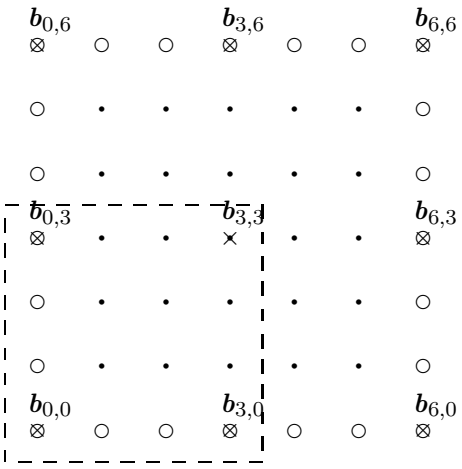


Abb. 12.13. ∘ vorzugebende Bézier-Punkte, • errechnete Bézier-Punkte und × Interpolationsstellen in der Parameterebene. Das Rechteckgitter gibt hier nur Aufschluss über die Reihenfolge der Bézier-Punkte, nicht über ihre relative Lage zueinander im Raum

Korrektur bikubischer Bézier-Splines

Verschiebt man einen Gewichtspunkt $\boldsymbol{d}_{ik}$ eines bikubischen Bézier-Splines um $36\boldsymbol{d}$, so ändern sich die Bézier-Punkte $\boldsymbol{b}_{j,s}$ in den angrenzenden Pflastern in der folgenden Weise: Aus den Gleichungen (12.34) ergeben sich die neuen Bézier-Punkte $\boldsymbol{b}^*_{js}$ zu

$$
\begin{aligned}
\boldsymbol{b}^*_{3i-2,3k-2} &= [4\boldsymbol{d}_{i-1,k-1} + 2\boldsymbol{d}_{i-1,k} + 2\boldsymbol{d}_{i,k-1} + \boldsymbol{d}_{i,k} + 36\boldsymbol{d}]\,/\,9 &&= \boldsymbol{b}_{3i-2,3k-2} + 4\boldsymbol{d},\\
\boldsymbol{b}^*_{3i-2,3k+2} &= [4\boldsymbol{d}_{i-1,k+1} + 2\boldsymbol{d}_{i-1,k} + 2\boldsymbol{d}_{i,k+1} + \boldsymbol{d}_{i,k} + 36\boldsymbol{d}]\,/\,9 &&= \boldsymbol{b}_{3i-2,3k+2} + 4\boldsymbol{d},\\
\boldsymbol{b}^*_{3i+2,3k-2} &= [4\boldsymbol{d}_{i+1,k-1} + 2\boldsymbol{d}_{i,k-1} + 2\boldsymbol{d}_{i+1,k} + \boldsymbol{d}_{i,k} + 36\boldsymbol{d}]\,/\,9 &&= \boldsymbol{b}_{3i+2,3k-2} + 4\boldsymbol{d},\\
\boldsymbol{b}^*_{3i+2,3k+2} &= [4\boldsymbol{d}_{i+1,k+1} + 2\boldsymbol{d}_{i,k+1} + 2\boldsymbol{d}_{i+1,k} + \boldsymbol{d}_{i,k} + 36\boldsymbol{d}]\,/\,9 &&= \boldsymbol{b}_{3i+2,3k+2} + 4\boldsymbol{d},\\
\boldsymbol{b}^*_{3i-2,3k} &= [2\boldsymbol{d}_{i-1,k-1} + 8\boldsymbol{d}_{i-1,k} + 2\boldsymbol{d}_{i-1,k+1} + \boldsymbol{d}_{i,k-1}\\
&\quad + 4(\boldsymbol{d}_{i,k} + 36\boldsymbol{d}) + \boldsymbol{d}_{i,k+1}]\,/\,18 &&= \boldsymbol{b}_{3i-2,3k} + 8\boldsymbol{d},\\
\boldsymbol{b}^*_{3i,3k-2} &= [2\boldsymbol{d}_{i-1,k-1} + 8\boldsymbol{d}_{i,k-1} + 2\boldsymbol{d}_{i+1,k-1} + \boldsymbol{d}_{i-1,k}\\
&\quad + 4(\boldsymbol{d}_{i,k} + 36\boldsymbol{d}) + \boldsymbol{d}_{i+1,k}]\,/\,18 &&= \boldsymbol{b}_{3i,3k-2} + 8\boldsymbol{d},\\
\boldsymbol{b}^*_{3i,3k+2} &= [2\boldsymbol{d}_{i-1,k+1} + 8\boldsymbol{d}_{i,k+1} + 2\boldsymbol{d}_{i+1,k+1} + \boldsymbol{d}_{i-1,k}\\
&\quad + 4(\boldsymbol{d}_{i,k} + 36\boldsymbol{d}) + \boldsymbol{d}_{i+1,k}]\,/\,18 &&= \boldsymbol{b}_{3i,3k+2} + 8\boldsymbol{d},\\
\boldsymbol{b}^*_{3i+2,3k} &= [2\boldsymbol{d}_{i+1,k-1} + 8\boldsymbol{d}_{i+1,k} + 2\boldsymbol{d}_{i+1,k+1} + \boldsymbol{d}_{i,k-1}\\
&\quad + 4(\boldsymbol{d}_{i,k} + 36\boldsymbol{d}) + \boldsymbol{d}_{i,k+1}]\,/\,18 &&= \boldsymbol{b}_{3i+2,3k} + 8\boldsymbol{d},\\
\boldsymbol{b}^*_{3i,3k} &= [\boldsymbol{d}_{i-1,k-1} + 4\boldsymbol{d}_{i,k-1} + \boldsymbol{d}_{i+1,k-1} + 4\boldsymbol{d}_{i-1,k} + 16(\boldsymbol{d}_{i,k} + 36\boldsymbol{d})\\
&\quad + 4\boldsymbol{d}_{i+1,k} + \boldsymbol{d}_{i-1,k+1} + 4\boldsymbol{d}_{i,k+1} + \boldsymbol{d}_{i+1,k+1}]\,/\,36 &&= \boldsymbol{b}_{3i,3k} + 16\boldsymbol{d}.
\end{aligned}
$$

Die Änderungen der restlichen, in der nachstehenden Tabelle aufgeführten Bézier-Punkte errechnet man analog.

$$\left\{\begin{array}{c|ccccccc} 3i+3 & +\boldsymbol{d} & +2\boldsymbol{d} & +4\boldsymbol{d} & +4\boldsymbol{d} & +4\boldsymbol{d} & +2\boldsymbol{d} & +\boldsymbol{d} \\ 3i+2 & +2\boldsymbol{d} & +4\boldsymbol{d} & +8\boldsymbol{d} & +8\boldsymbol{d} & +8\boldsymbol{d} & +4\boldsymbol{d} & +2\boldsymbol{d} \\ 3i+1 & +4\boldsymbol{d} & +8\boldsymbol{d} & +16\boldsymbol{d} & +16\boldsymbol{d} & +16\boldsymbol{d} & +8\boldsymbol{d} & +4\boldsymbol{d} \\ 3i & +4\boldsymbol{d} & +8\boldsymbol{d} & +16\boldsymbol{d} & +16\boldsymbol{d} & +16\boldsymbol{d} & +8\boldsymbol{d} & +4\boldsymbol{d} \\ 3i-1 & +4\boldsymbol{d} & +8\boldsymbol{d} & +16\boldsymbol{d} & +16\boldsymbol{d} & +16\boldsymbol{d} & +8\boldsymbol{d} & +4\boldsymbol{d} \\ 3i-2 & +2\boldsymbol{d} & +4\boldsymbol{d} & +8\boldsymbol{d} & +8\boldsymbol{d} & +8\boldsymbol{d} & +4\boldsymbol{d} & +2\boldsymbol{d} \\ 3i-3 & +\boldsymbol{d} & +2\boldsymbol{d} & +4\boldsymbol{d} & +4\boldsymbol{d} & +4\boldsymbol{d} & +2\boldsymbol{d} & +\boldsymbol{d} \\ \hline & 3k-3 & 3k-2 & 3k-1 & 3k & 3k+1 & 3k+2 & 3k+3 \end{array}\right. \tag{12.35}$$

Eine Änderung der äußeren Bézier-Punkte $\boldsymbol{b}_{j,s}$ ($j \in \{0, ..., 3m\}$, $s \in \{0, \ldots, 3n\}$) ändert die übrigen Bézier-Punkte nicht.

Korrigiert man nun statt des Gewichtes $\boldsymbol{d}_{ik}$ den Bézier-Punkt $\boldsymbol{b}_{3i,3k}$ um $16\boldsymbol{d}$, so ändern sich der Gewichtspunkt $\boldsymbol{d}_{ik}$ um $36\boldsymbol{d}$ und die entsprechenden Bézier-Punkte $\boldsymbol{b}_{j,s}$ um die in der Tabelle (12.35) aufgezeigten Werte. Als wesentliches Ergebnis der Untersuchung des Verhaltens von bikubischen Bézier-Splines gegenüber Änderungen der Vorgabepunkte kann man feststellen, dass eine mittels des bikubischen Bézier-Verfahrens errechnete Splinefläche (im Gegensatz zu den bikubischen Splines in Abschnitt 12.1) *lokal änderbar* ist, was dem bikubischen Bézier-Verfahren für die praktische Anwendung gegenüber den bikubischen Splines erhebliche Vorteile einräumt.

Nachteil: Bei der praktischen Anwendung des kubischen bzw. bikubischen Bézier-Verfahrens wird dem Benutzer auffallen, dass er zwar den Verlauf der jeweils zu bestimmenden Splinekurve (-fläche) an den Rändern durch Vorgabe der äußeren Bézier-Punkte relativ exakt bestimmen kann, dass er aber auf den Verlauf der Splinekurven im Inneren der Fläche nur indirekt, d. h. durch die Wahl der sehr unanschaulichen Gewichtspunkte, Einfluss ausüben kann. Es liegt also nahe, diese Verfahren so zu modifizieren, dass man statt der Gewichtspunkte die Interpolationsstellen des kubischen bzw. bikubischen Bézier-Spline vorgeben kann, um ein für die Anwendung besser einsetzbares, echtes Interpolationsverfahren zu erhalten, vgl. Abschnitt 12.3.3.

Algorithmus 12.7. (*Berechnung eines bikubischen Bézier-Spline*)

Gegeben: 1. $\boldsymbol{b}_{0,s}$, $\boldsymbol{b}_{3m,s}$, $\boldsymbol{b}_{j,0}$, $\boldsymbol{b}_{j,3n}$ für $s = 0(1)3n$, $j = 0(1)3m$ als äußere Bézier-Punkte.
2. $\boldsymbol{d}_{ik}$, $i = 0(1)m$, $k = 0(1)n$, als Gewichtspunkte.

Gesucht: Bikubischer Bézier-Spline (12.30).

1. Schritt: Bestimmung der $9(m-1)(n-1)$ inneren Bézier-Punkte mit (12.34).

2. Schritt: Aufstellung der $m \cdot n$ bikubischen Polynome (12.31) für $i = 0(1)m-1$, $k = 0(1)n-1$.

Im Anschluss werden noch Beispiele für Flächen angegeben, die sich mit Hilfe des bikubischen bzw. modifizierten bikubischen Bézier-Verfahrens darstellen lassen.

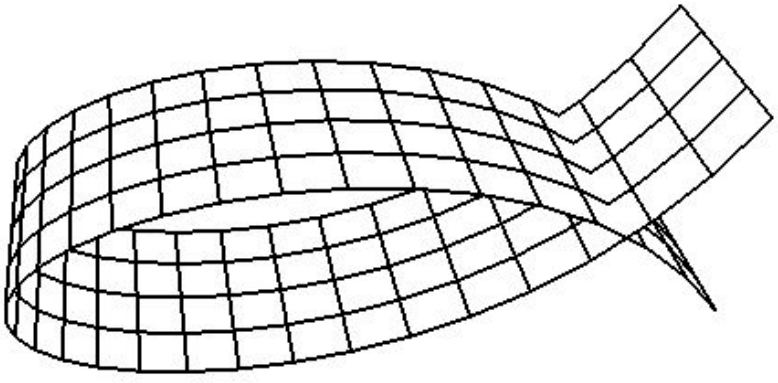


Abb. 12.14. Erzeugung einer sich selbst durchdringenden Fläche

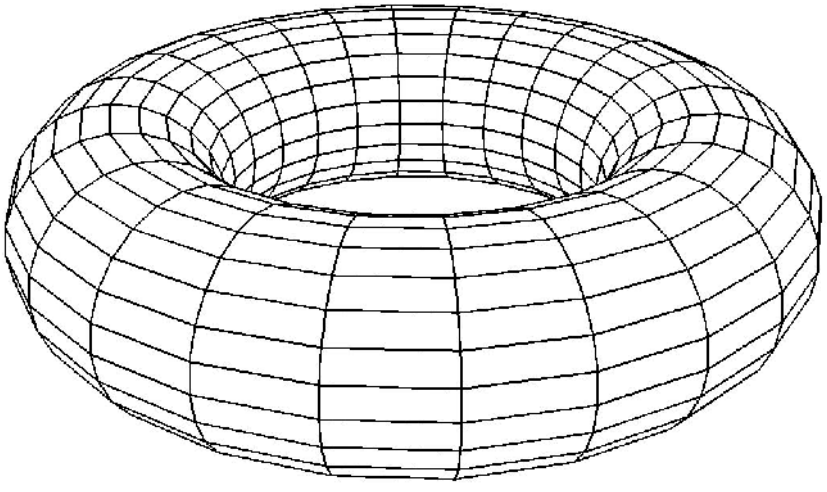
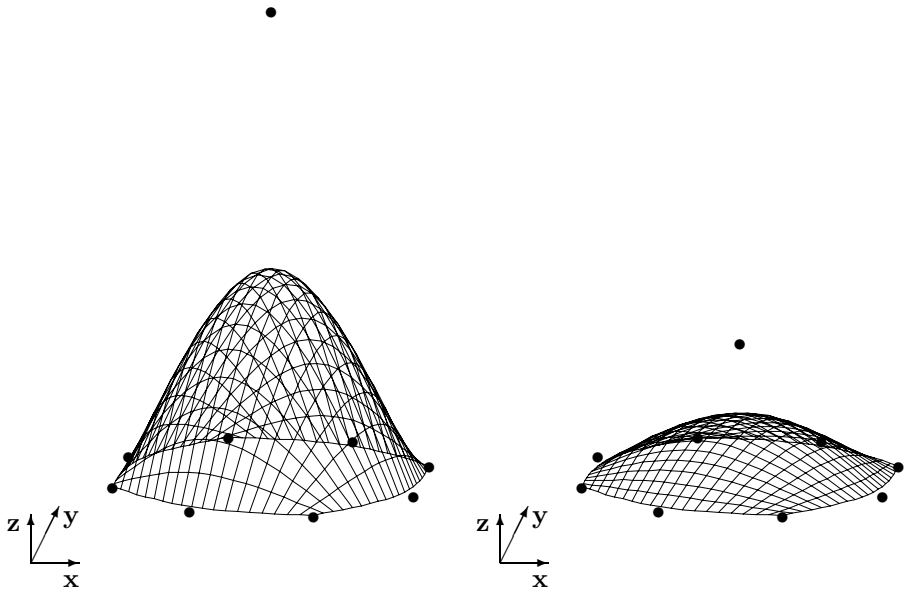


Abb. 12.15. Erzeugung einer geschlossenen Fläche (Ringtorus)



a) b)

Abb. 12.16. a) Konstruierte Bézier-Spline-Fläche zu den eingetragenen Interpolationsstellen **b**) Änderung einer Interpolationsstelle in a) und die dazugehörige Bézier-Spline-Fläche

12.3.3 Modifizierte (interpolierende) kubische Bézier-Splines

Eine wegen der in Abschnitt 12.3.2 genannten Nachteile sinnvolle Modifikation der bisher behandelten Bézier-Verfahren ist z. B. in [STEU1979] und [HOSC1989] angegeben. In [STEU1979] wird das bikubische Bézier-Verfahren zu einem „echten“ Interpolationsverfahren ausgebaut, indem statt der Gewichtspunkte $\boldsymbol{d}_{ik}$ neben den $6(m+n)$ äußeren Bézier-Punkten $\boldsymbol{b}_{0,s}$, $\boldsymbol{b}_{3m,s}$, $\boldsymbol{b}_{j,0}$, $\boldsymbol{b}_{j,3n}$, $s = 0(1)3n$, $j = 1(1)3m-1$, die $(m-1)(n-1)$ Bézier-Punkte $\boldsymbol{b}_{3i,3k}$, $i = 1(1)m-1$, $k = 1(1)n-1$, als Interpolationspunkte des bikubischen Bézier-Spline vorgegeben werden.

Zunächst werden diese Interpolationspunkte als Gewichtspunkte betrachtet, zu welchen man sich durch Mittelbildung sogenannte Pseudointerpolationspunkte errechnet, die unter Beachtung des Korrekturverhaltens bikubischer Bézier-Splines so lange verschoben werden, bis sie um weniger als ein vorgegebenes $\varepsilon > 0$ von den gegebenen Interpolationspunkten abweichen.

12.4 B-Splines

Wie die Bézier-Splines sind auch die B-Splines parametrisch und nicht interpolierend. Gegenüber den Bézier-Splines sind die B-Splines leichter in der Handhabung und deshalb im Allgemeinen vorzuziehen. Ganz besonders macht sich das bei den geschlossenen Kurven bzw. Flächen bemerkbar, für diese sind nur die B-Splines zu empfehlen.

12.4.1 B-Spline-Kurven

Für eine B-Spline-Kurve sind $n+1$ de Boor-Punkte[1](Kontrollpunkte) $\boldsymbol{d}_i$, $i = 0(1)n$, $n \geq 1$, $\boldsymbol{d}_i \in \mathbb{R}^2$ oder $\boldsymbol{d}_i \in \mathbb{R}^3$, vorzugeben. Diese Punkte $\boldsymbol{d}_0$, $\boldsymbol{d}_1$, ..., $\boldsymbol{d}_n$ sind die Ecken des de Boor-Polygons. Zu gegebenen $n+1$ de Boor-Punkten wird eine B-Spline-Kurve der Ordnung k, $1 \leq k \leq n+1$, mit Hilfe normalisierter B-Spline-Funktionen $N_{i,k}$ wie folgt erzeugt:

$$\boldsymbol{P}(t) = \sum_{i=0}^{n} N_{i,k}(t)\boldsymbol{d}_i, \quad t \in I. \tag{12.36}$$

Zunächst werden die Definition und einige Eigenschaften der B-Spline-Funktionen $N_{i,k}$ behandelt.
Gegeben seien $k+1$ reelle Parameterwerte t_j, $j = i(1)i+k$, $k \geq 1$, mit

$$t_i < t_{i+1} < \ldots < t_{i+k};$$

die t_j werden auch Knoten genannt. Je zwei benachbarte Knoten bestimmen ein Intervall. Die zu diesen Knoten gehörige normalisierte B-Spline-Funktion $N_{i,k}$ der Ordnung k wird wie folgt definiert:

$$\begin{cases} k = 1: \quad N_{i,1}(t) = \begin{cases} 1 & \text{für} \quad t_i \leq t < t_{i+1} \\ 0 & \text{für} \quad t < t_i \quad \text{und} \quad t \geq t_{i+1} \end{cases} \\ k \geq 2: \quad N_{i,k}(t) = \frac{t-t_i}{t_{i+k-1}-t_i} N_{i,k-1}(t) + \frac{t_{i+k}-t}{t_{i+k}-t_{i+1}} N_{i+1,k-1}(t). \end{cases} \tag{12.37}$$

Das Intervall $[t_i, t_{i+k}]$ heißt Träger von $N_{i,k}$. Die Ordnung k gibt an, aus wievielen Teilintervallen, beginnend bei t_i, der Träger besteht.
Die B-Spline-Funktion $N_{i,k}$ ist nur auf ihrem Träger von Null verschieden. Für $k \geq 2$ gilt:

$$N_{i,k}(t) > 0 \quad \text{für} \quad t_i < t < t_{i+k},$$

$$N_{i,k}(t) = 0 \quad \text{für} \quad t \leq t_i \text{ und } t \geq t_{i+k}.$$

Aufgrund der rekursiven Definition (12.37) gilt für $k \geq 2$:
$N_{i,k}(t)$ setzt sich stückweise aus k Polynomen $(k-1)$ten Grades zusammen (je eines in jedem Teilintervall des Trägers), die an den inneren Knoten $t_j, t_i < t_j < t_{i+k}$, des Trägers C^{k-2}-stetig aneinander schließen.

[1] Die de Boor-Punkte $\boldsymbol{d}_i$ sind verschieden von den ebenso bezeichneten Gewichtspunkten der Bézier-Spline-Kurven.

Für einen glatten Anschluss muss $k-2 \geq 1$, also $k \geq 3$ sein.

Nach (12.36) werden für eine B-Spline-Kurve mit den $n+1$ Kontrollpunkten $\boldsymbol{d}_i, i = 0(1)n$, ebenso viele B-Spline-Funktionen $N_{i,k}$ mit den insgesamt $n+1+k$ Knoten

$$t_0, t_1, \ldots, t_n, t_{n+1}, \ldots, t_{n+k}$$

benötigt.
Die Knoten können mittels $t_j := j,\ j = 0(1)n+k$, normiert werden:

$$0, 1, \ldots, n, n+1, \ldots, n+k;$$

alle Teilintervalle haben dann die Länge 1, und (12.37) lautet

$$\begin{cases} k=1: & N_{i,1}(t) = \begin{cases} 1 & \text{für} \quad i \leq t < i+1 \\ 0 & \text{für} \quad t < i \quad \text{und} \quad t \geq i+1 \end{cases} \\ k \geq 2: & N_{i,k}(t) = \frac{t-i}{k-1} N_{i,k-1}(t) + \frac{i+k-t}{k-1} N_{i+1,k-1}(t). \end{cases}$$

Eine B-Spline-Kurve mit normierten Knoten heißt uniforme B-Spline-Kurve.

Nun werde von einer B-Spline-Kurve (12.36) der Punkt $\boldsymbol{P}(t)$ mit $t \in [t_r, t_{r+1})$ berechnet. In der Summe müssen nur solche $N_{i,k}$ berücksichtigt werden, deren Träger das Teilintervall $[t_r, t_{r+1})$ enthalten. Diese sind $N_{r-k+1,k}$ mit dem Träger $[t_{r-k+1}, t_{r+1}]$ bis $N_{r,k}$ mit dem Träger $[t_r, t_{r+k}]$. Also ist

$$\boldsymbol{P}(t) = \sum_{i=r-k+1}^{r} N_{i,k}(t)\boldsymbol{d}_i, \quad t \in [t_r, t_{r+1}). \tag{12.38}$$

Übrigens gilt $\sum\limits_{i=0}^{n} N_{i,k}(t) = 1$, für $t \in [t_r, t_{r+1})$ also $\sum\limits_{i=r-k+1}^{r} N_{i,k}(t) = 1$, d. h. an jeder Stelle dieses Teilintervalls wird das Gesamtgewicht 1 auf die k de Boor-Punkte $\boldsymbol{d}_{r-k+1}$ bis $\boldsymbol{d}_r$ verteilt.

Die Berechnung von $\boldsymbol{P}(t)$ erfolgt zweckmäßig mit dem de Boor-Algorithmus, für dessen Formulierung

$$\alpha_{i,k}^{j} := \frac{t - t_i}{t_{i+k-j} - t_i} \tag{12.39}$$

und

$$\boldsymbol{D}_i^j := \alpha_{i,k}^j \boldsymbol{D}_i^{j-1} + (1 - \alpha_{i,k}^j)\boldsymbol{D}_{i-1}^{j-1}, \qquad \boldsymbol{D}_i^0 := \boldsymbol{d}_i \tag{12.40}$$

für $k \geq 2$, $j \geq 1$ benötigt werden. Mit (12.39) lautet (12.37) für $k \geq 2$

$$N_{i,k}(t) = \alpha_{i,k}^1 N_{i,k-1}(t) + (1 - \alpha_{i+1,k}^1) N_{i+1,k-1}(t). \tag{12.41}$$

Damit ergibt sich für (12.38)

$$\boldsymbol{P}(t) = \sum_{i=r-k+1}^{r} [N_{i,k-1}(t)\alpha_{i,k}^1 \boldsymbol{d}_i + N_{i+1,k-1}(t)(1 - \alpha_{i+1,k}^1)\boldsymbol{d}_i].$$

Wird im zweiten Summanden $i+1$ durch i ersetzt und beachtet, dass für die $N_{i,k-1}$ wegen $t \in [t_r, t_{r+1})$ die Summation ab $i = r-(k-1)+1$ genügt, so folgen

$$\boldsymbol{P}(t) = \sum_{i=r-(k-1)+1}^{r} N_{i,k-1}(t)[\alpha_{i,k}^1 \boldsymbol{d}_i + (1-\alpha_{i,k}^1)\boldsymbol{d}_{i-1}]$$

und mit (12.40) für $j = 1$

$$\boldsymbol{P}(t) = \sum_{i=r-(k-1)+1}^{r} N_{i,k-1}(t)\boldsymbol{D}_i^1.$$

Dieser Prozess lässt sich fortsetzen. Mit $k-j+1$ anstelle von k lautet (12.41)

$$N_{i,k-j+1}(t) = \alpha_{i,k-j+1}^1 N_{i,k-j}(t) + (1-\alpha_{i+1,k-j+1}^1)N_{i+1,k-j}(t)$$

und wegen $\alpha_{i,k-j+1}^1 = \alpha_{i,k}^j$, vgl. (12.39),

$$N_{i,k-j+1}(t) = \alpha_{i,k}^j N_{i,k-j}(t) + (1-\alpha_{i+1,k}^j)N_{i+1,k-j}(t).$$

Damit ergibt sich wie oben im ersten Schritt mit $r-(k-j+1)+1 = r-(k-j)$

$$\begin{aligned}
\boldsymbol{P}(t) &= \sum_{i=r-(k-j)}^{r} N_{i,k-(j-1)}(t)\boldsymbol{D}_i^{j-1} \\
&= \sum_{i=r-(k-j)+1}^{r} N_{i,k-j}(t)[\alpha_{i,k}^j \boldsymbol{D}_i^{j-1} + (1-\alpha_{i,k}^j)\boldsymbol{D}_{i-1}^{j-1}] \\
&= \sum_{i=r-(k-j)+1}^{r} N_{i,k-j}(t)\boldsymbol{D}_i^j.
\end{aligned}$$

Das Verfahren bricht ab, wenn die Summe nur noch einen Summanden enthält, wenn also $r-(k-j)+1 = r$ und somit $j = k-1$ ist. Dann ist wegen (12.37)

$$\boldsymbol{P}(t) = N_{r,k-j}(t)\boldsymbol{D}_r^j = N_{r,1}(t)\boldsymbol{D}_r^{k-1} = \boldsymbol{D}_r^{k-1}.$$

Der de Boor-Algorithmus lässt sich übersichtlich in dem folgenden Schema darstellen.

	$j=1$	$j=2$	$\dots$	$j=k-2$	$j=k-1$
$\boldsymbol{d}_{r-k+1} = \boldsymbol{D}_{r-k+1}^0$					
$\boldsymbol{d}_{r-k+2} = \boldsymbol{D}_{r-k+2}^0$	$\boldsymbol{D}_{r-k+2}^1$				
$\boldsymbol{d}_{r-k+3} = \boldsymbol{D}_{r-k+3}^0$	$\boldsymbol{D}_{r-k+3}^1$	$\boldsymbol{D}_{r-k+3}^2$			
$\vdots$					
$\boldsymbol{d}_{r-1} = \boldsymbol{D}_{r-1}^0$	$\boldsymbol{D}_{r-1}^1$	$\boldsymbol{D}_{r-1}^2$	$\dots$	$\boldsymbol{D}_{r-1}^{k-2}$	
$\boldsymbol{d}_r = \boldsymbol{D}_r^0$	$\boldsymbol{D}_r^1$	$\boldsymbol{D}_r^2$	$\dots$	$\boldsymbol{D}_r^{k-2}$	$\boldsymbol{D}_r^{k-1} = \boldsymbol{P}(t)$ $t \in [t_r, t_{r+1})$

Ein Punkt $\boldsymbol{D}_i^j$ der Spalte j entsteht aus dem links und links oben notierten Punkt $\boldsymbol{D}_i^{j-1}$ bzw. $\boldsymbol{D}_{i-1}^{j-1}$ der Spalte $j-1$ nach (12.40) mit (12.39). Beim horizontalen Fortschreiten wird mit $\alpha_{i,k}^j$, beim absteigenden mit $(1-\alpha_{i,k}^j)$ multipliziert. (Da in den Endformeln die $\alpha_{i,k}^j$ nur mit demselben Index k auftreten, kann auf die Angabe dieses Index verzichtet werden.)

Für den ersten de Boor-Punkt $\boldsymbol{d}_0$ ist $r-k+1=0$, also $r=k-1$, für $\boldsymbol{d}_n$, den letzten, ist $r=n$. Daher können mit dem de Boor-Algorithmus Punkte $\boldsymbol{P}(t)$ mit $t \in [t_r, t_{r+1})$ für $r=(k-1)(1)n$ berechnet werden, d. h. für $t_{k-1} \le t < t_{n+1}$. Da stets k de Boor-Punkte benötigt werden, muss $n+1 \ge k$ sein.

Der de Boor-Algorithmus benutzt für j=1 $\alpha_{r-k+2,k}^1$ bis $\alpha_{r,k}^1$, d. h. wegen (12.39) die Knoten

$$t_{r-k+2}, \ldots, t_r, t_{r+1}, \ldots, t_{r+k-1}.$$

Für $r=k-1$ bis $r=n$ sind das also $t_1, \ldots, t_{k-1}, t_k, \ldots, t_n, t_{n+1}, \ldots, t_{n+k-1}$. Die Randknoten t_0 und t_{n+k} werden demnach nicht benötigt.

Es können sowohl offene, d. h. nicht geschlossene, als auch geschlossene B-Spline-Kurven erzeugt werden. Da diese Kurven glatt sein sollen, wird $k \ge 3$ vorausgesetzt, die Kurven werden also mit mindestens 3 de Boor-Punkten erzeugt.

Offene B-Spline-Kurven

Eine offene B-Spline-Kurve soll ebenso wie eine Bézier-Kurve durch den ersten und letzten de Boor-Punkt $\boldsymbol{d}_0$ bzw. $\boldsymbol{d}_n$ gehen, und die Verbindungsgeraden von $\boldsymbol{d}_0$ und $\boldsymbol{d}_1$ bzw. von $\boldsymbol{d}_{n-1}$ und $\boldsymbol{d}_n$ sollen Tangenten der Kurve sein. Das lässt sich erreichen, indem die ersten und letzten k Knoten einander gleichgesetzt werden. Die Knoten einer offenen B-Spline-Kurve sind also

$$t_0 = t_1 = \ldots = t_{k-1} < t_k < \ldots < t_n < t_{n+1} = t_{n+2} = \ldots = t_{n+k}.$$

Zwischen t_{k-1} und t_{n+1} liegen $n-k+2$ Intervalle, deren Länge von Null verschieden ist. Eine Kurve maximaler Ordnung $k=n+1$ ist eine Bézier-Kurve mit dem Intervall $[t_n, t_{n+1}]$.
Für eine uniforme, offene B-Spline-Kurve der Ordnung $k \ge 3$ können die normierten Knoten wie folgt erzeugt werden:

$$\begin{cases} t_j = k-1 & \text{für} \quad j = 0(1)k-1, \\ t_j = j & \text{für} \quad j = k(1)n, \text{ falls } k \le n \text{ ist}, \\ t_j = n+1 & \text{für} \quad j = n+1(1)n+k. \end{cases} \tag{12.42}$$

Falls beim de Boor-Algorithmus wegen zusammenfallender Knoten ein $\alpha_{i,k}^j$ mit verschwindendem Nenner auftritt, ist $\alpha_{i,k}^j = 0$ zu setzen.

Geschlossene B-Spline-Kurven

Eine geschlossene B-Spline-Kurve wird erzeugt mit

$$\boldsymbol{d}_{n+1} := \boldsymbol{d}_0, \quad \boldsymbol{d}_{n+2} := \boldsymbol{d}_1, \ldots,$$

so dass die de Boor-Punkte zyklisch durchlaufen werden können. Der de Boor-Algorithmus kann dann für insgesamt $n+1$ k-Tupel aufeinander folgender de Boor-Punkte $(\boldsymbol{d}_{r-k+1}, \ldots, \boldsymbol{d}_r)$, $r = k-1(1)n+k-1$, durchgeführt werden:

$$\begin{array}{l}(\boldsymbol{d}_0, \boldsymbol{d}_1, \ldots, \boldsymbol{d}_{k-1}),\\ \quad\vdots\\ (\boldsymbol{d}_{n-k+1}, \ldots, \boldsymbol{d}_{n-1}, \boldsymbol{d}_n),\\ (\boldsymbol{d}_{n-k+2}, \ldots, \boldsymbol{d}_n, \boldsymbol{d}_{n+1}),\\ \quad\vdots\\ (\boldsymbol{d}_n, \boldsymbol{d}_{n+1}, \ldots, \boldsymbol{d}_{n+k-1}).\end{array}$$

Dabei werden die Knoten $t_1, \ldots, t_{n+k-1}, t_{n+k}, \ldots, t_{n+2k-2}$ benötigt, die alle verschieden sein müssen.
Für eine geschlossene B-Spline-Kurve sind den $n+1$ de Boor-Punkten $\boldsymbol{d}_0$, …, $\boldsymbol{d}_n$ also die $k-1$ weiteren Punkte $\boldsymbol{d}_{n+1} = \boldsymbol{d}_0, \boldsymbol{d}_{n+2} = \boldsymbol{d}_1, \ldots, \boldsymbol{d}_{n+k-1} = \boldsymbol{d}_{k-2}$ hinzuzufügen.
Für t mit $t_{k-1} \le t \le t_{n+k}$ ergeben sich alle Punkte $\boldsymbol{P}(t)$ der geschlossenen B-Spline-Kurve. Die Anzahl der Teilintervalle ist $n+1$, unabhängig von k.
Für eine uniforme, geschlossene B-Spline-Kurve der Ordnung $k \ge 3$ sind die normierten Knoten:

$$t_j = j \quad \text{für} \quad j = 0(1)n+2k-1; \tag{12.43}$$

der erste und der letzte Knoten werden nicht benötigt.

Algorithmus 12.8. (*Uniforme B-Spline-Kurve*)

Gegeben: $n+1$ de Boor-Punkte $\boldsymbol{d}_i$, $i = 0(1)n$, $n \ge 2$, $\boldsymbol{d}_i \in \mathbb{R}^2$ oder $\boldsymbol{d}_i \in \mathbb{R}^3$; Ordnung k, $3 \le k \le n+1$; Typ der Kurve: offen oder geschlossen.

Gesucht: Punkte $\boldsymbol{P}(t)$ der uniformen B-Spline-Kurve.

1. Offene Kurve:

   Bereitstellung der Knoten (12.42). Das Intervall $I = [k-1, n+1]$ enthält $n-k+2$ Teilintervalle.

   Geschlossene Kurve:

   Bereitstellung der Knoten (12.43). Das Intervall $I = [k-1, n+k]$ enthält $n+1$ Teilintervalle. Bereitstellung von $k-1$ weiteren de Boor-Punkten $\boldsymbol{d}_{n+1} = \boldsymbol{d}_0, \boldsymbol{d}_{n+2} = \boldsymbol{d}_1, \ldots, \boldsymbol{d}_{n+k-1} = \boldsymbol{d}_{k-2}$.

2. Für $t \in I$ wird das Teilintervall $[r, r+1]$ mit $r \le t < r+1$ ermittelt.

3. de Boor-Algorithmus:

   Für $j = 1(1)k-1$ sind zu berechnen, jeweils für $i = (r-k+j+1)(1)r$

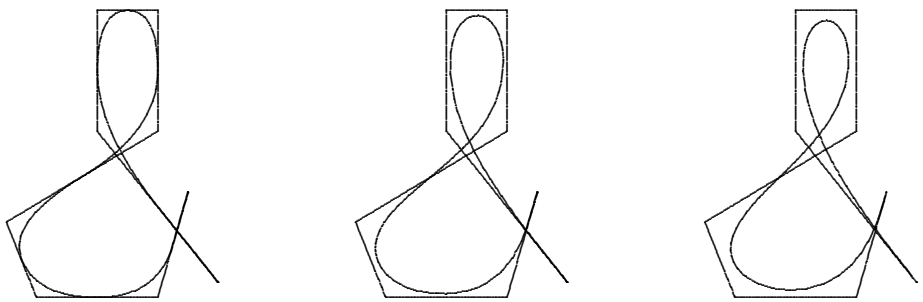
   $$\alpha_i^j = \frac{t-t_i}{t_{i+k-j}-t_i}, \qquad (12.39)$$

   $$\boldsymbol{D}_i^j = \alpha_i^j \boldsymbol{D}_i^{j-1} + (1-\alpha_i^j)\boldsymbol{D}_{i-1}^{j-1}, \quad (12.40)$$

   $$\boldsymbol{D}_i^0 = \boldsymbol{d}_i$$

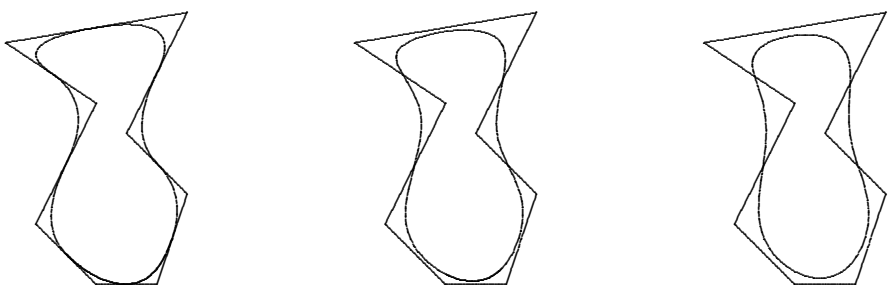
4. $\boldsymbol{P}(t) = \boldsymbol{D}_r^{k-1}$.

Die folgenden Beispiele zeigen offene und geschlossene B-Spline-Kurven der Ordnungen 3, 4 und 5. Die Kurven der Ordnung 3 berühren die Seiten des de Boor-Polygons; mit zunehmender Ordnung entfernen sich die Kurven vom Kontrollpolygon.



a) Ordnung 3 b) Ordnung 4 c) Ordnung 5

Abb. 12.17. Offene B-Spline-Kurven

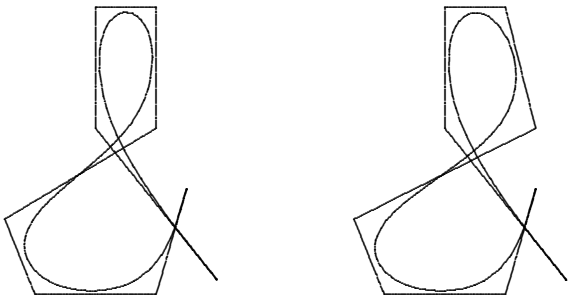


a) Ordnung 3 b) Ordnung 4 c) Ordnung 5

Abb. 12.18. Geschlossene B-Spline-Kurven

Wie der de Boor-Algorithmus erkennen lässt, wird ein Kurvensegment einer B-Spline-Kurve zum Intervall $[t_r, t_{r+1}]$ nur von k de Boor-Punkten beeinflusst. Umgekehrt hat ein de Boor-Punkt $\boldsymbol{d}_i$ nur auf k Intervalle $[t_i, t_{i+1}]$ bis $[t_{i+k-1}, t_{i+k}]$ Einfluss.
Somit wirkt sich die Änderung eines Kontrollpunktes in vorteilhafter Weise nur lokal auf den Kurvenverlauf aus, und zwar umso weniger, je niedriger die Ordnung k ist. Außerdem verläuft eine Kurve niedriger Ordnung in der Nähe des Kontrollpolygons, so dass sich der Kurvenverlauf gut abschätzen lässt. Günstig ist die Verwendung kubischer B-Spline-Kurven der Ordnung 4, die C^2-stetig sind.

Das Verhalten einer offenen B-Spline-Kurve der Ordnung 4 bei Änderung eines Kontrollpunktes zeigt die folgende Abbildung.



a) b)

Abb. 12.19.

Im Gegensatz zu den B-Spline-Kurven sind bei den Bézier-Kurven die Anzahl $n+1$ der Kontrollpunkte $\boldsymbol{b}_i$ und der Grad n der Bernsteinpolynome B_i^n wechselseitig festgelegt. Die kubischen Bézier-Spline-Kurven erfordern die Vorgabe der Gewichtspunkte, deren Einfluss nicht so leicht überschaubar ist.

Ein Vorteil der B-Spline-Kurven ist die bequeme Erzeugung geschlossener Kurven.

12.4.2 B-Spline-Flächen

Analog zu den Bézier-Flächen können mit den B-Spline-Funktionen $N_{i,k}(v)$ und $N_{j,k}(w)$ der Ordnung k sowie den $(m+1)\cdot(n+1)$ Kontrollpunkten $\boldsymbol{d}_{ij} \in \mathbb{R}^3$ Tensor-Produkt-B-Spline-Flächen mit der Darstellung

$$\boldsymbol{P}(v,w) = \sum_{i=0}^{m}\sum_{j=0}^{n} N_{i,k}(v)N_{j,k}(w)\boldsymbol{d}_{ij}, \qquad v \in I_v, w \in I_w$$

erzeugt werden. Die de Boor-Punkte $\boldsymbol{d}_{ij}$ sind die Ecken des de Boor-Polyeders; vgl. das Bézier-Polyeder in Abb. 12.11.

Für $w = w^* = \text{const}$ ergibt sich eine v-Kurve der B-Spline-Fläche

$$\begin{aligned} \boldsymbol{P}(v, w^*) &= \sum_{i=0}^{m} N_{i,k}(v)\Big(\sum_{j=0}^{n} N_{j,k}(w^*)\boldsymbol{d}_{ij}\Big) \\ &= \sum_{i=0}^{m} N_{i,k}(v)\boldsymbol{d}_i(w^*), \quad v \in I_v \ ; \end{aligned} \tag{12.44}$$

sie ist eine B-Spline-Kurve der Ordnung k zu den $m+1$ de Boor-Punkten

$$\boldsymbol{d}_i(w^*) = \sum_{j=0}^{n} N_{j,k}(w^*)\boldsymbol{d}_{ij}, \quad i = 0(1)m \ . \tag{12.45}$$

Ebenso entsteht für $v = v^* = \text{const}$ eine w-Kurve

$$\begin{aligned} \boldsymbol{P}(v^*, w) &= \sum_{j=0}^{n} N_{j,k}(w)\Big(\sum_{i=0}^{m} N_{i,k}(v^*)\boldsymbol{d}_{ij}\Big) \\ &= \sum_{j=0}^{n} N_{j,k}(w)\boldsymbol{d}_j(v^*), \quad w \in I_w \ ; \end{aligned} \tag{12.46}$$

sie ist eine B-Spline-Kurve der Ordnung k zu den $n+1$ de Boor-Punkten

$$\boldsymbol{d}_j(v^*) = \sum_{i=0}^{m} N_{i,k}(v^*)\boldsymbol{d}_{ij}, \quad j = 0(1)n \ .$$

Wie bei den B-Spline-Kurven (vgl. Algorithmus 12.8) sei $3 \le k \le \min(m+1,\, n+1)$. Also muss für die Mindestordnung $k = 3$ ein de Boor-Polyeder mit mindestens 9 Ecken gegeben sein.

Die Ordnungen der v- und w-Kurven können auch unterschiedlich gewählt werden:

$$\begin{aligned} &v\text{-Kurven mit der Ordnung } k_v, \quad 3 \le k_v \le m{+}1, \\ &w\text{-Kurven mit der Ordnung } k_w, \quad 3 \le k_w \le n{+}1. \end{aligned}$$

Hier wird zwecks Vereinfachung $k_v = k_w = k$ gesetzt. Bei den Programmen können k_v und k_w gesondert angegeben werden.

Im Folgenden werden uniforme, offene B-Spline-Flächen betrachtet, deren v- und w-Kurven uniforme, offene B-Spline-Kurven der Ordnung k sind.
Nach (12.42) sind die normierten Knoten für $N_{i,k}(v)$

$$\begin{cases} v_\ell = k-1 & \text{für} \quad \ell = 0(1)k{-}1, \\ v_\ell = \ell & \text{für} \quad \ell = k(1)m, \text{ falls } k \le m \text{ ist}, \\ v_\ell = m+1 & \text{für} \quad \ell = m{+}1(1)m{+}k \end{cases} \tag{12.47}$$

und für $N_{j,k}(w)$

$$\begin{cases} w_\ell = k-1 & \text{für} \quad \ell = 0(1)k{-}1, \\ w_\ell = \ell & \text{für} \quad \ell = k(1)n, \text{ falls } k \le n \text{ ist}, \\ w_\ell = n+1 & \text{für} \quad \ell = n{+}1(1)n{+}k \ . \end{cases} \tag{12.48}$$

Die Intervalle für v und w sind $I_v = [k{-}1, m{+}1]$, $I_w = [k{-}1, n{+}1]$.

Nun werde von einer uniformen, offenen B-Spline-Fläche der Punkt $\boldsymbol{P}(v^*, w^*)$ mit $v^* \in I_v$ und $w^* \in I_w$ berechnet. Dann gibt es $r, s \in \mathbb{N}$, so dass gilt

$$k - 1 \le r \le v^* < r + 1 \le m + 1,$$

$$k - 1 \le s \le w^* < s + 1 \le n + 1.$$

Als Punkt der v-Kurve (12.44) ist wegen $v^* \in [r, r+1)$, vgl. (12.38),

$$\boldsymbol{P}(v^*, w^*) = \sum_{i=r-k+1}^{r} N_{i,k}(v^*)\boldsymbol{d}_i(w^*) \; . \tag{12.49}$$

Es werden also k de Boor-Punkte (12.45) benötigt, die wegen $w^* \in [s, s+1)$ mittels

$$\boldsymbol{d}_i(w^*) = \sum_{j=s-k+1}^{s} N_{j,k}(w^*)\boldsymbol{d}_{ij}, \quad i = r - k + 1(1)r \; , \tag{12.50}$$

berechnet werden können. Sowohl diese k de Boor-Punkte als auch der Flächenpunkt $\boldsymbol{P}(v^*, w^*)$ können mit dem de Boor-Algorithmus ermittelt werden; vgl. (12.38) und Algorithmus 12.8. $\boldsymbol{P}(v^*, w^*)$ kann auch als Punkt der w-Kurve (12.46) berechnet werden:

$$\boldsymbol{P}(v^*, w^*) = \sum_{j=s-k+1}^{s} N_{j,k}(w^*)\boldsymbol{d}_j(v^*) \; , \tag{12.51}$$

$$\boldsymbol{d}(v^*) = \sum_{i=r-k+1}^{r} N_{i,k}(v^*)\boldsymbol{d}_{ij}, \quad j = s - k + 1(1)s \; . \tag{12.52}$$

Auf beiden Wegen sind für $\boldsymbol{P}(v^*, w^*)$ $k+1$ Anwendungen des de Boor-Algorithmus erforderlich.

Algorithmus 12.9. (*Uniforme, offene B-Spline-Fläche*)

Gegeben: $(m+1)\cdot(n+1)$ de Boor-Punkte $\boldsymbol{d}_{ij} \in \mathbb{R}^3$, $i = 0(1)m$, $m \ge 2$, $j = 0(1)n$, $n \ge 2$; Ordnung k, $3 \le k \le \min(m+1, n+1)$.

Gesucht: Punkte $\boldsymbol{P}(v^*, w^*)$ der uniformen, offenen B-Spline-Fläche.

1. Bereitstellung der Knoten (12.47) und (12.48). Intervalle $I_v = [k{-}1, m{+}1]$, $I_w = [k{-}1, n{+}1]$.
2. Für $v^* \in I_v$ wird das Teilintervall $[r, r+1]$ mit $r \le v^* < r+1$, für $w^* \in I_w$ das Teilintervall $[s, s+1]$ mit $s \le w^* < s+1$ ermittelt.
3. $\boldsymbol{P}(v^*, w^*)$ wird als Punkt der v-Kurve $w = w^*$ mit (12.50), (12.49) berechnet.

   3.1 Erzeugung der k de Boor-Punkte $\boldsymbol{d}_i(w^*)$ für $i = (r-k+1)(1)r$:

Für $q = 1(1)k{-}1$ sind zu berechnen, jeweils für
$p = (s - k + q + 1)(1)s$

$$\alpha_p^q = \frac{w^* - w_p}{w_{p+k-q} - w_p},$$

$$\boldsymbol{D}_p^q = \alpha_p^q \boldsymbol{D}_p^{q-1} + (1 - \alpha_p^q)\boldsymbol{D}_{p-1}^{q-1},$$
$$\boldsymbol{D}_p^0 = \boldsymbol{d}_{ip},$$

$$\boldsymbol{d}_i(w^*) = \boldsymbol{D}_s^{k-1}.$$

3.2 Berechnung des Punktes $\boldsymbol{P}(v^*, w^*)$:
Für $q = 1(1)k{-}1$ sind zu berechnen, jeweils für
$p = (r - k + q + 1)(1)r$

$$\alpha_p^q = \frac{v^* - v_p}{v_{p+k-q} - v_p},$$

$$\boldsymbol{D}_p^q = \alpha_p^q \boldsymbol{D}_p^{q-1} + (1 - \alpha_p^q)\boldsymbol{D}_{p-1}^{q-1},$$

$$\boldsymbol{D}_p^0 = \boldsymbol{d}_p(w^*),$$

$$\boldsymbol{P}(v^*, w^*) = \boldsymbol{D}_r^{k-1}.$$

4. $\boldsymbol{P}(v^*, w^*)$ wird als Punkt der w-Kurve $v = v^*$ mit (12.52), (12.51) berechnet.

4.1 Erzeugung der k de Boor-Punkte $\boldsymbol{d}_j(v^*)$ für $j = (s - k + 1)(1)s$:
Für $q = 1(1)k{-}1$ sind zu berechnen, jeweils für
$p = (r - k + q + 1)(1)r$

$$\alpha_p^q = \frac{v^* - v_p}{v_{p+k-q} - v_p},$$

$$\boldsymbol{D}_p^q = \alpha_p^q \boldsymbol{D}_p^{q-1} + (1 - \alpha_p^q)\boldsymbol{D}_{p-1}^{q-1},$$

$$\boldsymbol{D}_p^0 = \boldsymbol{d}_{pj},$$

$$\boldsymbol{d}_j(v^*) = \boldsymbol{D}_r^{k-1}.$$

4.2 Berechnung des Punktes $\boldsymbol{P}(v^*, w^*)$:
Für $q = 1(1)k{-}1$ sind zu berechnen, jeweils für
$p = (s - k + q + 1)(1)s$

$$\alpha_p^q = \frac{w^* - w_p}{w_{p+k-q} - w_p},$$

$$\boldsymbol{D}_p^q = \alpha_p^q \boldsymbol{D}_p^{q-1} + (1 - \alpha_p^q)\boldsymbol{D}_{p-1}^{q-1},$$

$$\boldsymbol{D}_p^0 = \boldsymbol{d}_p(v^*),$$

$$\boldsymbol{P}(v^*, w^*) = \boldsymbol{D}_s^{k-1}.$$

Wenn viele Punkte einer v-Kurve erzeugt werden sollen, ist es zweckmäßig, zunächst alle de Boor-Punkte (12.45) zu ermitteln, um deren wiederholte Berechnung im Algorithmus 12.9 zu vermeiden. Für $v^* \in [r, r+1)$ sind dann unter 3.1 nur die in 3.2 benötigten de Boor-Punkte $\boldsymbol{d}_{r-k+1}(w^*)$ bis $\boldsymbol{d}_r(w^*)$ bereitzustellen. Analog ist vorzugehen, wenn viele Punkte einer w-Kurve zu ermitteln sind.

Ebenso wie bei den B-Spline-Kurven wirken sich Änderungen der de Boor-Punkte $\boldsymbol{d}_{ij}$ nur lokal auf den Verlauf der uniformen B-Spline-Fläche aus. Ein Punkt $\boldsymbol{P}(v^*, w^*)$ mit $v^* \in [r, r+1)$, $w^* \in [s, s+1)$ wird nach (12.49) bis (12.52) mit Hilfe der de Boor-Punkte $\boldsymbol{d}_{ij}$, $i = r-k+1(1)r$, $j = s-k+1(1)s$ berechnet. Die Punkte $\boldsymbol{P}(v^*, w^*)$ des Flächensegmentes $r \le v^* < r+1$, $s \le w^* < s+1$ werden also nur von den genannten $k \cdot k$ de Boor-Punkten beeinflusst. Umgekehrt hat ein de Boor-Punkt $\boldsymbol{d}_{ij}$ Einfluss auf die v-Intervalle $[i, i+1]$ bis $[i+k-1, i+k]$ und auf die w-Intervalle $[j, j+1]$ bis $[j+k-1, j+k]$, also auf $k \cdot k$ Flächensegmente.
Eine rohrförmige uniforme B-Spline-Fläche kann erzeugt werden, wenn z. B. für die v-Kurven offene und für die w-Kurven geschlossene uniforme B-Spline-Kurven der Ordnung k gewählt werden. Für die de Boor-Punkte der geschlossenen w-Kurven sind dann für $i = 0(1)m$ zu setzen:

$$\boldsymbol{d}_{i,n+1} := \boldsymbol{d}_{i,0}, \boldsymbol{d}_{i,n+2} := \boldsymbol{d}_{i,1}, \ldots, \boldsymbol{d}_{i,n+k-1} := \boldsymbol{d}_{i,k-2} \ ,$$

und zu den B-Spline-Funktionen $N_{j,k}(w)$ gehören analog zu (12.43) die Knoten $w_\ell = \ell$ für $\ell = 0(1)n+2k-1$.

Beispiel 12.10.

Eine B-Spline-Fläche wird zusammen mit ihrem Kontrollpolyeder dargestellt, das aus 4 Polygonen mit je 5 Ecken in x-Richtung und 5 Polygonen mit je 4 Ecken in y-Richtung besteht. Die Kurven der B-Spline-Fläche sind nicht geschlossen und sind alle von dritter Ordnung.

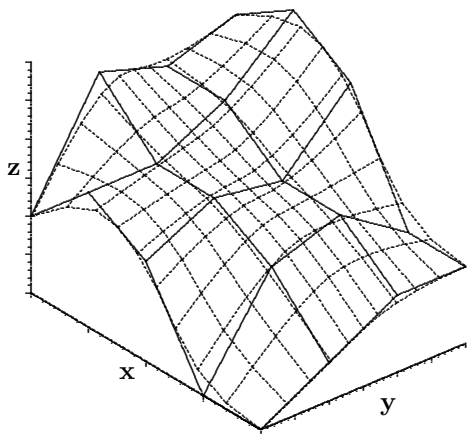


Abb. 12.20. B-Spline-Fläche mit Kontrollpolyeder

□

12.5 Anwendungsbeispiel

Im Brennkammerlabor der FH Aachen wurden Versuche im Zusammenhang mit der Wasserstoffverbrennung in Flugtriebwerken gefahren. Zum Beispiel wurde an Messstellen (x_i, y_i) in der Ebene des Brennkammeraustrittes (Abb. 12.22) die Temperatur z_i gemessen. Um eine Aussage über die Temperaturverteilung machen zu können, wird durch die Messpunkte (x_i, y_i, z_i) ein interpolierender Oberflächenspline der Ableitungsordnung $M = 2$ gelegt.

Versuchsaufbau:

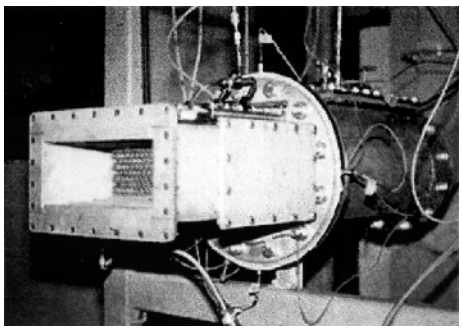


Abb. 12.21. Brennkammer

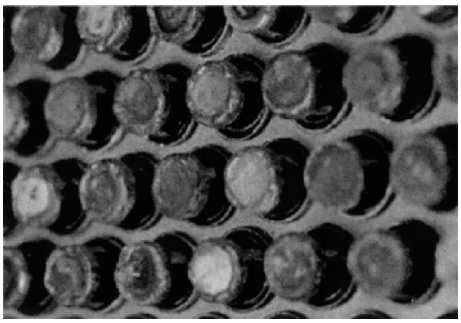


Abb. 12.22. Brennkammeraustritt

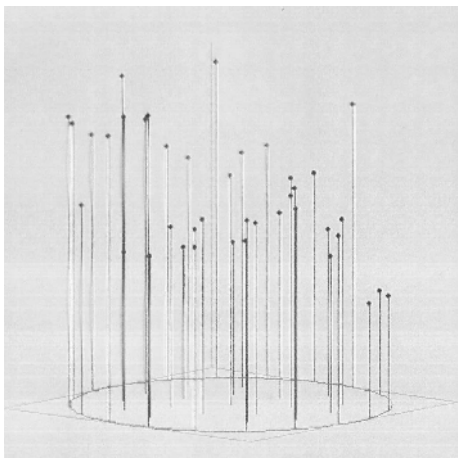


Abb. 12.23. Messpunkte über einer kreisförmigen Grundfläche

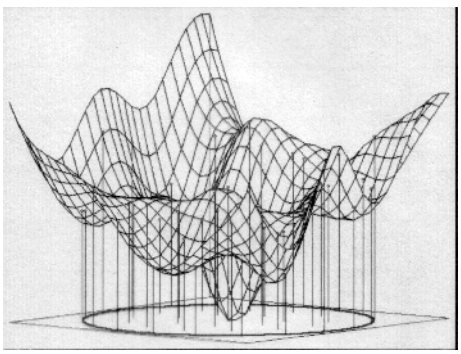


Abb. 12.24. Oberflächenspline ($M = 2$) zu den beliebig angeordneten Messpunkten aus Abb. 12.23

Die Messstellen können bei Verwendung eines Oberflächenspline völlig beliebig angeordnet sein. Für den Einsatz eines bikubischen Spline müssen sie in einem Rechteckgitter liegen.
Den Anwender interessieren nun Linien gleicher Temperatur, d. h. die Höhenlinien $z = \text{const.}$ des Oberflächenspline, und die Bereiche, in denen die Temperatur innerhalb bestimmter Grenzen schwankt. Dazu werden die Höhenlinien berechnet (Abb. 12.25) und in die x, y-Ebene projiziert (Abb. 12.26).

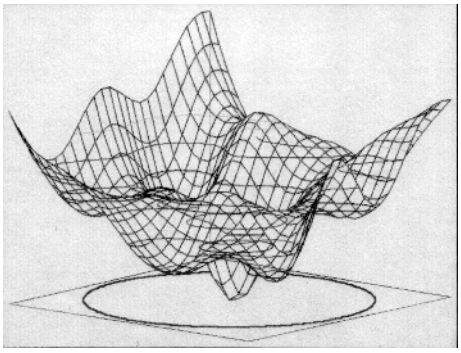


Abb. 12.25. Höhenlinien $z = \text{const.}$ auf dem Oberflächenspline Abb. 12.24

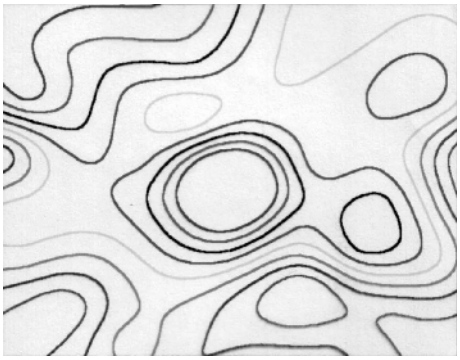


Abb. 12.26. In die Ebene projizierte Höhenlinien von Abb. 12.25

In dieser Ebene (Abb. 12.26) werden dann die Bereiche zwischen den Linien konstanter Temperatur farbig ausgefüllt, um die Temperaturverteilung anschaulich darzustellen; hier ist das natürlich nur in verschiedenen Grauabstufungen möglich.

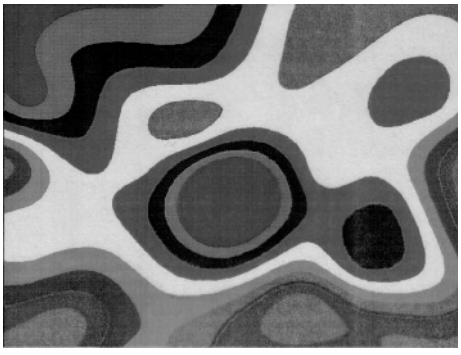


Abb. 12.27. Temperaturbereiche

Bemerkung. Auf der CD-ROM mit den C-Programmen ist ein Kurvenverfolgungsprogramm angegeben, mit dem speziell die Höhenlinien der Oberflächensplines, ihre Projektion in die Ebene und das Ausfüllen der Flächen zwischen den Höhenlinien realisiert werden können. Die Bedienoberfläche von „CurvTrac“ mit dem Modul Höhenlinien ist in Java geschrieben, sie nutzt die C-Programme dieses Buches und ist für Windows und Linux getestet.

Folgende Daten (x_i, y_i, z_i) wurden in einem Versuch mit einer Brennkammer rechteckigen Querschnitts (siehe Bild 12.21) ermittelt. Die Temperaturdaten z_i wurden hier über einem Rechteckgitter mit Mess-Sonden ermittelt. Insofern ist für dieses Beispiel neben den Oberflächensplines auch der Einsatz bikubischer Splines möglich, die im Gegensatz zu den Oberflächensplines an ein Rechteckgitter gebunden sind.

Die folgende Tabelle gibt die $12 \cdot 4 = 48$ Stützstellen (Gitterpunkte) (x_i, y_i) und die zugehörigen Stützwerte (Temperaturen) z_i an:

i	x_i / mm	y_i / mm	z_i / °C	i	x_i / mm	y_i / mm	z_i / °C
1	11.625	13.875	582.0955	25	151.125	13.875	787.9407
2	11.625	41.625	852.1718	26	151.125	41.625	1236.8262
3	11.625	69.375	966.4116	27	151.125	69.375	1280.6493
4	11.625	97.125	582.8441	28	151.125	97.125	1268.7828
5	34.875	13.875	898.5392	29	174.375	13.875	898.4575
6	34.875	41.625	1099.0728	30	174.375	41.625	1112.8422
7	34.875	69.375	1230.4410	31	174.375	69.375	1139.5120
8	34.875	97.125	1205.7226	32	174.375	97.125	1262.8046
9	58.125	13.875	867.5090	33	197.625	13.875	851.7656
10	58.125	41.625	1261.8866	34	197.625	41.625	1198.9421
11	58.125	69.375	1376.0741	35	197.625	69.375	1192.3113
12	58.125	97.125	1205.5225	36	197.625	97.125	1211.9989
13	81.375	13.875	1016.2651	37	220.875	13.875	820.0187
14	81.375	41.625	1340.9722	38	220.875	41.625	1205.3432
15	81.375	69.375	1310.8810	39	220.875	69.375	1153.1227
16	81.375	97.125	1287.4576	40	220.875	97.125	1311.9108
17	104.625	13.875	803.9880	41	244.125	13.875	851.8790
18	104.625	41.625	1286.6032	42	244.125	41.625	1064.6590
19	104.625	69.375	1340.9722	43	244.125	69.375	1085.9858
20	104.625	97.125	1250.0519	44	244.125	97.125	1086.3762
21	127.875	13.875	851.8760	45	267.375	13.875	536.2519
22	127.875	41.625	1340.9275	46	267.375	41.625	811.5963
23	127.875	69.375	1358.7555	47	267.375	69.375	820.0769
24	127.875	97.125	1268.7461	48	267.375	97.125	609.1909

Gesucht: Zu diesen Interpolationsstellen (x_i, y_i, z_i) eine interpolierende Fläche, erzeugt mit

- einer bikubischen Splinefunktion ohne Vorgabe von Randwerten (Abschnitt 12.1)
- zweidimensonalen interpolierenden Oberflächensplines der Ableitungsordnungen $M = 2$, $M = 3$, $M = 4$ (Abschnitt 12.2)
- einer modifizierten kubischen Bézier-Spline-Fläche (Abschnitt 12.3.3)

Lösung:

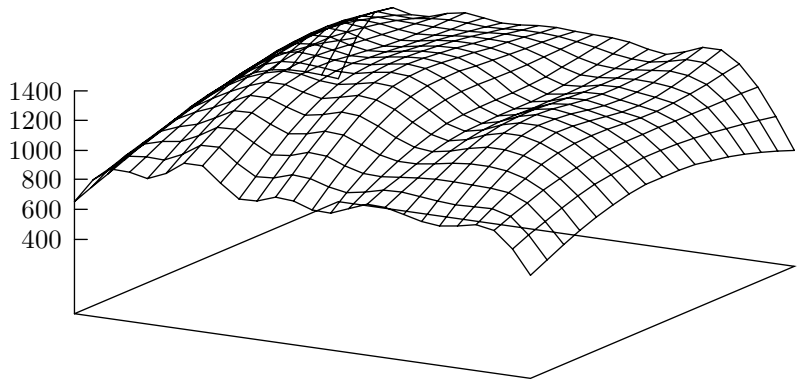


Abb. 12.28. Bikubische Splinefunktion ohne Vorgabe von Randwerten

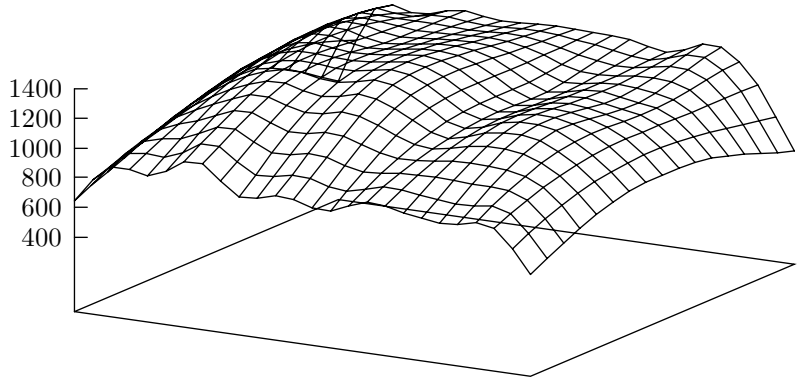


Abb. 12.29. Zweidimensionaler interpolierender Oberflächenspline mit der Ableitungsordnung $M = 2$

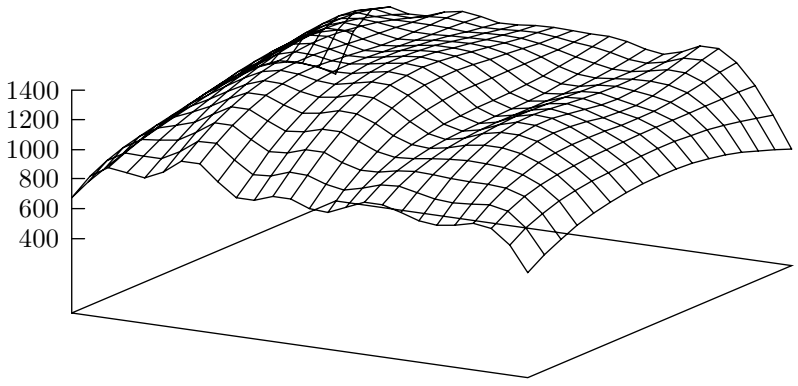


Abb. 12.30. Zweidimensionaler interpolierender Oberflächenspline mit der Ableitungsordnung $M = 3$

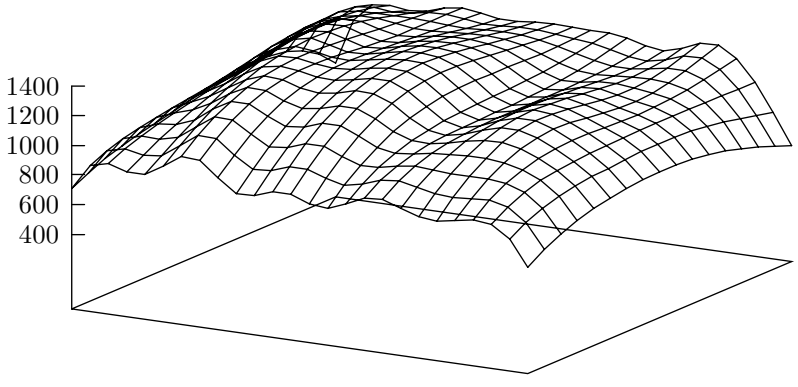


Abb. 12.31. Zweidimensionaler interpolierender Oberflächenspline mit der Ableitungsordnung $M = 4$

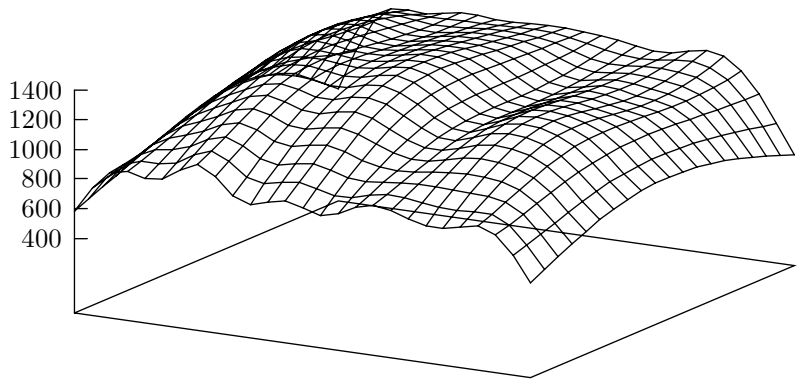


Abb. 12.32. Bézier-Spline-Fläche mit dem modifizierten Verfahren, $\varepsilon = 0.0001$.

□

12.6 Entscheidungshilfen

Bézier- und B-Spline-Kurven

Beide Kurven-Typen sind parametrisch und nicht interpolierend. Bei den Bézier-Splines ist in Abschnitt 12.3.3 eine interpolierende Variante angegeben (modifizierte interpolierende kubische Bézier-Splines). In der Orientierungstabelle für Spline-Kurven in Abschnitt 10.4 sind die kubischen Bézier-Splines enthalten.
Die uniformen B-Spline-Kurven sind gegenüber den Bézier-Spline-Kurven leichter zu handhaben, weil nur das de Boor-Polygon vorgegeben werden muss, das den Kurvenverlauf festlegt. Geschlossene Kurven sollte man nur mit B-Spline-Kurven erzeugen.

Flächendarstellung mit Splines

Bei der Konstruktion glatter Flächen haben die bikubischen Splinefunktionen (Abschnitt 12.1) den Nachteil, dass sie nur dann angewendet werden können, wenn die Funktionswerte auf einem Rechteckgitter mit monoton angeordneten Knoten in x- und y-Richtung gegeben sind; dies ist eine starke Einschränkung.

In der Reihenfolge völlig beliebig können die Wertetripel $\big(x_i,\ y_i,\ z_i = f(x_i,\ y_i)\big)$ für die interpolierenden Oberflächensplines (Abschnitt 12.2) vorgegeben werden; dies ist für den praktischen Einsatz äußerst vorteilhaft. Allerdings muss gewährleistet sein, dass zu jeder Stützstelle $(x_i,\ y_i)$ genau ein Stützwert z_i existiert. Die Ableitungsordnung für die Oberflächensplines ist frei wählbar. Aus Testrechnungen hat sich ergeben, dass im

Allgemeinen die Ableitungsordnungen 3, 4 oder 5 empfehlenswert sind. Bei wachsender Ableitungsordnung und zunehmender Zahl von Interpolationsstellen verschlechtert sich die Kondition der zu lösenden linearen Gleichungssysteme stark.

Bei den Bézier-Splines (Abschnitt 12.3) kann wegen ihrer parametrischen Darstellung die Monotonieforderung entfallen. Sie können deshalb auch zur Darstellung von geschlossenen, sich selbst durchdringenden Flächen benutzt werden, außerdem können bewusst Flächen mit Knick erzeugt werden, und lokale Änderungen sind möglich.

Die Vorteile der B-Spline-Kurven übertragen sich auch auf die B-Spline-Flächen; sie sind den Bézier-Flächen vorzuziehen, weil sie überall mindestens einmal stetig differenzierbar sind (bei Ordnung $k \geq 3$).

Ergänzende Literatur zu Kapitel 12

[BEZI1972]; [BOHM1984]; [DONG1993]; [HAMM1994]; [PIEG1997]; [SPAT1986], 8; [STEU1979].

Kapitel 13

Numerische Differentiation

13.1 Aufgabenstellung und Motivation

Durch Anwendung von Differentiationsregeln kann praktisch jeder Ausdruck aus differenzierbaren Funktionen geschlossen abgeleitet werden. Eine näherungsweise Berechnung der Ableitungen ist nur dann unumgänglich, wenn die zu differenzierende Funktion empirisch gegeben ist.

Es gibt beispielsweise folgende Möglichkeiten, Näherungswerte für die Ableitungen einer differenzierbaren Funktion f zu berechnen:

- Differentiation eines Interpolationspolynoms Φ,
- Differentiation einer interpolierenden Splinefunktion S,
- Differentiation mit dem Romberg-Verfahren (Richardson-Extrapolation),
- Adaptive numerische Differentiation.

Die ersten drei Methoden werden hier besprochen, die adaptiven Verfahren können analog zur Beschreibung der adaptiven Quadraturverfahren (vgl. Abschnitt 14.12) konstruiert werden, siehe dazu [STEP1979], [BJOR1979]. Von den angegebenen Methoden ist die Differentiation eines Interpolationspolynoms am wenigsten zu empfehlen. Insgesamt ist im Vergleich zur numerischen Quadratur eine weitaus geringere Genauigkeit der Ergebnisse zu erwarten.

Für eine empirische Funktion f, deren Werte $f(x_i)$ an nicht äquidistanten Stellen x_i gegeben sind, eignet sich nur die Differentiation mit Hilfe einer interpolierenden Splinefunktion.

13.2 Differentiation mit Hilfe eines Interpolationspolynoms

Gegeben seien Wertepaare $\big(x_i, y_i = f(x_i)\big), x_i \in [a,b]$, $i = 0(1)n$, einer hinreichend oft differenzierbaren Funktion f.
Gesucht sind Näherungswerte für die Ableitungen von f. Dazu interpoliert man f durch ein algebraisches Polynom Φ vom Höchstgrad n, welches anstelle von f an einer beliebigen Stelle $x \in [a,b]$ differenziert wird. Man kann dazu jede Darstellung des eindeutig bestimmten Interpolationspolynoms Φ benutzen. Das Restglied der Differentiation ergibt sich durch Differentiation des Restglieds der Interpolationsformel. Rechnungsfehler wirken sich infolge Auslöschung sicherer Stellen stark aus. Sogar in den Stützstellen x_i, in denen f und das Interpolationspolynom Φ übereinstimmen, kann Φ' stark von f' abweichen. Man muss also mindestens voraussetzen, dass Φ eine gute Polynomapproximation für f ist; trotzdem ist im Allgemeinen die Genauigkeit von Φ' schlechter als die von Φ; man spricht von der „aufrauenden“ Wirkung der Differentiation.

Als Beispiel für eine abgeleitete Interpolationsformel wird die in Abschnitt 9.5 angegebene Newtonsche Interpolationsformel $N_+(t)$ für äquidistante Stützstellen mit $x = x_0 + ht$, $\mathrm{d}x = h\,\mathrm{d}t$ verwendet:

$$\Phi'(x) \equiv N'_+(x) = \frac{1}{h}\widetilde{N}'_+(t) = \frac{1}{h}\Bigg\{\Delta^1_{1/2} + \frac{2t-1}{2!}\Delta^2_1 + \frac{3t^2-6t+2}{3!}\Delta^3_{3/2}$$
$$+\frac{4t^3-18t^2+22t-6}{4!}\Delta^4_2 + \ldots + \frac{1}{n!}\sum_{k=0}^{n-1}\frac{\prod_{i=0}^{n-1}(t-i)}{t-k}\Delta^n_{n/2}\Bigg\}.$$

Analog lässt sich jede andere Interpolationsformel differenzieren.

Die durch Ableitung eines algebraischen Interpolationspolynoms Φ für eine Funktion f erhaltenen Werte $\Phi'(x)$ können wegen der großen Welligkeit von Polynomen höheren Grades erheblich von den Werten $f'(x)$ abweichen. Das veranschaulicht Abb. 13.1. Während die Werte $f(x)$ und $\Phi(x)$ an den Stützstellen übereinstimmen, unterscheiden sich die Tangentensteigungen erheblich. Im Inneren eines Teilintervalls $[x_i, x_{i+1}]$ gibt es eine Stelle ξ_i, so dass $f'(\xi_i)$ durch $\Phi'(\xi_i)$ besser angenähert wird als $f'(x_i)$ durch $\Phi'(x_i)$.

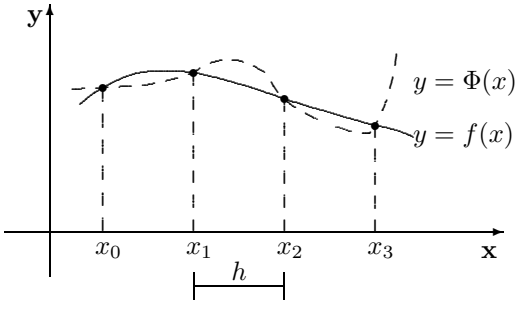


Abb. 13.1.

Vor allem Rechnungsfehler infolge Auslöschung sicherer Stellen bei der Bildung der Differenzen und der gleichzeitigen Division durch h wirken sich stark aus. Einerseits soll h klein sein, um den Verfahrensfehler klein zu halten, andererseits führt ein kleiner Wert von h zu großen Rechnungsfehlern. („Aufrauende“ Wirkung der Differentiation mit Hilfe von Interpolationspolynomen gegenüber der „glättenden“ Wirkung der Interpolationsquadratur, Kapitel 9.) Bei Verkleinerung von h müssen die Werte f_i mit mehr Dezimalstellen zur Verfügung stehen ([STIE1976], S. 127).

Zur Ermittlung einer optimalen Schrittweite h für die Differentiation mit Hilfe von Differenzenquotienten siehe [HERM2001] 8.1.3 und [PREU2001] 7.1. Nach [HERM2001] S. 419 ist für den zentralen Differenzenquotienten $\big(f(x_0+h)-f(x_0-h)\big)/2h$ ein Richtwert für die optimale Schrittweite $h = 10^{-(q/3+1)}$, worin q durch die Maschinengenauigkeit $\varrho \approx 10^{-q}$ gegeben ist.

Eine Fehlerabschätzung mit Hilfe des Restgliedes (vgl. Abschnitt 9.6) der Interpolationsformel ist in der Praxis schwierig, weil dazu die Kenntnis des Wertes einer höheren Ableitung an einer unbekannten Zwischenstelle erforderlich wäre.

Im Anschluss wird eine Tabelle zur näherungsweisen Berechnung der ersten und zweiten Ableitungen an äquidistanten Stützstellen x_i angegeben. Gesucht sind Näherungswerte Y_i', Y_i'' für die Ableitungen $y_i' = f'(x_i)$ und $y_i'' = f''(x_i)$; es gilt

$$\begin{array}{lcll} y_i' & = & Y_i' & + \quad \text{Restglied}\,, \\ y_i'' & = & Y_i'' & + \quad \text{Restglied}\,. \end{array}$$

Die Tabelle gibt die gesuchten Näherungswerte Y_i' bzw. Y_i'' für $i = 0(1)n$, $n = 2(1)6$ bzw. $n = 2, 3, 4$ an. Sie werden über ein Interpolationspolynom gewonnen. Die Anzahl $n+1$ der verwendeten Stützstellen ist in der Tabelle angegeben. Die Restgliedkoeffizienten sind jeweils in den mittleren Stützstellen des Interpolationsintervalls $[x_0, x_n]$ am kleinsten. Es empfiehlt sich daher, wenn genügend Interpolationsstellen vorliegen, diese von Schritt zu Schritt durch Erhöhung des Index i um 1 so umzunummerieren, dass zur Ermittlung von Y_i' bzw. Y_i'' jeweils die Formeln für die mittleren Stützstellen verwendet werden. Doch auch hier wirken sich Rechnungsfehler infolge Auslöschung sicherer Stellen stark aus. Deshalb ist die Ableitung eines Interpolationspolynoms weniger zu empfehlen als die in den folgenden Abschnitten beschriebenen Ableitungen von Splinefunktionen und das Romberg-Verfahren für die Differentiation.

Anzahl d. Interpolationsstellen	Näherungswerte Y_i', Y_i''	Restglied ($\xi, \xi_1, \xi_2 \in [x_0, x_n]$)
3	$Y_0' = \frac{1}{2h}(-3y_0 + 4y_1 - y_2)$	$\frac{h^2}{3} f'''(\xi)$
	$Y_1' = \frac{1}{2h}(-y_0 + y_2)$	$-\frac{h^2}{6} f'''(\xi)$
	$Y_2' = \frac{1}{2h}(y_0 - 4y_1 + 3y_2)$	$\frac{h^2}{3} f'''(\xi)$
	$Y_0'' = \frac{1}{h^2}(y_0 - 2y_1 + y_2)$	$-hf'''(\xi_1) + \frac{h^2}{6} f^{(4)}(\xi_2)$
	$Y_1'' = \frac{1}{h^2}(y_0 - 2y_1 + y_2)$	$-\frac{h^2}{12} f^{(4)}(\xi)$
	$Y_2'' = \frac{1}{h^2}(y_0 - 2y_1 + y_2)$	$hf'''(\xi_1) + \frac{h^2}{6} f^{(4)}(\xi_2)$
4	$Y_0' = \frac{1}{6h}(-11y_0 + 18y_1 - 9y_2 + 2y_3)$	$-\frac{h^3}{4} f^{(4)}(\xi)$
	$Y_1' = \frac{1}{6h}(-2y_0 - 3y_1 + 6y_2 - y_3)$	$\frac{h^3}{12} f^{(4)}(\xi)$
	$Y_2' = \frac{1}{6h}(y_0 - 6y_1 + 3y_2 + 2y_3)$	$-\frac{h^3}{12} f^{(4)}(\xi)$
	$Y_3' = \frac{1}{6h}(-2y_0 + 9y_1 - 18y_2 + 11y_3)$	$\frac{h^3}{4} f^{(4)}(\xi)$
	$Y_0'' = \frac{1}{6h^2}(12y_0 - 30y_1 + 24y_2 - 6y_3)$	$\frac{11}{12} h^2 f^{(4)}(\xi_1) - \frac{h^3}{10} f^{(5)}(\xi_2)$
	$Y_1'' = \frac{1}{6h^2}(6y_0 - 12y_1 + 6y_2)$	$-\frac{h^2}{12} f^{(4)}(\xi_1) + \frac{h^3}{30} f^{(5)}(\xi_2)$
	$Y_2'' = \frac{1}{6h^2}(6y_1 - 12y_2 + 6y_3)$	$-\frac{h^2}{12} f^{(4)}(\xi_1) - \frac{h^3}{30} f^{(5)}(\xi_2)$
	$Y_3'' = \frac{1}{6h^2}(-6y_0 + 24y_1 - 30y_2 + 12y_3)$	$\frac{11}{12} h^2 f^{(4)}(\xi_1) + \frac{h^3}{10} f^{(5)}(\xi_2)$
5	$Y_0' = \frac{1}{12h}(-25y_0 + 48y_1 - 36y_2 + 16y_3 - 3y_4)$	$\frac{h^4}{5} f^{(5)}(\xi)$
	$Y_1' = \frac{1}{12h}(-3y_0 - 10y_1 + 18y_2 - 6y_3 + y_4)$	$-\frac{h^4}{20} f^{(5)}(\xi)$
	$Y_2' = \frac{1}{12h}(y_0 - 8y_1 + 8y_3 - y_4)$	$\frac{h^4}{30} f^{(5)}(\xi)$
	$Y_3' = \frac{1}{12h}(-y_0 + 6y_1 - 18y_2 + 10y_3 + 3y_4)$	$-\frac{h^4}{20} f^{(5)}(\xi)$
	$Y_4' = \frac{1}{12h}(3y_0 - 16y_1 + 36y_2 - 48y_3 + 25y_4)$	$\frac{h^4}{5} f^{(5)}(\xi)$
	$Y_0'' = \frac{1}{24h^2}(70y_0 - 208y_1 + 228y_2 - 112y_3 + 22y_4)$	$-\frac{5}{6} h^3 f^{(5)}(\xi_1) + \frac{h^4}{15} f^{(6)}(\xi_2)$
	$Y_1'' = \frac{1}{24h^2}(22y_0 - 40y_1 + 12y_2 + 8y_3 - 2y_4)$	$\frac{h^3}{12} f^{(5)}(\xi_1) - \frac{h^4}{60} f^{(6)}(\xi_2)$
	$Y_2'' = \frac{1}{24h^2}(-2y_0 + 32y_1 - 60y_2 + 32y_3 - 2y_4)$	$\frac{h^4}{90} f^{(6)}(\xi)$
	$Y_3'' = \frac{1}{24h^2}(-2y_0 + 8y_1 + 12y_2 - 40y_3 + 22y_4)$	$-\frac{h^3}{12} f^{(5)}(\xi_1) - \frac{h^4}{60} f^{(6)}(\xi_2)$
	$Y_4'' = \frac{1}{24h^2}(22y_0 - 112y_1 + 228y_2 - 208y_3 + 70y_4)$	$\frac{5}{6} h^3 f^{(5)}(\xi_1) + \frac{h^4}{15} f^{(6)}(\xi_2)$

Anzahl d. Interpolationsstellen	Näherungswerte Y_i', Y_i''	Restglied ($\xi, \xi_1, \xi_2 \in [x_0, x_n]$)
6	$Y_0' = \frac{1}{60h}(-137y_0 + 300y_1 - 300y_2 + 200y_3 - 75y_4 + 12y_5)$	$-\frac{h^5}{6} f^{(6)}(\xi)$
	$Y_1' = \frac{1}{60h}(-12y_0 - 65y_1 + 120y_2 - 60y_3 + 20y_4 - 3y_5)$	$\frac{h^5}{30} f^{(6)}(\xi)$
	$Y_2' = \frac{1}{60h}(3y_0 - 30y_1 - 20y_2 + 60y_3 - 15y_4 + 2y_5)$	$-\frac{h^5}{60} f^{(6)}(\xi)$
	$Y_3' = \frac{1}{60h}(-2y_0 + 15y_1 - 60y_2 + 20y_3 + 30y_4 - 3y_5)$	$\frac{h^5}{60} f^{(6)}(\xi)$
	$Y_4' = \frac{1}{60h}(3y_0 - 20y_1 + 60y_2 - 120y_3 + 65y_4 + 12y_5)$	$-\frac{h^5}{30} f^{(6)}(\xi)$
	$Y_5' = \frac{1}{60h}(-12y_0 + 75y_1 - 200y_2 + 300y_3 - 300y_4 + 137y_5)$	$\frac{h^5}{6} f^{(6)}(\xi)$
7	$Y_0' = \frac{1}{60h}(-147y_0 + 360y_1 - 450y_2 + 400y_3 - 225y_4 + 72y_5 - 10y_6)$	$\frac{h^6}{7} f^{(7)}(\xi)$
	$Y_1' = \frac{1}{60h}(-10y_0 - 77y_1 + 150y_2 - 100y_3 + 50y_4 - 15y_5 + 2y_6)$	$-\frac{h^6}{42} f^{(7)}(\xi)$
	$Y_2' = \frac{1}{60h}(2y_0 - 24y_1 - 35y_2 + 80y_3 - 30y_4 + 8y_5 - y_6)$	$\frac{h^6}{105} f^{(7)}(\xi)$
	$Y_3' = \frac{1}{60h}(-y_0 + 9y_1 - 45y_2 + 45y_4 - 9y_5 + y_6)$	$-\frac{h^6}{140} f^{(7)}(\xi)$
	$Y_4' = \frac{1}{60h}(y_0 - 8y_1 + 30y_2 - 80y_3 + 35y_4 + 24y_5 - 2y_6)$	$\frac{h^6}{105} f^{(7)}(\xi)$
	$Y_5' = \frac{1}{60h}(-2y_0 + 15y_1 - 50y_2 + 100y_3 - 150y_4 + 77y_5 + 10y_6)$	$-\frac{h^6}{42} f^{(7)}(\xi)$
	$Y_6' = \frac{1}{60h}(10y_0 - 72y_1 + 225y_2 - 400y_3 + 450y_4 - 360y_5 + 147y_6)$	$\frac{h^6}{7} f^{(7)}(\xi)$

13.3 Differentiation mit Hilfe interpolierender kubischer Polynom-Splines

Die angenäherte Differentiation einer Funktion $f : [a, b] \to \mathbb{R}$ mittels der Ableitungen einer interpolierenden Splinefunktion S zu Stützpunkten $\big(x_i, f(x_i)\big)$, $i = 0(1)n$, mit $a = x_0 < x_1 < \ldots < x_n = b$ beruht darauf, dass S gegen f, S' gegen f', und S'' gegen f'' konvergiert, wenn mit wachsender Anzahl der Teilintervalle von $[a, b]$ deren maximale Länge gegen Null strebt (Abschnitt 10.1.9). Das gilt für die Splinefunktion S mit vorgegebenen 1. oder 2. Randableitungen und für die natürliche Splinefunktion. Mit

$$\begin{aligned} S(x) \equiv S_i(x) &= a_i + b_i\,(x - x_i) + c_i\,(x - x_i)^2 + d_i\,(x - x_i)^3, \\ S'(x) \equiv S_i'(x) &= b_i + 2\,c_i\,(x - x_i) + 3\,d_i\,(x - x_i)^2, \\ S''(x) \equiv S_i''(x) &= 2\,c_i + 6\,d_i\,(x - x_i) \\ &\text{für } x \in [x_i, x_{i+1}],\ i = 0(1)n{-}1, \end{aligned}$$

sind S' und S'' Näherungsfunktionen für f' bzw. f''.

Die Genauigkeit der Annäherung für f'' lässt sich erhöhen, wenn man mit den erhaltenen Wertepaaren für $\big(x_i, S'(x_i)\big)$, $i = 0(1)n$, $S'(x_i) \approx f'(x_i)$, eine weitere Spline-Interpolation durchführt und die dann erhaltene Splinefunktion ableitet („spline on spline“, [AHLB1986], S.43 und 49).

Die numerische Differentiation mit Hilfe kubischer Splines lässt im Allgemeinen eine bessere Übereinstimmung von S' und f' erwarten, als sie durch die Differentiation eines Interpolationspolynoms erreicht werden kann.

Beispiel 13.1. ([AHLB1986], S. 48)

Gegeben: Die Funktion $f(x) = \sin x$. Die äquidistanten Knoten $x_i = i\pi/18$, $i = 0(1)9$.

Gesucht: Näherungswerte für $f'(x_i)$ mit Hilfe der natürlichen kubischen Splinefunktion zu den gegebenen Knoten. Es werden hier nur die numerischen Resultate (nicht die Splinefunktion selbst) angegeben.

Lösung:

x_i	$\sin x_i$	$S'(x_i)$	$f'(x_i) = \cos x_i$ auf 5 Dezimalen	max. absoluter Fehler
0	0.00000	1.00005	1.00000	$5 \cdot 10^{-5}$
$\pi/18$	0.17365	0.98478	0.98481	$3 \cdot 10^{-5}$
$2\pi/18$	0.34202	0.93966	0.93969	$3 \cdot 10^{-5}$
$3\pi/18$	0.50000	0.86606	0.86603	$3 \cdot 10^{-5}$
$4\pi/18$	0.64279	0.76603	0.76604	$1 \cdot 10^{-5}$
$5\pi/18$	0.76604	0.64279	0.64279	
$6\pi/18$	0.86603	0.49999	0.50000	$1 \cdot 10^{-5}$
$7\pi/18$	0.93969	0.34201	0.34202	$1 \cdot 10^{-5}$
$8\pi/18$	0.98481	0.17366	0.17365	$1 \cdot 10^{-5}$
$\pi/2$	1.00000	0.00000	0.00000	

□

Beispiel 13.2. ([AHLB1986], S. 49)

Gegeben: Die Funktion $f(x) = \sin x$, $x_i = i\pi/18$, $i = 0(1)9$, $h_i = \pi/18$.

Gesucht: Näherungswerte für $f''(x_i)$ mit Hilfe der periodischen Splinefunktion S.

Lösung:

x_i	$S'(x_i)$	$S''(x_i)$ mit $S'(x_i) = b_i$ $S''(x_i) = 2c_i$	$S''(x_i)$ durch „spline-on-spline“	$f''(x_i) = -\sin x_i$ auf 5 Dezimalen
0	1.00005	0.00000	0.00000	0.00000
$\pi/18$	0.98478	−0.17494	−0.17375	−0.17365
$2\pi/18$	0.93966	−0.34212	−0.34196	−0.34202
$3\pi/18$	0.86606	−0.50119	−0.50002	−0.50000
$4\pi/18$	0.76603	−0.64516	−0.64280	−0.64279
$5\pi/18$	0.64279	−0.76706	−0.76600	−0.76604
$6\pi/18$	0.49999	−0.86924	−0.86594	−0.86603
$7\pi/18$	0.34201	−0.94107	−0.93970	−0.93969
$8\pi/18$	0.17366	−0.98814	−0.98479	−0.98481
$\pi/2$	0.00000	−1.00184	−0.99996	−1.00000

□

Die Tabellen in Beispiel 13.2 sind (unter Beachtung des Vorzeichens) symmetrisch zu $x = \pi/2$ bis zu $x = \pi$ fortzusetzen. Je mehr man sich von den Randpunkten $x_0 = 0$ und $x_n = \pi$ entfernt, umso geringer werden die Fehler.

Wählt man statt $h_i = \pi/18$ die Schrittweite $h_i = \pi/9$, so stellt sich bei der Berechnung von $S'(x_i)$ als maximaler absoluter Fehler der Wert $11{\cdot}10^{-5}$ gegenüber $5{\cdot}10^{-5}$ für $h_i = \pi/18$, also etwa eine Verdoppelung, ein. Die verschiedenen Randbedingungen wirken sich auf $S'(x_i)$ mit Ausnahme der letzten Dezimale eines Wertes nahe x_0 bzw. x_n überhaupt nicht aus.

13.4 Differentiation mit dem Romberg-Verfahren

Gegeben seien eine Funktion $f \in C^{2n}[a,b]$, $n \geq 1$, und eine Schrittweite h, $h > 0$. Gesucht ist an der Stelle $x_0 \in (a,b)$ ein Näherungswert für $f'(x_0)$; x_0 sei so gewählt, dass $a \leq x_0 - h$ und $x_0 + h \leq b$ sind. Wenn f genügend oft stetig differenzierbar ist, sind mit $f(x_0) = f_0$, $f^{(p)}(x_0) = f_0^{(p)}$, $p = 1, 2, \ldots$, sowie mit $\xi_1 \in (x_0, x_0 + h)$, $\xi_2 \in (x_0 - h, x_0)$,

$$f(x_0 + h) = f_0 + hf_0' + \frac{h^2}{2!}f_0'' + \frac{h^3}{3!}f_0''' + \frac{h^4}{4!}f_0^{(4)} + \frac{h^5}{5!}f^{(5)}(\xi_1)\,, \tag{13.1}$$

$$f(x_0 - h) = f_0 - hf_0' + \frac{h^2}{2!}f_0'' - \frac{h^3}{3!}f_0''' + \frac{h^4}{4!}f_0^{(4)} - \frac{h^5}{5!}f^{(5)}(\xi_2)\,. \tag{13.2}$$

Subtrahiert man (13.2) von (13.1), so folgt für den zentralen Differenzenquotienten zur Schrittweite h

$$\frac{f(x_0+h)-f(x_0-h)}{2h} = f'(x_0) + \frac{h^2}{3!}f_0''' + \frac{h^4}{2\cdot 5!}\left(f^{(5)}(\xi_1)+f^{(5)}(\xi_2)\right). \tag{13.3}$$

Der zentrale Differenzenquotient zur Schrittweite h_j wird mit

$$D_j^{(0)}(f) := \frac{f(x_0+h_j)-f(x_0-h_j)}{2h_j} \tag{13.4}$$

bezeichnet. Für (13.3) ergibt sich mit h_j anstelle von h, mit (13.4) und mit $c_2 = -\frac{1}{3!}f_0'''$ kurz

$$f'(x_0) = D_j^{(0)}(f) + c_2 h_j^2 + O(h_j^4). \tag{13.5}$$

Dabei ist c_2 unabhängig von h_j.

Mit Verwendung der höheren Ableitungen von f in (13.1) und (13.2) lässt sich (13.5) in einer allgemeinen Form schreiben mit von h_j unabhängigen c_{2k}:

$$f'(x_0) = D_j^{(0)}(f) + \sum_{k=1}^{n-1} c_{2k} h_j^{2k} + O(h_j^{2n}). \tag{13.6}$$

Im Folgenden wird (13.6) verwendet zu den Schrittweiten $h_j = h/(2^j)$, $j = 0, 1, 2, \ldots$, die also durch fortgesetzte Halbierung von $h_0 = h$ entstehen. Für die Schrittweiten h_j und $h_{j+1} = h_j/2$ lautet (13.6) ausführlich

$$f'(x_0) = D_j^{(0)}(f) + c_2 h_j^2 + c_4 h_j^4 + \ldots + c_{2n-2} h_j^{2n-2} + O\left(h_j^{2n}\right), \tag{13.7}$$

$$\begin{aligned} f'(x_0) = \quad & D_{j+1}^{(0)}(f) + c_2\left(\tfrac{h_j}{2}\right)^2 + c_4\left(\tfrac{h_j}{2}\right)^4 \\ & + \ldots + c_{2n-2}\left(\tfrac{h_j}{2}\right)^{2n-2} + O\left((\tfrac{h_j}{2})^{2n}\right). \end{aligned} \tag{13.8}$$

Um aus diesen Gleichungen $c_2 h_j^2$ zu eliminieren, wird (13.7) von der mit $2^2 = 4$ multiplizierten Gleichung (13.8) subtrahiert. Mit dieser Linearkombination ergibt sich

$$\begin{aligned} f'(x_0) = \quad & \tfrac{1}{3}\big(4D_{j+1}^{(0)}(f) - D_j^{(0)}(f)\big) + c_4^{(1)} h_j^4 \\ & + \ldots + c_{2n-2}^{(1)} h_j^{2n-2} + O(h_j^{2n}) \end{aligned}$$

und mit

$$D_j^{(1)}(f) := \frac{1}{3}\big(4D_{j+1}^{(0)}(f) - D_j^{(0)}(f)\big)$$

$$\begin{aligned} f'(x_0) &= D_j^{(1)}(f) + c_4^{(1)} h_j^4 + \ldots + c_{2n-2}^{(1)} h_j^{2n-2} + O(h_j^{2n}) \\ &= D_j^{(1)}(f) + O(h_j^4); \end{aligned} \tag{13.9}$$

also ist $D_j^{(1)}(f)$ eine Näherung der Fehlerordnung $O(h_j^4)$ für $f'(x_0)$.

Durch eine Linearkombination zweier Näherungen $D_j^{(0)}$ und $D_{j+1}^{(0)}$ für $f'(x_0)$ wurde somit für $f'(x_0)$ eine neue Näherung $D_j^{(1)}$ der Fehlerordnung $O(h_j^4)$ erzeugt.

Dies ist das Prinzip des Romberg-Verfahrens (Richardson-Extrapolation). Das Verfahren lässt sich wie folgt fortsetzen.

Um eine Näherung der Fehlerordnung $O(h_j^6)$ zu gewinnen, wird (13.9) für die Schrittweiten h_j und $h_{j+1} = h_j/2$ notiert

$$\begin{aligned} f'(x_0) = \quad & D_j^{(1)}(f) + c_4^{(1)} h_j^4 + c_6^{(1)} h_j^6 \\ & + \ldots + c_{2n-2}^{(1)} h_j^{2n-2} + O\left(h_j^{2n}\right) , \end{aligned} \tag{13.10}$$

$$\begin{aligned} f'(x_0) = \quad & D_{j+1}^{(1)}(f) + c_4^{(1)} \left(\tfrac{h_j}{2}\right)^4 + c_6^{(1)} \left(\tfrac{h_j}{2}\right)^6 \\ & + \ldots + c_{2n-2}^{(1)} \left(\tfrac{h_j}{2}\right)^{2n-2} + O\left((\tfrac{h_j}{2})^{2n}\right) . \end{aligned} \tag{13.11}$$

Um $c_4^{(1)} h_j^4$ zu eliminieren, wird (13.10) von der mit 2^4 multiplizierten Gleichung (13.11) subtrahiert. Diese Linearkombination ergibt

$$f'(x_0) = \frac{1}{2^4 - 1}\left(2^4 D_{j+1}^{(1)}(f) - D_j^{(1)}(f)\right) + c_6^{(2)} h_j^6 + \ldots + c_{2n-2}^{(2)} h_j^{2n-2} + O(h_j^{2n}) .$$

Mit

$$D_j^{(2)}(f) := \frac{1}{2^4 - 1}\left(2^4 D_{j+1}^{(1)}(f) - D_j^{(1)}(f)\right)$$

folgt

$$f'(x_0) = D_j^{(2)}(f) + O(h_j^6) .$$

Indem man so fortfahrend je einen Näherungswert für $f'(x_0)$ zu den Schrittweiten h_j und $h_j/2$ erstellt und anschließend eine geeignete Linearkombination der beiden Näherungswerte bildet, erhält man durch dieses Zusammenspiel von Schrittweitenhalbierung und Linearkombination mit $2^{2k} = 4^k$ für $f'(x_0)$ die Darstellung

$$f'(x_0) = D_j^{(k)}(f) + O(h_j^{2k+2})$$

mit

$$\begin{aligned} D_j^{(k)}(f) \quad &:= \quad \frac{1}{4^k - 1}\left(4^k D_{j+1}^{(k-1)}(f) - D_j^{(k-1)}(f)\right) \\ &= \quad D_{j+1}^{(k-1)}(f) + \frac{1}{4^k - 1}\left(D_{j+1}^{(k-1)}(f) - D_j^{(k-1)}(f)\right) \end{aligned} \tag{13.12}$$

für $j = 0, 1, 2, \ldots$, $h_j = h/(2^j)$, $k = 1(1)n - 1$; n ergibt sich aus der Voraussetzung $f \in C^{2n}[a, b]$.

Die Rechnung wird zweckmäßig zeilenweise nach dem folgenden Schema durchgeführt:

Rechenschema 13.3. (*Romberg-Verfahren zur numerischen Differentiation*)

$D_j^{(0)}$	$D_j^{(1)}$	$D_j^{(2)}$	$\cdots$	$D_j^{(m-1)}$	$D_j^{(m)}$
$D_0^{(0)}$					
$D_1^{(0)}$	$D_0^{(1)}$				
$D_2^{(0)}$	$D_1^{(1)}$	$D_0^{(2)}$			
$\vdots$	$\vdots$	$\vdots$	$\ddots$		
$D_{m-1}^{(0)}$	$D_{m-2}^{(1)}$	$D_{m-3}^{(2)}$	$\cdots$	$D_0^{(m-1)}$	
$D_m^{(0)}$	$D_{m-1}^{(1)}$	$D_{m-2}^{(2)}$	$\cdots$	$D_1^{(m-1)}$	$D_0^{(m)}$

Dabei werden die $D_j^{(0)}$ nach (13.4), die $D_j^{(k)}$ für $k \geq 1$ nach (13.12) berechnet. Das Schema ist so lange fortzusetzen, bis zu vorgegebenem $\varepsilon > 0$ gilt: $|D_0^{(m)} - D_1^{(m-1)}| < \varepsilon$, sofern $m \leq n$ ist.

Wie auch bei den anderen Verfahren zur näherungsweisen Differentiation wird hier auf eine Fehlerabschätzung und auf Aussagen über die Konvergenzordnung verzichtet. Denn bei zu kleinen Werten von h wird das Resultat durch Rechnungsfehler infolge Auslöschung sicherer Ziffern bei der Bildung der $D_j^{(k)}$ im Rechenschema 13.3 ohnehin verfälscht. Solange die Werte $D_j^{(k)}$ mit wachsendem j sich monoton verhalten, kann man die Rechnung fortsetzen. Wenn die $D_j^{(k)}$ zu oszillieren beginnen, ist die Rechnung abzubrechen, auch wenn die geforderte Genauigkeit nicht erreicht ist. Will man trotzdem eine größere Genauigkeit erreichen, so müssen die Funktionswerte genauer angegeben werden. Ein wesentlicher Vorzug des Verfahrens liegt darin, dass sich durch die fortgesetzte Halbierung der Schrittweite schließlich einmal der Wert h_j einstellt, für den Verfahrensfehler und Rundungsfehler etwa gleich groß werden, vorausgesetzt, dass mit einer genügend großen Schrittweite $h_0 = h$ begonnen wird. Wird h_j noch kleiner, so überwiegen die Rundungsfehler.

Beispiel 13.4. (Fortsetzung von Beispiel 13.1)

Gegeben: Die Funktion $f(x) = \sin x$.

Gesucht: Näherungswerte für $f'(0) = \cos(0) = 1$ und $f'\left(\frac{\pi}{6}\right) = \cos\left(\frac{\pi}{6}\right) = \frac{\sqrt{3}}{2} \approx 0.866\,025$ mit dem Romberg-Verfahren zur Startschrittweite $h_0 = \pi/18$ (6-stellige Mantisse).

Lösung: $\boxed{x_0 = 0}$

$D_j^{(0)}$	$D_j^{(1)}$	$D_j^{(2)}$	$D_j^{(3)}$
$D_0^{(0)} = 0.994931$			
$D_1^{(0)} = 0.998731$	$D_0^{(1)} = 0.999998$		
$D_2^{(0)} = 0.999683$	$D_1^{(1)} = 1.00000$	$D_0^{(2)} = 1.00000$	
$D_3^{(0)} = 0.999921$	$D_2^{(1)} = 1.00000$	$D_1^{(2)} = 1.00000$	$D_0^{(3)} = 1.00000$

$\boxed{x_0 = \pi/6}$

$D_j^{(0)}$	$D_j^{(1)}$	$D_j^{(2)}$	$D_j^{(3)}$
$D_0^{(0)} = 0.861635$			
$D_1^{(0)} = 0.864927$	$D_0^{(1)} = 0.866024$		
$D_2^{(0)} = 0.865751$	$D_1^{(1)} = 0.866025$	$D_0^{(2)} = 0.866025$	
$D_3^{(0)} = 0.865957$	$D_2^{(1)} = 0.866025$	$D_1^{(2)} = 0.866025$	$D_0^{(3)} = 0.866025$

Nach drei Romberg-Schritten wird dieselbe Genauigkeit erreicht wie bei Anwendung der Splinefunktion (Beispiel 13.1). Für $x_0 = \pi/6$ ist die Genauigkeit sogar noch größer. □

13.5 Entscheidungshilfen

Wegen der „aufrauenden“ Wirkung der Differentiation sind hier bei weitem nicht so gute Ergebnisse zu erwarten wie bei der numerischen Quadratur. Die Berechnung von Ableitungen über Splines ist der über Interpolationspolynome vorzuziehen. Lassen sich die Funktionswerte an den für das Romberg-Verfahren erforderlichen Stützstellen berechnen, so ist dieses Verfahren einzusetzen. Es ist empfehlenswert, hier analog zur Adaption bei der Quadratur (vgl. Abschnitt 14.12) vorzugehen, um die Anzahl der erforderlichen Funktionsauswertungen möglichst klein zu halten.

Ergänzende Literatur zu Kapitel 13

[HAMM1978], 13; [HERM2001]; [KNOR2003], 7.1; [PREU2001]; [RALS1979] Bd.II, 8.2; [STOE1989], 2.4.3; [STUM1982], 3.3-4; [WERN1993], III §5, §8.

Kapitel 14

Numerische Quadratur

14.1 Vorbemerkungen

Jede auf einem Intervall I_x stetige Funktion f besitzt dort Stammfunktionen F, die sich nur durch eine additive Konstante unterscheiden, mit

$$\frac{\mathrm{d}}{\mathrm{d}x} F(x) = F'(x) = f(x), \qquad x \in I_x.$$

Die Zahl $I(f;\alpha,\beta)$ heißt das *bestimmte Integral* der Funktion f über $[\alpha,\beta]$; es gilt der *Hauptsatz der Integralrechnung*

$$I(f;\alpha,\beta) := \int_\alpha^\beta f(x)\,\mathrm{d}x = F(\beta) - F(\alpha), \quad [\alpha,\beta] \subset I_x,$$

f heißt *integrierbar* auf $[\alpha,\beta]$.

In der Praxis ist man in den meisten Fällen auf eine näherungsweise Berechnung bestimmter Integrale $I(f;\alpha,\beta)$ mit Hilfe sogenannter *Quadraturformeln* Q angewiesen. Die Ursachen dafür können sein:

1. f hat eine Stammfunktion F, die nicht in geschlossener (integralfreier) Form darstellbar ist (z. B. $f(x) = (\sin x)/x$, $f(x) = \mathrm{e}^{-x^2}$, $f(x) = \sqrt{1 - k^2 \sin^2 x}$ mit $0 < k^2 < 1$).
2. f ist nur an diskreten Stellen $x_k \in [\alpha,\beta]$ bekannt.
3. F ist in geschlossener Form darstellbar, jedoch ist die Ermittlung von F oder auch die Berechnung von $F(\alpha)$ und $F(\beta)$ mit Aufwand verbunden.

Beispiel 14.1. (zu 1.)

Ein sogenanntes elliptisches Integral 2. Gattung

$$\int_0^{\pi/2} \sqrt{1 - k^2 \sin^2 t}\,\mathrm{d}t\,, \quad 0 < k^2 < 1\,,$$

das nicht elementar integrierbar ist, ergibt sich bei der Berechnung des Umfanges U einer Ellipse mit der Parameterdarstellung

$$x = b\cos t\,, \quad y = a\sin t\,, \quad a > b > 0\,.$$

Es gilt nämlich

$$\begin{aligned} U &= 4 \int_0^{\pi/2} \sqrt{\dot{x}^2(t) + \dot{y}^2(t)}\,\mathrm{d}t\,, \quad \dot{x} := \tfrac{\mathrm{d}x}{\mathrm{d}t} \\ &= 4 \int_0^{\pi/2} \sqrt{a^2\cos^2 t + b^2\sin^2 t}\,\mathrm{d}t \\ &= 4a \int_0^{\pi/2} \sqrt{\cos^2 t + \tfrac{b^2}{a^2}\sin^2 t}\,\mathrm{d}t \\ &= 4a \int_0^{\pi/2} \sqrt{1 - \tfrac{a^2-b^2}{a^2}\sin^2 t}\,\mathrm{d}t \\ &= 4a \int_0^{\pi/2} \sqrt{1 - k^2\sin^2 t}\,\mathrm{d}t \quad \text{mit} \quad k^2 = \tfrac{a^2-b^2}{a^2} \end{aligned}$$

und $0 < k^2 < 1$ wegen $a > b$. □

Beispiel 14.2. (zu 2. Spritzgießen)

Beim Spritzgussverfahren in der Kunststofftechnik werden Thermoplaste in eine gekühlte Form gespritzt. Das heiße Formteil bleibt so lange im Werkzeug, bis es eine Temperatur erreicht hat, bei der es nach der Entformung nicht mehr deformiert werden kann. Nimmt man beispielsweise als Form eine beidseitig gekühlte Platte, bei der beide Oberflächen die konstante Wandtemperatur ϑ_W (Werkzeugwandtemperatur) besitzen, so ergeben sich sinusförmige Temperaturprofile $\vartheta_M(x)$ (Massetemperatur in Abhängigkeit vom Ort zur Zeit t) durch Integration der Differentialgleichung für die Wärmeleitung $\vartheta_t = a\Delta\vartheta$ (a = Temperaturleitfähigkeit) $\left(\vartheta_t := \frac{\partial}{\partial t}\vartheta\,,\ \Delta := \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right)$. Löst man die DGL etwa mit dem Differenzenverfahren, so erhält man die Temperaturprofile $\vartheta_M(x)$ in Form einer Wertetabelle zu diskreten Stellen x_k.

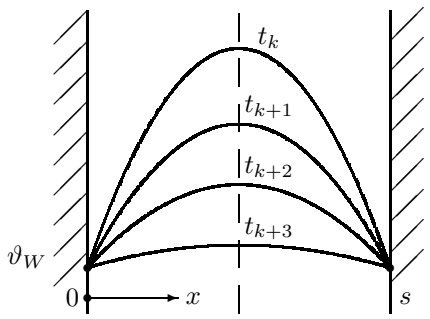


Abb. 14.1. Temperaturprofile beim Spritzgussverfahren

Gesucht ist nun die mittlere Massetemperatur $\overline{\vartheta}_M$ über dem Formteilquerschnitt

$$\overline{\vartheta}_M = \frac{1}{s} \int_0^s \vartheta_M(x)\,\mathrm{d}x\,.$$

Da $\vartheta_M(x)$ nur empirisch vorliegt, muss numerisch integriert werden mit einer Quadraturformel, die die Funktionswerte an den Stellen x_k verwendet. □

In allen Fällen muss die vorliegende Aufgabe der Analysis, die nur mit Funktionen einer stetigen Veränderlichen arbeitet, durch eine approximierende Aufgabe (Ersatzproblem) der numerischen Mathematik ersetzt werden.

Mögliche Ersatzprobleme für die Integration sind z. B. Linearkombinationen Q aus Funktionswerten $f(x_k)$ des Integranden an diskreten *Stützstellen* (Knoten) x_k des Integrationsintervalls $[\alpha,\beta]$ mit *Gewichten* A_k

$$Q(f;\alpha,\beta) = \sum_k A_k f(x_k) \approx \int_\alpha^\beta f(x)\,\mathrm{d}x, \quad x_k \in [\alpha,\beta]$$

oder entsprechende Linearkombinationen aus Funktionswerten und Ableitungswerten von f an den Stellen x_k. Die Quadraturformeln Q liefern Näherungswerte für das bestimmte Integral $I(f;\alpha,\beta)$. Zum Beispiel ist

$$Q^R(f;a,b) = (b-a)f(a) \approx \int_a^b f(x)\,\mathrm{d}x$$

die sogenannte *Rechteckformel*, konstruiert für das sogenannte *Referenzintervall* $[a,b]$; es könnte als Referenzintervall z. B. auch $[-1,1]$, $[0,h]$, $[-h,h]$ gewählt werden. Will man mit Hilfe der Rechteckformel einen Näherungswert für $I(f;\alpha,\beta)$ ermitteln, so zerlegt man das Integrationsintervall $[\alpha,\beta]$ mit

$$Z : \alpha = t_0 < t_1 < t_2 < \ldots < t_N = \beta$$

in Teilintervalle $[t_k,t_{k+1}]$ der Länge $h_k := t_{k+1} - t_k$. Z heißt *Zerlegung* des Integrationsintervalls. Wegen

$$\int_\alpha^\beta f(x)\,\mathrm{d}x = \int_{t_0}^{t_1} f(x)\,\mathrm{d}x + \int_{t_1}^{t_2} f(x)\,\mathrm{d}x + \ldots + \int_{t_{N-1}}^{t_N} f(x)\,\mathrm{d}x$$

wendet man auf jedes Teilintervall $[t_k,t_{k+1}]$ die auf dieses Intervall als Referenzintervall transformierte Rechteckformel Q^R an. Man erhält so als Näherungswert für $I(f;\alpha,\beta)$ die *summierte Rechteckformel*

$$Q^R_{h_k}(f;\alpha,\beta) = \sum_{k=0}^{N-1} (t_{k+1} - t_k) f(t_k) \approx \int_\alpha^\beta f(x)\,\mathrm{d}x.$$

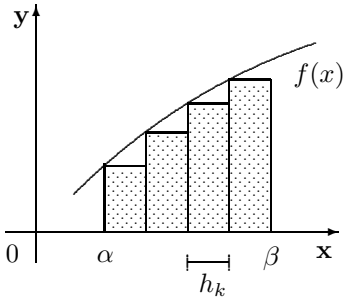


Abb. 14.2. Zusammengesetzte (summierte) Rechteckformel

Eine so aus einer für ein Referenzintervall konstruierten Quadraturformel zusammengesetzte Formel heißt *summierte* oder *zusammengesetzte Quadraturformel*.

Die Differenz aus Integralwert und Quadraturformel für das Referenzintervall liefert den sogenannten *lokalen Quadraturfehler*, die Differenz aus Integralwert und summierter Quadraturformel den *globalen Quadraturfehler*. Bei geeigneter Zerlegung des Integrationsintervalls kann der Fehler beliebig klein gemacht werden. Insofern könnte die Kenntnis *einer* Quadraturformel theoretisch ausreichen. Da aber gleichzeitig der Rechenaufwand minimiert werden soll, werden im Folgenden verschiedene Quadraturformeln angegeben. Eine feinere Unterteilung würde mehr Funktionsauswertungen erfordern. Deshalb muss man zur Minimierung des Rechenaufwandes optimale Gewichte A_k und geeignete Knoten x_k wählen, so dass $Q(f;\alpha,\beta) = \sum\limits_k A_k\, f(x_k)$ das Integral $\int\limits_\alpha^\beta f(x)\,\mathrm{d}x$ schon mit einer geringen Anzahl von Summanden gut approximiert. Entscheidungshilfen für die Auswahl der geeigneten Methode sind in Abschnitt 14.15 gegeben.

14.2 Konstruktion von Interpolationsquadraturformeln

Es sollen nun Quadraturformeln Q für ein Referenzintervall $[a,b]$ konstruiert werden, die Polynome möglichst hohen Grades exakt integrieren. Von dem Integranden f eines bestimmten Integrals $I(f;a,b)$ seien an $n+1$ paarweise verschiedenen und nicht notwendig äquidistanten Stützstellen (Knoten) x_k, $k = 0(1)n$, des Referenzintervalles $[a,b]$, die Stützwerte $y_k = f(x_k)$ bekannt.

Dann liegt es nahe, durch die $n+1$ Stützpunkte $\big(x_k,\, y_k = f(x_k)\big)$ das zugehörige Interpolationspolynom Φ vom Höchstgrad n zu legen und das bestimmte Integral von Φ über $[a,b]$: $I(\Phi;a,b)$ als Näherungswert für das gesuchte Integral $I(f;a,b)$ zu benutzen. Mit dem Restglied $R(x)$ der Interpolation gilt

$$f(x) = \Phi(x) + R(x), \qquad x \in [a,b].$$

Für das Integral $I(f;a,b)$ erhält man somit

$$\left\{\begin{array}{lcl} I(f;a,b) & = & \int\limits_a^b f(x)\,\mathrm{d}x = Q(f;a,b) + E(f;a,b) \quad \text{mit} \\ Q(f;a,b) & = & I(\Phi;a,b) = \int\limits_a^b \Phi(x)\,\mathrm{d}x, \\ E(f;a,b) & = & I(R;a,b) = \int\limits_a^b R(x)\,\mathrm{d}x. \end{array}\right. \tag{14.1}$$

Nach Ausführung der Integration über Φ bzw. R liefert $Q(f;a,b)$ die Quadraturformel und $E(f;a,b)$ das zugehörige *Restglied der Quadratur*. Die Summe aus Q und E wird als *Integrationsregel* bezeichnet. Für das Restglied gilt mit (14.1)

$$E(f;a,b) = \int_a^b R(x)\,\mathrm{d}x = \int_a^b \big(f(x) - \Phi(x)\big)\,\mathrm{d}x.$$

Falls also $f - \Phi$ in $[a,b]$ das Vorzeichen mehrfach wechselt, heben sich positive und negative Fehler teilweise auf, so dass der resultierende Fehler selbst dann klein werden kann, wenn Φ *keine* gute Approximation von f darstellt, d. h. zwischen den Stützstellen stark von f abweicht. Durch Integration werden also Fehler geglättet.

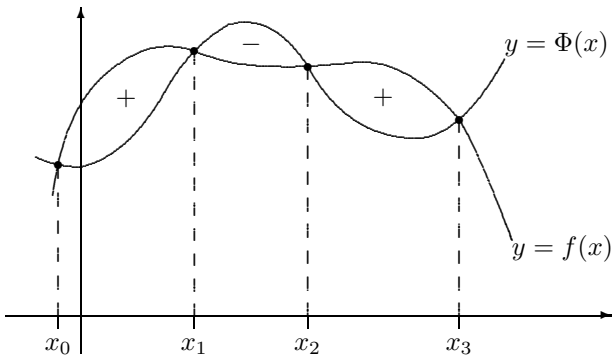


Abb. 14.3. Aufhebung der Fehler

Benutzt man die Interpolationsformel von Lagrange (3) mit

$$\Phi(x) = L(x) = \sum_{k=0}^{n} L_k(x_k)\, f(x_k)\,,$$

so erhält man für $Q(f;a,b)$ die Darstellung

$$Q(f;a,b) = \int_a^b L(x)\,\mathrm{d}x = \sum_{k=0}^{n} f(x_k) \int_a^b L_k(x)\,\mathrm{d}x$$

und mit der Abkürzung

$$A_k := \int_a^b L_k(x)\,\mathrm{d}x\,, \qquad k = 0(1)n\,, \tag{14.2}$$

die Darstellung

$$Q(f;a,b) = \sum_{k=0}^{n} A_k f(x_k) = A_0 f(x_0) + A_1 f(x_1) + \ldots + A_n f(x_n). \tag{14.3}$$

Die Koeffizienten A_k heißen *Gewichte* der Quadraturformel; sie hängen von den gegebenen Stützstellen x_k und dem Referenzintervall $[a,b]$ ab. Die Integrale in (14.2) können stets berechnet werden, denn die Integranden L_k sind Polynome, deren Stammfunktionen in geschlossener Form darstellbar sind. Die Aufgabe, das Integral $I(f;a,b)$ zu bestimmen, ist also zurückgeführt worden auf die Berechnung von elementar ausführbaren Integralen $I(L_k;a,b)$.

Aus der Forderung, dass die Quadraturformel (14.3) auf dem Referenzintervall $[a,b]$ Polynome P_m bis zum Grade $M \geq n$ exakt integrieren soll

$$\int_a^b P_m(x)\,\mathrm{d}x \;\stackrel{!}{=}\; Q(P_m;a,b)$$

ergibt sich mit $P_m(x) = x^m$

$$\begin{aligned} \int_a^b x^m\,\mathrm{d}x &= \tfrac{1}{m+1}\,(b^{m+1} - a^{m+1}) \\ Q(x^m;a,b) &= \sum_{k=0}^{n} A_k\, {x_k}^m\,. \end{aligned}$$

Man erhält so das lineare Gleichungssystem

$$\sum_{k=0}^{n} A_k x_k^m = \frac{1}{m+1}\,(b^{m+1} - a^{m+1})\,, \quad m = 0(1)M\,, \tag{14.4}$$

welches linear in den A_k und nichtlinear in den x_k für $k = 0(1)n$ ist. Je nachdem, ob die x_k oder die A_k oder weder die x_k noch die A_k vorgegeben werden, erhält man bestimmte Arten von interpolatorischen Quadraturformeln unterschiedlicher Konvergenzordnung.

Interpolationsquadraturformeln (14.3) können für Polynome höchstens bis zum Grad $M = 2n+1$ exakt sein (Beweis siehe [NIED1987], 11.3). Dies ist genau dann der Fall, wenn in (14.4) die x_k und die A_k frei sind, d. h. $2n+2$ Größen zu berechnen sind. Dazu werden $2n+2$ Bedingungen benötigt, die man für $m = 0(1)2n+1$ erhält.

Lokale und globale Fehlerordnung

Ist eine für ein Referenzintervall $[a,b]$ konstruierte Quadraturformel Q exakt für alle Polynome vom Grad M und gilt $f \in C^{M+1}\,[a,b]$, so beträgt die lokale Fehlerordnung $q_\ell = M+2$.

Ist $f \in C^{M+1}\,[\alpha,\beta]$ und ist $h_{\max}$ die Länge des größten Teilintervalls der Zerlegung Z von $[\alpha,\beta]$, so beträgt die Fehlerordnung der aus Q zusammengesetzten (summierten) Quadraturformel $O(h_{\max}^{M+1})$, d. h. die globale Fehlerordnung ist $q_g = M+1$.

Sind a, b und sämtliche Stützstellen x_k, $k = 0(1)n$, im Referenzintervall $[a, b]$ gegeben, so ist (14.4) ein lineares Gleichungssystem von $n + 1$ Gleichungen für die $n + 1$ Gewichte A_k; es ergeben sich die *Newton-Cotes-Formeln*, die für eine gerade Anzahl $(n + 1)$ von Stützstellen Polynome bis zum n-ten Grade exakt integrieren (d. h. der Exaktheitsgrad $M = n$, die lokale Fehlerordnung $q_\ell = n + 2$, die globale Fehlerordnung $q_g = n + 1$) und für ungerades $(n + 1)$ Polynome bis zum $(n + 1)$-ten Grade (d. h. $M = n + 1$, $q_\ell = n + 3$, $q_g = n + 2$).

Sind a, b mit $a = -h$, $b = h$ und sämtliche Gewichte A_k mit $A_k = 2h/(n+1)$ vorgegeben, so ist (14.4) ein nichtlineares Gleichungssystem für die Stützstellen x_k; man erhält die *Tschebyscheffschen Quadraturformeln*, die Polynome bis zum $(n + 1)$-ten Grade exakt integrieren (d. h. $M = n + 1$, $q_\ell = n + 3$, $q_g = n + 2$).

Schreibt man für ein Referenzintervall weder die Stützstellen x_k noch die Gewichte A_k vor und fordert, dass die Quadraturformel (14.3) Polynome bis zum $(2n + 1)$-ten Grade exakt integriert, so ist (14.4) ein nichtlineares Gleichungssystem für die x_k und A_k, und es ergeben sich die *optimalen Gauß-Formeln* (d. h. $M = 2n+1$, $q_\ell = 2n+3$, $q_g = 2n+2$).

Die genannten und weitere Quadraturverfahren werden in den folgenden Abschnitten angegeben.

14.3 Newton-Cotes-Formeln

Mit Hilfe des linearen Gleichungssystems (14.4) lassen sich für ein Referenzintervall $[a, b]$ spezielle *Quadraturformeln für äquidistante Stützstellen* aufstellen. Die Randpunkte des Referenzintervalls $[a, b]$ fallen dabei jeweils mit Stützstellen des zu integrierenden Interpolationspolynoms zusammen. So konstruierte Formeln gehören zur Klasse der Newton-Cotes-Formeln.
Man erhält die $n + 1$ Gewichte A_k der Newton-Cotes-Formeln bei vorgegebenen x_k aus (14.4) für $m = 0(1)n$. Das lineare Gleichungssystem lautet ausgeschrieben

$$\begin{pmatrix} 1 & 1 & 1 & & 1 \\ x_0 & x_1 & x_2 & \cdots & x_n \\ x_0^2 & x_1^2 & x_2^2 & \cdots & x_n^2 \\ \vdots & \vdots & \vdots & & \vdots \\ x_0^n & x_1^n & x_2^n & \cdots & x_n^n \end{pmatrix} \begin{pmatrix} A_0 \\ A_1 \\ A_2 \\ \vdots \\ A_n \end{pmatrix} = \begin{pmatrix} b-a \\ \frac{1}{2}(b^2-a^2) \\ \frac{1}{3}(b^3-a^3) \\ \vdots \\ \frac{1}{n+1}(b^{n+1}-a^{n+1}) \end{pmatrix} \tag{14.5}$$

Die Matrix des Systems (14.5) heißt *Vandermonde-Matrix*. Für paarweise verschiedene Stützstellen x_k ist ihre Determinante verschieden von Null und das System (14.4) bzw. (14.5) eindeutig lösbar, d. h. die Gewichte A_k sind durch die Vorgabe der Stützstellen x_k eindeutig bestimmt.

Mit dieser Methode kann also zu einem gegebenen Referenzintervall $[a,b]$ und $n+1$ gegebenen paarweise verschiedenen und nicht notwendig äquidistanten Stützstellen $x_k \in [a,b]$ jeweils eine Interpolationsquadraturformel hergeleitet werden. Sind dann an diesen Stützstellen $x_k \in [a,b]$ Funktionswerte $f(x_k)$ einer über $[a,b]$ zu integrierenden Funktion f bekannt, so liefert die Quadraturformel (14.3) einen Näherungswert für das Integral $I(f;a,b)$.

Mit dem Restglied der Interpolation (12) und $f \in C^{n+1}\,[a,b]$ erhält man für $E(f;a,b)$ die Darstellung

$$E(f;a,b) \;=\; \frac{1}{(n+1)!}\int\limits_a^b f^{(n+1)}(\xi)\;\pi(x)\,\mathrm{d}x \quad \text{mit}$$
$$\xi = \xi(x,x_0,x_1,\ldots,x_n) \in [a,b], \quad \pi(x) = (x-x_0)(x-x_1)\ldots(x-x_n)$$

bzw. nach dem verallgemeinerten Mittelwertsatz der Integralrechnung

$$E(f;a,b) = \frac{1}{(n+1)!} f^{(n+1)}(\xi^*) \int\limits_a^b \pi(x)\,\mathrm{d}x, \quad \xi^* \in [a,b]\,, \tag{14.6}$$

falls überall in $[a,b]$ gilt $\pi(x) \geq 0$ oder $\pi(x) \leq 0$, d. h. $\pi(x)$ in $[a,b]$ nicht das Vorzeichen wechselt. Im ganzen Intervall $[a,b]$ gilt $\pi(x) \geq 0$ (bzw. $\pi(x) \leq 0$) aber nur dann, wenn kein Knoten x_k in (a,b), d. h. die Knoten mit den Randpunkten a, b des Referenzintervalles zusammenfallen. Für alle übrigen muss das Restglied $E(f;a,b)$ auf andere Weise als mit dem Mittelwertsatz der Integralrechnung bestimmt werden; eine solche Herleitung wird für die Simpsonsche Formel (siehe Abschnitt 14.3.2) und die Tangententrapezformel (siehe Abschnitt 14.4.1) durchgeführt.
Aus der Darstellung des Restgliedes folgt, dass das Restglied für $f(x) = P_n(x)$ verschwindet, weil die $(n+1)$-te Ableitung eines Polynoms P_n n-ten Grades Null ist. Das bedeutet, dass die zugehörige Quadraturformel Q zu $n+1$ vorgegebenen Knoten im Referenzintervall Polynome bis zum Grade n exakt integriert. Bei geeigneter Wahl der Stützstellen x_k können außerdem durch dieselbe Quadraturformel auch Polynome noch höheren Grades integriert werden (vgl. Simpsonsche Formel).

Allgemein gelten die folgenden Aussagen, die ohne Beweis angegeben werden: Für $2n$-mal stetig differenzierbare Funktionen f hat unter Verwendung von $2n-1$ bzw. $2n$ Knoten x_k im Referenzintervall $[a,b]$ für das zu integrierende Interpolationspolynom das Restglied der Quadraturregel die Gestalt

$$E(f;a,b) \;=\; c_{2n-1} \quad \frac{(b-a)^{2n-1}}{(2n)!} \quad f^{(2n)}(\xi) \qquad \xi \in [a,b]$$

bzw.

$$E(f;a,b) \;=\; c_{2n} \quad \frac{(b-a)^{2n-1}}{(2n)!} \quad f^{(2n)}(\xi) \qquad \xi \in [a,b]\,.$$

Die Koeffizienten c_{2n-1} bzw. c_{2n} hängen nur von den $2n-1$ bzw. $2n$ Knoten im Referenzintervall ab.

Mit einem oberen Index an Q und E wird im Folgenden der Name der Quadraturformel gekennzeichnet, mit dem unteren Index die gewählte Schrittweite.

14.3.1 Die Sehnentrapezformel

Sehnentrapezformel für das Referenzintervall [a,b]

Betrachtet man das Integral von f über das Referenzintervall $[a,b]=[0,h]$ und wählt die Randpunkte $x_0=0$, $x_1=h$ als Stützstellen, so ergeben sich aus (14.4) wegen $n=1$, $a=0$, $b=h$ die Gewichte $A_0=A_1=h/2$, so dass die Quadraturformel (14.3) lautet

$$Q^{ST}(f;0,h)=A_0f(x_0)+A_1f(x_1)=\frac{h}{2}\big(f(0)+f(h)\big). \tag{14.7}$$

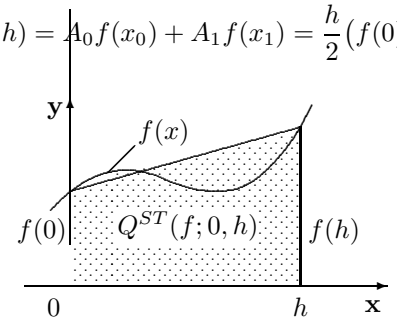


Abb. 14.4. Zur Sehnentrapezformel

$Q^{ST}(f;0,h)$ heißt *Sehnentrapezformel* (ST-Formel). Für das zugehörige Restglied folgt mit (14.6) die Darstellung

$$E^{ST}(f;0,h)=\frac{1}{2}f''(\xi^*)\int_0^h x(x-h)\,\mathrm{d}x=-\frac{h^3}{12}f''(\xi^*),\ \xi^*\in[0,h],\ f\in C^2[0,h]. \tag{14.8}$$

Die ST-Formel besitzt somit die *lokale Fehlerordnung* $O(h^3)$, sie integriert Polynome 1. Grades exakt. Geometrisch bedeutet $Q^{ST}(f;0,h)$ die Fläche des der Kurve $y=f(x)$ für $x\in[0,h]$ einbeschriebenen Sehnentrapezes. Zusammengefasst folgt die *Sehnentrapezregel*

$$\int_0^h f(x)\,\mathrm{d}x=Q^{ST}(f;0,h)+E^{ST}(f;0,h)=\frac{h}{2}\big(f(0)+f(h)\big)-\frac{h^3}{12}f''(\xi^*),\quad \xi^*\in[0,h].$$

Summierte Sehnentrapezformel bei äquidistanter Zerlegung

Ist die Integration über ein ausgedehntes Intervall $[\alpha,\beta]$ auszuführen, so zerlegt man $[\alpha,\beta]$ in N Teilintervalle der Länge $h=(\beta-\alpha)/N$; h heißt *Schrittweite*. Wegen

$$I(f;\alpha,\beta)=\int_\alpha^\beta f(x)\,\mathrm{d}x=\int_\alpha^{\alpha+h} f(x)\,\mathrm{d}x+\int_{\alpha+h}^{\alpha+2h} f(x)\,\mathrm{d}x+\quad\ldots\quad+\int_{\alpha+(N-1)h}^{\beta=\alpha+Nh} f(x)\,\mathrm{d}x$$

kann man auf jedes Integral über ein Teilintervall der Länge h die Sehnentrapezformel (14.7) anwenden und anschließend addieren. So ergibt sich die *summierte Sehnentrapezformel*

$$\begin{cases} Q_h^{ST}(f;\alpha,\beta) = \frac{h}{2}\big(f(\alpha)+2f(\alpha{+}h)+2f(\alpha{+}2h)+\ldots+2f(\alpha(N{-}1)h)+f(\beta)\big) \\ \qquad\qquad = \frac{h}{2}\big(f(\alpha)+f(\beta)+2\sum\limits_{i=1}^{N-1} f(\alpha{+}ih)\big) \end{cases} \tag{14.9}$$

und mit Hilfe von (14.8) das *summierte Restglied*

$$\begin{cases} E_h^{ST}(f;\alpha,\beta) = -\frac{h^3}{12}\big(f''(\xi_0^*)+f''(\xi_1^*)+\ldots+f''(\xi_{N-1}^*)\big) \\ \text{mit} \quad \xi_j^* \in [a+jh,\ \alpha+(j+1)h], \quad j=0(1)N{-}1, \quad \beta=\alpha+Nh. \end{cases} \tag{14.10}$$

Ist f'' stetig in $[a,b]$ und bezeichnet man mit m den kleinsten und mit M den größten Wert von f'' in $[a,b]$, so gilt

$$m \quad \le \quad \frac{1}{N}\big(f''(\xi_0^*)+f''(\xi_1^*)+\ldots+f''(\xi_{N-1}^*)\big) \quad \le \quad M$$

und nach dem Zwischenwertsatz gibt es mindestens ein $\eta \in [a,b]$, so dass $f''(\eta) = \frac{1}{N}\big(f''(\xi_0^*)+f''(\xi_1^*)+\ldots+f''(\xi_{N-1}^*)\big)$ ist. Damit erhält man für (14.10)

$$E_h^{ST}(f;\alpha,\beta) = -\frac{h^2}{12}\,\frac{\beta-\alpha}{N}\,N f''(\eta) = -\frac{\beta-\alpha}{12}\,h^2\,f''(\eta)\,, \quad \eta\in[a,b]\,; \tag{14.11}$$

die *globale Fehlerordnung* ist somit $O(h^2)$.

Für das Integral von f über $[\alpha,\beta]$ gilt dann die *summierte Sehnentrapezformel*

$$\begin{aligned} \int_\alpha^\beta f(x)\,\mathrm{d}x &= Q_h^{ST}(f;\alpha,\beta) + E_h^{ST}(f;\alpha,\beta) \quad \text{mit} \\ Q_h^{ST}(f;\alpha,\beta) &= \frac{h}{2}\left(f(\alpha)+f(\beta)+2\sum_{k=1}^{N-1} f(\alpha+kh)\right) \\ E_h^{ST}(f;\alpha,\beta) &= -\frac{\beta-\alpha}{12}h^2 f''(\eta), \quad \eta\in[\alpha,\beta], \quad f\in C^2[\alpha,\beta]. \end{aligned} \tag{14.12}$$

Dabei sind $Q_h^{ST}\ (f;\alpha,\beta)$ die *summierte ST-Formel* und $E_h^{ST}\ (f;\alpha,\beta)$ das *Restglied der summierten ST-Formel*; die *globale Fehlerordnung* ist $O(h^2)$.

Geometrisch stellt die summierte Sehnentrapezformel (14.9) die Fläche des der Kurve $y=f(x)$ für $x\in[a,b]$ einbeschriebenen Polygonzuges mit den Eckpunkten $\big(\alpha+\nu h,\ f(\alpha+\nu h)\big)$ für $\nu=0(1)N$ dar, d. h. die Summe der Flächeninhalte der in Abbildung 14.5 eingezeichneten Trapeze.

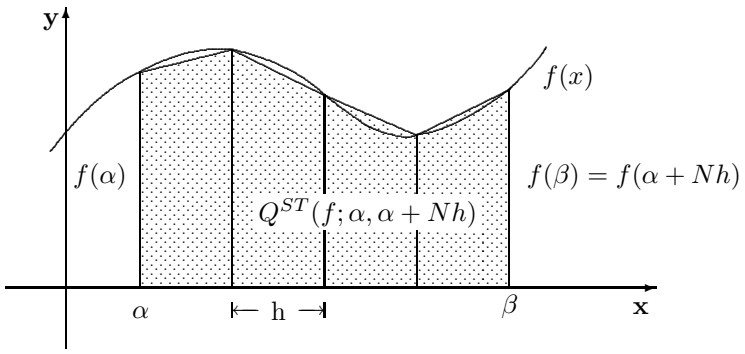


Abb. 14.5. Summierte Sehnentrapezformel

Beispiel 14.3.

Bei der Berechnung des Integrals von $f(x) = \mathrm{e}^{-x^2}$ über $[0,1]$ mit der Sehnentrapezformel $Q_h^{ST}(f;0,1)$ ist die Schrittweite h so zu wählen, dass der maximale absolute Verfahrensfehler den Wert $0.5 \cdot 10^{-6}$ nicht überschreitet. Gesucht ist ein Wert für h.

Lösung:

Es ist $\int\limits_{\alpha}^{\beta} f(x)\,\mathrm{d}x \;=\; \int\limits_{0}^{1} \mathrm{e}^{-x^2}\,\mathrm{d}x \;=\; Q_h^{ST}(f;0,1) + E_h^{ST}(f;0,1)\,.$

Der Betrag der zweiten Ableitung $f''(x) = 2\mathrm{e}^{-x^2}(2x^2-1)$ hat im Intervall $[0,1]$ sein Maximum bei $x = 0$, $|\,f''(0)\,| = 2$. Mit (14.11) muss dann

$$|\,E_h^{ST}(f;0,1)\,| \;\le\; \frac{h^2}{12} \max_{x\in[0,1]} |f''(x)| = \frac{h^2}{6} \;\le\; 0.5 \cdot 10^{-6} \quad \text{bzw.} \quad h \;\le\; \sqrt{3} \cdot 10^{-3} \text{ sein ;}$$

daraus folgt wegen $h = \frac{1}{N}\,(\beta - \alpha) = \frac{1}{N}$ für N die Bedingung $N \ge (1/\sqrt{3}) \cdot 10^3$, die für $N \ge 578$ erfüllt ist. □

Beispiel 14.4.

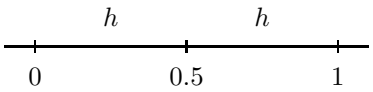
Gegeben: Die Funktion $f : f(x) = 1/(1+x^2)$ für $x \in [0,1]$

Gesucht: Ein Näherungswert für $\frac{\pi}{4}$, der durch Integration von f über $[0,1]$ mit Hilfe der Sehnentrapezformel zu bestimmen ist. Man wähle die Schrittweite $h = 0.5$ und gebe eine obere Schranke für den Verfahrensfehler an.

Lösung: Es gilt

$$\frac{\pi}{4} = \arctan 1 = \int\limits_{0}^{1} \frac{1}{1+x^2}\,\mathrm{d}x = Q_h^{ST}(f;0,1) + E_h^{ST}(f;0,1)\,.$$

Wegen $h = 0.5$ ist das Intervall $[0,1]$ zu halbieren:



Man wendet also die Sehnentrapezformel zweimal an, einmal transformiert man sie auf das Intervall $[0, 0.5]$, einmal auf das Intervall $[0.5, 1]$ und addiert beide Formeln. So ergibt sich die summierte Formel

$$\begin{aligned} Q_h^{ST}(f;0,1) &= Q^{ST}(f;0,0.5) + Q^{ST}(f;0.5,1) \\ &= \tfrac{h}{2}\big(f(0)+f(0.5)\big) + \tfrac{h}{2}\big(f(0.5)+f(1)\big) \\ &= \tfrac{h}{2}\big(f(0)+2\,f(0.5)+f(1)\big) \\ &= 0.25\,(1+2\cdot 0.8+0.5) = 0.775\,, \end{aligned}$$

die (14.9) entspricht für $N = 2$.

Eine obere Schranke für den Verfahrensfehler erhält man durch Abschätzung des zugehörigen Restgliedes (14.11). Aus $E_h^{ST}(f;0,1) = -\frac{1}{12}\cdot\frac{1}{4}f''(\eta)$ folgt

$$|E_h^{ST}(f;0,1)| \le \frac{1}{48}\max_{x\in[0,1]}|f''(x)| = \frac{1}{24} \le 0.42\cdot 10^{-1}\,,$$

da wegen $f''(x) = 2\cdot g(x)$ mit $g(x) = (3x^2-1)/(1+x^2)^3$ und $g'(x) = 12x\,(1-x^2)/(1+x^2)^4 = 0$ für $x = 0$ und $x = 1$ das Maximum von $|g(x)|$ für $x = 0$ angenommen wird, so dass $|g(x)| \le 1$, also $|f''(x)| \le 2$ gilt. Für den wahren Fehler folgt wegen $\frac{\pi}{4} = 0.785398\ldots$ die Abschätzung

$$0.1\cdot 10^{-1} = 0.785 - 0.775 \le \frac{\pi}{4} - 0.775 \le 0.786 - 0.775 = 0.11\cdot 10^{-1}\,.$$

Dieser Weg zur Abschätzung des Verfahrensfehlers ist für die praktische Anwendung uninteressant, da man die Ableitungen des Integranden kennen und abschätzen muss. Man verwendet besser eine Fehlerschätzung (siehe Abschnitt 14.11). □

Beispiel 14.5.

Gegeben: Das Integral

$$I\left(\frac{\sin x}{x};0,\frac{\pi}{2}\right) = \int_0^{\frac{\pi}{2}} \frac{\sin x}{x}\,\mathrm{d}x$$

Gesucht: Der Näherungswert Q^{ST} für das Integral mit Hilfe der summierten Sehnentrapezformel für $h = \dfrac{\pi}{12}$.

Lösung: Mit $f(x) = (\sin x)/x$ ist

$$\begin{aligned} Q^{ST}_{\pi/12}\left(\frac{\sin x}{x}; 0, \frac{\pi}{2}\right) &= \frac{\pi/12}{2}\Bigg(f(0) + f\left(\frac{\pi}{2}\right) \\ &\quad + 2\left(f\left(\frac{\pi}{12}\right) + f\left(\frac{2\pi}{12}\right) + f\left(\frac{3\pi}{12}\right) + f\left(\frac{4\pi}{12}\right) + f\left(\frac{5\pi}{12}\right)\right)\Bigg) \\ &= \frac{\pi}{24}\left(1 + \frac{2}{\pi} + 2\left(6\,\frac{\sqrt{3}-1}{\pi\sqrt{2}} + \frac{3}{\pi} + \frac{2\sqrt{2}}{\pi} + \frac{3\sqrt{3}}{2\pi} + 6\,\frac{\sqrt{3}+1}{5\pi\sqrt{2}}\right)\right) \\ &\approx 1.368445849 \end{aligned}$$

Der exakte Integralwert ist $I_{ex} = 1.370\,762\,168$. □

Beispiel 14.6.

Gegeben: Die Funktion $f(x) = \sqrt{1 - k^2 \sin^2 x},\ k = 0.75$

Gesucht: Ergebnisse der Quadratur von f im Intervall $[0, \pi/2]$ zu verschiedenen Genauigkeitsschranken mit Hilfe der Sehnentrapez-Newton-Cotes-Formel.

Lösung:

Genauigkeit	$5 \cdot 10^{-3}$	$5 \cdot 10^{-6}$	$5 \cdot 10^{-9}$	$5 \cdot 10^{-12}$
Funktionsausw.	2	4	8	16
Ergebnis	1.31829879830	1.31847200818	1.31847210799	1.31847210799
Schätzwert	$4.47 \cdot 10^{-3}$	$4.72 \cdot 10^{-6}$	$9.52 \cdot 10^{-12}$	$1.61 \cdot 10^{-15}$

Zur Ermittlung der Schätzwerte für den Fehler siehe Abschnitt 14.11. □

Summierte Sehnentrapezformel bei nichtäquidistanter Zerlegung

Mit $h_k = t_{k+1} - t_k$ und $h_{\max} := \max\limits_{0 \le k \le N-1}\{h_k\}$ erhält man für ein Integrationsintervall $[\alpha,\beta]$ bei der Zerlegung $\alpha = t_0 < t_1 < \ldots < t_N = \beta$

$$\begin{aligned} \int_\alpha^\beta f(x)\,\mathrm{d}x &= Q^{ST}_{h_k}(f;\alpha,\beta) + E^{ST}_{h_k}(f;\alpha,\beta) \quad \text{mit} \\ Q^{ST}_{h_k}(f;\alpha,\beta) &= \frac{1}{2}\sum_{k=0}^{N-1}(t_{k+1} - t_k)\big(f(t_k) + f(t_{k+1})\big) \\ &= \frac{t_1 - t_0}{2}f(t_0) + \frac{1}{2}\sum_{k=1}^{N-1}(t_{k+1} - t_k)f(t_k) + \frac{t_N - t_{N-1}}{2}f(t_N) \\ E^{ST}_{h_k}(f;\alpha,\beta) &= O(h^2_{\max}). \end{aligned}$$

Sehnentrapezformel für periodische Funktionen f

Sei f eine auf $[\alpha,\beta]$ $(2m)$-mal stetig differenzierbare periodische Funktion mit der Periode $\beta-\alpha$ und sei $[\alpha,\beta]$ in N Teilintervalle der Länge $h=(\beta-\alpha)/N$ unterteilt, dann gilt mit der summierten Euler-Maclaurinformel (vgl. Abschnitt 14.5) wegen

$$f^{(2k-1)}(\alpha)=f^{(2k-1)}(\beta)$$

für die Sehnentrapezregel

$$\int_{\alpha}^{\beta} f(x)\,\mathrm{d}x = h\sum_{k=0}^{N-1} f(\alpha+kh)+O(h^{2m}).$$

14.3.2 Die Simpsonsche Formel

Simpsonsche Formel für das Referenzintervall [0,2h]

Betrachtet man das Integral von f über $[a,b]=[0,2h]$ und wählt $x_0=0$, $x_1=h$, $x_2=2h$ als Stützstellen, so ergeben sich wegen $n=2$, $a=0$, $b=2h$ aus dem Gleichungssystem (14.4)

$$\left\{\begin{array}{rcrcrcl} A_0 & + & A_1 & + & A_2 & = & 2h,\\ & & A_1 & + & 2A_2 & = & 2h,\\ & & A_1 & + & 4A_2 & = & \frac{8}{3}h; \end{array}\right.$$

die Gewichte $A_0=A_2=h/3$, $A_1=4h/3$, so dass die Quadraturformel (14.3) lautet

$$Q^S(f;0,2h)=A_0f(x_0)+A_1f(x_1)+A_2f(x_2)=\frac{h}{3}\big(f(0)+4f(h)+f(2h)\big). \quad (14.13)$$

$Q^S(f;0,2h)$ heißt *Simpsonsche Formel* (S-Formel). Da $\pi(x)=x\,(x-h)\,(x-2h)$ das Vorzeichen in $[0,2h]$ wechselt, kann das Restglied $E^S(f;0,2h)$ nicht in der Form (14.6) dargestellt werden. Daher wird ein anderer Weg eingeschlagen. Im Folgenden wird vorausgesetzt, dass $f^{(4)}(x)\in C\,[-h,h]$ ist, und das Integral $I(f;-h,h)$ betrachtet.
Für das Restglied gilt dann

$$\begin{aligned} E(h) \;&:=\; E^S(f;-h,h) \;=\; I(f;-h,h)-Q^S(f;-h,h) \quad (14.14)\\ &=\; \int_{-h}^{h} f(x)\,\mathrm{d}x-\frac{h}{3}\big(f(-h)+4f(0)+f(h)\big)\,. \end{aligned}$$

Man bildet

$$\begin{aligned} E'(h) &= \tfrac{1}{3}\big(2f(h)-4f(0)+2f(-h)\big)-\tfrac{h}{3}\big(f'(h)-f'(-h)\big)\,,\\ E''(h) &= \tfrac{1}{3}\big(f'(h)-f'(-h)\big)-\tfrac{h}{3}\big(f''(h)-f''(-h)\big)\,,\\ E'''(h) &= -\tfrac{2h^2}{3}\cdot\tfrac{1}{2h}\big(f'''(h)-f'''(-h)\big)\,. \end{aligned}$$

Mit dem Mittelwertsatz der Differentialrechnung folgt für $E'''(h)$ die Darstellung

$$E'''(h) = -\frac{2h^2}{3} f^{(4)}(\xi)\,, \qquad \xi \in (-h, h)\,.$$

Wegen $E(0) = 0$, $E'(0) = 0$, $E''(0) = 0$, $E'''(0) = 0$ und

$$E(h) = \int_0^h E'(t)\,\mathrm{d}t\,, \quad E'(h) = \int_0^h E''(t)\,\mathrm{d}t\,, \quad E''(h) = \int_0^h E'''(t)\,\mathrm{d}t$$

erhält man mit dem verallgemeinerten Mittelwertsatz die Beziehung

$$\begin{aligned} E''(h) &= \int_0^h E'''(t)\,\mathrm{d}t = \int_0^h \left(-\frac{2t^2}{3} f^{(4)}(\xi)\right) \mathrm{d}t \\ &= -\frac{2}{3} f^{(4)}(\xi_1) \int_0^h t^2\,\mathrm{d}t = -\frac{2}{9} h^3 f^{(4)}(\xi_1) \end{aligned} \tag{14.15}$$

und analog zu (14.15)

$$\begin{aligned} E'(h) &= -\frac{2}{9} f^{(4)}(\xi_2) \quad \int_0^h t^3\,\mathrm{d}t \quad = -\frac{h^4}{18} f^{(4)}(\xi_2)\,, \\ E(h) &= -\frac{1}{18} f^{(4)}(\xi_3) \quad \int_0^h t^4\,\mathrm{d}t \quad = -\frac{h^5}{90} f^{(4)}(\xi_3)\,. \end{aligned}$$

Für das Restglied (14.14) gilt somit

$$E^S(f; -h, h) = -\frac{h^5}{90} f^{(4)}(\xi^*) \qquad \xi^* \in [-h, h] \tag{14.16}$$

Für das Restglied der S-Formel gilt bezogen auf das Intervall $[0, 2h]$

$$E^S(f; 0, 2h) = -\frac{h^5}{90} f^{(4)}(\xi^*), \quad \xi^* \in [0, 2h], \quad f^{(4)} \in C[0, 2h].$$

Die S-Formel besitzt somit die lokale Fehlerordnung $O(h^5)$, sie integriert Polynome bis 3-ten Grades exakt. Diese Tatsache ist bemerkenswert, denn die Sehnentrapezformel, die zwei Stützstellen verwendet, ist von der Ordnung $O(h^3)$ und die Simpsonsche Formel mit nur einer Stützstelle mehr von der Ordnung $O(h^5)$. Aus dieser Tatsache resultiert die Beliebtheit der Simpsonschen Formel. Zusammengefasst folgt die *Simpsonsche Regel*

$$\begin{aligned} \int_0^{2h} f(x)\,\mathrm{d}x &= Q^S(f; 0, 2h) + E^S(f; 0, 2h) \\ &= \frac{h}{3}\big(f(0) + 4f(h) + f(2h)\big) - \frac{h^5}{90} f^{(4)}(\xi^*), \quad \xi^* \in [0, 2h]. \end{aligned}$$

Summierte Simpsonsche Formel für äquidistante Zerlegung

Zur Bestimmung des Integrals von f über ein ausgedehntes Intervall $[\alpha,\beta]$ zerlegt man $[\alpha,\beta]$ in $2N$ Teilintervalle der Länge $h = (\beta-\alpha)/(2N)$. Man erhält

$$I(f;\alpha,\beta) = \int_\alpha^\beta f(x)\,\mathrm{d}x = \int_\alpha^{\alpha+2h} f(x)\,\mathrm{d}x + \int_{\alpha+2h}^{\alpha+4h} f(x)\,\mathrm{d}x + \ldots + \int_{\alpha+2(N-1)h}^{\beta} f(x)\,\mathrm{d}x\,.$$

Auf jedes Integral über ein Teilintervall der Länge $2h$ wendet man die Simpsonsche Formel (14.13) an und erhält so die *summierte Simpsonsche Formel* mit $\beta = \alpha + 2N \cdot h$

$$\begin{aligned} Q_h^S(f;\alpha,\beta) \quad &= \quad \frac{h}{3}\Big(f(\alpha) + 4f(\alpha+h) + 2f(\alpha+2h) + 4f(\alpha+3h) + \ldots \\ &\quad\; +4f\big(\alpha+(2N-1)h\big) + f(\beta)\Big) \end{aligned}$$

Unter der Voraussetzung $f^{(4)} \in C[\alpha,\beta]$ erhält man mit (14.16) das *summierte Restglied*, indem man analoge Überlegungen wie bei (14.11) anstellt. Für $\eta \in [\alpha,\beta]$ gilt

$$E_h^S(f;\alpha,\beta) = -\frac{h^5}{90}\, f^{(4)}(\eta) + N\,O(h^6) = -\frac{\beta-\alpha}{180}\, h^4\, f^{(4)}(\eta) + O(h^5)\,. \tag{14.17}$$

Die *globale Fehlerordnung* der Simpsonschen Formel ist $O(h^4)$. Für das Integral von f über $[\alpha,\beta]$ gilt schließlich zusammengefasst die Simpsonsche Regel

$$\begin{aligned} \int_\alpha^\beta f(x)\,\mathrm{d}x \quad &= \quad Q_h^S(f;\alpha,\beta) + E_h^S(f;\alpha,\beta) \quad \text{mit} \\ Q_h^S(f;\alpha,\beta) \quad &= \quad \frac{h}{3}\left(f(\alpha) + f(\beta) + 4\sum_{k=0}^{N-1} f(\alpha+(2k+1)h) + 2\sum_{k=1}^{N-1} f(\alpha+2kh)\right), \\ E_h^S(f;\alpha,\beta) \quad &= \quad -\frac{\beta-\alpha}{180}h^4 f^{(4)}(\eta), \quad \eta \in [\alpha,\beta], \quad f^{(4)} \in C[\alpha,\beta], \quad h = \frac{\beta-\alpha}{2N}. \end{aligned}$$

Dabei ist $Q_h^S(f;\alpha,\beta)$ die *summierte S-Formel* und $E_h^S(f;\alpha,\beta)$ das *Restglied der summierten S-Formel*. Die summierte S-Formel besitzt die globale Fehlerordnung $O(h^4)$.

Ein Nachteil der S-Formel für äquidistante Zerlegung ist, dass immer eine gerade Anzahl von Teilintervallen der Länge h erforderlich ist, um die Formel anwenden zu können. Dieser Nachteil lässt sich aber durch Kombination der S-Formel mit der 3/8-Formel im Falle einer ungeraden Zahl von Teilintervallen immer vermeiden, vgl. dazu die Bemerkung in Abschnitt 14.3.3.

Beispiel 14.7. (Fortsetzung von Beispiel 14.3)

Bei der Berechnung des Integrals von f mit $f(x) = \mathrm{e}^{-x^2}$ über $[0,1]$ mit der Simpsonschen Formel $Q_h^S(f;0,1)$ ist die Schrittweite h so zu wählen, dass der maximale absolute Verfahrensfehler den Wert $0.5 \cdot 10^{-6}$ nicht überschreitet. Die Schrittweite h ist zu ermitteln.

Lösung:

$f^{(4)} = \mathrm{e}^{-x^2}(16x^4 - 48x^2 + 12)$ hat für $x = 0$ den maximalen Wert $f^{(4)}(0) = 12$. Mit (14.17) muss dann

$$|E_h^S(f;0,1)| \le \frac{12\,h^4}{180} = \frac{1}{15}\,h^4 \le 0.5 \cdot 10^{-6} \quad \text{bzw.} \quad h \le \frac{\sqrt[4]{7.5}}{10\sqrt{10}} \le 0.0523$$

sein; also reicht $h = 0.05$ aus, um die geforderte Genauigkeit zu erreichen. Wegen $h = 1/(2N)$ ergeben sich $N = 10$ Simpsonschritte gegenüber $N \ge 580$ Schritten bei der Berechnung mit der Sehnentrapezformel (vgl. Beispiel 14.3) □

Beispiel 14.8. (Fortsetzung von Beispiel 14.4)

Für das Integral von $f : f(x) = 1/(1 + x^2)$, $x \in [0, 1]$, ist mit der Simpsonschen Formel für $h = 0.5$ ein Näherungswert zu bestimmen. Das Ergebnis ist mit dem des Beispiels 14.4 zu vergleichen. Wegen $h = 0.5$ entspricht $[0, 1]$ dem Referenzintervall $[0, 2h]$. Man erhält

$$Q^S(f;0,1) = \frac{0.5}{3}\big(f(0) + 4f(h) + f(2h)\big) = \frac{0.5}{3}\,(1 + 4 \cdot 0.8 + 0.5) = 0.7833.$$

Wegen $\frac{\pi}{4} = 0.785398\ldots$ ist $Q^S(f;0,2h)$ schon ein bedeutend besserer Näherungswert für $\frac{\pi}{4}$ als der Wert $Q^{ST}(f;0,2h) = 0.775$, der mit der Sehnentrapezformel ermittelt wurde. Es gilt für den wahren Fehler die Abschätzung

$$0.19 \cdot 10^{-2} \le \frac{\pi}{4} - 0.7833 \le 0.7854 - 0.7833 = 0.21 \cdot 10^{-2}\,.$$

Eine Abschätzung des Verfahrensfehlers mit Hilfe des Restgliedes $E^S(f;0,2h)$ hätte wegen der erforderlichen Berechnung und Abschätzung von $f^{(4)}(x)$ große Mühe bereitet. Da der wahre Wert des Integrals im Allgemeinen nicht bekannt ist, so dass eine Abschätzung des wahren Fehlers, wie in dem vorliegenden Beispiel, nicht praktikabel ist, muss eine bessere Möglichkeit zur Bestimmung des Fehlers gesucht werden; vgl. dazu Abschnitt 14.11 (Fehlerschätzung). □

Beispiel 14.9. (Fortsetzung von Beispiel 14.5)

Gegeben: Das Integral

$$I\left(\frac{\sin x}{x};0,\frac{\pi}{2}\right) = \int_0^{\frac{\pi}{2}} \frac{\sin x}{x}\,\mathrm{d}x$$

Gesucht: Der Näherungswert Q^S für das Integral mit Hilfe der summierten Simpsonschen Formel für $h = \dfrac{\pi}{12}$.

Lösung: Mit $f(x) = (\sin x)/x$ ist

$$\begin{aligned} Q^S_{\pi/12}\left(\frac{\sin x}{x}; 0, \frac{\pi}{2}\right) &= \frac{\pi/12}{3}\Bigg(f(0) + f\left(\frac{\pi}{2}\right) + 4\left(f\left(\frac{\pi}{12}\right) + f\left(\frac{3\pi}{12}\right)\right. \\ &\quad \left. + f\left(\frac{5\pi}{12}\right)\right) + 2\left(f\left(\frac{2\pi}{12}\right) + f\left(\frac{4\pi}{12}\right)\right)\Bigg) \\ &= \frac{\pi}{36}\Bigg(1 + \frac{2}{\pi} + 4\left(6\,\frac{\sqrt{3}-1}{\pi\sqrt{2}} + \frac{2\sqrt{2}}{\pi} + 6\,\frac{\sqrt{3}+1}{5\pi\sqrt{2}}\right) \\ &\quad + 2\left(\frac{3}{\pi} + \frac{3\sqrt{3}}{2\pi}\right)\Bigg) \\ &\approx 1.370768213 \end{aligned}$$

Der exakte Wert des Integrals ist $I_{ex} = 1.370\,762\,168$. □

Beispiel 14.10. (Fortsetzung von Beispiel 14.7)

Gegeben: Die Funktion $f(x) = \sqrt{1 - k^2 \sin^2 x},\ k = 0.75$

Gesucht: Ergebnisse der Quadratur von f im Intervall $[0, \pi/2]$ zu verschiedenen Genauigkeitsschranken mit Hilfe der Simpson-Newton-Cotes-Formel.

Lösung:

Genauigkeit	$5 \cdot 10^{-3}$	$5 \cdot 10^{-6}$	$5 \cdot 10^{-9}$	$5 \cdot 10^{-12}$
Funktionsausw.	2	4	8	8
Ergebnis	1.31852974481	1.31847214127	1.31847210799	1.31847210799
Schätzwert	$-2.83 \cdot 10^{-4}$	$-5.66 \cdot 10^{-7}$	$-1.38 \cdot 10^{-12}$	$-1.38 \cdot 10^{-12}$

Zur Ermittlung der Schätzwerte für den Fehler siehe Abschnitt 14.11. □

Summierte Simpsonsche Formel für nichtäquidistante Zerlegung

Mit der Zerlegung

$$Z : \alpha = t_0 < t_1 < t_2 < \ldots < t_m = \beta$$

des Integrationsintervalls $[\alpha,\beta]$ und $h_k := t_{k+1} - t_k$, $h_{\max} = \max\limits_{0 \le k \le m-1}\{h_k\}$, erhält man die Simpsonsche Formel

$$\begin{aligned} \int_\alpha^\beta f(x)\,\mathrm{d}x &= Q^S_{h_k}(f;\alpha,\beta) + E^S_{h_k}(f;\alpha,\beta) \quad \text{mit} \\ Q^S_{h_k}(f;\alpha,\beta) &= \frac{1}{6}\sum_{k=0}^{m-1}(t_{k+1} - t_k)\left[f(t_k) + 4f\left(\frac{t_k + t_{k+1}}{2}\right) + f(t_{k+1})\right], \\ E^S_{h_k}(f;\alpha,\beta) &= O(h^4_{\max}). \end{aligned}$$

14.3.3 Die 3/8-Formel

3/8-Formel für das Referenzintervall [0,3h]

Betrachtet man das Integral von f über $[a,b] = [0,3h]$ und wählt $x_0 = 0$, $x_1 = h$, $x_2 = 2h$, $x_3 = 3h$ als Stützstellen, so ergibt sich aus (14.4) wegen $n = 3$, $a = 0$, $b = 3h$ das Gleichungssystem

$$\begin{aligned} A_0 + A_1 + A_2 + A_3 &= 3h\,, \\ A_1 + 2A_2 + 3A_3 &= \tfrac{9}{2}h\,, \\ A_1 + 4A_2 + 9A_3 &= 9h\,, \\ A_1 + 8A_2 + 27A_3 &= \tfrac{81}{4}h; \end{aligned}$$

es hat als Lösung die Gewichte $A_0 = \frac{3}{8}h$, $A_1 = \frac{9}{8}h$, $A_2 = \frac{9}{8}h$, $A_3 = \frac{3}{8}h$. Die Quadraturformel (14.3) lautet damit

$$\begin{aligned} Q^{3/8}(f;0,3h) &= A_0 f(x_0) + A_1 f(x_1) + A_2 f(x_2) + A_3 f(x_3) \\ &= \frac{3h}{8}\big(f(0) + 3f(h) + 3f(2h) + f(3h)\big). \end{aligned}$$

$\mathrm{Q}^{3/8}(f;0,3h)$ heißt *3/8-Formel*. Für das Restglied der 3/8-Formel gilt

$$E^{3/8}(f;0,3h) = -\frac{3}{80}h^5 f^{(4)}(\xi^*), \quad \xi^* \in [0,3h], \quad f^{(4)} \in C[0,3h].$$

Die lokale Fehlerordnung ist somit $O(h^5)$, die Quadraturformel integriert Polynome 3. Grades exakt. Zusammengefasst folgt die *3/8-Regel*

$$\begin{aligned} \int_0^{3h} f(x)\,\mathrm{d}x &= Q^{3/8}(f;0,3h) + E^{3/8}(f;0,3h) \\ &= \tfrac{3h}{8}\big(f(0) + 3f(h) + 3f(2h) + f(3h)\big) - \tfrac{3}{80}h^5 f^{(4)}(\xi^*), \quad \xi^* \in [0,3h]. \end{aligned}$$

Summierte 3/8-Formel bei äquidistanter Zerlegung

Zur Bestimmung des Integrals von f über ein ausgedehntes Intervall $[\alpha,\beta]$ zerlegt man $[\alpha,\beta]$ in $3N$ Teilintervalle der Länge $h = (\beta-\alpha)/(3N)$, so dass die *summierte 3/8-Regel* lautet

$$\begin{aligned} \int_\alpha^\beta f(x)\,\mathrm{d}x &= Q_h^{3/8}(f;\alpha,\beta) + E_h^{3/8}(f;\alpha,\beta) \quad \text{mit} \\ Q_h^{3/8}(f;\alpha,\beta) &= \frac{3h}{8}\Bigg(f(\alpha) + f(\beta) + 3\sum_{k=1}^{N} f(\alpha+(3k-2)h) \\ &\quad + 3\sum_{k=1}^{N} f(\alpha+(3k-1)h) + 2\sum_{k=1}^{N-1} f(\alpha+3kh)\Bigg), \\ E_h^{3/8}(f;\alpha,\beta) &= -\frac{\beta-\alpha}{80}h^4 f^{(4)}(\eta), \quad \eta \in [\alpha,\beta], \quad f^{(4)} \in C[\alpha,\beta]. \end{aligned}$$

Dabei ist $Q_h^{3/8}(f;\alpha,\beta)$ die *summierte 3/8-Formel* und $E_h^{3/8}(f;\alpha,\beta)$ das *Restglied der summierten 3/8-Formel*. Die summierte 3/8-Formel besitzt die globale Fehlerordnung $O(h^4)$.

Beispiel 14.11. (Fortsetzung von Beispiel 14.5)

Gegeben: Das Integral

$$I\left(\frac{\sin x}{x};0,\frac{\pi}{2}\right)=\int\limits_0^{\frac{\pi}{2}}\frac{\sin x}{x}\,\mathrm{d}x$$

Gesucht: Der Näherungswert $Q^{3/8}$ für das Integral mit Hilfe der 3/8-Formel für $h=\frac{\pi}{12}$.

Lösung: Mit $f(x)=(\sin x)/x$ ist

$$\begin{aligned}
Q_{\pi/12}^{3/8}\left(\frac{\sin x}{x};0,\frac{\pi}{2}\right) &= \frac{3}{8}\cdot\frac{\pi/12}{3}\Bigg(f(0)+f\left(\frac{\pi}{2}\right)+3\left(f\left(\frac{\pi}{12}\right)\right.\\
&\quad \left.+f\left(\frac{2\pi}{12}\right)+f\left(\frac{4\pi}{12}\right)+f\left(\frac{5\pi}{12}\right)\right)+2\left(f\left(\frac{3\pi}{12}\right)\right)\Bigg)\\
&= \frac{\pi}{32}\Bigg(1+\frac{2}{\pi}+3\left(6\,\frac{\sqrt{3}-1}{\pi\sqrt{2}}+\frac{3}{\pi}+\frac{3\sqrt{3}}{2\pi}+6\,\frac{\sqrt{3}+1}{5\pi\sqrt{2}}\right)\\
&\quad +2\cdot\frac{2\sqrt{2}}{\pi}\Bigg)\\
&\approx 1.370775847
\end{aligned}$$

Der exakte Integralwert ist $I_{ex}=1.370\,762\,168$. □

Bemerkung. Soll das Integral $I(f;\alpha,\beta)$ von f über $[\alpha,\beta]$ mit der globalen Fehlerordnung $O(h^4)$ bei vorgegebenem konstantem h berechnet werden, und ist es nicht möglich, das Intervall $[\alpha,\beta]$ in $2N$ oder $3N$ Teilintervalle der Länge h zu zerlegen, so empfiehlt es sich, die Simpsonsche Formel mit der 3/8-Formel zu kombinieren, da beide die Fehlerordnung $O(h^4)$ besitzen.

Beispiel 14.12. (Fortsetzung von Beispiel 14.4)

Das Integral von $f: f(x)=1/(1+x^2)$ über $[0,1]$ ist mit $h=0.2$ mit der globalen Fehlerordnung $O(h^4)$ zu berechnen.

Lösung:

Es sind $N=5$ Teilintervalle der Länge $h=0.2$, d. h. eine ungerade Anzahl. Man kann also die Simpsonsche Formel, die die geforderte Fehlerordnung besitzt, nicht allein anwenden. Die 3/8-Formel ist ebenfalls nicht allein anwendbar, da die Anzahl der Teilintervalle nicht durch 3 teilbar ist. Man kann aber beide Formeln kombinieren, indem man z. B. die Simpsonsche Formel auf $[0,0.4]$ und die 3/8-Formel auf $[0.4,1]$ anwendet.

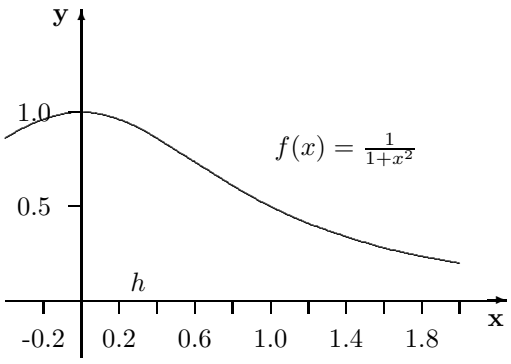
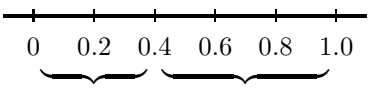


Abb. 14.6.

Man erhält



$$Q(f;0,1) \quad = \quad Q^S(f;0,0.4) \; + \; Q^{3/8}(f;0.4,1)$$

Die Wertetabelle für $f : f(x) = 1/(1+x^2)$ mit 6-stelliger Mantisse lautet:

i	0	1	2	3	4	5
x_i	0	0.2	0.4	0.6	0.8	1.0
$f(x_i)$	1.00000	0.961538	0.862069	0.735294	0.609756	0.500000

$$\begin{aligned}
Q^S(f;0,0.4) &= \frac{0.2}{3}\big(f(0)+4f(0.2)+f(0.4)\big) \\
&= \frac{0.2}{3}(1.00000+4\cdot 0.961538+0.862069) = 0.380548\,, \\
Q^{3/8}(f;0.4,1) &= \frac{3}{8}\,0.2\,\big(f(0.4)+3f(0.6)+3f(0.8)+f(1.0)\big) \\
&= \frac{3}{8}\,0.2\,(0.862069+3\cdot 0.735294+3\cdot 0.609756+0.500000) \\
&= 0.404791\,,
\end{aligned}$$

$$\Longrightarrow \qquad Q(f;0,1) = 0.380548 + 0.404791 = 0.785339\,.$$

Der exakte Werte des Integrals ist $\frac{\pi}{4}$, damit ist der Fehler $\frac{\pi}{4} - Q(f;0,1) = 0.59\cdot 10^{-4}$ □

Beispiel 14.13.

Gegeben: Die Funktion $f(x) = \sqrt{1-k^2\sin^2 x},\ k = 0.75$

Gesucht: Ergebnisse der Quadratur von f im Intervall $[0, \pi/2]$ zu verschiedenen Genauigkeitsschranken mit Hilfe der 3/8-Newton-Cotes-Formel.

Lösung:

Genauigkeit	$5 \cdot 10^{-3}$	$5 \cdot 10^{-6}$	$5 \cdot 10^{-9}$	$5 \cdot 10^{-12}$
Funktionsausw.	2	4	8	8
Ergebnis	1.31849377160	1.31847212047	1.31847210799	1.31847210799
Schätzwert	$-1.11 \cdot 10^{-4}$	$-2.12 \cdot 10^{-7}$	$-5.16 \cdot 10^{-13}$	$-5.16 \cdot 10^{-13}$

□

Summierte 3/8-Formel bei nichtäquidistanter Zerlegung

Mit der Zerlegung:

$$Z : \alpha = t_0 < t_1 < t_2 < \ldots < t_m = \beta$$

des Integrationsintervalls $[\alpha,\beta]$ und $h_k := t_{k+1} - t_k$, $h_{\max} = \max\limits_{0 \le k \le m-1} \{h_k\}$, erhält man die summierte 3/8 Formel

$$\begin{aligned}
\int_\alpha^\beta f(x)\,\mathrm{d}x &= Q_{h_k}^{3/8}(f;\alpha,\beta) + E_{h_k}^{3/8}(f;\alpha,\beta) \quad \text{mit} \\
Q_{h_k}^{3/8}(f;\alpha,\beta) &= \frac{3}{8} \sum_{k=0}^{m-1} (t_{k+1} - t_k) \Big[f(t_k) + 3f\left(\frac{2t_k + t_{k+1}}{3}\right) \\
&\quad + 3f\left(\frac{t_k + 2t_{k+1}}{3}\right) + f(t_{k+1}) \Big] \\
E_{h_k}^{3/8}(f;\alpha,\beta) &= O(h_{\max}^4).
\end{aligned}$$

14.3.4 Weitere Newton-Cotes-Formeln

Bisher wurden drei Newton-Cotes-Formeln angegeben, die sich jeweils durch Integration des Interpolationspolynoms für f zu 2 bzw. 3 bzw. 4 Stützstellen ergaben. Hier werden vier weitere Formeln angegeben zu 5,6,7 und 8 Stützstellen. Diese Formeln werden sofort zusammen mit den Restgliedern aufgeschrieben, so dass sich folgende Regeln ergeben:

4/90-Regel (5 Stützstellen für das Referenzintervall $[0, 4h]$)

$$\begin{aligned}
\int_0^{4h} f(x)\,\mathrm{d}x &= \frac{4h}{90}\big(7f(0) + 32f(h) + 12f(2h) + 32f(3h) + 7f(4h)\big) \\
&\quad - \frac{8h^7}{945} f^{(6)}(\xi^*), \quad \xi^* \in [0, 4h], \quad f^{(6)} \in C[0, 4h].
\end{aligned}$$

Summierte 4/90-Regel bei äquidistanter Zerlegung Mit $h = \frac{(\beta-\alpha)}{4N}$ ist

$$\begin{aligned}\int_\alpha^\beta f(x)\,\mathrm{d}x &= \frac{4h}{90}\Big(7f(\alpha)+7f(\beta)+32\sum_{k=1}^{N} f\big(\alpha+(4k-3)h\big)\\ &+12\sum_{k=1}^{N} f\big(\alpha+(4k-2)h\big)+32\sum_{k=1}^{N} f\big(\alpha+(4k-1)h\big)\\ &+14\sum_{k=1}^{N-1} f(\alpha+4kh)\Big)-\frac{2(\beta-\alpha)}{945}h^6 f^{(6)}(\eta),\\ &\eta\in[\alpha,\beta],\quad f^{(6)}\in C[\alpha,\beta].\end{aligned}$$

5/288-Regel für das Referenzintervall [0,5h] (6 Stützstellen)

$$\begin{aligned}\int_0^{5h} f(x)\,\mathrm{d}x &= \frac{5h}{288}\big(19f(0)+75f(h)+50f(2h)+50f(3h)+75f(4h)\\ &+19f(5h)\big)-\frac{275}{12096}h^7 f^{(6)}(\xi^*),\\ &\xi^*\in[0,5h],\quad f^{(6)}\in C[0,5h].\end{aligned}$$

Summierte 5/288-Regel bei äquidistanter Zerlegung Mit $h = \frac{(\beta-\alpha)}{5N}$ ist

$$\begin{aligned}\int_\alpha^\beta f(x)\,\mathrm{d}x &= \frac{5h}{288}\Big(19f(\alpha)+19f(\beta)+75\sum_{k=1}^{N} f\big(\alpha+(5k-4)h\big)\\ &+50\sum_{k=1}^{N} f\big(\alpha+(5k-3)h\big)+50\sum_{k=1}^{N} f\big(\alpha+(5k-2)h\big)\\ &+75\sum_{k=1}^{N} f\big(\alpha+(5k-1)h\big)+38\sum_{k=1}^{N-1} f(\alpha+5kh)\Big)\\ &-\frac{55(\beta-\alpha)}{12096}h^6 f^{(6)}(\eta),\quad \eta\in[\alpha,\beta],\quad f^{(6)}\in C[\alpha,\beta].\end{aligned}$$

6/840-Regel für das Referenzintervall [0,6h] (7 Stützstellen)

$$\begin{aligned}\int_0^{6h} f(x)\,\mathrm{d}x &= \frac{6h}{840}\big(41f(0)+216f(h)+27f(2h)+272f(3h)\\ &+27f(4h)+216f(5h)+41f(6h)\big)-\frac{9}{1400}h^9 f^{(8)}(\xi^*),\\ &\xi^*\in[0,6h],\quad f^{(8)}\in C[0,6h].\end{aligned}$$

Summierte 6/840-Regel bei äquidistanter Zerlegung Mit $h = \frac{(\beta-\alpha)}{6N}$ ist

$$\begin{aligned}\int_\alpha^\beta f(x)\,\mathrm{d}x &= \frac{6h}{840}\Big(41f(\alpha)+41f(\beta)+216\sum_{k=1}^{N} f\big(\alpha+(6k-5)h\big)\\ &+27\sum_{k=1}^{N} f\big(\alpha+(6k-4)h\big)+272\sum_{k=1}^{N} f\big(\alpha+(6k-3)h\big)\\ &+27\sum_{k=1}^{N} f\big(\alpha+(6k-2)h\big)+216\sum_{k=1}^{N} f\big(\alpha+(6k-1)h\big)\\ &+82\sum_{k=1}^{N-1} f(\alpha+6kh)\Big)-\frac{3(\beta-\alpha)}{2800}h^8 f^{(8)}(\eta),\\ &\eta\in[\alpha,\beta],\quad f^{(8)}\in C[\alpha,\beta].\end{aligned}$$

7/17280-Regel für das Referenzintervall [0,7h] (8 Stützstellen)

$$\begin{aligned}\int_0^{7h} f(x)\,\mathrm{d}x &= \frac{7h}{17280}\big(751f(0)+3577f(h)+1323f(2h)+2989f(3h)\\ &\quad +2989f(4h)+1323f(5h)+3577f(6h)+751f(7h)\big)\\ &\quad -\frac{8163}{518400}h^9 f^{(8)}(\xi^*),\quad \xi^*\in[0,7h],\quad f^{(8)}\in C[0,7h].\end{aligned}$$

Summierte 7/17280-Regel bei äquidistanter Zerlegung Mit $h=\frac{(\beta-\alpha)}{7N}$ ist

$$\begin{aligned}\int_\alpha^\beta f(x)\,\mathrm{d}x &= \frac{7h}{17280}\Big(751f(\alpha)+751f(\beta)+3577\sum_{k=1}^{N} f\big(\alpha+(7k-6)h\big)\\ &\quad +1323\sum_{k=1}^{N} f\big(\alpha+(7k-5)h\big)+2989\sum_{k=1}^{N} f\big(\alpha+(7k-4)h\big)\\ &\quad +2989\sum_{k=1}^{N} f\big(\alpha+(7k-3)h\big)+1323\sum_{k=1}^{N} f\big(\alpha+(7k-2)h\big)\\ &\quad +3577\sum_{k=1}^{N} f\big(\alpha+(7k-1)h\big)+1502\sum_{k=1}^{N-1} f(\alpha+7kh)\Big)\\ &\quad -\frac{8163(\beta-\alpha)}{3628800}h^8 f^{(8)}(\eta),\quad \eta\in[\alpha,\beta],\quad f^{(8)}\in C[\alpha,\beta].\end{aligned}$$

Eine Herleitung der Restglieder aller Newton-Cotes-Formeln ist in [FADD1979] Bd.1, 3.4.2; [ISAA1973], S.323/4; [KRYL1991], 6.1 und [WILL1971], S.144-146 zu finden.

Beispiel 14.14.

Gegeben: Die Funktion $f(x)=\sqrt{1-k^2\sin^2 x},\ k=0.75$

Gesucht: Ergebnisse der Quadratur von f im Intervall $[0,\pi/2]$ zu verschiedenen Genauigkeitsschranken mit Hilfe der 4/90-Newton-Cotes-Formel .

Lösung:

Genauigkeit	$5\cdot10^{-3}$	$5\cdot10^{-6}$	$5\cdot10^{-9}$	$5\cdot10^{-12}$
Funktionsausw.	2	2	8	8
Ergebnis	1.31846830103	1.31846830103	1.31847210799	1.31847210799
Schätzwert	$3.51\cdot10^{-6}$	$3.51\cdot10^{-6}$	$5.28\cdot10^{-14}$	$5.28\cdot10^{-14}$

□

Beispiel 14.15.

Gegeben: Die Funktion $f(x)=\sqrt{1-k^2\sin^2 x},\ k=0.75$

Gesucht: Ergebnisse der Quadratur von f im Intervall $[0,\pi/2]$ zu verschiedenen Genauigkeitsschranken mit Hilfe der 5/288-Newton-Cotes-Formel.

Lösung:

Genauigkeit	$5 \cdot 10^{-3}$	$5 \cdot 10^{-6}$	$5 \cdot 10^{-9}$	$5 \cdot 10^{-12}$
Funktionsausw.	2	2	8	8
Ergebnis	1.31847005077	1.31847005077	1.31847210799	1.31847210799
Schätzwert	$2.01 \cdot 10^{-6}$	$2.01 \cdot 10^{-6}$	$2.87 \cdot 10^{-14}$	$2.87 \cdot 10^{-14}$

□

Beispiel 14.16.

Gegeben: Die Funktion $f(x) = \sqrt{1 - k^2 \sin^2 x},\ k = 0.75$

Gesucht: Ergebnisse der Quadratur von f im Intervall $[0, \pi/2]$ zu verschiedenen Genauigkeitsschranken mit Hilfe der 6/840-Newton-Cotes-Formel.

Lösung:

Genauigkeit	$5 \cdot 10^{-3}$	$5 \cdot 10^{-6}$	$5 \cdot 10^{-9}$	$5 \cdot 10^{-12}$
Funktionsausw.	2	2	4	8
Ergebnis	1.31847230107	1.31847230107	1.31847210811	1.31847210799
Schätzwert	$1.80 \cdot 10^{-8}$	$1.80 \cdot 10^{-8}$	$-4.85 \cdot 10^{-10}$	$-1.74 \cdot 10^{-15}$

□

Beispiel 14.17.

Gegeben: Die Funktion $f(x) = \sqrt{1 - k^2 \sin^2 x},\ k = 0.75$

Gesucht: Ergebnisse der Quadratur von f im Intervall $[0, \pi/2]$ zu verschiedenen Genauigkeitsschranken mit Hilfe der 7/17280-Newton-Cotes-Formel.

Lösung:

Genauigkeit	$5 \cdot 10^{-3}$	$5 \cdot 10^{-6}$	$5 \cdot 10^{-9}$	$5 \cdot 10^{-12}$
Funktionsausw.	2	2	4	8
Ergebnis	1.31847222443	1.31847222443	1.31847210807	1.31847210799
Schätzwert	$8.14 \cdot 10^{-9}$	$8.14 \cdot 10^{-9}$	$-2.91 \cdot 10^{-10}$	$-9.30 \cdot 10^{-16}$

□

14.3.5 Zusammenfassung zur Fehlerordnung von Newton-Cotes-Formeln

Sei $n+1$ die Anzahl der Stützstellen x_k im Referenzintervall und sei f in $[\alpha,\beta]$ genügend oft stetig differenzierbar. Die lokale Fehlerordnung sei $O(h^q)$.

Dann gelten folgende Aussagen:

1. *Für gerades n+1:* $q = n + 2$, d. h. es werden Polynome bis zum n-ten Grade exakt integriert, die globale Fehlerordnung ist $O(h^{n+1})$.

   Beispiel 1 (ST-Formel): $n + 1 = 2$ Stützstellen, d. h. $n = 1$, lokale Fehlerordnung $O(h^3)$, globale Fehlerordnung $O(h^2)$.

   Beispiel 2 (3/8-Formel): $n + 1 = 4$ Stützstellen, d. h. $n = 3$, lokale Fehlerordnung $O(h^5)$, globale Fehlerordnung $O(h^4)$.

2. *Für ungerades n+1:* $q = n+3$, d. h. es werden Polynome bis zum $(n+1)$-ten Grade exakt integriert, die globale Fehlerordnung ist $O(h^{n+2})$.

   Beispiel (S-Formel): $n + 1 = 3$, d. h. $n = 2$, daraus ergibt sich die lokale Fehlerordnung $O(h^5)$, die globale Fehlerordnung $O(h^4)$. Zur Fehlerschätzung siehe Abschnitt 14.11.

Die genannten Newton-Cotes-Formeln sind Formeln vom geschlossenen Typ; Newton-Cotes-Formeln vom offenen Typ sind in [CARN1990], S.75 zu finden. Bei wachsendem Grad des integrierten Interpolationspolynoms, d. h. bei wachsender Anzahl (>8) verwendeter Stützstellen, treten negative Gewichte auf, so dass die Quadraturkonvergenz nicht mehr gesichert ist (s. Abschnitt 14.13). Außerdem differieren die Koeffizienten bei zunehmendem Grad immer stärker voneinander, was zum unerwünschten Anwachsen von Rundungsfehlern führen kann. Deshalb werden zur Integration über große Intervalle anstelle von Formeln höherer Ordnung besser summierte Formeln niedrigerer Fehlerordnung mit hinreichend feiner Zerlegung oder ein anderes Verfahren verwendet.

14.4 Quadraturformeln von Maclaurin

Bei den Formeln von Maclaurin liegen die Stützstellen jeweils in der Mitte eines Teilintervalls der Länge h, es sind Formeln vom offenen Typ. Gewichte und Restglieder können z. B. mittels Taylorabgleich bestimmt werden.

14.4.1 Die Tangententrapezformel

Tangententrapezformel für das Referenzintervall [0,h]

Betrachtet man das Integral von f über das Referenzintervall $[a,b]=[0,h]$, wählt nur eine Stützstelle x_0 in $[0,h]$ und fordert, dass diese möglichst günstig liegt, so dass sich Polynome vom Grad 0 und 1 exakt integrieren lassen, so ergeben sich aus (14.4) mit $n=1$ und $m=0,1$ die Gleichungen $A_0=h$ und $A_0\,x_0=h^2/2$ mit den Lösungen $A_0=h$, $x_0=h/2$. Die Quadraturformel (14.3) lautet somit

$$Q^{TT}(f;0,h)=A_0 f(x_0)=hf(h/2).$$

$\mathrm{Q}^{TT}(f;0,h)$ heißt *Tangententrapezformel* (TT-Formel), da sie geometrisch den Flächeninhalt des Trapezes bedeutet, dessen vierte Seite von der Tangente an $f(x)$ im Punkt $\big(h/2,f(h/2)\big)$ gebildet wird.

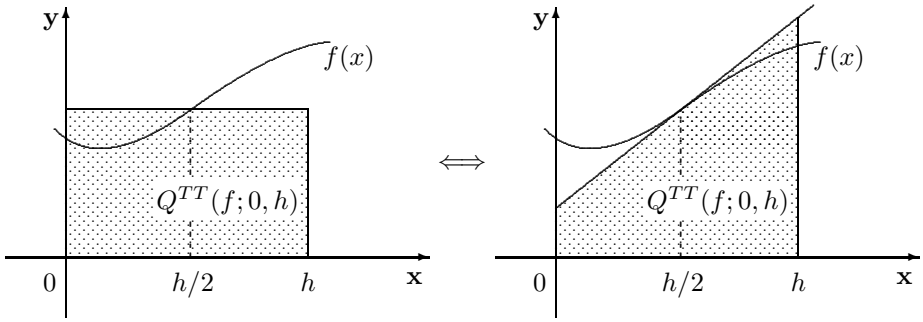


Abb. 14.7. Zur Tangententrapezformel

Das Gewicht A_0 und das Restglied $E^{TT}(f;0,h)$ können auch durch Taylorabgleich bestimmt werden. Dazu entwickelt man f an der Stelle $x=\frac{h}{2}$:

$$f(x)=f\left(\frac{h}{2}\right)+\left(x-\frac{h}{2}\right)f'\left(\frac{h}{2}\right)+\left(x-\frac{h}{2}\right)^2\frac{1}{2}f''(\xi)\,,\quad \xi=\xi(x)\in[0,h] \tag{14.18}$$

und integriert über $[0,h]$:

$$\begin{aligned}\int_0^h f(x)\,\mathrm{d}x &= f\left(\frac{h}{2}\right)\int_0^h \mathrm{d}x+f'\left(\frac{h}{2}\right)\int_0^h\left(x-\frac{h}{2}\right)\mathrm{d}x+\frac{1}{2}\int_0^h\left(x-\frac{h}{2}\right)^2 f''(\xi)\,\mathrm{d}x\\ &= A_0\,f\left(\frac{h}{2}\right)+E^{TT}(f;0,h)\,.\end{aligned}$$

Mit $\int_0^h\left(x-\frac{h}{2}\right)\mathrm{d}x=0$ und dem verallgemeinerten Mittelwertsatz erhält man

$$\begin{aligned}A_0\,f\left(\frac{h}{2}\right)+E^{TT}(f;0,h) &= h\,f\left(\frac{h}{2}\right)+\frac{1}{2}\,f''(\xi^*)\int_0^h\left(x-\frac{h}{2}\right)^2\mathrm{d}x\\ &= h\,f\left(\frac{h}{2}\right)+\frac{h^3}{24}\,f''(\xi^*)\,,\quad \xi^*\in[0,h]\,.\end{aligned}$$

Für das zugehörige Restglied folgt

$$E^{TT}(f;0,h) = \frac{h^3}{24} f''(\xi^*), \quad \xi^* \in [0,h], \quad f'' \in C[0,h].$$

Die lokale Fehlerordnung ist somit $O(h^3)$.

Zusammengefasst folgt die *Tangententrapezregel* für das Referenzintervall $[0,h]$

$$\begin{aligned} \int_0^h f(x)\,\mathrm{d}x &= Q^{TT}(f;0,h) + E^{TT}(f;0,h) \quad \text{mit} \\ Q^{TT}(f;0,h) &= h f\Big(\frac{h}{2}\Big), \\ E^{TT}(f;0,h) &= \frac{h^3}{24} f''(\xi^*), \quad \xi^* \in [0,h], \quad f'' \in C[0,h]. \end{aligned}$$

Bemerkung. Würde man die Entwicklung (14.18) schon nach dem linearen Glied abbrechen, dann würde die Integration auf ein Integral der Form

$$\int_0^h \left(x - \frac{h}{2}\right) f'(\xi)\,\mathrm{d}x\,, \quad \xi \in [0,h]\,,$$

führen, auf das der verallgemeinerte Mittelwertsatz nicht anwendbar ist, da $\left(x - \frac{h}{2}\right)$ in $[0,h]$ das Vorzeichen wechselt. Die Entwicklung bis zum quadratischen Glied führt wegen $\int_0^h \left(x - \frac{h}{2}\right)\,\mathrm{d}x = 0$ zum Erfolg.

Summierte Tangententrapezformel für äquidistante Zerlegung

Zur Bestimmung des Integrals von f über ein ausgedehntes Intervall $[\alpha,\beta]$ zerlegt man $[\alpha,\beta]$ in N Teilintervalle der Länge $h = (\beta-\alpha)/N$, so dass die *summierte Tangententrapezregel* bei äquidistanter Zerlegung lautet

$$\begin{aligned} \int_\alpha^\beta f(x)\,\mathrm{d}x &= Q_h^{TT}(f;\alpha,\beta) + E_h^{TT}(f;\alpha,\beta) \quad \text{mit} \\ Q_h^{TT}(f;\alpha,\beta) &= h \sum_{k=0}^{N-1} f\left(\alpha + (2k+1)\frac{h}{2}\right), \\ E_h^{TT}(f;\alpha,\beta) &= \frac{h^2}{24}(\beta-\alpha) f''(\eta), \quad \eta \in [\alpha,\beta], f'' \in C[\alpha,\beta]. \end{aligned}$$

Die globale Fehlerordnung ist somit $O(h^2)$.

Beispiel 14.18. (Fortsetzung von Beispiel 14.5)

Gegeben: Das Integral

$$I\left(\frac{\sin x}{x}; 0, \frac{\pi}{2}\right) = \int_0^{\frac{\pi}{2}} \frac{\sin x}{x}\,\mathrm{d}x$$

Gesucht: Der Näherungswert Q^{TT} für das Integral mit Hilfe der summierten Tangententrapezformel für $h = \dfrac{\pi}{12}$.

Lösung: Mit $f(x) = (\sin x)/x$ ist

$$Q^{TT}_{\pi/12}\left(\tfrac{\sin x}{x}; 0, \tfrac{\pi}{2}\right) = \tfrac{\pi}{12}\left(f\left(\tfrac{\pi}{24}\right) + f\left(\tfrac{3\pi}{24}\right) + f\left(\tfrac{5\pi}{24}\right) + f\left(\tfrac{7\pi}{24}\right) + f\left(\tfrac{9\pi}{24}\right) + f\left(\tfrac{11\pi}{24}\right)\right)$$

$$= \tfrac{\pi}{12}\Big(0.9971467 + 0.9744954 + 0.9301189 + 0.8658247 + 0.7842133 + 0.6885528\Big)$$

$$\approx 1.371920892$$

Der exakte Integralwert ist $I_{ex} = 1.370\,762\,168$. □

Summierte Tangententrapezformel für nichtäquidistante Zerlegung

Mit der Zerlegung

$$Z : \alpha = t_0 < t_1 < t_2 < \ldots < t_N = \beta,$$

$h_k = \quad t_{k+1} - t_k, h_{\max} := \max\limits_{0 \le k \le N-1}\{h_k\}, \quad$ erhält man

$$\int_\alpha^\beta f(x)\,\mathrm{d}x = Q^{TT}_{h_k}(f;\alpha,\beta) + E^{TT}_{h_k}(f;\alpha,\beta) \quad \text{mit}$$

$$Q^{TT}_{h_k}(f;\alpha,\beta) = \sum_{k=0}^{N-1}(t_{k+1} - t_k) f\left(\frac{t_k + t_{k+1}}{2}\right),$$

$$E^{TT}_{h_k}(f;\alpha,\beta) = O(h^2_{\max}).$$

Bemerkung. Die beiden Trapezformeln (ST und TT) sind von derselben Fehlerordnung. Der Restgliedkoeffizient der TT-Formel ist nur halb so groß wie der der ST-Formel. Außerdem ist bei der Integration nach der TT-Formel stets ein Funktionswert weniger zu berechnen, da als Stützstellen die Intervallmitten genommen werden.

14.4.2 Weitere Maclaurin-Formeln

Im Folgenden werden noch die Formeln für 2,3,4 und 5 Stützstellen zusammen mit den zugehörigen Restgliedern als Integrationsregeln angegeben:

Regel zu zwei Stützstellen für das Referenzintervall [0,2h]

$$\int_0^{2h} f(x)\,\mathrm{d}x = h\left(f\left(\frac{h}{2}\right) + f\left(\frac{3h}{2}\right)\right) + \frac{h^3}{12} f''(\xi^*), \quad \xi^* \in [0,2h], \quad f'' \in C[0,2h].$$

Summierte Regel bei äquidistanter Zerlegung Mit $h = \frac{(\beta-\alpha)}{2N}$ ist

$$\int_\alpha^\beta f(x)\,\mathrm{d}x = h\sum_{k=1}^{2N} f\left(\alpha + (2k-1)\frac{h}{2}\right) + \frac{\beta-\alpha}{24} h^2 f''(\eta), \quad \begin{matrix}\eta \in [\alpha,\beta],\\ f'' \in C[\alpha,\beta].\end{matrix}$$

Regel zu drei Stützstellen für das Referenzintervall [0,3h]

$$\int_0^{3h} f(x)\,\mathrm{d}x = \frac{3h}{8}\left(3f\left(\frac{h}{2}\right)+2f\left(\frac{3h}{2}\right)+3f\left(\frac{5h}{2}\right)\right)+\frac{21}{640}h^5 f^{(4)}(\xi^*),$$
$$\xi^* \in [0,3h], \quad f^{(4)} \in C[0,3h], \quad \text{vgl. [LAUX1988].}$$

Summierte Regel bei äquidistanter Zerlegung Mit $h = \frac{(\beta-\alpha)}{3N}$ ist

$$\begin{aligned}\int_\alpha^\beta f(x)\,\mathrm{d}x &= \frac{3h}{8}\sum_{k=1}^{N-1}\left(3f\left(\alpha+(6k+1)\frac{h}{2}\right)+2f\left(\alpha+(6k+3)\frac{h}{2}\right)\right.\\ &\left.+3f\left(\alpha+(6k+5)\frac{h}{2}\right)\right)+\frac{1701}{20480}(\beta-\alpha)h^4 f^{(4)}(\eta),\\ &\eta\in[\alpha,\beta], \quad f^{(4)}\in C[\alpha,\beta].\end{aligned}$$

Regel zu vier Stützstellen für das Referenzintervall [0,4h]

$$\begin{aligned}\int_0^{4h} f(x)\,\mathrm{d}x &= \frac{h}{12}\left(13f\left(\frac{h}{2}\right)+11f\left(\frac{3h}{2}\right)+11f\left(\frac{5h}{2}\right)+13f\left(\frac{7h}{2}\right)\right)\\ &+\frac{103}{1440}h^5 f^{(4)}(\xi^*), \quad \xi^*\in[0,4h], \quad f^{(4)}\in C[0,4h].\end{aligned}$$

Summierte Regel bei äquidistanter Zerlegung Mit $h = \frac{(\beta-\alpha)}{4N}$ ist

$$\begin{aligned}\int_\alpha^\beta f(x)\,\mathrm{d}x &= \frac{h}{12}\sum_{k=0}^{N-1}\left(13f\left(\alpha+(8k+1)\frac{h}{2}\right)+11f\left(\alpha+(8k+3)\frac{h}{2}\right)\right.\\ &\left.+11f\left(\alpha+(8k+5)\frac{h}{2}\right)+13f\left(\alpha+(8k+7)\frac{h}{2}\right)\right)\\ &+\frac{103}{5760}(\beta-\alpha)h^4 f^{(4)}(\eta), \quad \eta\in[\alpha,\beta], \quad f^{(4)}\in C[\alpha,\beta].\end{aligned}$$

Regel zu fünf Stützstellen für das Referenzintervall [0,5h]

$$\begin{aligned}\int_0^{5h} f(x)\,\mathrm{d}x &= \frac{5h}{1152}\left(275f\left(\frac{h}{2}\right)+100f\left(\frac{3h}{2}\right)+402f\left(\frac{5h}{2}\right)+100f\left(\frac{7h}{2}\right)\right.\\ &\left.+275f\left(\frac{9h}{2}\right)\right)+\frac{435\,546\,875}{3170\,893\,824}h^7 f^{(6)}(\xi^*),\\ &\xi^*\in[0,5h], \quad f^{(6)}\in C[0,5h].\end{aligned}$$

Summierte Regel bei äquidistanter Zerlegung Mit $h = \frac{(\beta-\alpha)}{5N}$ ist

$$\begin{aligned}\int_\alpha^\beta f(x)\,\mathrm{d}x &= \frac{5h}{1152}\sum_{k=0}^{N-1}\left(275f\left(\alpha+(10k+1)\frac{h}{2}\right)+100f\left(\alpha+(10k+3)\frac{h}{2}\right)\right.\\ &+402f\left(\alpha+(10k+5)\frac{h}{2}\right)+100f\left(\alpha+(10k+7)\frac{h}{2}\right)\\ &\left.+275f\left(\alpha+(10k+9)\frac{h}{2}\right)\right)+\frac{87\,109\,375}{3170\,893\,824}(\beta-\alpha)h^6 f^{(6)}(\eta),\\ &\eta\in[\alpha,\beta], \quad f^{(6)}\in C[\alpha,\beta].\end{aligned}$$

Aus der Aufstellung ist erkennbar, dass die Formeln mit ungerader Stützstellenzahl ebenso wie bei den Newton-Cotes-Formeln die günstigeren Formeln sind. Die Formel für $n = 6$ wird nicht mehr angegeben, da sie dieselbe Fehlerordnung hat wie die für $n = 5$. In der Formel für $n = 7$ ist bereits ein negatives Gewicht (nämlich A_0), so dass die Quadraturkonvergenz nicht mehr gesichert ist.

14.5 Die Euler-Maclaurin-Formeln

Euler-Maclaurin-Formeln für das Referenzintervall [0,h]

Die Euler-Maclaurin-Formeln entstehen durch Integration der Newtonschen Interpolationsformel $\widetilde{N}_+(t)$ für absteigende Differenzen (vgl. Abschnitt9.2). Es sei f $2n$-mal stetig differenzierbar auf dem Referenzintervall $[0, h]$. Betrachtet man das Integral von f über $[0, h]$ und wählt als Stützstellen $x_0 = 0$, $x_1 = h$, so ergibt sich für jedes $n \in \mathbb{N}$ mit $f \in C^{2n}[0, h]$ eine *Euler-Maclaurin-Formel* (EM_n-Formel)

$$Q^{EM_n}(f; 0, h) = \frac{h}{2}\big(f(0) + f(h)\big) + \sum_{j=1}^{n-1} \frac{B_{2j}}{(2j)!} h^{2j} \left(f^{(2j-1)}(0) - f^{(2j-1)}(h)\right) \tag{14.19}$$

mit den Bernoullischen Zahlen

$$B_0 = 1, B_1 = -\frac{1}{2}, B_2 = \frac{1}{6}, B_4 = -\frac{1}{30}, B_6 = \frac{1}{42}, \ldots; \quad B_{2j+1} = 0 \text{ für } j = 1, 2, \ldots.$$

Das zugehörige Restglied lautet

$$E^{EM_n}(f; 0, h) = -\frac{B_{2n}}{(2n)!} h^{2n+1} f^{(2n)}(\xi^*), \quad \xi^* \in [0, h], \tag{14.20}$$

d. h. die lokale Fehlerordnung ist $O(h^{2n+1})$.

Zusammengefasst folgt mit (14.19) und (14.20) für jedes n eine *Euler-Maclaurin-Regel*

$$\int_0^h f(x)\,\mathrm{d}x = Q^{EM_n}(f; 0, h) + E^{EM_n}(f; 0, h).$$

Summierte Euler-Maclaurin-Formeln für äquidistante Zerlegung

Ist die Integration über ein ausgedehntes Intervall $[\alpha,\beta]$ zu erstrecken, so zerlegt man $[\alpha,\beta]$ in N Teilintervalle der Länge $h = (\beta - \alpha)/N$ und wendet eine EM_n-Formel und das zugehörige Restglied auf jedes Teilintervall an. Man erhält so die *summierte Euler-Maclaurin-Regel*

$$\int_\alpha^\beta f(x)\,\mathrm{d}x = Q_h^{EM_n}(f; \alpha, \beta) + E_h^{EM_n}(f; \alpha, \beta),$$

mit der *summierten Euler-Maclaurin-Formel*

$$\begin{aligned} Q_h^{EM_n}(f;\alpha,\beta) &= \frac{h}{2}\left(f(\alpha)+2\sum_{\nu=1}^{N-1} f(\alpha+\nu h)+f(\beta)\right) \\ &+ \sum_{j=1}^{n-1}\frac{B_{2j}}{(2j)!}h^{2j}\left(f^{(2j-1)}(\alpha)-f^{(2j-1)}(\beta)\right) \end{aligned} \tag{14.21}$$

und dem *Restglied der summierten Euler-Maclaurin-Formel*

$$E_h^{EM_n}(f;\alpha,\beta) = -\frac{\beta-\alpha}{(2n)!}B_{2n}h^{2n}f^{(2n)}(\eta), \quad \eta\in[\alpha,\beta],$$

die globale Fehlerordnung ist somit $O(h^{2n})$.

Bemerkung. Mit der Sehnentrapezformel kann man für (14.19) auch schreiben

$$Q^{EM_n}(f;0,h) = Q^{ST}(f;0,h)+\sum_{k=1}^{n-1}\widetilde{c}_{2k}h^{2k}$$

und mit der summierten Sehnentrapezformel für (14.21)

$$Q_h^{EM_n}(f;\alpha,\beta) = Q_h^{ST}(f;\alpha,\beta)+\sum_{k=1}^{n-1}c_{2k}h^{2k}, \tag{14.22}$$

wobei die $\widetilde{c}_{2k}$ und c_{2k} unabhängig von h sind. Die einfache und summierte Euler-Maclaurin-Formel setzt sich also aus der einfachen bzw. summierten Sehnentrapezformel und einem Korrekturglied zusammen. Für $n=1$ sind die ST-Formel und die EM_n-Formel identisch. Wegen der Notwendigkeit der Berechnung von Ableitungen haben die Euler-Maclaurin-Formeln weniger praktische Bedeutung.

Beispiel 14.19.

Gegeben: Das Integral

$$I\left(\frac{\sin x}{x};0,\frac{\pi}{2}\right) = \int_0^{\frac{\pi}{2}}\frac{\sin x}{x}\,\mathrm{d}x.$$

Gesucht: Der Näherungswert für das Integral mit der Euler-Maclaurin-Formel für $n=3$.

Lösung: Diese Formel lautet

$$Q^{EM_3}(f;0,h) = \frac{h}{2}\big(f(0)+f(h)\big)+\frac{h^2}{12}\big(f'(0)-f'(h)\big)-\frac{h^4}{720}\big(f'''(0)-f'''(h)\big).$$

Mit

$$\begin{aligned} f(x) &= \frac{\sin x}{x}, \\ f'(x) &= \frac{\cos x}{x}-\frac{\sin x}{x^2} \quad \text{und} \\ f'''(x) &= \left(\frac{3}{x^2}-\frac{6}{x^4}\right)\sin x-\left(\frac{1}{x}-\frac{6}{x^3}\right)\cos x \end{aligned}$$

ergeben sich

$$f(0) = 1\,, \qquad f'(0) = f'''(0) = 0$$
$$f\left(\frac{\pi}{2}\right) = \frac{2}{\pi}\,, \quad f'\left(\frac{\pi}{2}\right) = -\frac{4}{\pi^2}\,, \quad f'''\left(\frac{\pi}{2}\right) = \frac{12}{\pi^2} - \frac{96}{\pi^4}\,.$$

Damit ist

$$\begin{aligned} Q^{EM_3}\left(\frac{\sin x}{x}; 0, \frac{\pi}{2}\right) &= \frac{\pi}{4}\left(1 + \frac{2}{\pi}\right) + \frac{\pi^2}{48}\,\frac{4}{\pi^2} - \frac{\pi^4}{11520}\left(\frac{96}{\pi^4} - \frac{12}{\pi^2}\right) \\ &= 1.370\,679\,001\,. \end{aligned}$$

Der exakte Wert des Integrals ist $I_{ex} = 1.370\,762\,168$. □

14.6 Tschebyscheffsche Quadraturformeln

Bei der Konstruktion aller bisher behandelten Quadraturformeln vom Typ (14.3) wurden die $n+1$ Stützstellen $x_k \in [a,b]$ vorgegeben und die Gewichte A_k als Lösungen des für sie linearen Gleichungssystems (14.4) erhalten. Sind die Funktionswerte $f(x)$ des Integranden mit Rundungsfehlern behaftet, so wird der dadurch bedingte Fehler des Integralwertes am kleinsten, wenn alle Gewichte der Quadraturformel gleich sind. Die Tschebyscheffschen Formeln haben die Form (14.3) mit gleichen Gewichten.

Tschebyscheff-Formeln für das Referenzintervall [−h,h]

Man betrachtet das Integral von f über das Referenzintervall $[-h,h]$ und setzt die *Tschebyscheffschen Regeln* in der Form an:

$$I(f;-h,h) = \int_{-h}^{h} f(x)\,\mathrm{d}x = Q^{Ch_{n+1}}(f;-h,h) + E^{Ch_{n+1}}(f;-h,h),$$

wobei $n+1$ die Anzahl der Stützstellen $x_k \in [-h,h]$ ist. $Q^{Ch_{n+1}}(f;-h,h)$ heißt *Tschebyscheffsche Formel* (Ch_{n+1}-Formel) zu $n+1$ Stützstellen und $E^{Ch_{n+1}}(f;-h,h)$ ist das *Restglied der Ch_{n+1}-Formel*. Die Gewichte A_k werden gleich groß vorgegeben:

$$A_k = \frac{2h}{n+1}, \quad k = 0(1)n.$$

Es wird gefordert, dass die Quadraturformel $Q^{Ch_{n+1}}(f;-h,h)$ Polynome bis zum Grad $M = n+1$ exakt integriert. So erhält man aus (14.4) mit $m = 1(1)n+1$ für die $n+1$ Stützstellen x_k $n+1$ nichtlineare Gleichungen. Es muss also vorausgesetzt werden, dass sich die Funktionswerte $f(x)$ an den Stützstellen x_k berechnen oder aus einer Tabelle ablesen lassen; ist von f nur eine Wertetabelle bekannt, so sind die Tschebyscheffschen Formeln im Allgemeinen nicht anwendbar.

Für $n = 1$ sind in (14.4) $a = -h$, $b = h$, $m = 1, 2$, und wegen $A_k = 2h/(n+1)$, $A_0 = A_1 = h$ zu setzen. Man erhält die Lösungen $x_0 = -h/\sqrt{3}$, $x_1 = h/\sqrt{3}$, so dass die zugehörige *Tschebyscheffsche Regel für 2 Stützstellen* lautet:

$$\begin{aligned}
\int_{-h}^{h} f(x)\,\mathrm{d}x &= Q^{Ch_2}(f;-h,h) + E^{Ch_2}(f;-h,h) \quad \text{mit} \\
Q^{Ch_2}(f;-h,h) &= A_0 f(x_0) + A_1 f(x_1) = h\big(f(-h/\sqrt{3}) + f/(h/\sqrt{3})\big), \\
E^{Ch_2}(f;-h,h) &= O(h^5);
\end{aligned}$$

sie besitzt die lokale Fehlerordnung $O(h^5)$.

Allgemein haben die Tschebyscheffschen Formeln mit 2ν und $2\nu+1$ Stützstellen die lokale Fehlerordnung $O(h^{2\nu+3})$. Die Restgliedkoeffizienten sind in [BERE1971] Bd.1, S.219 zu finden.

Tabelle der Stützstellenwerte für das Referenzintervall $[-h, h]$:

n	x_k für $k = 0(1)n$		
1	$x_{0,1} = \pm 0.577350\,h$		
2	$x_{0,2} = \pm 0.707107\,h$	$x_1 = 0$	
3	$x_{0,3} = \pm 0.794654\,h$	$x_{1,2} = \pm 0.187592\,h$	
4	$x_{0,4} = \pm 0.832498\,h$	$x_{1,3} = \pm 0.374541\,h$	$x_2 = 0$
5	$x_{0,5} = \pm 0.866247\,h$ $x_{2,3} = \pm 0.266635\,h$	$x_{1,4} = \pm 0.422519\,h$	
6	$x_{0,6} = \pm 0.883862\,h$ $x_{2,4} = \pm 0.323912\,h$	$x_{1,5} = \pm 0.529657\,h$ $x_3 = 0$	

Reelle Werte x_k ergeben sich nur für $n = 0(1)6$ und $n = 8$.

Summierte Tschebyscheff-Formeln für äquidistante Zerlegung.

Ist die Integration über ein ausgedehntes Intervall $[\alpha,\beta]$ zu erstrecken, so teilt man $[\alpha,\beta]$ in N Teilintervalle der Länge $2h$ mit $h = (\beta-\alpha)/2N$ und wendet auf jedes Teilintervall die entsprechende Ch_{n+1}-Formel an. Die Stützstellen x_k des Referenzintervalls sind dabei wie folgt zu transformieren:

$$x_k \mapsto \alpha + (2j+1)h + x_k, \quad j = 0(1)N-1, k = 0(1)n.$$

Man erhält so z. B. für $n = 1$ bei äquidistanter Zerlegung folgende *summierte Tschebyscheffsche Regel*

$$\begin{aligned}
\int_{\alpha}^{\beta} f(x)\,\mathrm{d}x &= Q_h^{Ch_2}(f;\alpha,\beta) + E_h^{Ch_2}(f;\alpha,\beta) \quad \text{mit} \\
Q_h^{Ch_2}(f;\alpha,\beta) &= h\sum_{j=0}^{N-1}\Big(f\big(\alpha+(2j+1)h - h/\sqrt{3}\big) + f\big(\alpha+(2j+1)h + h/\sqrt{3}\big)\Big), \\
E_h^{Ch_2}(f;\alpha,\beta) &= O(h^4).
\end{aligned}$$

Dabei ist $Q_h^{Ch_2}(f;\alpha,\beta)$ die *summierte Tschebyscheffsche Formel zu zwei Stützstellen* und $E_h^{Ch_2}(f;\alpha,\beta)$ das *Restglied der summierten Ch_2-Formel*, die globale Fehlerordnung beträgt somit $O(h^4)$.

Summierte Tschebyscheff-Formeln für nichtäquidistante Zerlegung

Mit der Zerlegung

$$\alpha = t_0 < t_1 < t_2 < \ldots < t_m = \beta,$$

$h_j = t_{j+1} - t_j$, $h_{\max} := \max\limits_{0\le j\le m-1}\{h_j\}$, sind die Stützstellen x_k des Referenzintervalles wie folgt zu transformieren:

$$x_k \mapsto t_j + h_j/2 + x_k \quad \text{für} \quad j = 0(1)m{-}1, \quad k = 0(1)n.$$

Für $n = 1$ z. B. erhält man die summierte Regel

$$\begin{aligned}
\int_\alpha^\beta f(x)\,\mathrm{d}x &= Q_{h_j}^{Ch_2}(f;\alpha,\beta) + E_{h_j}^{Ch_2}(f;\alpha,\beta) \quad \text{mit}\\
Q_{h_j}^{Ch_2}(f;\alpha,\beta) &= \sum_{j=0}^{m-1} h_j\left[f\left(t_j + \frac{h_j}{2} - \frac{h_j}{2\sqrt{3}}\right) + f\left(t_j + \frac{h_j}{2} + \frac{h_j}{2\sqrt{3}}\right)\right],\\
E_{h_j}^{Ch_2}(f;\alpha,\beta) &= O(h_{\max}^4).
\end{aligned}$$

Die Tschebyscheffschen Formeln haben für eine gerade Anzahl von Stützstellen eine günstigere Fehlerordnung als die Newton-Cotes-Formeln. Zur Fehlerschätzung vergleiche Abschnitt 14.11.

14.7 Quadraturformeln von Gauß

Um die Gaußschen Formeln optimaler Genauigkeit zu erhalten, werden weder die Stützstellen x_k noch die Gewichte A_k in (14.3) vorgeschrieben, so dass in (14.4) insgesamt $2(n+1) = 2n+2$ freie Parameter enthalten sind. Die Forderung, dass die Quadraturformel Polynome bis zum Grad $M = 2n+1$ exakt integriert, führt hier auf ein System von $2n+2$ Gleichungen für die $n+1$ Gewichte A_k und die $n+1$ Stützstellen x_k, $k = 0(1)n$; es lautet

$$\frac{1}{m+1}(b^{m+1} - a^{m+1}) = \sum_{k=0}^{n} A_k x_k^m, \quad m = 0(1)2n+1; \tag{14.23}$$

es ist linear bzgl. der Gewichte A_k und nichtlinear bzgl. der Stützstellen x_k. Man muss hier also voraussetzen, dass sich die Funktionswerte $f(x)$ an den sogenannten *Gaußschen Stützstellen* $x_k \in [a,b]$ berechnen oder aus einer Tabelle ablesen lassen. Ist von der Funktion f nur eine Wertetabelle bekannt, in der die Gaußschen Stützstellen im Allgemeinen nicht auftreten werden, so berechnet man das Integral bei äquidistanten Stützstellen mittels einer Newton-Cotes-Formel oder einer Maclaurin-Formel und bei beliebigen Stützstellen mittels einer Quadraturformel, die mit Hilfe des Systems (14.4) konstruiert wird.

Für das Integral von f über $[a,b]=[-1,+1]$ lässt sich zeigen, dass die $n+1$ Gaußschen Stützstellen x_k gerade die Nullstellen der *Legendreschen Polynome* $P_{n+1}(x)$ in $[-1,+1]$ sind (s. hierzu z. B. [POLO1964], S.209; [STRO1966], 1.2; [STUM1982], S.86/87 sowie [ENGE1996], 8.1.2).

Gaußsche Formeln für das Referenzintervall [−h,+h]

Betrachtet man nun das Integral von f über das Referenzintervall $[-h,+h]$ und setzt

$$\int_{-h}^{+h} f(x)\,\mathrm{d}x = Q^{G_{n+1}}(f;-h,h) + E^{G_{n+1}}(f;-h,h) = \sum_{k=0}^{n} A_k f(x_k) + O(h^q),$$

so bezeichnet man diese Beziehung als *Gaußsche Regel*, $Q^{G_{n+1}}(f;-h,+h)$ als *Gaußsche Formel* (G_{n+1}-Formel) und $E^{G_{n+1}}(f;-h,+h)$ als *Restglied der G_{n+1}-Formel* zu $n+1$ Gaußschen Stützstellen.

Wie oben erwähnt, sind die $n+1$ Gaußschen Stützstellen x_k zum Intervall $[-1,1]$ die Nullstellen der *Legendreschen Polynome* P_{n+1} in $[-1,1]$.

Die Legendreschen Polynome P_n sind Polynome vom Grad n und bilden ein in $[-1,1]$ orthogonales Funktionensystem mit $w(x)=1$; es gilt

$$\int_{-1}^{+1} P_n(x)\,P_m(x)\,\mathrm{d}x = \begin{cases} 0 & \text{für } n \neq m\,, \\ 2/(2n+1) & \text{für } n = m\,. \end{cases}$$

Die Legendreschen Polynome P_n für $n=0(1)4$ sind

$$P_0(x)=1\,; \quad P_1(x)=x\,; \quad P_2(x)=\tfrac{1}{2}\,(3x^2-1)\,;$$
$$P_3(x)=\tfrac{1}{2}\,(5x^3-3x)\,; \quad P_4(x)=\tfrac{1}{8}\,(35x^4-30x^2+3)\,.$$

Weitere Polynome lassen sich aus der Rekursionsformel

$$(n+1)\,P_{n+1}(x) = (2n+1)\,x\,P_n(x) - n\,P_{n-1}(x)\,, \quad n \geq 1$$

ermitteln. Die Nullstellen der P_n liegen symmetrisch zum Nullpunkt; alle Polynome ungerader Ordnung besitzen $x=0$ als Nullstelle.

Das Intervall $[-1,+1]$ muss zunächst auf $[-h,+h]$ transformiert werden. Dann ergeben sich für einige spezielle Gaußsche Quadraturformeln die folgenden Gewichte A_k und Stützstellen x_k, $k=0(1)n$.

Tabelle der Gaußschen Stützstellenwerte und Gewichte:

n	$x_k, \quad k = 0(1)n$	$A_k, \quad k = 0(1)n$
0	$x_0 = 0$	$A_0 = 2h$
1	$x_{0,1} = \pm\frac{h}{\sqrt{3}} = \pm 0.577350269\,h$	$A_0 = A_1 = h$
2	$x_{0,2} = \pm\sqrt{\frac{3}{5}}\,h = \pm 0.774596669\,h$ $x_1 = 0$	$A_0 = A_2 = \frac{5}{9}\,h = 0.\bar{5}\,h$ $A_1 = \frac{8}{9}\,h = 0.\bar{8}\,h$
3	$x_{0,3} = \pm 0.86113631\,h$ $x_{1,2} = \pm 0.33998104\,h$	$A_0 = A_3 = 0.34785485\,h$ $A_1 = A_2 = 0.65214515\,h$
4	$x_{0,4} = \pm 0.90617985\,h$ $x_{1,3} = \pm 0.53846931\,h$ $x_2 = 0$	$A_0 = A_4 = 0.23692689\,h$ $A_1 = A_3 = 0.47862867\,h$ $A_2 = \frac{128}{225}\,h = 0.56\bar{8}\,h$
5	$x_{0,5} = \pm 0.93246951\,h$ $x_{1,4} = \pm 0.66120939\,h$ $x_{2,3} = \pm 0.23861919\,h$	$A_0 = A_5 = 0.17132449\,h$ $A_1 = A_4 = 0.36076157\,h$ $A_2 = A_3 = 0.46791393\,h$

Weitere Werte sind in [ABRA1986], Tabelle 25.4 angegeben.

Das Restglied besitzt die allgemeine Form

$$E^{G_{n+1}}(f;-h,h) = \frac{2^{2n+3}\left((n+1)!\right)^4}{(2n+3)\left((2n+2)!\right)^3}\,h^{2n+3}\,f^{(2n+2)}(\xi^*),$$
$$\xi^* \in [-h,h], \quad f^{(2n+2)} \in C[-h,h],$$

d. h. die *lokale Fehlerordnung* bei $n+1$ Stützstellen im Referenzintervall $[-h,+h]$ ist $O(h^{2n+3})$.

Im Folgenden werden zwei der Gaußschen Regeln explizit aufgeschrieben, und zwar die für 2 und 3 Stützstellen $x_k \in [-h,+h]$:

1. $n = 1$ (2 Stützstellen):

$$\int_{-h}^{+h} f(x)\,\mathrm{d}x = Q^{G_2}(f;-h,h) + E^{G_2}(f;-h,h) \quad \text{mit}$$
$$Q^{G_2}(f;-h,h) = h\left(f(-h/\sqrt{3}) + f(h/\sqrt{3})\right),$$
$$E^{G_2}(f;-h,h) = \frac{h^5}{135} f^{(4)}(\xi^*), \quad \xi^* \in [-h,h], \quad f^{(4)} \in C[-h,h].$$

2. $n = 2$ (3 Stützstellen):

$$\begin{aligned}
\int_{-h}^{+h} f(x)\,\mathrm{d}x &= Q^{G_3}(f;-h,h) + E^{G_3}(f;-h,h) \quad \text{mit} \\
Q^{G_3}(f;-h,h) &= \frac{h}{9}\left(5f(-\sqrt{0.6}h) + 8f(0) + 5f(\sqrt{0.6}h)\right), \\
E^{G_3}(f;-h,h) &= \frac{h^7}{15750} f^{(6)}(\xi^*), \quad \xi^* \in [-h,h], \quad f^{(6)} \in C[-h,h].
\end{aligned}$$

Mit zwei Stützstellen erhält man eine Formel der lokalen Fehlerordnung $O(h^5)$, mit drei Stützstellen eine Formel der lokalen Fehlerordnung $O(h^7)$. Die Newton-Cotes-Formeln der lokalen Fehlerordnungen $O(h^5)$ und $O(h^7)$ erfordern dagegen drei bzw. fünf Stützstellen.

Für $n = 4$ und $n = 5$ lassen sich die Formeln an Hand der Tabelle der x_k, A_k leicht bilden. Dabei ist

$$E^{G_4} = \frac{h^9}{3472875} f^{(8)}(\xi^*), \quad E^{G_5} = \frac{h^{11}}{1237732650} f^{(10)}(\xi^*), \quad \xi^* \in [-h,+h].$$

Summierte Gaußsche Quadraturformeln bei äquidistanter Zerlegung

Zur Bestimmung des Integrals von f über ein Intervall $[\alpha,\beta]$ teilt man $[\alpha,\beta]$ in N Teilintervalle der Länge $2h$: $h = (\beta - \alpha)/2N$. Die Stützstellen sind dabei wie folgt zu transformieren:

$$x_k \mapsto \alpha + (2j+1)h + x_k, \quad j = 0(1)N{-}1, \quad k = 0(1)n.$$

Man erhält für $n = 1$ und $n = 2$ die folgenden summierten Gaußschen Regeln:

$$\begin{aligned}
\int_{\alpha}^{\beta} f(x)\,\mathrm{d}x &= Q_h^{G_2}(f;\alpha,\beta) + E_h^{G_2}(f;\alpha,\beta) \quad \text{mit} \\
Q_h^{G_2}(f;\alpha,\beta) &= h \sum_{j=0}^{N-1} \Big(f\big(\alpha + (2j+1)h - h/\sqrt{3}\big) + f\big(\alpha + (2j+1)h + h/\sqrt{3}\big) \Big), \\
E_h^{G_2}(f;\alpha,\beta) &= \frac{\beta-\alpha}{270} h^4 f^{(4)}(\eta), \quad \eta \in [\alpha,\beta], \quad f^{(4)} \in C[\alpha,\beta].
\end{aligned}$$

$$\begin{aligned}
\int_{\alpha}^{\beta} f(x)\,\mathrm{d}x &= Q_h^{G_3}(f;\alpha,\beta) + E_h^{G_3}(f;\alpha,\beta) \quad \text{mit} \\
Q_h^{G_3}(f;\alpha,\beta) &= \frac{h}{9} \sum_{j=0}^{N-1} \Big(5f\big(\alpha + (2j+1)h - \sqrt{3/5}h\big) + 8f\big(\alpha + (2j+1)h\big) \\
&\qquad + 5f\big(\alpha + (2j+1)h + \sqrt{3/5}h\big) \Big), \\
E_h^{G_3}(f;\alpha,\beta) &= \frac{\beta-\alpha}{31500} h^6 f^{(6)}(\eta), \quad \eta \in [\alpha,\beta], \quad f^{(6)} \in C[\alpha,\beta].
\end{aligned}$$

Summierte Gaußsche Quadraturformeln bei nichtäquidistanter Zerlegung

Mit der Zerlegung

$$\alpha = t_0 < t_1 < t_2 < \ldots < t_m = \beta,$$

$h_j = t_{j+1} - t_j$, $h_{\max} = \max\limits_{0 \le j \le m-1} \{h_j\}$ sind die Stützstellen x_k des Referenzintervalles wie folgt zu transformieren:

$$x_k \mapsto t_j + h_j/2 + x_k \quad \text{für} \quad j = 0(1)m{-}1, \quad k = 0(1)n.$$

Man erhält z. B. für $n = 1$ die summierte Regel:

$$\begin{aligned} n = 1: \quad \int_\alpha^\beta f(x)\,\mathrm{d}x &= Q_{h_j}^{G_2}(f;\alpha,\beta) + E_{h_j}^{G_2}(f;\alpha,\beta) \quad \text{mit} \\ Q_{h_j}^{G_2}(f;\alpha,\beta) &= \sum_{j=0}^{m-1} h_j \left[f\left(t_j + \frac{h_j}{2} - \frac{h_j}{2\sqrt{3}}\right) + f\left(t_j + \frac{h_j}{2} + \frac{h_j}{2\sqrt{3}}\right)\right] \\ E_{h_j}^{G_2}(f;\alpha,\beta) &= O(h_{\max}^4), \end{aligned}$$

und für $n = 2$ die Regel

$$\begin{aligned} n = 2: \quad \int_\alpha^\beta f(x)\,\mathrm{d}x &= Q_{h_j}^{G_3}(f;\alpha,\beta) + E_{h_j}^{G_3}(f;\alpha,\beta) \quad \text{mit} \\ Q_{h_j}^{G_3}(f;\alpha,\beta) &= \sum_{j=0}^{m-1} h_h \Bigg(\frac{5h_j}{18} \Big(f\big(t_j + h_j/2 - \sqrt{0.6}/2h_j\big) \\ &\quad + f\left(t_j + h_j/2 + \sqrt{0.6}/2h_j\right)\Big) + 4/9 h_j f\left(t_j + h_j/2\right)\Bigg) \\ E_{h_j}^{G_3}(f;\alpha,\beta) &= O(h_{\max}^6). \end{aligned}$$

Die Gaußschen Formeln $Q^{G_{n+1}}$ sind optimal bezüglich der Fehlerordnung. Gegenüber den Newton-Cotes-Formeln gleicher Fehlerordnung benötigen sie nur etwa die Hälfte des Rechenaufwandes. Schwierig ist jedoch die Berechnung der Gewichte A_k und der Stützstellen x_k aus dem nichtlinearen Gleichungssystem (14.23). Die Fehlerschätzung verläuft gemäß der Beschreibung in Abschnitt 14.11.

14.8 Berechnung von Gewichten und Stützstellen verallgemeinerter Gauß-Quadraturformeln

Im Folgenden ist eine Berechnungsmethode für die Gewichte und Stützstellen der verallgemeinerten Gaußschen Quadraturformeln für das Referenzintervall $[a, b]$ angegeben, die die Lösung eines nichtlinearen Gleichungssystems umgeht.

Die Quadraturformel

$$Q(f, g; a, b) = \sum_{i=1}^{n} A_i f(x_i)$$

soll eine Näherung für das Integral $\int_a^b f(x)\,g(x)\,\mathrm{d}x$ liefern, die alle Polynome bis zum Grad $2n-1$, $n \geq 1$, exakt integriert. Darin ist g eine Gewichtsfunktion mit $g(x) > 0$ in (a,b); $\int g(x)\,\mathrm{d}x$ sei berechenbar für alle Teilintervalle aus $[a,b]$.

Das Polynom

$$Q_n(x) = x^n + \sum_{k=0}^{n-1} q_k x^k$$

wird so bestimmt, dass Q_n orthogonal zu allen Monomen x^j mit $0 \leq j \leq n-1$ ist:

$$\int_a^b Q_n(x)\,x^j\,g(x)\,\mathrm{d}x = \int_a^b x^{n+j}\,g(x)\,\mathrm{d}x + \sum_{k=0}^{n-1} q_k \int_a^b x^{k+j}\,g(x)\,\mathrm{d}x = 0.$$

Dies ist ein lineares Gleichungssystem zur Bestimmung der q_k, wenn die Integrale $A_j = \int_a^b x^j\,g(x)\,\mathrm{d}x$ für $0 \leq j \leq 2n-1$ vorgegeben sind. Nach der Lösung dieses Systems werden die Nullstellen x_i, $i = 1(1)n$, von $Q_n(x)$ bestimmt.

Ein beliebiges Polynom $P(x)$ bis zum Grad $2n-1$ kann mit Polynomdivision in der Form

$$P(x) = S(x)Q_n(x) + R(x)$$

dargestellt werden, dabei sind $S(x)$ und $R(x)$ Polynome vom Grad $\leq n-1$. Setzt man hier die Nullstellen x_i von Q_n ein, so gilt: $P(x_i) = R(x_i)$, da $Q_n(x_i) = 0$. Da $R(x)$ maximal vom Grad $n-1$ ist, ist die Lagrangesche Interpolation exakt:

$$R(x) = \sum_{i=1}^{n} R(x_i)L_i(x) \quad \text{mit} \quad L_i(x) = \prod_{\substack{j=1 \\ j\neq i}}^{n} \frac{x - x_j}{x_i - x_j}.$$

Aus dem Grad $\leq n-1$ von $S(x)$ folgt wegen der Orthogonalität

$$\int_a^b S(x)Q_n(x)\,g(x)\,\mathrm{d}x = \sum_{j=0}^{n-1} s_j \int_a^b x^j Q_n(x)\,g(x)\,\mathrm{d}x = 0,$$

und es gilt

$$\begin{aligned}\int_a^b P(x)\,g(x)\,\mathrm{d}x \quad &= \int_a^b R(x)\,g(x)\,\mathrm{d}x = \sum_{i=1}^{n} R(x_i) \int_a^b L_i(x)\,g(x)\,\mathrm{d}x \\ &=: \sum_{i=1}^{n} P(x_i)A_i \quad \text{mit} \quad A_i = \int_a^b L_i(x)\,g(x)\,\mathrm{d}x.\end{aligned}$$

Die Lagrangeschen Interpolationspolynome können wegen

$$Q_n(x) = \prod_{i=1}^{n}(x - x_i)$$

folgendermaßen bestimmt werden:

$$\begin{aligned}L_i^*(x) \quad &:= Q_n(x)/(x - x_i) \\ L_i(x) \quad &= L_i^*(x)/L_i^*(x_i) = \sum_{j=0}^{n-1} L_{i,j}x^j,\end{aligned}$$

wobei die $L_{i,j}$ die Koeffizienten des Polynoms $L_i(x)$ sind.

Für die Gewichte A_i gilt darum:

$$A_i = \int_a^b L_i(x)\, g(x)\, \mathrm{d}x = \sum_{j=0}^{n-1} L_{i,j} \int_a^b x^j\, g(x)\, \mathrm{d}x.$$

Algorithmus 14.20. (*Verallgemeinerte Gauß-Formeln*)

Gegeben: n, $n \geq 1$, die exakten Integralwerte $\int_a^b x^i\, g(x)\, \mathrm{d}x$ für $i = 0(1)2n-1$, mit einer Gewichtsfunktion $g(x) > 0$ in (a, b).

Gesucht: Gaußsche Quadraturformel für das Referenzintervall $[a, b]$

$$Q(f, g; a, b) = \sum_{i=1}^{n} A_i f(x_i) \quad \text{mit} \quad \int_a^b f(x)\, g(x)\, \mathrm{d}x \approx Q(f, g; a, b).$$

1. Lösung des linearen Gleichungssystems $\boldsymbol{A}\boldsymbol{q} - \boldsymbol{a} = \boldsymbol{0}$ mit

$$\boldsymbol{A} = \begin{pmatrix} \int g(x)\,\mathrm{d}x & \int x g(x)\,\mathrm{d}x & \dots & \int x^{n-1} g(x)\,\mathrm{d}x \\ \int x g(x)\,\mathrm{d}x & \int x^2 g(x)\,\mathrm{d}x & & \int x^n g(x)\,\mathrm{d}x \\ \vdots & \vdots & & \vdots \\ \int x^{n-1} g(x)\,\mathrm{d}x & \int x^n g(x)\,\mathrm{d}x & & \int x^{2n-2} g(x)\,\mathrm{d}x \end{pmatrix},$$

$$\boldsymbol{q} = \begin{pmatrix} q_0 \\ q_1 \\ \vdots \\ q_{n-1} \end{pmatrix}, \quad \boldsymbol{a} = \begin{pmatrix} \int x^n g(x)\,\mathrm{d}x \\ \int x^{n+1} g(x)\,\mathrm{d}x \\ \vdots \\ \int x^{2n-1} g(x)\,\mathrm{d}x \end{pmatrix}.$$

2. Berechnung der Nullstellen x_i, $i = 1(1)n$, von

$$Q_n(x) = x^n + \sum_{k=0}^{n-1} q_k x^k.$$

3. (a) Berechnung der Koeffizienten $L^*_{i,j}$ von $L^*_i(x)$ mit Hilfe des Hornerschemas (Division von Q_n durch $x - x_i$),

   (b) Berechnung von $L^*_i(x_i)$ unter Verwendung der Koeffizienten $L^*_{i,j}$ mit dem Hornerschema,

   (c) Berechnung der $L_{i,j} = L^*_{i,j} / L^*_i(x_i)$.

4. Berechnung der Gewichte A_i aus den Formeln

$$A_i = \sum_{j=0}^{n-1} L_{i,j} \int_a^b x^j g(x)\, \mathrm{d}x.$$

Bemerkung. Da die Randpunkte des Referenzintervalles $[a, b]$ nicht in die Berechnung der Gewichte A_i eingehen, sind auch Gaußsche Quadraturformeln für uneigentliche Integrale möglich.

14.9 Quadraturformeln von Clenshaw-Curtis

Will man Quadraturformeln verwenden, deren Knoten sich im Falle höherer Fehlerordnung leichter berechnen lassen als die der Gaußschen Quadraturformeln, so ist der Einsatz von *Clenshaw-Curtis-Formeln* zu empfehlen.

Clenshaw-Curtis-Formeln für das Referenzintervall [−1,+1]

Es handelt sich dabei um Interpolationsquadraturformeln der Form

$$Q^{CC_n}(f; -1, 1) := \sum_{k=0}^{n} A_k^{(n)} f(x_k), \quad n \geq 2, \quad n \text{ gerade},$$

für das Referenzintervall $[-1, +1]$ mit den Tschebyscheff-Knoten

$$x_k = \cos(k\pi/n), \qquad k = 0(1)n,$$

als Stützstellen, die alle Polynome f vom Grad $n + 1$

$$I(f; -1, 1) = \int_{-1}^{1} f(x)\,\mathrm{d}x$$

exakt integrieren. Die Gewichte $A_k^{(n)}$ sind sämtlich positiv; es gilt

$$\begin{aligned} A_0^{(n)} &= A_n^{(n)} = \frac{1}{n^2 - 1}, \\ A_k^{(n)} &= 2\frac{n^2 - 1 - (-1)^k}{n(n^2 - 1)} - \frac{4}{n} \sum_{j=1}^{n/2-1} \frac{\cos(2jk\pi/n)}{4j^2 - 1}, \quad k = 1(1)n{-}1. \end{aligned}$$

Will man die Formeln für ein Referenzintervall $[a, b]$ statt $[-1, 1]$ verwenden, so erhält man die transformierte Clenshaw-Curtis-Formel

$$Q^{CC_n}(f; a, b) = \frac{b-a}{2} \sum_{k=0}^{n} A_k^{(n)} f\left(\frac{b-a}{2} \cos\left(\frac{k\pi}{n}\right) + \frac{b+a}{2}\right).$$

Die lokale Fehlerordnung beträgt $O(h^{n+3})$ bei geradem n und $f \in C^{n+2}([a, b])$.

Zusammengesetzte Clenshaw-Curtis-Formeln

Mit der Zerlegung Z des Integrationsintervalles $[\alpha,\beta]$:

$$Z : \alpha = t_0 < t_1 < t_2 < \ldots < t_m = \beta, \quad h_j := t_{j+1} - t_j, h_{\max} = \max_{0 \leq j \leq m-1} \{h_j\}$$

sind die Stützstellen x_k des Referenzintervalles $[-1,1]$ wie folgt zu transformieren

$$x_k \mapsto \frac{t_{j+1}+t_j}{2} + \frac{h_j}{2} x_k,$$

und man erhält die zusammengesetzten Clenshaw-Curtis-Formeln

$$Q_{h_j}^{CC_n}(f;\alpha,\beta) = \frac{1}{2}\sum_{j=0}^{m-1}(t_{j+1}-t_j)\sum_{k=0}^{n} A_k^{(n)} f\left(\frac{t_{j+1}-t_j}{2}\cos(k\pi/n) + \frac{t_{j+1}+t_j}{2}\right).$$

Für den globalen Fehler gilt bei f $\in$ C^{n+2} ([α,β])

$$E_{h_j}^{CC_n}(f;\alpha,\beta) = O(h_{\max}^{n+2}).$$

Zur Fehlerschätzung siehe Abschnitt 14.11.

14.10 Das Verfahren von Romberg

Das Verfahren von Romberg beruht auf der Approximation des Integrals $I(f;\alpha,\beta)$ durch die Sehnentrapezformel. Durch fortgesetzte Halbierung der Schrittweite und geeignete Linearkombination zugehöriger Approximationen für das Integral gelingt es, Quadraturformeln von höherer Fehlerordnung zu erzeugen.

Setzt man f $2n$-mal stetig differenzierbar voraus, so kann zur Entwicklung des Verfahrens die Euler-Maclaurin-Formel verwendet werden. Es lässt sich zeigen, dass das Verfahren auch dann konvergiert, wenn nur die Stetigkeit von f gefordert wird.
Man zerlegt [α,β] zunächst in N Teilintervalle der Länge $h = (\beta-\alpha)/N$. Für $I(f;\alpha,\beta)$ gilt dann mit der summierten Sehnentrapezformel (14.9) die Darstellung

$$\left\{\begin{array}{rcl} \int\limits_{\alpha}^{\beta} f(x)\,\mathrm{d}x & = & Q_h^{ST}(f;\alpha,\beta) + O(h^2) \qquad \text{mit} \\ Q_h^{ST}(f;\alpha,\beta) & = & \frac{h}{2}\left(f(\alpha) - f(\beta) + 2\sum\limits_{\nu=1}^{N-1} f(\alpha+\nu h)\right), \end{array}\right.$$

und mit der summierten Euler-Maclaurin-Formel (14.22) die Darstellung

$$\int_{\alpha}^{\beta} f(x)\,\mathrm{d}x = Q_h^{ST}(f;\alpha,\beta) + \sum_{k=1}^{n-1} c_{2k}\, h^{2k} + O(h^{2n}), \tag{14.24}$$

wobei die c_{2k} unabhängig von h sind.

Im Folgenden wird das Verhalten von (14.24) bei fortgesetzter Halbierung der Schrittweite untersucht. Es wird gesetzt

$$N_j = 2^j N, \quad h_j = \frac{\beta-\alpha}{2^j N} = \frac{h}{2^j}, \quad h_0 = h, \quad N_0 = N. \tag{14.25}$$

Die summierte Sehnentrapezformel mit der Schrittweite h_j bezeichnet man mit

$$L_j^{(0)}(f) := Q_{h_j}^{ST}(f;\alpha,\beta) = \frac{h_j}{2}\Big(f(\alpha)+f(\beta)+2\sum_{\nu=1}^{N_j-1} f(\alpha+\nu h_j)\Big)\,. \tag{14.26}$$

Für die Schrittweiten h_j und $h_{j+1}=\frac{h_j}{2}$ lautet (14.24) mit (14.26)

$$\begin{aligned}\int_\alpha^\beta f(x)\,\mathrm{d}x &= L_j^{(0)}(f)+c_2\,h_j^2+c_4\,h_j^4+\ldots+c_{2n-2}\,h_j^{2n-2}+O(h_j^{2n})\,, &(14.27)\\ \int_\alpha^\beta f(x)\,\mathrm{d}x &= L_{j+1}^{(0)}(f)+c_2\left(\frac{h_j}{2}\right)^2+c_4\left(\frac{h_j}{2}\right)^4+\ldots\\ &\quad +c_{2n-2}\left(\frac{h_j}{2}\right)^{2n-2}+O\left(\left(\frac{h_j}{2}\right)^{2n}\right)\,. &(14.28)\end{aligned}$$

Man bildet nun eine Linearkombination dieser beiden Gleichungen, indem man Gleichung (14.28) mit 4 multipliziert und davon Gleichung (14.27) subtrahiert. Man erhält

$$\int_\alpha^\beta f(x)\,\mathrm{d}x = \frac{1}{3}\left(4L_{j+1}^{(0)}(f)-L_j^{(0)}(f)\right)+c_4^{(1)}\,h_j^4+\ldots+c_{2n-2}^{(1)}\,h_j^{2n-2}+O(h_j^{2n})\,. \tag{14.29}$$

Mit

$$L_j^{(1)}(f) := \frac{1}{3}\left(4L_{j+1}^{(0)}(f)-L_j^{(0)}(f)\right) = L_{j+1}^{(0)}(f)+\frac{1}{3}\left(L_{j+1}^{(0)}(f)-L_j^{(0)}(f)\right)$$

gilt somit

$$\int_\alpha^\beta f(x)\,\mathrm{d}x = L_j^{(1)}(f)+O(h_j^4)\,;$$

also ist $L_j^{(1)}(f)$ eine Approximation der Fehlerordnung $O(h_j^4)$ für das Integral.

Somit wurde durch eine erste Linearkombination $L_j^{(1)}$ zweier Approximationen $L_j^{(0)}$ und $L_{j+1}^{(0)}$ für das Integral eine Quadraturformel $L_j^{(1)}$ der Fehlerordnung $O(h_j^4)$ erzeugt.

Um eine Quadraturformel der Fehlerordnung $O(h_j^6)$ zu gewinnen, wird eine Linearkombination der (14.29) entsprechenden Gleichungen für die Schrittweiten h_j und $h_{j+1}=h_j/2$ gebildet:

$$\begin{aligned}\int_\alpha^\beta f(x)\,\mathrm{d}x &= L_j^{(1)}(f)+c_4^{(1)}\,h_j^4+c_6^{(1)}\,h_j^6+\ldots\\ &\quad +c_{2n-2}^{(1)}\,h_j^{2n-2}+O(h_j^{2n})\,, &(14.30)\\ \int_\alpha^\beta f(x)\,\mathrm{d}x &= L_{j+1}^{(1)}(f)+c_4^{(1)}\left(\frac{h_j}{2}\right)^4+c_6^{(1)}\left(\frac{h_j}{2}\right)^6+\ldots\\ &\quad +c_{2n-2}^{(1)}\left(\frac{h_j}{2}\right)^{2n-2}+O\left(\left(\frac{h_j}{2}\right)^{2n}\right)\,. &(14.31)\end{aligned}$$

Man multipliziert Gleichung (14.31) mit 2^4, subtrahiert davon Gleichung (14.30) und erhält

$$\int_\alpha^\beta f(x)\,\mathrm{d}x = \frac{1}{2^4-1}\left(2^4 L_{j+1}^{(1)}(f) - L_j^{(1)}(f)\right) + c_6^{(2)}\,h_j^6 + \ldots + c_{2n-2}^{(2)}\,h_j^{2n-2} + O(h_j^{2n})\,.$$

Mit

$$L_j^{(2)}(f) := \frac{1}{2^4-1}\left(2^4 L_{j+1}^{(1)}(f) - L_j^{(1)}(f)\right) = L_{j+1}^{(1)}(f) + \frac{1}{15}\left(L_{j+1}^{(1)}(f) - L_j^{(1)}(f)\right)$$

folgt

$$\int_\alpha^\beta f(x)\,\mathrm{d}x = L_j^{(2)}(f) + O(h^6)\,.$$

So fortfahrend erhält man für das Integral die Darstellung

$$\int_\alpha^\beta f(x)\,\mathrm{d}x = L_j^{(k)}(f) + O(h_j^{2(k+1)})$$

mit

$$\begin{aligned} L_j^{(k)}(f) &= \frac{1}{2^{2k}-1}\left(2^{2k} L_{j+1}^{(k-1)}(f) - L_j^{(k-1)}(f)\right) \\ &= L_{j+1}^{(k-1)}(f) + \frac{1}{2^{2k}-1}\left(L_{j+1}^{(k-1)}(f) - L_j^{(k-1)}(f)\right) \end{aligned}$$

für $j = 0, 1, 2, \ldots$, $k = 1, 2, \ldots, n-1$, $h_j = h/2^j$, $h_0 = h$; dabei ergibt sich n aus der Voraussetzung $f \in C^{2n}[\alpha, \beta]$.

Die Rechnung wird zweckmäßig *zeilenweise* nach dem folgenden Schema durchgeführt:

Rechenschema 14.21. (*Verfahren von Romberg*)

h_j	$L_j^{(0)} = Q_{h_j}^{ST}(f;\alpha,\beta)$	$L_j^{(1)}$	$L_j^{(2)}$	$\cdots$	$L_j^{(m-1)}$	$L_j^{(m)}$
h_0	$L_0^{(0)}$					
$h_1 = \frac{h_0}{2}$	$L_1^{(0)}$	$L_0^{(1)}$				
$h_2 = \frac{h_1}{2}$	$L_2^{(0)}$	$L_1^{(1)}$	$L_0^{(2)}$			
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$		
$h_{m-1} = \frac{h_{m-2}}{2}$	$L_{m-1}^{(0)}$	$L_{m-2}^{(1)}$	$L_{m-3}^{(2)}$	$\cdots$	$L_0^{(m-1)}$	
$h_m = \frac{h_{m-1}}{2}$	$L_m^{(0)}$	$L_{m-1}^{(1)}$	$L_{m-2}^{(2)}$	$\cdots$	$L_1^{(m-1)}$	$L_0^{(m)}$

Dabei können die $L_j^{(0)}$ nach der Formel (14.26)

$$L_j^{(0)}(f) := Q_{h_j}^{ST}(f;\alpha,\beta) = \frac{h_j}{2}\Big(f(\alpha) + f(\beta) + 2\sum_{\nu=1}^{N_j-1} f(\alpha+\nu h_j)\Big)$$

berechnet werden. Besser und schneller (mit nur etwa dem halben Rechenaufwand) ist es, diese Formel nur für $j = 0$ zu verwenden und für $j = 1, 2, 3, \ldots$ die sich daraus ergebende Formel

$$\begin{aligned} L_j^{(0)}(f) &= \tfrac{1}{2} L_{j-1}^{(0)} + h_j \big(f(\alpha + h_j) + f(\alpha + 3h_j) + \ldots + f(\beta - h_j) \big) \\ &= \tfrac{1}{2} L_{j-1}^{(0)} + h_j \sum_{k=0}^{N_{j-1}-1} f\big(\alpha + (2k+1)h_j\big). \end{aligned}$$

Die $L_j^{(k)}$ für $k \geq 1$ und $j = 0, 1, 2, \ldots$ werden mit $2^{2k} = 4^k$ nach der Formel

$$L_j^{(k)}(f) = L_{j+1}^{(k-1)}(f) + \frac{1}{4^k - 1} \left(L_{j+1}^{(k-1)}(f) - L_j^{(k-1)}(f) \right)$$

berechnet. Das Schema wird so lange fortgesetzt, bis zu vorgegebenem $\varepsilon > 0$ gilt: $|L_0^{(m)} - L_1^{(m-1)}| < \varepsilon$. Dann wird $L_0^{(m)}(f)$ als bester erreichter Näherungswert für $I(f; \alpha, \beta)$ verwendet; es gilt mit $m \leq n-1$ die Romberg-Regel

$$\begin{aligned} I(f; \alpha, \beta) &= \int_\alpha^\beta f(x)\,\mathrm{d}x = L_0^{(m)}(f) + E^{R_m}(f; \alpha, \beta) \quad \text{mit} \\ E^{R_m}(f; \alpha, \beta) &= (-1)^{m+1} \frac{\beta - \alpha}{2^{m(m+1)}} \frac{B_{2m+2}}{(2m+2)!} h_0^{2m+2} f^{(2m+2)}(\xi), \quad \xi \in [\alpha, \beta], \end{aligned}$$

bzw. umgerechnet mit $h_0 = 2^m h_m$ gilt

$$\begin{aligned} E^{R_m}(f; \alpha, \beta) &= (-1)^{m+1} (\beta - \alpha) \frac{B_{2m+2}}{(2m+2)!} 2^{m(m+1)} h_m^{2m+2} f^{(2m+2)}(\xi) \\ &= O(h_m^{2m+2}), \quad \xi \in [\alpha, \beta], \end{aligned}$$

d. h. das Restglied E^{R_m} ist von der Ordnung $O(h_m^{2m+2})$ für $h_m \to 0$.

Unter der Voraussetzung $f \in C^{2n}[\alpha, \beta]$ konvergieren die Spalten $L_j^{(k)}$ des Schemas für jedes feste k und $j \to \infty$ linear gegen $I(f; \alpha, \beta)$. Ist f analytisch, so konvergieren die absteigenden Diagonalen des Schemas $L_j^{(k)}$ für jedes j und $k \to \infty$ superlinear gegen $I(f; \alpha, \beta)$. Es lässt sich zeigen, dass sowohl die Spalten als auch die absteigenden Diagonalen $L_j^{(k)}$ gegen $I(f; \alpha, \beta)$ konvergieren, wenn nur die Stetigkeit von f vorausgesetzt wird.

Beispiel 14.22.

Gegeben: Das Integral

$$I\left(\frac{\sin x}{x}; 0, \frac{\pi}{2}\right) = \int_0^{\frac{\pi}{2}} \frac{\sin x}{x}\,\mathrm{d}x$$

Gesucht: Der Näherungswert des Integrals mit dem Romberg-Verfahren auf 5 Dezimalen genau. Die Anfangsschrittweite sei $h_0 = \frac{\pi}{4}$. Abgebrochen werden soll, wenn $|L_0^{(m)} - L_1^{(m-1)}| < 0.5 \cdot 10^{-5}$ ist.

Lösung: Mit $f(x) = \dfrac{\sin x}{x}$ ergibt sich

$$h_0 = \frac{\pi}{4}: \qquad 0 \quad \frac{\pi}{4} \quad \frac{\pi}{2}$$

$$\begin{aligned} L_0^{(0)} &= Q^{ST}_{\pi/4}(f;0,\tfrac{\pi}{2}) = \frac{\pi}{8}\left(f(0) + 2f\left(\frac{\pi}{4}\right) + f\left(\frac{\pi}{2}\right)\right) \\ &= \tfrac{\pi}{8}\left(1 + \tfrac{4\sqrt{2}}{\pi} + \tfrac{2}{\pi}\right) = 1.349\,805\,863 \end{aligned}$$

$$h_1 = \frac{\pi}{8}: \qquad 0 \quad \frac{\pi}{8} \quad \frac{\pi}{4} \quad \frac{3\pi}{8} \quad \frac{\pi}{2}$$

$$L_1^{(0)} = \tfrac{1}{2}L_0^{(0)} + \frac{\pi}{8}\left(f\left(\frac{\pi}{8}\right) + f\left(\frac{3\pi}{8}\right)\right) = 1.365\,546\,208$$

1. Linearkombination: $L_0^{(1)} = L_1^{(0)} + \tfrac{1}{3}(L_1^{(0)} - L_0^{(0)}) = 1.370\,792\,990$

Abfrage : $|L_0^{(1)} - L_1^{(0)}| = 0.005\,24 < 0.5 \cdot 10^{-5}$?

$\Rightarrow$ Abbruchbedingung nicht erfüllt!

$$h_2 = \frac{\pi}{16}: \qquad 0 \quad \frac{\pi}{16} \quad \frac{\pi}{8} \quad \frac{3\pi}{16} \quad \frac{\pi}{4} \quad \frac{5\pi}{16} \quad \frac{3\pi}{8} \quad \frac{7\pi}{16} \quad \frac{\pi}{2}$$

$$\begin{aligned} L_2^{(0)} &= \tfrac{1}{2}L_1^{(0)} + \frac{\pi}{16}\left(f\left(\frac{\pi}{16}\right) + f\left(\frac{3\pi}{16}\right) + f\left(\frac{5\pi}{16}\right) + f\left(\frac{7\pi}{16}\right)\right) \\ &= 1.369\,459\,609 \end{aligned}$$

2. Linearkombination: $L_1^{(1)} = L_2^{(0)} + \tfrac{1}{3}(L_2^{(0)} - L_1^{(0)}) = 1.370\,764\,076$

3. Linearkombination: $L_0^{(2)} = L_1^{(1)} + \tfrac{1}{15}(L_1^{(1)} - L_0^{(1)}) = 1.370\,762\,149$

Abfrage : $|L_0^{(2)} - L_1^{(1)}| = 1.93 \cdot 10^{-6} < 0.5 \cdot 10^{-5}$?

$\Rightarrow$ Abbruchbedingung erfüllt!

$\Rightarrow$ Der Näherungswert des Integrals ist 1.370 76. $\square$

Beispiel 14.23.

Gegeben: Das Integral

$$I\left(\frac{2}{\sqrt{\pi}}e^{-x^2};0,0.5\right)=\frac{2}{\sqrt{\pi}}\int_0^{0.5}\mathrm{e}^{-x^2}\,\mathrm{d}x\,.$$

Gesucht: Ein Näherungswert für das Integral mit Hilfe des Romberg-Verfahrens.

Lösung: Die Sehnentrapezformel liefert zu $h=0.5$ den Wert $L_0^{(0)}=0.5017904365$ bei Rundung auf 10-stellige Mantisse.

Nach dem Rechenschema 14.21 erhält man für die $L_j^{(k)}$ bis zu $k=4$ die folgenden Werte bei 10-stelliger Mantisse:

j	$L_j^{(0)}$	$L_j^{(1)}$	$L_j^{(2)}$	$L_j^{(3)}$	$L_j^{(4)}$
0	0.5017904365				
1	0.5158987506	0.5206015220			
2	0.5193541352	0.5205059301	0.5204995573		
3	0.5202137226	0.5205002517	0.5204998732	0.5204998782	
4	0.5204283565	0.5204999011	0.5204998777	0.5204998778	0.5204998778

Bei $k=4$ kommt die Rechnung mit dem Wert 0.5204998778 zum Stehen. □

14.11 Fehlerschätzung und Rechnungsfehler

In der Regel ist eine Abschätzung des Quadraturfehlers nicht möglich, da die benötigten Ableitungen des Integranden unbekannt sind oder nur mit erheblichem Aufwand abgeschätzt werden können. Die genaue Kenntnis des Restgliedkoeffizienten ist somit nur von theoretischem Nutzen. Wesentlich ist aber die Kenntnis der globalen Fehlerordnung $O(h^q)$ einer Quadraturformel; sie reicht aus, um unter Verwendung von zwei zu verschiedenen Schrittweiten berechneten Näherungswerten für $I(f;\alpha,\beta)$ einen *Schätzwert für den wahren Fehler* angeben zu können.

Fehlerschätzung bei äquidistanter Zerlegung

Wurde etwa das Integral $I(f;\alpha,\beta)$ näherungsweise mit der Schrittweite h_i nach einer Quadraturformel der globalen Fehlerordnung $O(h_i^q)$ berechnet, so gilt

$$\begin{aligned} I(f;\alpha,\beta) &= Q_{h_i}(f;\alpha,\beta)+E_{h_i}(f;\alpha,\beta) \quad \text{mit}\\ E_{h_i}(f;\alpha,\beta) &= O(h_i^q)=c_i\,h_i^q\,, \quad c_i = \text{ Restgliedkoeffizient.}\end{aligned}$$

Für $i = 1$ und $i = 2$, q fest, erhält man die folgende Fehlerschätzungsformel für den globalen Fehler $E_{h_1}(f;\alpha,\beta)$ des mit der Schrittweite h_1 berechneten Näherungswertes $Q_{h_1}(f;\alpha,\beta)$ für $I(f;\alpha,\beta)$:

$$E_{h_1}(f;\alpha,\beta) \approx \frac{Q_{h_1}(f;\alpha,\beta) - Q_{h_2}(f;\alpha,\beta)}{\left(\frac{h_2}{h_1}\right)^q - 1} = E^*_{h_1}(f;\alpha,\beta). \tag{14.32}$$

Mit (14.32) lässt sich ein gegenüber $Q_{h_1}(f;\alpha,\beta)$ verbesserter Näherungswert $Q^*_{h_1}(f;\alpha,\beta)$ für $I(f;\alpha,\beta)$ angeben; es gilt

$$\begin{aligned} Q^*_{h_1}(f;\alpha,\beta) &= Q_{h_1}(f;\alpha,\beta) + E^*_{h_1}(f;\alpha,\beta) \\ &= Q_{h_1}(f;\alpha,\beta) + \tfrac{1}{\left(\frac{h_2}{h_1}\right)^q - 1}\big(Q_{h_1}(f;\alpha,\beta) - Q_{h_2}(f;\alpha,\beta)\big). \end{aligned} \tag{14.33}$$

Wählt man speziell $h_2 = 2h_1$ und setzt $h_1 = h$, so erhält (14.32) die Form

$$E_h(f;\alpha,\beta) \approx \frac{Q_h(f;\alpha,\beta) - Q_{2h}(f;\alpha,\beta)}{2^q - 1} \tag{14.34}$$

und für $Q^*_h(f;\alpha,\beta)$ ergibt sich aus (14.33) die Beziehung

$$Q^*_h(f;\alpha,\beta) = Q_h(f;\alpha,\beta) + \frac{1}{2^q - 1}\big(Q_h(f;\alpha,\beta) - Q_{2h}(f;\alpha,\beta)\big). \tag{14.35}$$

Dabei sind $Q_h(f;\alpha,\beta)$ der mit der Schrittweite h berechnete Näherungswert, $Q_{2h}(f;\alpha,\beta)$ der mit der doppelten Schrittweite berechnete Näherungswert und $Q^*_h(f;\alpha,\beta)$ der gegenüber $Q_h(f;\alpha,\beta)$ verbesserte Näherungswert für $I(f;\alpha,\beta)$.

Für die Trapezformeln, die Simpsonsche Formel und die 3/8-Formel lauten die (14.34) entsprechenden Fehlerschätzungsformeln und die (14.35) entsprechenden verbesserten Näherungswerte Q^*_h:

Sehnen- und Tangententrapezformel $(q = 2)$

$$\begin{aligned} E^{ST}_h &\approx \tfrac{1}{3}(Q^{ST}_h - Q^{ST}_{2h}) \quad , \qquad E^{TT}_h \approx \tfrac{1}{3}(Q^{TT}_h - Q^{TT}_{2h}), \\ Q^{*ST}_h &= Q^{ST}_h + \tfrac{1}{3}(Q^{ST}_h - Q^{ST}_{2h}), \qquad Q^{*TT}_h = Q^{TT}_h + \tfrac{1}{3}(Q^{TT}_h - Q^{TT}_{2h}); \end{aligned}$$

Simpsonsche Formel und 3/8-Formel $(q = 4)$

$$\begin{aligned} E^{S}_h &\approx \tfrac{1}{15}(Q^{S}_h - Q^{S}_{2h}) \quad , \qquad E^{3/8}_h \approx \tfrac{1}{15}(Q^{3/8}_h - Q^{3/8}_{2h}), \\ Q^{*S}_h &= Q^{S}_h + \tfrac{1}{15}(Q^{S}_h - Q^{S}_{2h}), \qquad Q^{*3/8}_h = Q^{3/8}_h + \tfrac{1}{15}(Q^{3/8}_h - Q^{3/8}_{2h}). \end{aligned}$$

Mit Hilfe der Euler-Maclaurin-Formeln lässt sich zeigen, dass bei Verwendung der gegenüber Q^{ST}_h und Q^{S}_h verbesserten Näherungswerte Q^{*ST}_h und Q^{*S}_h für I sogar zwei h-Potenzen in der Fehlerordnung gewonnen werden; es gilt

$$\begin{aligned} I(f;\alpha,\beta) &= Q_h^{*ST}(f;\alpha,\beta) + O(h^4) \quad \text{bzw.} \\ I(f;\alpha,\beta) &= Q_h^{*S}(f;\alpha,\beta) + O(h^6), \end{aligned}$$

vgl. auch Abschnitt 14.10.

Fehlerschätzung bei nichtäquidistanter Zerlegung

Mit allen für ein Referenzintervall bestimmten und dann zusammengesetzten Quadraturformeln kann eine Fehlerschätzung so vorgenommen werden, dass man $I(f;\alpha,\beta)$ einmal für die Zerlegung $Z : \alpha = t_0 < t_1 < \ldots < t_m = \beta$ und einmal für die Zerlegung $Z/2$, wo jeweils die Intervallmitten von $[t_i, t_{i+1}]$ als zusätzliche Stützstellen verwendet werden, näherungsweise berechnet.

Dann gilt bei Verwendung einer Quadraturformel der globalen Fehlerordnung q, die Polynome bis zum Grade $q-1$ exakt integriert, für den globalen Fehler von $Q_{Z/2}$

$$E_{Z/2}(f;\alpha,\beta) \approx \frac{Q_{Z/2}(f;\alpha,\beta) - Q_Z(f;\alpha,\beta)}{2^q - 1} = E^*_{Z/2}(f;\alpha,\beta)$$

und man erhält einen gegenüber $Q_{Z/2}$ verbesserten Näherungswert

$$Q^*_{Z/2}(f;\alpha,\beta) = Q_{Z/2}(f;\alpha,\beta) + E^*_{Z/2}(f;\alpha,\beta)$$

von der globalen Fehlerordnung $q+1$.

Rechnungsfehler. Während der globale Verfahrensfehler z. B. im Falle der ST-Regel bzw. S-Regel von zweiter bzw. von vierter Ordnung mit $h \to 0$ abnimmt, wächst der Rechnungsfehler in beiden Fällen von der Ordnung $O(1/h)$, so dass der Gesamtfehler (Verfahrensfehler plus Rechnungsfehler) nicht beliebig klein gehalten werden kann. Diese Aussage gilt auch für andere Quadraturformeln. Es ist empfehlenswert, die Schrittweite h so zu wählen, dass Verfahrensfehler und Rechnungsfehler etwa von gleicher Größenordnung sind.

Im Falle der ST-Regel ergibt sich nach [MCCR1987] für den globalen Rechnungsfehler die Beziehung

$$r_h(f;\alpha,\beta) = \frac{1}{2h}(\beta-\alpha)^2\varepsilon,$$

wobei ε der maximale absolute Rechnungsfehler pro Rechenschritt ist.

14.12 Adaptive Quadraturverfahren

Bei den bisher behandelten Quadraturverfahren wird im Allgemeinen das Integrationsintervall in äquidistante Teilintervalle zerlegt. Die Länge der Teilintervalle ergibt sich aus der Genauigkeitsforderung.

Je nach der Gestalt des Graphen der zu integrierenden Funktion kann es aber durchaus sinnvoll sein, bei gleichbleibender Genauigkeitsanforderung mit unterschiedlichen Schrittweiten zu arbeiten.

Berechnet man z. B. das Integral $I(f;\alpha,\beta)$ über das Romberg-Verfahren mit Adaption, so wird die fortlaufende Schrittweitenhalbierung einzelner Teilintervalle dann gestoppt, wenn die lokale Fehlerschätzung ausreichende Genauigkeit anzeigt. Es werden also nur noch diejenigen Teilintervalle weiter verfeinert, deren geschätzter Fehler oberhalb einer vorgegebenen Genauigkeitsschranke liegt. Im ungünstigsten Falle erhält man das in Abschnitt 14.10 beschriebene Romberg-Verfahren. Bei der Adaption wird also versucht, die Anzahl der Teilintervalle (und damit der Funktionsauswertungen) möglichst klein zu halten. Man kann natürlich ganz analog zur Sehnentrapezformel, die dem Romberg-Verfahren zugrunde liegt, Gaußsche oder andere Quadraturformeln zur Adaption benutzen.

Beispiel 14.24.

Vergleich verschiedener Quadraturverfahren zur Berechnung des elliptischen Integrals

$$\int_0^{\pi/2} \sqrt{1-\left(\frac{3}{4}\right)^2 \sin^2 x}\,\mathrm{d}x$$

mit der relativen Genauigkeit $5\cdot 10^{-6}$.

Verfahren	ermittelte Näherung	geschätzter Fehler absolut	geschätzter Fehler relativ	Anz. Funkt.-Auswertungen
NC-ST	1.3184720082	$4.71784\cdot 10^{-6}$	$3.6\cdot 10^{-6}$	14
NC-Simps	1.3184721413	$-5.66085\cdot 10^{-7}$	$-4.3\cdot 10^{-7}$	24
NC-3	1.3184721205	$-2.12303\cdot 10^{-7}$	$-1.6\cdot 10^{-7}$	34
NC-4	1.3184683010	$3.50997\cdot 10^{-6}$	$2.7\cdot 10^{-6}$	14
NC-5	1.3184700508	$2.00529\cdot 10^{-6}$	$1.5\cdot 10^{-6}$	17
NC-6	1.3184723011	$1.79922\cdot 10^{-8}$	$1.4\cdot 10^{-8}$	20
NC-7	1.3184722244	$8.13637\cdot 10^{-9}$	$6.2\cdot 10^{-9}$	23
Romberg	1.3184683010	$-3.84024\cdot 10^{-6}$	$-2.9\cdot 10^{-6}$	9
Ad-Gau2	1.3184720840	$2.77101\cdot 10^{-6}$	$2.1\cdot 10^{-6}$	18
Ad-Gau3	1.3184759445	$-3.30398\cdot 10^{-6}$	$-2.5\cdot 10^{-6}$	9
Ad-Gau4	1.3184719355	$-1.91333\cdot 10^{-8}$	$-1.5\cdot 10^{-8}$	12
Ad-Gau5	1.3184721117	$1.38011\cdot 10^{-9}$	$1.0\cdot 10^{-9}$	15
Ad-CC-2	1.3184721413	$-3.84024\cdot 10^{-6}$	$-2.9\cdot 10^{-6}$	27
Ad-CC-4	1.3184738586	$-1.31933\cdot 10^{-6}$	$-1.0\cdot 10^{-6}$	15
Ad-CC-6	1.3184720981	$1.57008\cdot 10^{-9}$	$1.2\cdot 10^{-9}$	21
Ad-CC-8	1.3184721081	$2.57731\cdot 10^{-11}$	$2.0\cdot 10^{-11}$	27
Ad-NC-1	1.3184721080	$3.32718\cdot 10^{-8}$	$2.5\cdot 10^{-8}$	35
Ad-NC-2	1.3184721413	$-3.84024\cdot 10^{-6}$	$-2.9\cdot 10^{-6}$	24
Ad-NC-3	1.3184721205	$-1.44341\cdot 10^{-6}$	$-1.1\cdot 10^{-6}$	33
Ad-NC-4	1.3184683010	$3.50997\cdot 10^{-6}$	$2.7\cdot 10^{-6}$	14
Ad-NC-5	1.3184700508	$2.00529\cdot 10^{-6}$	$1.5\cdot 10^{-6}$	17
Ad-NC-6	1.3184723011	$1.79922\cdot 10^{-8}$	$1.4\cdot 10^{-8}$	20
Ad-NC-7	1.3184722244	$8.13637\cdot 10^{-9}$	$6.2\cdot 10^{-9}$	23

□

14.13 Konvergenz der Quadraturformeln

Aus (14.12) folgt im Falle der Sehnentrapezformel unter der Voraussetzung, dass f'' in $[a,b]$ existiert und beschränkt ist

$$\Big| \int_\alpha^\beta f(x)\,\mathrm{d}x - Q^{ST}(f;\alpha,\alpha+Nh) \Big| = \frac{h^2}{12}(\beta-\alpha)|f''(\eta)| \quad \text{mit} \quad \eta \in [\alpha,\beta]\,.$$

Setzt man wie in (14.25) $N_k = 2^k N$, $h_k = h/2^k$, so wird bei fortgesetzter Intervallhalbierung für hinreichend großes $k > K_1(\varepsilon_1)$

$$\Big| \int_\alpha^\beta f(x)\,\mathrm{d}x - Q^{ST}(f;\alpha,\alpha+N_k h) \Big| = \frac{1}{12}\left(\frac{h}{2^k}\right)^2 (\beta-\alpha)|f''(\eta)| < \varepsilon_1\,. \tag{14.36}$$

(14.36) gilt auch noch, wenn man statt der speziellen Nullfolge $h/2^k$ eine beliebige Nullfolge $h_1, h_2, \ldots, h_p, \ldots$ mit $\lim\limits_{p\to\infty} h_p = 0$ wählt, wobei anstelle von $2^k N$ eine Folge $\{N_p\}$ mit $N_p \to \infty$ tritt. Analog gilt im Falle der Simpsonschen Formel

$$\Big| \int_\alpha^\beta f(x)\,\mathrm{d}x - Q^{S}(f;\alpha,\alpha+N_k\cdot 2h) \Big| < \varepsilon_2 \quad \text{für} \quad k > K_2(\varepsilon_2)$$

unter der Voraussetzung, dass $f^{(4)}(x)$ in $[\alpha,\beta]$ existiert und beschränkt ist. Damit ist die Konvergenz der Sehnentrapezformel und der Simpsonschen Formel unter den genannten Voraussetzungen nachgewiesen. In gleicher Weise lässt sich die Konvergenz der Tangententrapezformel zeigen.

Satz 14.25.
Eine Quadraturformel der Form

$$Q^{(n)}(f;\alpha,\beta) = \sum_{k=0}^{n} A_k^{(n)} f(x_k^{(n)}), \quad x_k^{(n)} \in [\alpha,\beta], \tag{14.37}$$

konvergiert für $n \to \infty$ und für jede in $[\alpha,\beta]$ stetige Funktion f genau dann gegen $I(f;\alpha,\beta)$, d. h.

$$\lim_{n\to\infty} Q^{(n)}(f;\alpha,\beta) = \lim_{n\to\infty} \sum_{k=0}^{n} A_k^{(n)} f(x_k^{(n)}) = I(f;\alpha,\beta), \tag{14.38}$$

wenn

1. (14.38) für jedes Polynom $f \equiv \Phi$ der Form $\Phi(x,\boldsymbol{c}) = \sum\limits_{k=0}^{n} c_k x^k$ erfüllt ist und
2. eine Konstante K existiert, so dass $\sum\limits_{k=0}^{n} |A_k^{(n)}| < K$ für jedes n gilt.

Wendet man (14.37) auf $f(x) = 1$ an, so erhält man mit 1.

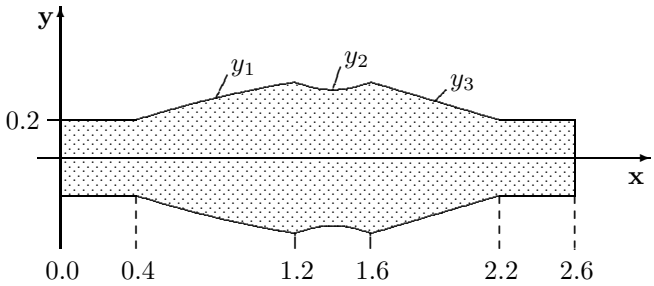
$$Q^{(n)}(1;\alpha,\beta) = \sum_{k=0}^{n} A_k^{(n)} = \int_\alpha^\beta \mathrm{d}x = \beta - \alpha.$$

Sind alle Gewichte $A_k^{(n)} > 0$, so ist 2. sicher erfüllt; treten dagegen negative Gewichte auf, so kann $|A_0^{(n)}| + |A_1^{(n)}| + \ldots + |A_n^{(n)}|$ bei genügend großem n beliebig groß werden.

Für Interpolationsquadraturformeln mit äquidistanten Stützstellen treten negative Gewichte erstmals bei 9 Stützstellen auf. Bis zu 8 Stützstellen stimmen die Newton-Cotes-Formeln mit den Formeln überein, die man bei dem Romberg-Verfahren für die entsprechende Stützstellenzahl erhält. Die Konvergenz der Romberg-Integration wurde bereits in Abschnitt 14.10 behandelt. Da die Werte von $|A_k^{(n)}|$ für Interpolationsquadraturformeln mit äquidistanten Stützstellen für wachsendes n unbegrenzt anwachsen, ist 2. nicht erfüllt.
Für die Gaußschen Quadraturformeln sind die Gewichte bei beliebiger Stützstellenzahl stets positiv, ebenso ist die 1. Bedingung erfüllt, diese Formel konvergieren also für $n \to \infty$.

14.14 Anwendungsbeispiel

Gegeben: Der folgende Walzenkörper:



Gesucht: Die Mantelfläche M (ohne Stirnflächen) und das Volumen V des Walzenkörpers mit Hilfe der Simpsonschen Formel. Pro Abschnitt sollen 51 Stützstellen berechnet werden.

Profilfunktionen in den einzelnen Abschnitten:

$$y_1(x) = \sqrt{0.15 \cdot x - 0.02}$$

$$y_2(x) = 0.36 + (x - 1.4)^2$$

$$y_3(x) = \frac{2.8 - x}{3}$$

Formeln:

$$M = 2\pi \int_0^\ell y\sqrt{1 + y'^2}\,\mathrm{d}x, \quad V = \pi \int_0^\ell y^2\,\mathrm{d}x$$

Lösung: Die Mantelflächen der einzelnen Teilbereiche sind:

$$\begin{aligned}
M_0 &= 1.005309626678314 \cdot 10^{-1}\ \mathrm{m}^2\\
M_1 &= 1.610142717146783\ \mathrm{m}^2\\
M_2 &= 3.603120565805025 \cdot 10^{-1}\ \mathrm{m}^2\\
M_3 &= 4.128254066379155 \cdot 10^{-1}\ \mathrm{m}^2\\
M_4 &= 1.005309626678314 \cdot 10^{-1}\ \mathrm{m}^2
\end{aligned}$$

Die Gesamtmantelfläche des Rotationskörpers beträgt:

$$M_{\mathrm{ges}} = 2.584342105700864\,\mathrm{m}^2.$$

Die Volumina der einzelnen Teilbereiche sind:

$$\begin{aligned}
V_0 &= 5.026548133391571 \cdot 10^{-2}\ \mathrm{m}^3\\
V_1 &= 2.513274253949293 \cdot 10^{-1}\ \mathrm{m}^3\\
V_2 &= 1.753260162734987 \cdot 10^{-1}\ \mathrm{m}^3\\
V_3 &= 1.759291706246914 \cdot 10^{-1}\ \mathrm{m}^3\\
V_4 &= 5.026548133391569 \cdot 10^{-2}\ \mathrm{m}^3
\end{aligned}$$

Das Gesamtvolumen beträgt $V_{\mathrm{ges}} = 7.031135749609508 \cdot 10^{-1}\ \mathrm{m}^3$. □

14.15 Entscheidungshilfen für die Auswahl der geeigneten Methode

Die Güte der Näherung für ein bestimmtes Integral hängt ab von

- der Fehlerordnung q der Quadraturformel,
- der Feinheit der Zerlegung Z,
- der Glattheit des Integranden.

Ist der Integrand beliebig oft differenzierbar, so wählt man die Formeln nach der Fehlerordnung aus und passt die Zerlegung dem Verlauf des Integranden an: Man zerlegt dort feiner, wo sich der Integrand stark ändert und wählt größere Teilintervalle, wo sich der Integrand nur langsam ändert (siehe Beispiele in [NIED1987] 11.2).

Für geringe Genauigkeitsansprüche sind von den behandelten Newton-Cotes-Formeln, die Simpsonsche Formel und die 3/8-Formel zu empfehlen; sie sind der Trapezformel (Ausnahme periodische Funktionen: siehe Abschnitt 14.3.1) wegen der günstigeren Fehlerordnung vorzuziehen. Newton-Cotes-Formeln höherer Ordnung sind nur bedingt zu empfehlen, weil sich bei ihnen Rundungsfehler wegen der großen Gewichte stärker auswirken.

Die Tschebyscheffschen Quadraturformeln sind in Bezug auf die Auswirkung von Rundungsfehlern günstig, weil sie mit gleichen Gewichten arbeiten. Vergleicht man die Tschebyscheffschen Formeln, die Clenshaw-Curtis-Formeln und die Gaußschen Formeln miteinander, so erfordert die Berechnung der Gewichte und Stützstellen bei den Gaußschen Formeln den größten Aufwand. Sieht man davon ab, so sind die Gaußschen Formeln am effektivsten. Bei höherer Ordnung sind aus den genannten Gründen die Clenshaw-Curtis-Formeln vorzuziehen, weil sich Stützstellen und Gewichte sehr leicht berechnen lassen.

Ein sehr einfaches und dennoch äußerst effektives und stabiles Verfahren ist das Romberg-Verfahren.

Auf den Verlauf des Integranden nehmen die adaptiven Quadraturverfahren automatisch Rücksicht. Arbeitet man etwa mit den Gaußschen Formeln adaptiv, so ist dies bezüglich des Rechenaufwandes und der Genauigkeit die optimale Vorgehensweise.

Zur Bestimmung der Länge einer Kurve und des Flächeninhalts einer ebenen geschlossenen Kurve siehe auch Abschnitt 11.4 bzw. Abschnitt 11.5.

Ergänzende Literatur zu Kapitel 14

[BART2001] 12, 12.5; [BJOR1979], 7.4; [BRON1991], 1.1.3.3, 1.1.3.4, 3.1.7; [DEUF2002] Bd.1, Kap.9; [ENGE1980], 6.1, 7.7.4; [HAMM1978], 10-12, S.103; [HAMM1994], 7; [KAHA1983]; [KNOR2003], 7; [KROM1994]; [MAES1985]; [MAES1988], 7; [OPFE2002], Kap.5; [PLAT2000], Kap.6; [PREU2001], Kap.7; [QUAD1983]; [QUAR2002], Kap.9; [RALS2001], 4.11; [RALS1979], IV.; [STOE1989], 3.1-5; [STOE1999], Kap.3; [WEIS1984], 4.1-4.3, 5.3; [WERN1993], III §6 V, III §7, III §8.

Kapitel 15

Numerische Kubatur

15.1 Problemstellung

Es werden Integrale über beschränkte ebene Bereiche betrachtet. B sei ein Bereich der x, y-Ebene mit stückweise glattem Rand

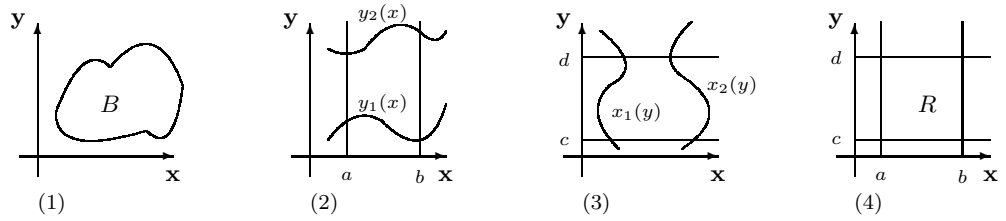


Abb. 15.1.

Die Funktion $f : B \to \mathbb{R}$, $B \subset \mathbb{R}^2 = \mathbb{R} \times \mathbb{R}$ sei stetig auf B. Dann heißt

$$I(f; B) = \iint\limits_B f(x, y)\, \mathrm{d}x\, \mathrm{d}y \tag{15.1}$$

das *Flächenintegral von f über B*.

Wenn $f(x, y) \geq 0$ für alle $(x, y) \in B$ gilt, so ist (15.1) das Volumen des Zylinders mit der Grundfläche B und mit der Deckfläche $z = f(x, y)$ für $(x, y) \in B$.

Das Flächenintegral (15.1) lässt sich in den Fällen (2), (3), (4) aus Abbildung 15.1 durch Hintereinanderschaltung zweier eindimensionaler Integrale berechnen. Ist B ein Bereich vom Typ (2) mit

$$B = \{(x, y) | \quad a \leq x \leq b,\ y_1(x) \leq y \leq y_2(x)\}\,,$$

so gilt

$$\iint\limits_B f(x, y)\, \mathrm{d}x\, \mathrm{d}y = \int\limits_a^b \left(\int\limits_{y_1(x)}^{y_2(x)} f(x, y)\, \mathrm{d}y \right) \mathrm{d}x,$$

hat B die Gestalt (3) mit

$$B = \{(x,y)| \quad x_1(y) \le x \le x_2(y),\ c \le y \le d\}\,,$$

so gilt

$$\iint\limits_B f(x,y)\,\mathrm{d}x\,\mathrm{d}y = \int\limits_c^d \left(\int\limits_{x_1(y)}^{x_2(y)} f(x,y)\,\mathrm{d}x \right) \mathrm{d}y,$$

und im Falle eines rechteckigen Bereiches $B = R$ mit

$$R = \{(x,y)| \quad a \le x \le b,\ \ c \le y \le d\} =: [a,b\,;\,c,d\,]$$

gilt

$$\iint\limits_R f(x,y)\,\mathrm{d}x\,\mathrm{d}y = \int\limits_a^b \left(\int\limits_c^d f(x,y)\,\mathrm{d}y \right) \mathrm{d}x,$$

(vgl. [FICH1987] III, XVI).

Zur näherungsweisen Berechnung von Flächenintegralen werden sogenannte *Kubaturformeln* $C(f;B)$ verwendet. Der Name erklärt sich daraus, dass diese Formeln auch zur Volumenberechnung eingesetzt werden.

Zum Beispiel erhält man einen Näherungswert für $I(f;B)$, wenn man B in Teilbereiche B_j, $j = 1(1)N$, zerlegt mit der Grundfläche F_j und zur Berechnung des Gesamtvolumens die N Volumina der einzelnen Zylinder über den B_j addiert.

Mit je einem beliebigen Punkt $(x_j,y_j) \in B_j$ erhält man für das Volumen

$$I(f;B) \approx \sum_{j=1}^{N} F_j f(x_j,y_j);$$

die rechte Seite ist eine (summierte) Kubaturformel. Wählt man für die Punkte (x_j,y_j) die Schwerpunkte der Teilflächen B_j, so ergibt sich die sogenannte *Schwerpunkt-Kubaturformel*.

Analog zum eindimensionalen Fall lassen sich Kubaturformeln für Referenzbereiche (Elementarbereiche) B_r z. B. in der Form

$$C(f;B_r) = \sum_{i=0}^{m}\sum_{k=0}^{n} a_{ik} f(x_i,x_k) \approx \iint\limits_{B_r} f(x,y)\,\mathrm{d}x\,\mathrm{d}y = I(f;B_r) \tag{15.2}$$

oder

$$C(f;B_r) = \sum_{j=1}^{N} A_j f(x_j,y_j) \approx \iint\limits_{B_r} f(x,y)\,\mathrm{d}x\,\mathrm{d}y = I(f;B_r) \tag{15.3}$$

als Linearkombination aus $N = (m+1)\cdot(n+1)$ Funktionswerten des Integranden an N diskreten Knoten aus dem Referenzbereich B_r mit N Gewichten darstellen.

Definition 15.1. (*Genauigkeitsgrad*)
Die Kubaturformel (15.2) bzw. (15.3) besitzt als Näherung für das Flächenintegral $I(f;B)$ den Genauigkeitsgrad L, wenn sie für *alle* Polynome P_L mit

$$P_L(x,y) = \sum_{0\le s+t\le L} c_{st} x^s y^t \tag{15.4}$$

höchstens L-ten Grades exakt ist, jedoch nicht mehr für ein beliebiges Polynom P_{L+1} vom Grad $L+1$:

$$\begin{array}{lcl} C(P_L;B) & \stackrel{!}{=} & I(P_L;B)\ , \\ C(P_{L+1};B) & \neq & I(P_{L+1};B) \end{array}$$

für mindestens ein Polynom P_{L+1} mit $s+t=L+1$.

Ist eine für einen Referenzbereich B_r konstruierte Kubaturformel C exakt für alle Polynome P_L der Form (15.4) vom Grad $\le L$ und ist f genügend oft differenzierbar in B_r, so beträgt die lokale Fehlerordnung $q_\ell = L+2$.

Will man ein Flächenintegral (15.1) über einem beliebigen Bereich B berechnen, so zerlegt man B in Teilbereiche B_j, die die Form des Referenzbereiches B_r besitzen, und wendet auf jeden Teilbereich B_j die für B_r konstruierte und auf B_j transformierte Kubaturformel an und summiert über j (dies ist eventuell erst nach einer nichtlinearen Transformation möglich). Die so zusammengesetzte Formel heißt *summierte* oder *zusammengesetzte Kubaturformel*, sie besitzt die globale Fehlerordnung $q_g = L+1$.

Im Folgenden werden nur Kubaturformeln zu rechteckigen und dreieckigen Referenzbereichen betrachtet. Bekanntlich lassen sich auch andere Bereiche durch nichtlineare Variablentransformationen auf Referenzrechtecke bzw. -dreiecke abbilden, siehe dazu [ENGE1980]; [FICH1987], §4; [MAES1988], 7.4.2, [NIEM1991], 7.

15.2 Konstruktion von Interpolationskubaturformeln

B_r sei der Referenzbereich, für den eine Kubaturformel $C(f;B_r)$ durch Integration eines Interpolationspolynoms konstruiert werden soll. Dazu legt man durch $(m+1)\cdot(n+1)$ Stützpunkte

$$\big(x_i, y_k, f(x_i,y_k)\big), \quad i = 0(1)m, \quad k = 0(1)n$$

das zugehörige Interpolationspolynom Φ mit $\Phi(x_i, y_k) = f(x_i, y_k)$. Für ein Rechteckgitter ist diese Interpolationsaufgabe eindeutig gelöst (vgl. Abschnitt 9.7.1). Im Folgenden wird als Referenzbereich ein Rechteck R_r gewählt. In der Form von Lagrange lautet Φ für R_r

$$\begin{array}{rcl} \Phi(x,y) & = & \sum\limits_{i=0}^{m}\sum\limits_{k=0}^{n} L_i(x)\ L_k(y)\ f(x_i,y_k) \quad \text{mit} \\ L_i(x) & = & \prod\limits_{\substack{j=0\\ j\neq i}}^{m} \frac{x-x_j}{x_i-x_j}, \quad L_k(y) = \prod\limits_{\substack{j=0\\ j\neq k}}^{n} \frac{y-y_j}{y_k-y_j}. \end{array}$$

Daraus folgt

$$\begin{aligned}
\int_a^b\int_c^d f(x,y)\,\mathrm{d}x\,\mathrm{d}y &\approx \int_a^b\int_c^d \Phi(x,y)\,\mathrm{d}x\,\mathrm{d}y \\
&= \int_a^b\int_c^d \sum_{i=0}^{m}\sum_{k=0}^{n} L_i(x)\;L_k(y)\;f(x_i,y_k)\,\mathrm{d}x\,\mathrm{d}y \\
&= \sum_{i=0}^{m}\sum_{k=0}^{n} \underbrace{\int_a^b L_i(x)\,\mathrm{d}x}_{A_i}\;\underbrace{\int_c^d L_k(y)\,\mathrm{d}y}_{B_k}\,f(x_i,y_k) \qquad (15.5)\\
&= \sum_{i=0}^{m}\sum_{k=0}^{n} A_i\;B_k\;f(x_i,y_k) \\
&=: \sum_{i=0}^{m}\sum_{k=0}^{n} a_{ik}\;f_{ik} =: C(f;R_r)\,.
\end{aligned}$$

Die A_i bzw. B_k entsprechen den Gewichten eindimensionaler Interpolationsquadraturformeln. Die Kubaturgewichte $a_{ik} := A_iB_k$ ergeben sich als Matrizenprodukt $\boldsymbol{A}\boldsymbol{B}^{\mathsf{T}}$ aus der $(m+1,1)$-Matrix $\boldsymbol{A}$ und der $(1,n+1)$-Matrix $\boldsymbol{B}^{\mathsf{T}}$

$$\boldsymbol{A} = \begin{pmatrix} A_0 \\ A_1 \\ \vdots \\ A_m \end{pmatrix}, \quad \boldsymbol{B}^{\mathsf{T}} = (B_0B_1\cdots B_n)$$

mit den Quadraturgewichten A_i, B_k; es gilt

$$(a_{ik}) := \begin{pmatrix} A_0B_0 & A_0B_1 & \cdots & A_0B_n \\ A_1B_0 & A_1B_1 & \cdots & A_1B_n \\ \vdots & & & \\ A_mB_0 & A_mB_1 & \cdots & A_mB_n \end{pmatrix} = \begin{pmatrix} A_0 \\ A_1 \\ \vdots \\ A_m \end{pmatrix} (B_0B_1\cdots B_n) = \boldsymbol{A}\boldsymbol{B}^{\mathsf{T}}. \qquad (15.6)$$

Beispiel 15.2.

Wenn

$$\boldsymbol{A} = \begin{pmatrix} 2 \\ 1 \\ 3 \end{pmatrix} \text{ und } \boldsymbol{B} = \begin{pmatrix} 1 \\ 2 \\ 0 \\ 3 \end{pmatrix}, \text{ also } \boldsymbol{B}^{\mathsf{T}} = \begin{pmatrix} 1 & 2 & 0 & 3 \end{pmatrix},$$

dann ist

$$a_{ik} := \boldsymbol{A}\boldsymbol{B}^{\mathsf{T}} = \begin{pmatrix} 2 \\ 1 \\ 3 \end{pmatrix} \begin{pmatrix} 1 & 2 & 0 & 3 \end{pmatrix} = \begin{pmatrix} 2 & 4 & 0 & 6 \\ 1 & 2 & 0 & 3 \\ 3 & 6 & 0 & 9 \end{pmatrix}.$$

□

Die interpolatorische Kubaturformel (15.5) stellt somit eine Linearkombination aus $(m+1)\cdot(n+1)$ Gewichten a_{ik} sowie Funktionswerten f_{ik} von f an den $(m+1)\cdot(n+1)$ verschiedenen Knoten (x_i,y_k) dar, die auf einem Referenz-Rechteckgitter definiert sind.

Zur Berechnung der Knoten und Gewichte wird analog zum eindimensionalen Fall die Forderung gestellt, dass Polynome P_L möglichst hohen Grades L exakt integriert werden. Gibt man in (15.5) z. B. sämtliche Knoten des Referenzrechteckes vor, so ergibt sich aus der oben genannten Forderung für die Gewichte der Newton-Cotes-Kubaturformeln ein lineares Gleichungssystem mit $N=(m+1)\cdot(n+1)$ Gleichungen.

Schreibt man die Gewichte vor, so ergeben sich die Knoten als Lösungen eines nichtlinearen Gleichungssystems; man erhält die *Tschebyscheffschen Kubaturformeln.*

Lässt man schließlich Knoten und Gewichte frei, so lassen sich über ein nichtlineares Gleichungssystem die optimalen *Gaußschen Kubaturformeln* ermitteln.

Wenn die Kubaturformel (15.5) alle Polynome P_L bis zu möglichst hohem Grad L exakt integrieren soll, so können speziell die Monome $x^s y^t$ zur Berechnung der Knoten und Gewichte verwendet werden, und aus der Forderung

$$C(x^s y^t; R_r) \stackrel{!}{=} I(x^s y^t; R_r) \quad \text{für} \quad s+t \le L$$

ergibt sich mit (15.5)

$$\begin{aligned}\sum_{i=0}^{m}\sum_{k=0}^{n} a_{ik}\, x_i^s\, y_k^t \quad &\stackrel{!}{=} \quad \int_a^b\int_c^d x^s y^t \,\mathrm{d}y\,\mathrm{d}x = \int_a^b x^s\,\mathrm{d}x \int_c^d y^t\,\mathrm{d}y \\ &= \quad \frac{1}{(s+1)(t+1)}\,(b^{s+1}-a^{s+1})\,(d^{t+1}-c^{t+1}) \\ &\qquad \text{mit} \quad s+t=0,1,2,\ldots,L.\end{aligned}$$

Das sind für $s+t=0,1,2,\ldots,L$ die Gleichungen

$$\sum_{i=0}^{m}\sum_{k=0}^{n} a_{ik}\, x_i^s\, y_k^t = \frac{1}{(s+1)(t+1)}\,(b^{s+1}-a^{s+1})\,(d^{t+1}-c^{t+1}) \tag{15.7}$$

bzw. mit $N=(m+1)\cdot(n+1)$

$$\sum_{j=1}^{N} A_j\, x_j^s\, y_j^t = \frac{1}{(s+1)(t+1)}\,(b^{s+1}-a^{s+1})\,(d^{t+1}-c^{t+1}). \tag{15.8}$$

Dies sind $\frac{1}{2}(L+1)(L+2)$ Bedingungen, die zum Erreichen des gewünschten Genauigkeitsgrades L erfüllt sein müssen. Die zu jedem Genauigkeitsgrad L gehörigen Monome $x^s y^t$ lassen sich aus der folgenden Tabelle ablesen:

Genauigkeits-grad L	Monome $x^s y^t$ mit $s+t \le L$	Anzahl $\frac{1}{2}(L+1)(L+2)$ der Monome $s+t \le L$
0	1	1
1	$x \quad y$	3
2	$x^2 \quad xy \quad y^2$	6
3	$x^3 \quad x^2y \quad xy^2 \quad y^3$	10
4	$x^4 \quad x^3y \quad x^2y^2 \quad xy^3 \quad y^4$	15
5	$x^5 \quad x^4y \quad x^3y^2 \quad x^2y^3 \quad xy^4 \quad y^5$	21

Um z. B. den Genauigkeitsgrad $L = 1$ zu erreichen, muss die Kubaturformel alle Polynome vom Grad $s + t \le 1$ exakt integrieren. Mit den Monomen 1, x, y ergeben sich daraus drei Bedingungen für die freien Parameter. Eventuell fehlende Gleichungen stellt man mit Monomen $x^s y^t$ für $s + t = L + 1$ auf, ohne dass die zu konstruierende Kubaturformel für alle Monome mit $s + t = L + 1$ erfüllt ist (siehe dazu Abschnitt 15.3).

15.3 Newton-Cotes-Formeln für rechteckige Integrationsbereiche

Die Newton-Cotes-Kubaturformeln für einen rechteckigen Integrationsbereich R_r sind interpolatorische Formeln der Gestalt

$$C(f; R_r) = \sum_{i=0}^{m} \sum_{k=0}^{n} a_{ik}\, f(x_i, y_k) \approx \iint\limits_{R_r} f(x, y)\, \mathrm{d}x\, \mathrm{d}y$$

bei vorgegebenen $N = (m + 1) \cdot (n + 1)$ voneinander verschiedenen Knoten $(x_i,\ y_k)$, die auf einem Referenz-Rechteck R_r mit konstanter Schrittweite h_x in x-Richtung bzw. h_y in y-Richtung definiert sind. Die N Gewichte a_{ik} ergeben sich aus dem linearen Gleichungssystem (15.7). Gleichzeitig lassen sich jedoch die Gewichte aus den entsprechenden Quadraturformeln ableiten.

Zunächst werden beispielhaft für die Referenzrechtecke in Abbildung 15.2 die Trapez-, Simpson- und 3/8-Kubaturformeln aufgestellt.

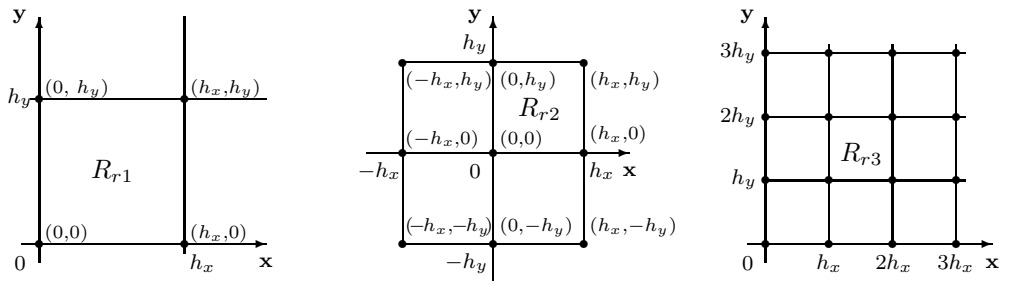


Abb. 15.2.

Trapez-Formel für das Referenzrechteck R_{r1}

Die $N = 4$ Gewichte a_{ik}, $i, k = 0, 1$, ergeben sich aus den Gewichten der entsprechenden Quadraturformeln gemäß (15.6)

$$(a_{ik}) = \boldsymbol{A}\boldsymbol{B}^{\mathsf{T}} = \frac{h_x}{2}\begin{pmatrix} 1 \\ 1 \end{pmatrix}\frac{h_y}{2}(1\ ,\ 1) = \frac{h_x h_y}{4}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}.$$

Damit lautet die Trapez-Kubaturformel

$$C^{\mathsf{T}}(f; R_{r1}) = \frac{h_x h_y}{4}\big[f(0,0) + f(h_x,0) + f(0,h_y) + f(h_x,h_y)\big]. \tag{15.9}$$

Diese Kubaturformel integriert alle bilinearen Polynome P_1 mit $s + t \leq 1$ exakt sowie zusätzlich alle Polynome mit $s = 1$, $t = 1$, aber nicht mehr die mit $s = 0$, $t = 2$ bzw. $t = 0$, $s = 2$, d. h. sie besitzt den Genauigkeitsgrad $L = 1$.

Beispiel 15.3.

Gegeben: Die sechs Polynome unterschiedlichen Grades

$$\begin{aligned} f(x,y) &= 2\,x + 2 \\ g(x,y) &= -4\,y + 3 \\ h(x,y) &= 2\,x - y + 1 \\ i(x,y) &= x^2 - 2 \\ j(x,y) &= 3\,x\,y + 2 \\ k(x,y) &= y^2 + 1 \end{aligned}$$

Gesucht: Die Unterschiede zwischen den Integralwerten und den Ergebnissen der Trapez-Formel

Lösung: Dazu werden im Referenzrechteck $h_x = h_y = 1$ gesetzt und zunächst die Doppelintegrale berechnet:

$$\begin{aligned} \int_0^1\int_0^1 f(x,y)\,\mathrm{d}x\,\mathrm{d}y &= 3 \\ \int_0^1\int_0^1 g(x,y)\,\mathrm{d}x\,\mathrm{d}y &= 1 \\ \int_0^1\int_0^1 h(x,y)\,\mathrm{d}x\,\mathrm{d}y &= \tfrac{3}{2} \\ \int_0^1\int_0^1 i(x,y)\,\mathrm{d}x\,\mathrm{d}y &= -\tfrac{5}{3} \\ \int_0^1\int_0^1 j(x,y)\,\mathrm{d}x\,\mathrm{d}y &= \tfrac{11}{4} \\ \int_0^1\int_0^1 k(x,y)\,\mathrm{d}x\,\mathrm{d}y &= \tfrac{5}{3} \end{aligned}$$

Beim Vergleich dieser Ergebnisse mit denen der Trapezformel

$$\begin{array}{lcllcl} C^{\mathsf{T}}(f;R) & = & \frac{1}{4}(2+4+2+4) & = & 3 & \\ C^{\mathsf{T}}(g;R) & = & \frac{1}{4}(3+3-1-1) & = & 1 & \\ C^{\mathsf{T}}(h;R) & = & \frac{1}{4}(1+3+0+2) & = & \frac{3}{2} & \\ C^{\mathsf{T}}(i;R) & = & \frac{1}{4}(-2-1-2-1) & = & -\frac{3}{2} & \neq \quad -\frac{5}{3} \\ C^{\mathsf{T}}(j;R) & = & \frac{1}{4}(2+2+2+5) & = & \frac{11}{4} & \\ C^{\mathsf{T}}(k;R) & = & \frac{1}{4}(1+1+3+3) & = & 2 & \neq \quad \frac{5}{3} \end{array}$$

erkennt man, dass die Anwendung auf $h(x,y)$ und sogar $j(x,y)$ tatsächlich noch Übereinstimmung zeigt, die auf $i(x,y)$ und $k(x,y)$ dagegen nicht mehr!

□

Simpsonsche Formel für das Referenzrechteck R_{r2}

Die $N=9$ Gewichte a_{ik}, $i,k=0,1,2$, ergeben sich aus

$$(a_{ik}) = \boldsymbol{A}\boldsymbol{B}^{\mathsf{T}} = \frac{h_x}{3}\begin{pmatrix} 1 \\ 4 \\ 1 \end{pmatrix}\frac{h_y}{3}(1,\ 4,\ 1) = \frac{h_x h_y}{9}\begin{pmatrix} 1 & 4 & 1 \\ 4 & 16 & 4 \\ 1 & 4 & 1 \end{pmatrix},$$

und daraus folgt die Simpsonsche Kubaturformel (vgl. Abb. 15.2)

$$\begin{array}{lcl} C^S(f;R_{r2}) & = & \frac{h_x h_y}{9}\Big\{f(-h_x,-h_y)+f(-h_x,h_y)+f(h_x,-h_y) \\ & + & f(h_x,h_y)+4\big[f(0,-h_y)+f(h_x,0)+f(0,h_y) \\ & + & f(-h_x,0)\big]+16f(0,0)\Big\} \end{array} \tag{15.10}$$

Sie ist für alle Polynome P_3 mit $s+t \le L=3$ (d. h. alle bikubischen Polynome) exakt, jedoch nicht mehr für alle Polynome mit $L=4$.

Beispiel 15.4.

Gegeben: Das bikubische Polynom $f(x,y) = x^2y - 2\,xy^2 + xy + 3$

Gesucht: Kubaturwert gemäß Simpsonscher Formel zu
$R_{r2} = \big\{\,(x,y)\,\big|\,-1 \le x \le 1;\ -1 \le y \le 1\,\big\}$

Lösung: Bestimmung von h_x und h_y: Die Längen der Rechteckseiten sind in beiden Richtungen gleich 2. Für die Simpsonsche Formel werden je zwei Segmente benötigt, also ergibt sich $h_x = h_y = 1$ (vgl. dazu auch die Bemerkung 15.6). Damit berechnet man die Näherung zu

$$\begin{array}{lcl} C^S(f;R_{r2}) & = & \frac{1\cdot 1}{9}\big\{f(-1,-1)+f(-1,1)+f(1,-1)+f(1,1) \\ & & +\,4\,[f(0,-1)+f(1,0)+f(0,1)+f(-1,0)]+16\,f(0,0)\big\} \\ & = & \frac{1}{9}\big\{5+5+(-1)+3+4\,[3+3+3+3]+16\cdot 3\big\} \\ & = & \frac{108}{9} = 12\,. \end{array}$$

Als Vergleich dient der wahre Integralwert des Polynoms:

$$\begin{aligned}\iint\limits_{R_{r2}} f(x,y)\,\mathrm{d}x\,\mathrm{d}y &= \int\limits_{-1}^{1}\int\limits_{-1}^{1} f(x,y)\,\mathrm{d}x\,\mathrm{d}y \\ &= \int\limits_{-1}^{1} y\ \left[\tfrac{1}{3}x^3\right]_{-1}^{1} + (-2y^2+y)\ \underbrace{\left[\frac{1}{2}x^2\right]_{-1}^{1}}_{=0} + 3\ [x]_{-1}^{1}\ \mathrm{d}y \\ &= \int\limits_{-1}^{1} \tfrac{2}{3}\,y + 6\,\mathrm{d}y = \underbrace{\int\limits_{-1}^{1} \frac{2}{3}\,y\,\mathrm{d}y}_{=0} + 6\ [y]_{-1}^{1} = 6\cdot 2 = 12\end{aligned}$$

Die Kubaturformel hat also erwartungsgemäß das Polynom exakt integriert. □

Beispiel 15.5.

Gegeben: Die Funktion

$$f(x,y) = 4\ \sin x\ \cos y + 1$$

Gesucht: Die näherungsweise Berechnung des Flächenintegrals von f mit der Simpsonschen Formel (15.10) über dem Bereich

$$R = \left\{\ (x,y)\mid 0 \le x \le 1\,,\ -1 \le y \le 1\ \right\}$$

Lösung: Dazu werden ausgehend vom Referenzrechteck die Argumente der Funktion transformiert (statt $f(-1,y)$, $f(0,y)$, $f(+1,y)$ verwendet man $f(0,y)$, $f(\frac{1}{2},y)$, $f(1,y)$; $h_x = \frac{1}{2}$, $h_y = 1$), und man erhält (vgl. dazu auch die Bemerkung 15.6) mit der neuen Formel das Ergebnis:

$$\begin{aligned}C^S(f;R) = \frac{\frac{1}{2}\cdot 1}{9}\Big\{ & f(0,-1)+f(0,1)+f(1,-1)+f(1,1) \\ & + 4\left[f(\tfrac{1}{2},-1)+f(1,0)+f(\tfrac{1}{2},1)+f(0,0)\right] + 16\,f(\tfrac{1}{2},0)\ \Big\} \\ \approx\ \tfrac{1}{18}\quad & \{\ 1+1+2.819+2.819 \\ & + 4\,(2.036+4.366+2.036+1) + 16\cdot 2.918\ \} \\ =\quad & 5.115\end{aligned}$$

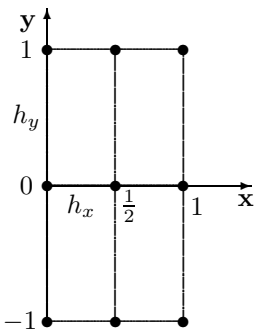


Abb. 15.3.

□

3/8-Formel für das Referenzrechteck R_{r3}

Die $N = 16$ Gewichte a_{ik}, $i, k = 0, 1, 2, 3$, ergeben sich aus

$$(a_{ik}) = \boldsymbol{A}\boldsymbol{B}^{\mathsf{T}} = \frac{3h_x}{8}\begin{pmatrix} 1 \\ 3 \\ 3 \\ 1 \end{pmatrix} \cdot \frac{3h_y}{8}(1, \quad 3, \quad 3, \quad 1) = \frac{9h_xh_y}{64}\begin{pmatrix} 1 & 3 & 3 & 1 \\ 3 & 9 & 9 & 3 \\ 3 & 9 & 9 & 3 \\ 1 & 3 & 3 & 1 \end{pmatrix},$$

und damit folgt die 3/8-Kubaturformel (vgl. Abb. 15.2)

$$\begin{aligned} C^{3/8}(f; R_{r3}) \quad = \quad & \frac{9h_xh_y}{64}\Big\{ f(0,0) + f(3h_x,0) + f(0,3h_y) + f(3h_x,3h_y) \\ & + 3\big[f(h_x,0) + f(2h_x,0) + f(3h_x,h_y) + f(3h_x,2h_y) \\ & \quad + f(2h_x,3h_y) + f(h_x,3h_y) + f(0,2h_y) + f(0,h_y)\big] \\ & + 9\big[f(h_x,h_y) + f(2h_x,h_y) + f(h_x,2h_y) + f(2h_x,2h_y)\big] \Big\} \end{aligned}$$

Die 3/8-Formel integriert alle Polynome P_3 mit $s+t \leq 3$ exakt sowie einzelne (aber nicht alle) Polynome mit $s + t = 4$. Sie besitzt somit den Genauigkeitsgrad $L = 3$.

Die *zusammengesetzten* bzw. *summierten* Newton-Cotes-Formeln ergeben sich wie folgt: Man zerlegt den rechteckigen Integrationsbereich

$$R = \{(x, y)|\ \ a \leq x \leq b,\ \ c \leq y \leq d\}$$

in Teilrechtecke, die jeweils auf das Referenzrechteck der entsprechenden Kubaturformel abgebildet werden. Die Summe der Kubaturformeln über alle Teilrechtecke ergibt dann die summierte Kubaturformel.

Summierte Trapez-Kubaturformel für das Rechteck R

Das Rechteck R wird in Teilrechtecke R_{pq} wie folgt zerlegt: Die Zerlegung in x-Richtung sei mit

$$Z_x : x_p = a + ph_x, \qquad p = 0(1)P, \qquad h_x = \frac{b-a}{P}$$

definiert, die Zerlegung in y-Richtung mit

$$Z_y : y_q = c + qh_y, \qquad q = 0(1)Q, \qquad h_y = \frac{d-c}{Q}.$$

Somit erhält man die $P \cdot Q$ Teilrechtecke

$$R_{pq} := \{(x, y)|\ \ x_p \leq x \leq x_{p+1}, \quad y_q \leq y \leq y_{q+1}\}\ , \quad p = 0(1)P\text{–}1, \quad q = 0(1)Q\text{–}1.$$

Wendet man nun auf jedes Teilrechteck R_{pq} die Trapez-Formel (15.9) an, so ergibt sich die summierte Trapez-Formel wie folgt:

$$\begin{aligned} C^{\mathsf{T}}_{h_x h_y}(f;R) &= \sum_{p=0}^{P-1}\sum_{q=0}^{Q-1} C^{\mathsf{T}}(f;R_{pq}) \\ &= \frac{h_x h_y}{4}\{f(a,c)+f(b,c)+f(a,d)+f(b,d) \\ &+2\sum_{p=1}^{P-1}\big(f(a+ph_x,c)+f(a+ph_x,d)\big) \\ &+2\sum_{q=1}^{Q-1}\big(f(a,c+qh_y)+f(b,c+qh_y)\big) \\ &+4\sum_{p=1}^{P-1}\sum_{q=1}^{Q-1} f(a+ph_x,c+qh_y)\} \end{aligned}$$

Die globale Fehlerordnung beträgt $q_g = 2$. Es gilt die Trapez-Regel

$$\begin{aligned} \iint_R f(x,y)\,\mathrm{d}x\,\mathrm{d}y &= C^{\mathsf{T}}_{h_x h_y}(f;R) + E^{\mathsf{T}}_{h_x h_y}(f;R) \qquad \text{mit} \\ |E^{\mathsf{T}}_{h_x h_y}(f;R)| &\le \frac{(b-a)(d-c)}{12}\left\{h_x^2 \max_{(x,y)\in R}|f_{xx}| + h_y^2 \max_{(x,y)\in R}|f_{yy}|\right\} \\ &= O(h_{\max}^2) \end{aligned}$$

mit $h_{\max} = \max\{h_x, h_y\}$.

Beweis zur Fehlerordnung in [MAES1988], 7.4.3; [HAMM1994], §6 .

Summierte Simpson-Kubaturformel

Hier benötigt man Teilrechtecke mit den Seitenlängen $2h_x$ und $2h_y$. Deshalb wird R wie folgt zerlegt:

$$\begin{aligned} Z_x: &\quad x_p = a + ph_x, \quad p = 0(1)2P, \quad h_x = \frac{b-a}{2P} \\ Z_y: &\quad y_q = c + qh_y, \quad q = 0(1)2Q, \quad h_y = \frac{d-c}{2Q} \end{aligned}$$

Man erhält so $P \cdot Q$ Teilrechtecke

$$R^S_{pq} := \left\{(x,y)|x_{2p} \le x \le x_{2(p+1)}, y_{2q} \le y \le y_{2(q+1)}\right\},\ p = 0(1)P{-}1,\ q = 0(1)Q{-}1,$$

auf die die Simpsonsche Formel (15.10) angewendet wird. Für die summierte Simpsonsche Formel ergibt sich

$$
\begin{aligned}
C^S_{h_x h_y}(f;R) &= \sum_{p=0}^{P-1}\sum_{q=0}^{Q-1} C^S(f;R^S_{pq}) \\
&= \tfrac{h_x h_y}{9}\Big\{ f(a,c)+f(b,c)+f(a,d)+f(b,d) \\
&+4\sum_{p=0}^{P-1}\Big(f\big(a+(2p+1)h_x,c\big)+f\big(a+(2p+1)h_x,d\big)\Big) \\
&+4\sum_{q=0}^{Q-1}\Big(f\big(a,c+(2q+1)h_y\big)+f\big(b,c+(2q+1)h_y\big)\Big) \\
&+2\sum_{p=1}^{P-1}\big(f(a+2ph_x,c)+f(a+2ph_x,d)\big) \\
&+2\sum_{q=1}^{Q-1}\big(f(a,c+2qh_y)+f(b,c+2qh_y)\big) \\
&+4\sum_{p=1}^{P-1}\sum_{q=1}^{Q-1} f(a+2ph_x,c+2qh_y) \\
&+16\sum_{p=0}^{P-1}\sum_{q=0}^{Q-1} f\big(a+(2p+1)h_x,c+(2q+1)h_y\big) \\
&+8\sum_{p=1}^{P-1}\sum_{q=0}^{Q-1} f\big(a+2ph_x,c+(2q+1)h_y\big) \\
&+8\sum_{p=0}^{P-1}\sum_{q=1}^{Q-1} f\big(a+(2p+1)h_x,c+2qh_y\big)\Big\}
\end{aligned}
$$

Die globale Fehlerordnung beträgt $q_g = 4$; es gilt die Simpson-Regel

$$
\begin{aligned}
\iint_R f(x,y)\,\mathrm{d}x\,\mathrm{d}y &= C^S_{h_x h_y}(f;R) + E^S_{h_x h_y}(f;R) \qquad \text{mit} \\
|E^S_{h_x h_y}(f;R)| &\le \frac{(b-a)(d-c)}{180}\left\{h_x^4 \max|f_{xxxx}| + h_y^4 \max|f_{yyyy}|\right\} \\
&= O(h^4_{\max})
\end{aligned}
$$

mit $h_{\max} = \max\{h_x, h_y\}$. Beweis zur Fehlerordnung in [MAES1988], 7.4.3.
Analog kann mit der 3/8-Formel verfahren werden.

Beispiel 15.6. (Anwendung der summierten Simpsonschen Formel)

Gegeben: Die Funktion $f(x,y) = \mathrm{e}^{-\frac{1}{2}(x^2+y^2)}$

Gesucht: Eine Annäherung des Doppelintegrals

$$\int_{-a}^{a}\int_{-a}^{a} \mathrm{e}^{-\frac{1}{2}(x^2+y^2)}\,\mathrm{d}x\,\mathrm{d}y$$

mit der summierten Simpsonschen Kubaturformel für verschiedene quadratische Integrationsbereiche $[-a,a;-a,a] := \{(x,y) \mid -a \le x \le a, -a \le y \le a\}$ sowie wachsende Unterteilungen der Flächen in $P \times P$ Teilflächen.

Das Integral soll für unterschiedliche Integrationsbereiche (verschieden große Quadrate mit $a = 5$, 10 und 100) ausgewertet werden. Dabei wird sich zeigen, dass die Gestalt der Funktion $f(x,y)$ starken Einfluss nimmt auf die Konvergenz der Näherung.

Lösung:

P	$a = 5$	$a = 10$	$a = 100$	N
1	44.444 610 073 629	177.777 777 777 778	17 777.777 777 777 777	9
2	3.839 976 384 413	11.111 442 371 640	1 111.111 111 111 111	25
4	5.932 254 308 547	3.840 000 726 790	277.777 777 777 778	81
8	6.283 149 053 446	5.932 269 489 908	69.444 444 444 444	289
16	6.283 177 924 445	6.283 157 977 454	17.361 111 568 571	1 089
32	6.283 178 089 910	6.283 185 307 180	4.607 307 670 781	4 225
64	6.283 178 102 002	6.283 185 307 180	5.229 367 116 254	16 641
128	6.283 178 102 789	6.283 185 307 180	6.280 604 681 717	66 049
256	6.283 178 102 839	6.283 185 307 179	6.283 185 307 180	263 169
512	6.283 178 102 842	6.283 185 307 179	6.283 185 307 180	1 050 625

Dabei ist N die Anzahl der jeweils benötigten Funktionsauswertungen. Die sehr unterschiedlichen Anfangswerte (vgl. obere Tabellenzeilen) werden klar, wenn man sich die Gestalt der Funktion vor Augen führt. Sie ist im Ursprung konvex und weiter außen konkav. Bei grober Unterteilung der (mit $a = 10$ und $a = 100$) doch sehr großen Integrationsbereiche wird die eigentliche Funktion gar nicht gut approximiert.

Zum Vergleich: $2\,\pi = 6.283\,185\,307\,179\,586\,...$

Anders sieht es aus, wenn man einen Bereich wählt, innerhalb dessen die Funktion (fast) konvex ist. Dann erkennt man bereits nach wenigen Unterteilungs-Verdoppelungen die gute Qualität der Lösung:

P	$a = 1$	N
1	3.019 556 480 010	9
2	2.931 535 548 730	25
4	2.928 556 708 982	81
8	2.928 383 725 411	289
16	2.928 373 104 874	1 089
32	2.928 372 444 012	4 225
64	2.928 372 402 753	16 641
128	2.928 372 400 175	66 049
256	2.928 372 400 014	263 169
512	2.928 372 400 004	1 050 625

□

Bemerkung 15.6.
Wenn eine Kubaturformel mit dem Referenzrechteck $[a,b;\,c,d]$ auf ein Rechteck $[\alpha,\beta;\,\gamma,\delta]$ angewendet werden soll, müssen die Koordinaten der Knoten in Bezug auf dieses Rechteck berechnet werden. Einem Knoten (x,y) in $[a,b;\,c,d]$ entspricht bezüglich $[\alpha,\beta;\,\gamma,\delta]$ der Knoten

$$\left(\frac{\beta-\alpha}{b-a}\,x+\frac{\alpha b-\beta a}{b-a}\,,\,\frac{\delta-\gamma}{d-c}\,y+\frac{\gamma d-\delta c}{d-c}\right).$$

Beispielsweise ist für die Simpsonsche Kubaturformel mit $[a,b;\,c,d]=[-h_x,h_x;\,-h_y,h_y]$

$$(x,y)\mapsto\left(\frac{\beta-\alpha}{2\,h_x}\,x+\frac{\alpha+\beta}{2}\,,\,\frac{\delta-\gamma}{2\,h_y}\,y+\frac{\gamma+\delta}{2}\right).$$

15.4 Das Romberg-Kubaturverfahren für Rechteckbereiche

Das Romberg-Verfahren (nach dem Richardson-Extrapolationsprinzip) ist für Flächenintegrale analog zum eindimensionalen Fall anwendbar. Durch fortgesetzte Intervallhalbierung und Berechnung von Näherungen für das Flächenintegral mit der Trapez-Formel und Extrapolation durch Linearkombinationen von Näherungen zu verschiedenen Schrittweiten erhält man fortlaufend verbesserte Näherungen für das Integral. Das Verfahren ist sehr gut in [ENGE1980] bewiesen.

Die Trapez-Formel für das Referenzrechteck R_r mit

$$R_r=\{(x,y)|\,-1\le x\le 1,-1\le y\le 1\}$$

lautet zur Schrittweite $h_{x_0}=h_{y_0}=2$

$$\begin{aligned}C^{\mathsf{T}}(f;R) &= f(-1,-1)+f(1,-1)+f(1,1)+f(-1,1)\\ &=: L_0^{(0)}(f)\end{aligned}$$

Halbiert man fortlaufend sowohl in x- als auch in y-Richtung die Schrittweite: $h_j=h/2^j$, $j=0,1,2,\ldots$, so ergibt sich für R_r die summierte Trapez-Kubaturformel zur Schrittweite h_j:

$$\begin{aligned}L_j^{(0)}(f) &= \frac{1}{4^j}\Big\{f(-1,-1)+f(-1,1)+f(1,-1)+f(1,1)\\ &\quad+2\sum_{k=1}^{2^j-1}\Big[f\left(-1,-1+\tfrac{k}{2^{j-1}}\right)+f\left(1,-1+\tfrac{k}{2^{j-1}}\right)\\ &\qquad+f\left(-1+\tfrac{k}{2^{j-1}},-1\right)+f\left(-1+\tfrac{k}{2^{j-1}},1\right)\Big]\\ &\quad+4\sum_{k=1}^{2^j-1}\sum_{\ell=1}^{2^j-1}f\left(-1+\tfrac{k}{2^{j-1}},-1+\tfrac{\ell}{2^{j-1}}\right)\Big\}.\end{aligned}$$

Hieraus ergibt sich die folgende Rekursionsformel für $j \geq 1$

$$\begin{aligned} L_j^{(0)}(f) \;=\; & \tfrac{1}{4} L_{j-1}^{(0)} + \frac{1}{2^{2j-1}} \Bigg\{ \sum_{k=0}^{2^{j-1}-1} \Big[f\left(-1, -1 + \frac{2k+1}{2^{j-1}}\right) + f\left(1, -1 + \frac{2k+1}{2^{j-1}}\right) \\ & + f\left(-1 + \frac{2k+1}{2^{j-1}}, -1\right) + f\left(-1 + \frac{2k+1}{2^{j-1}}, 1\right) \Big] \\ & + 2 \sum_{k=0}^{2^{j-1}-1} \sum_{\ell=1}^{2^j-1} f\left(-1 + \frac{2k+1}{2^{j-1}}, -1 + \frac{\ell}{2^{j-1}}\right) \\ & + 2 \sum_{k=0}^{2^{j-1}-2} \sum_{\ell=0}^{2^{j-1}-1} f\left(-1 + \frac{2k+2}{2^{j-1}}, -1 + \frac{2\ell+1}{2^{j-1}}\right) \Bigg\}. \end{aligned}$$

Analog zum Romberg-Quadraturverfahren hat das Romberg-Schema die Form

h_j	$L_j^{(0)}$	$L_j^{(1)}$	$L_j^{(2)}$	$\cdots$	$L_j^{(m-1)}$	$L_j^{(m)}$
h_0	$L_0^{(0)}$					
$h_1 = \frac{h_0}{2}$	$L_1^{(0)}$	$L_0^{(1)}$				
$h_2 = \frac{h_1}{2}$	$L_2^{(0)}$	$L_1^{(1)}$	$L_0^{(2)}$			
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$		
$h_{m-1} = \frac{h_{m-2}}{2}$	$L_{m-1}^{(0)}$	$L_{m-2}^{(1)}$	$L_{m-3}^{(2)}$	$\cdots$	$L_0^{(m-1)}$	
$h_m = \frac{h_{m-1}}{2}$	$L_m^{(0)}$	$L_{m-1}^{(1)}$	$L_{m-2}^{(2)}$	$\cdots$	$L_1^{(m-1)}$	$L_0^{(m)}$

Die extrapolierten Werte $L_j^{(k)}$ lassen sich wie folgt ermitteln:

$$L_j^{(k)} = L_{j+1}^{k-1} + \frac{1}{4^k - 1} \left(L_{j+1}^{k-1} - L_j^{k-1} \right) \quad \text{für} \quad k \geq 1 \quad \text{und} \quad j = 0, 1, 2, \ldots$$

Die Argumente $x' \in [-1, 1]$, $y' \in [-1, 1]$ in den Formeln für die $L_j^{(k)}$ müssen dann auf Wertepaare (x, y) aus dem aktuellen Rechteck $[a, b; c, d]$ transformiert werden mit

$$\begin{aligned} x &= \tfrac{h_x}{2} x' + (j_x - 0.5) h_x + a, \qquad j_x = 1(1) j_{\mathrm{xmax}} \\ y &= \tfrac{h_y}{2} y' + (j_y - 0.5) h_y + c, \qquad j_y = 1(1) j_{\mathrm{ymax}} \end{aligned}$$

$$h_x = \frac{b-a}{j_{\mathrm{xmax}}}, \qquad h_y = \frac{d-c}{j_{\mathrm{ymax}}}$$

Mit $A_x = h_x/2$, $A_y = h_y/2$, $B_x = (j_x - 0.5)h_x + a$, $B_y = (j_y - 0.5)h_y + c$ gelten dann die Transformationsgleichungen

$$\begin{aligned} x &= A_x x' + B_x \\ y &= A_y y' + B_y. \end{aligned}$$

Beispiel 15.7.

Gegeben: Die Funktion

$$f(x,y) = 2\,\mathrm{e}^{-\frac{1}{2}\,(x^2+y^2)}.$$

Gesucht: Mittels Romberg-Verfahren soll ein Näherungswert für das Doppelintegral

$$\int_{-1}^{1}\int_{-1}^{1} f(x,y)\,\mathrm{d}x\,\mathrm{d}y$$

bestimmt werden.

Es soll gezeigt werden, wie das Verfahren zeilenweise eine Matrix vergrößert und iterativ zu immer besseren Näherungswerten für das Integral führt.

Lösung: Zunächst wird für die Ausgangsschrittweite $h_0 = 2$ der Wert $L_0^{(0)} = 2.943036$ ermittelt und in die Tabelle eingetragen:

j	h_j	$L_j^{(0)}$
0	2	2.943036

Man verbessert nun diesen ersten Näherungswert, indem man zwei Schritte ausführt:

1. Für die Schrittweite $h_1 = h_0/2 = 1$ berechnet man den genaueren Näherungswert $L_1^{(0)} = 5.161882$:

j	h_j	$L_j^{(0)}$
0	2	2.943036
1	1	5.161882

2. Nun verfeinert man den Näherungswert, indem man die bisher ermittelten Werte miteinander verknüpft (ohne weitere Funktionsauswertungen):

$$L_0^{(1)} = L_1^{(0)} + \frac{1}{3}\left(L_1^{(0)} - L_0^{(0)}\right).$$

Man erweitert die Tabelle um eine neue Spalte und trägt $L_0^{(1)} = 5.901497$ ein:

j	h_j	$L_j^{(0)}$	$L_{j-1}^{(1)}$
0	2	2.943036	—
1	1	5.161882	5.901497

Man erkennt hier schon das weitere Vorgehen:

Es werden verbesserte Näherungswerte ermittelt, indem man zeilenweise zuerst mit weiteren Funktionsauswertungen einen genaueren Wert berechnet und dann die restlichen Tabellen-Werte durch Linearkombinationen der bereits bekannten Werte ergänzt. Dabei wächst die Tabelle jedes Mal auch um eine neue Spalte.

Hier zur Anschauung noch die nächste Zeile:
Man ermittelt $L_2^{(0)} = 5.683589$, kombiniert

$$L_1^{(1)} = L_2^{(0)} + \frac{1}{3}\left(L_2^{(0)} - L_1^{(0)}\right) \quad \text{zu} \quad L_1^{(1)} = 5.857491$$

und schließlich noch (für die neue Spalte)

$$L_0^{(2)} = L_1^{(1)} + \frac{1}{15}\left(L_1^{(1)} - L_0^{(1)}\right).$$

Also trägt man $L_0^{(2)} = 5.854557$ ein und ist fertig:

j	h_j	$L_j^{(0)}$	$L_{j-1}^{(1)}$	$L_{j-2}^{(2)}$
0	2	2.943036	—	—
1	1	5.161882	5.901497	—
2	1/2	5.683589	5.857491	5.854557

□

15.5 Gauß-Kubaturformeln für Rechteckbereiche

Ganz analog zur Ausführung bei den Newton-Cotes-Formeln lassen sich die Gauß-Formeln über Rechteckbereichen aus den eindimensionalen Gaußschen Quadraturformeln (vgl. Abschnitt 14.7) zusammensetzen.

Aus der eindimensionalen Quadraturformel $Q^{G_2}(f; -h, h)$ ergeben sich durch Multiplikation der Gewichtsmatrizen mit je zwei Elementen

$$(a_{ik}) = \boldsymbol{A}\boldsymbol{B}^{\mathsf{T}} = h_x \begin{pmatrix} 1 \\ 1 \end{pmatrix} h_y(1,1) = h_x h_y \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

die vier Gewichte a_{ik}, $i, k = 1, 2$, der entsprechenden Kubaturformel und aus den je zwei Knoten x_i, y_k

$$\text{in } x\text{-Richtung:} \quad x_0 = -\frac{h_x}{\sqrt{3}}, \quad x_1 = \frac{h_x}{\sqrt{3}}$$

$$\text{in } y\text{-Richtung:} \quad y_0 = -\frac{h_y}{\sqrt{3}}, \quad y_1 = \frac{h_y}{\sqrt{3}}$$

die vier Knoten der Kubaturformel. Man erhält die Gauß-Formel für $n = 1$, $m = 1$, d. h. $N = 4$

$$C^{G_2}(f; R_{r_2}) = h_x h_y \left\{ f\left(-\frac{h_x}{\sqrt{3}}, -\frac{h_y}{\sqrt{3}}\right) + f\left(-\frac{h_x}{\sqrt{3}}, \frac{h_y}{\sqrt{3}}\right) + f\left(\frac{h_x}{\sqrt{3}}, -\frac{h_y}{\sqrt{3}}\right) + f\left(\frac{h_x}{\sqrt{3}}, \frac{h_y}{\sqrt{3}}\right) \right\}$$

Sie besitzt den Genauigkeitsgrad $L = 3$ und integriert damit alle bikubischen Polynome vom Grad $s + t \leq 3$ exakt und hat die globale Fehlerordnung $O(h_{\max}^4)$ mit $h_{\max} = \max\{h_x, h_y\}$.

Aus der eindimensionalen Quadraturformel $Q^{G_3}(f;-h,h)$ ergeben sich die neun Gewichte der entsprechenden Kubaturformel wie folgt

$$(a_{ik}) = \boldsymbol{A}\boldsymbol{B}^{\mathsf{T}} = \frac{h_x}{9}\begin{pmatrix}5\\8\\5\end{pmatrix}\frac{h_y}{9}(5,\ 8,\ 5) = \frac{h_x h_y}{81}\begin{pmatrix}25 & 40 & 25\\ 40 & 64 & 40\\ 25 & 40 & 25\end{pmatrix}.$$

Die neun Knoten der Kubaturformel ergeben sich aus den je drei Knoten x_i bzw. y_k, $i,k=0,1,2$ in

$$\begin{aligned} &x\text{-Richtung:} \quad x_0 = -\sqrt{0.6}h_x, \quad x_1 = 0, \quad x_2 = \sqrt{0.6}h_x\\ &y\text{-Richtung:} \quad y_0 = -\sqrt{0.6}h_y, \quad y_1 = 0, \quad y_2 = \sqrt{0.6}h_y \end{aligned}$$

Man erhält die Gauß-Kubaturformel (15.5) für $n=2$, $m=2$, d. h. $N=9$

$$\begin{aligned} C^{G_3}(f;R_{r_2}) \quad &= \tfrac{h_x h_y}{81}\{25\left[f(-\sqrt{0.6}h_x,-\sqrt{0.6}h_y)\right.\\ &+f(-\sqrt{0.6}h_x,\sqrt{0.6}h_y)+f(\sqrt{0.6}h_x,-\sqrt{0.6}h_y)\\ &\left.+\ f(\sqrt{0.6}h_x,\sqrt{0.6}h_y)\right]+40\left[f(0,-\sqrt{0.6}h_y)+f(-\sqrt{0.6}h_x,0)\right.\\ &\left.+\ f(\sqrt{0.6}h_x,0)+f(0,\sqrt{0.6}h_y)\right]+\ 64f(0,0)\}. \end{aligned}$$

Sie besitzt den Genauigkeitsgrad $L=5$, integriert damit alle Polynome vom Grad $s+t\le 5$ exakt und besitzt die globale Fehlerordnung $O(h_{\max}^6)$ mit $h_{\max}=\max\{h_x,h_y\}$.

Weitere Formeln lassen sich analog aus den entsprechenden eindimensionalen Formeln herleiten, die in Abschnitt 14.7 „Tabelle der Gaußschen Knoten und Gewichte“ angegeben sind.
Die summierten Formeln ergeben sich in gleicher Weise wie in Abschnitt 15.3 beschrieben: Zur Berechnung eines Integrals über einem Rechteck $R:\{(x,y)|a\le x\le b, c\le y\le d\}$ wird R in $P\cdot Q$ Teilrechtecke R_{pq} der Größen $2h_x\cdot 2h_y$ zerlegt. Auf jedes Teilrechteck wird eine Gaußsche Formel angewandt. Dabei sind die Knoten des Referenzrechteckes wie folgt zu transformieren

$$\begin{aligned} x_i \quad &\mapsto \quad a+(2p+1)h_x+x_i, \quad p=0(1)P\text{–}1, \quad i=0(1)m\\ y_k \quad &\mapsto \quad c+(2q+1)h_y+y_k, \quad q=0(1)Q\text{–}1, \quad k=0(1)n \end{aligned}$$

Für $n=m=1$ erhält man so z. B. die summierte Formel

$$\begin{aligned} C^{G_2}(f;R) \quad &= h_x h_y \sum_{p=0}^{P-1}\sum_{q=0}^{Q-1}\Big[f\left(a+(2p+1)h_x-\tfrac{h_x}{\sqrt{3}},c+(2q+1)h_y-\tfrac{h_y}{\sqrt{3}}\right)\\ &+f\left(a+(2p+1)h_x-\tfrac{h_x}{\sqrt{3}},c+(2q+1)h_y+\tfrac{h_y}{\sqrt{3}}\right)\\ &+f\left(a+(2p+1)h_x+\tfrac{h_x}{\sqrt{3}},c+(2q+1)h_y-\tfrac{h_y}{\sqrt{3}}\right)\\ &+\ f\left(a+(2p+1)h_x+\tfrac{h_x}{\sqrt{3}},c+(2q+1)h_y+\tfrac{h_y}{\sqrt{3}}\right)\Big] \end{aligned}$$

mit der globalen Fehlerordnung $q_g=4$.

Die 17 Knoten und Gewichte (d. h. 51 Parameter x_j, y_j, A_j) der Gauß-Formel für $L = 9$ ergeben sich aus den $\frac{1}{2}(L+1)(L+2) = 55$ nichtlinearen Gleichungen (15.8) und sind aus [ENGE1980], S.257 entnommen:

I	$X(I)$	$Y(I)$	$A(I)$
1	$.96884996636198E+00$	$.63068011973167E+00$	$.88879378170200E-01$
2	$-.96884996636198E+00$	$-.63068011973167E+00$	$.88879378170200E-01$
3	$-.63068011973167E+00$	$.96884996636198E+00$	$.88879378170200E-01$
4	$.63068011973167E+00$	$-.96884996636198E+00$	$.88879378170200E-01$
5	$.75027709997890E+00$	$.92796164595957E+00$	$.11209960212960E+00$
6	$-.75027709997890E+00$	$-.92796164595957E+00$	$.11209960212960E+00$
7	$-.92796164595957E+00$	$.75027709997890E+00$	$.11209960212960E+00$
8	$.92796164595957E+00$	$-.75027709997890E+00$	$.11209960212960E+00$
9	$.52373582021443E+00$	$.45333982113565E+00$	$.39828243926207E+00$
10	$-.52373582021443E+00$	$-.45333982113565E+00$	$.39828243926207E+00$
11	$-.45333982113565E+00$	$.52373582021443E+00$	$.39828243926207E+00$
12	$.45333982113565E+00$	$-.52373582021443E+00$	$.39828243926207E+00$
13	$.76208328192620E-01$	$.85261572933366E+00$	$.26905133763978E+00$
14	$-.76208328192620E-01$	$-.85261572933366E+00$	$.26905133763978E+00$
15	$-.85261572933366E+00$	$.76208328192620E-01$	$.26905133763978E+00$
16	$.85261572933366E+00$	$-.76208328192620E-01$	$.26905133763978E+00$
17	$0.00000000000000E+00$	$0.00000000000000E+00$	$.52674897119342E+00$

Die Lage dieser 17 Knoten graphisch dargestellt:

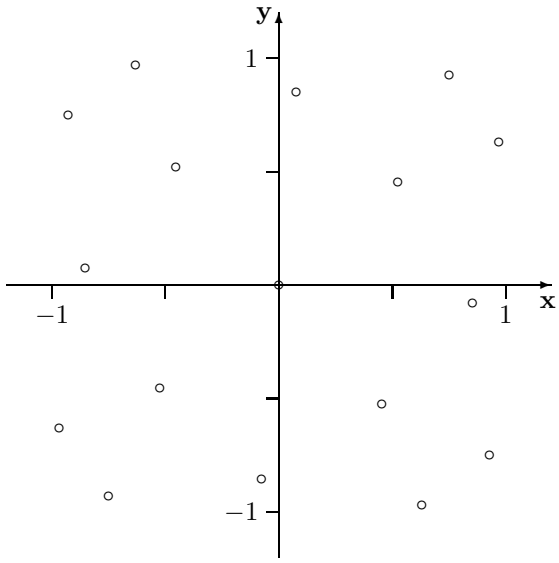


Abb. 15.4.

15.6 Berechnung des Riemannschen Flächenintegrals mit bikubischen Splines

Es sei das Flächenintegral

$$I(f;R) = \iint\limits_R f(x,y)\,\mathrm{d}x\,\mathrm{d}y$$

über dem Rechteck $R = \{(x,y)|a \le x \le b,\ c \le y \le d\}$ zu berechnen. Dies kann mit Hilfe bikubischer Splines

$$S = S(x,y) \equiv S_{ij}(x,y) \quad \text{für} \quad (x,y) \in R_{ij}$$

gemäß Darstellung in Abschnitt 12.1, Algorithmen 1, 3, 4, geschehen, die für das Rechteck R berechnet wurden.

Für jedes Teilrechteck R_{ij} gilt dann

$$\begin{aligned}
&\int_{x_i}^{x_{i+1}}\int_{y_j}^{y_{j+1}} f(x,y)\,\mathrm{d}y\,\mathrm{d}x \approx \int_{x_i}^{x_{i+1}}\int_{y_j}^{y_{j+1}} S_{ij}(x,y)\,\mathrm{d}y\,\mathrm{d}x\\
&= \int_{x_i}^{x_{i+1}}\int_{y_j}^{y_{j+1}} \sum_{k=0}^{3}\sum_{m=0}^{3} a_{ijkm}(x-x_i)^k(y-y_j)^m\,\mathrm{d}y\,\mathrm{d}x\\
&= \sum_{k=0}^{3}\sum_{m=0}^{3} a_{ijkm}\int_{x_i}^{x_{i+1}}(x-x_i)^k\mathrm{d}x\int_{y_j}^{y_{j+1}}(y-y_j)^m\mathrm{d}y\\
&= \sum_{k=0}^{3}\sum_{m=0}^{3} a_{ijkm}\frac{(x_{i+1}-x_i)^{k+1}}{k+1}\,\frac{(y_{j+1}-y_j)^{m+1}}{m+1}.
\end{aligned}$$

Der gesuchte Integralwert I ergibt sich näherungsweise aus der Summation der Integrale über alle Teilrechtecke.

15.7 Vergleich der Verfahren anhand von Beispielen

Beispiel 15.8.

Gegeben: Die Funktion $f(x) = \mathrm{e}^{\sin x\,\cos x}$

(Zur einfacheren Nachvollziehbarkeit wurde hier eine Funktion gewählt, die nur von x abhängt.)

Gesucht: Die Näherungen für das Integral

$$\int_{-1/2}^{1/2}\int_{-1/2}^{1/2} f(x)\,\mathrm{d}x\,\mathrm{d}y = 1.034\,408\,860\,73\ldots$$

unter verschiedenen Genauigkeitsanforderungen

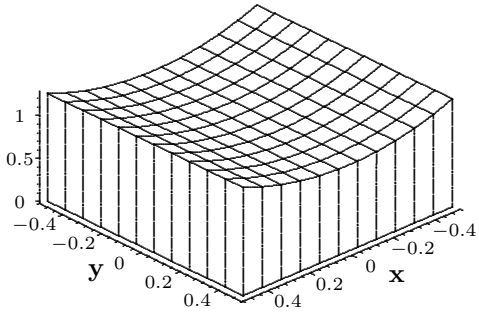


Abb. 15.5. Graph der Funktion $f(x) = \mathrm{e}^{\sin x \, \cos x}$

Lösung: Erläuterung zu den folgenden Tabellen:

Zur Spalte Verfahren:

NC-ST	Newton-Cotes-Sehnentrapez-Formel
NC-S	Newton-Cotes-Simpson-Formel
NC-3	Newton-Cotes-3/8-Formel
NC-4	Newton-Cotes-4/90-Formel
NC-5	Newton-Cotes-5/288-Formel
NC-6	Newton-Cotes-6/840-Formel
NC-7	Newton-Cotes-7/17280-Formel
Gau-n	Summierte Gauß-Verfahren nach Newton-Cotes-Formeln

Zu den Spalten geschätzter relativer Fehler:

Hier werden die letzten beiden Schätzungen angegeben. Dadurch wird klar, warum in Spalte F nicht unbedingt eine strenge Abnahme der Werte entsteht: Wenn die vorletzte Schätzung noch knapp oberhalb der geforderten Genauigkeit liegt, muss die Anzahl der Referenz-Rechtecke notwendigerweise erneut vervierfacht werden, wodurch sich die Anzahl der Funktionsauswertungen natürlich stark erhöht.

Zur Spalte N:

Hier wird die Anzahl der verwendeten Referenz-Rechtecke angegeben.

Zur Spalte F:

Hier steht die Anzahl der benötigten Funktionsauswertungen.

1. Geforderte Genauigkeit = $5.0{\cdot}10^{-09}$

Verfahren	ermittelte Näherung	geschätzter rel. Fehler vorletzter	geschätzter rel. Fehler letzter	N	F
NC-ST	2.034 408 863 05	$-9.3{\cdot}10^{-09}$	$-2.3{\cdot}10^{-09}$	1024^2	22 386 003
NC-S	1.034 408 858 18	$4.1{\cdot}10^{-08}$	$2.5{\cdot}10^{-09}$	8^2	5 684
NC-3	1.034 408 859 60	$1.8{\cdot}10^{-08}$	$1.1{\cdot}10^{-09}$	8^2	12 604
NC-4	1.034 408 861 00	$-2.1{\cdot}10^{-08}$	$-2.8{\cdot}10^{-10}$	2^2	1 378
NC-5	1.034 408 860 88	$-1.2{\cdot}10^{-08}$	$-1.6{\cdot}10^{-10}$	2^2	2 122
NC-6	1.034 408 860 70		$3.9{\cdot}10^{-11}$	1^2	625
NC-7	1.034 408 860 71		$2.4{\cdot}10^{-11}$	1^2	841
Gau-0	1.034 408 856 08	$1.9{\cdot}10^{-08}$	$4.7{\cdot}10^{-09}$	1024^2	6 990 505
Gau-1	1.034 408 860 83	$-8.5{\cdot}10^{-09}$	$-5.3{\cdot}10^{-10}$	32^2	27 300
Gau-2	1.034 408 860 72	$5.6{\cdot}10^{-09}$	$8.5{\cdot}10^{-11}$	8^2	3 825
Gau-3	1.034 408 860 75	$-3.5{\cdot}10^{-08}$	$-3.0{\cdot}10^{-09}$	2^2	400
Gau-4	1.034 408 860 73	$-2.3{\cdot}10^{-08}$	$1.8{\cdot}10^{-11}$	2^2	625
Gau-5	1.034 408 860 73		$8.1{\cdot}10^{-10}$	1^2	180
Gau-6	1.034 408 860 73		$-1.4{\cdot}10^{-11}$	1^2	245
Gau-7	1.034 408 860 73		$1.8{\cdot}10^{-14}$	1^2	320

2. Geforderte Genauigkeit = $5.0{\cdot}10^{-10}$

Verfahren	ermittelte Näherung	geschätzter rel. Fehler vorletzter	geschätzter rel. Fehler letzter	N	F
NC-S	1.034 408 860 57	$2.5{\cdot}10^{-09}$	$1.6{\cdot}10^{-10}$	16^2	22 325
NC-3	1.034 408 860 66	$1.1{\cdot}10^{-09}$	$7.1{\cdot}10^{-11}$	16^2	49 853
NC-4	1.034 408 861 00	$-2.1{\cdot}10^{-08}$	$-2.8{\cdot}10^{-10}$	2^2	1 378
NC-5	1.034 408 860 88	$-1.2{\cdot}10^{-08}$	$-1.6{\cdot}10^{-10}$	2^2	2 122
NC-6	1.034 408 860 70		$3.9{\cdot}10^{-11}$	1^2	625
NC-7	1.034 408 860 71		$2.4{\cdot}10^{-11}$	1^2	841
Gau-1	1.034 408 860 73	$-5.3{\cdot}10^{-10}$	$-3.3{\cdot}10^{-11}$	64^2	109 220
Gau-2	1.034 408 860 72	$5.6{\cdot}10^{-09}$	$8.5{\cdot}10^{-11}$	8^2	3 825
Gau-3	1.034 408 860 73	$-3.0{\cdot}10^{-09}$	$-9.0{\cdot}10^{-12}$	4^2	1 680
Gau-4	1.034 408 860 73	$-2.3{\cdot}10^{-08}$	$1.8{\cdot}10^{-11}$	2^2	625
Gau-5	1.034 408 860 73	$8.1{\cdot}10^{-10}$	$-6.6{\cdot}10^{-14}$	2^2	900
Gau-6	1.034 408 860 73		$-1.4{\cdot}10^{-11}$	1^2	245
Gau-7	1.034 408 860 73		$1.8{\cdot}10^{-14}$	1^2	320

3. Geforderte Genauigkeit = $5.0{\cdot}10^{-11}$

Verfahren	ermittelte Näherung	geschätzter rel. Fehler vorletzter	geschätzter rel. Fehler letzter	N	F
NC-S	1.034 408 860 72	$1.6{\cdot}10^{-10}$	$9.9{\cdot}10^{-12}$	32^2	88 374
NC-3	1.034 408 860 72	$7.1{\cdot}10^{-11}$	$4.4{\cdot}10^{-12}$	32^2	198 078
NC-4	1.034 408 860 73	$-2.8{\cdot}10^{-10}$	$-4.2{\cdot}10^{-12}$	4^2	5 603
NC-5	1.034 408 860 73	$-1.6{\cdot}10^{-10}$	$-2.4{\cdot}10^{-12}$	4^2	8 683
NC-6	1.034 408 860 70		$3.9{\cdot}10^{-11}$	1^2	625
NC-7	1.034 408 860 71		$2.4{\cdot}10^{-11}$	1^2	841
Gau-1	1.034 408 860 73	$-5.3{\cdot}10^{-10}$	$-3.3{\cdot}10^{-11}$	64^2	109 220
Gau-2	1.034 408 860 73	$8.5{\cdot}10^{-11}$	$1.3{\cdot}10^{-12}$	16^2	15 345
Gau-3	1.034 408 860 73	$-3.0{\cdot}10^{-09}$	$-9.0{\cdot}10^{-12}$	4^2	1 680
Gau-4	1.034 408 860 73	$-2.3{\cdot}10^{-08}$	$1.8{\cdot}10^{-11}$	2^2	625
Gau-5	1.034 408 860 73	$8.1{\cdot}10^{-10}$	$-6.6{\cdot}10^{-14}$	2^2	900
Gau-6	1.034 408 860 73		$-1.4{\cdot}10^{-11}$	1^2	245
Gau-7	1.034 408 860 73		$1.8{\cdot}10^{-14}$	1^2	320

4. Geforderte Genauigkeit = $5.0{\cdot}10^{-12}$

Verfahren	ermittelte Näherung	geschätzter rel. Fehler vorletzter	geschätzter rel. Fehler letzter	N	F
NC-S	1.034 408 860 73	$9.9{\cdot}10^{-12}$	$6.2{\cdot}10^{-13}$	64^2	351 543
NC-3	1.034 408 860 72	$7.1{\cdot}10^{-11}$	$4.4{\cdot}10^{-12}$	32^2	198 078
NC-4	1.034 408 860 73	$-2.8{\cdot}10^{-10}$	$-4.2{\cdot}10^{-12}$	4^2	5 603
NC-5	1.034 408 860 73	$-1.6{\cdot}10^{-10}$	$-2.4{\cdot}10^{-12}$	4^2	8 683
NC-6	1.034 408 860 73	$3.9{\cdot}10^{-11}$	$1.2{\cdot}10^{-13}$	2^2	3 026
NC-7	1.034 408 860 73	$2.4{\cdot}10^{-11}$	$7.4{\cdot}10^{-14}$	2^2	4 090
Gau-1	1.034 408 860 73	$-3.3{\cdot}10^{-11}$	$-2.0{\cdot}10^{-12}$	128^2	436 900
Gau-2	1.034 408 860 73	$8.5{\cdot}10^{-11}$	$1.3{\cdot}10^{-12}$	16^2	15 345
Gau-3	1.034 408 860 73	$-9.0{\cdot}10^{-12}$	$-3.4{\cdot}10^{-14}$	8^2	6 800
Gau-4	1.034 408 860 73	$1.8{\cdot}10^{-11}$	$1.4{\cdot}10^{-14}$	4^2	2 625
Gau-5	1.034 408 860 73	$8.1{\cdot}10^{-10}$	$-6.6{\cdot}10^{-14}$	2^2	900
Gau-6	1.034 408 860 73	$-1.4{\cdot}10^{-11}$	$7.4{\cdot}10^{-17}$	2^2	1 225
Gau-7	1.034 408 860 73		$1.8{\cdot}10^{-14}$	1^2	320

□

Beispiel 15.9.

Gegeben: Das Polynom $P(x,y) = -3\,x^{15}\,y^{17} - 2xy^2 + x^2y + 3y^2 - xy + 2y$

Gesucht: Die Näherungen für das Integral

$$\int_0^1 \int_0^1 P(x,y)\,\mathrm{d}x\,\mathrm{d}y = 1.57291\overline{6}$$

mit der geforderten Genauigkeit $2.2{\cdot}10^{-10}$

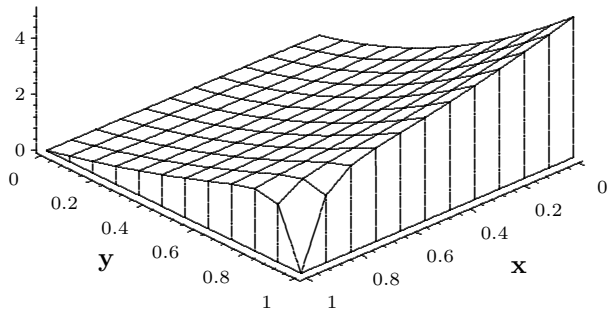


Abb. 15.6. Graph der Funktion $P(x,y)$

Lösung: (Erläuterungen zur Tabelle vgl. Beispiel 15.8)

Verfahren	ermittelte Näherung	geschätzter rel. Fehler vorletzter	geschätzter rel. Fehler letzter	N	F
NC-ST	1.572 916 666 45	$8.5{\cdot}10^{-10}$	$2.1{\cdot}10^{-10}$	4 096	357 979 477
NC-S	1.572 916 666 57	$1.6{\cdot}10^{-09}$	$9.9{\cdot}10^{-11}$	64	351 543
NC-3	1.572 916 666 62	$7.0{\cdot}10^{-10}$	$4.4{\cdot}10^{-11}$	64	789 439
NC-4	1.572 916 666 57	$5.8{\cdot}10^{-09}$	$9.4{\cdot}10^{-11}$	8	22 244
NC-5	1.572 916 666 61	$3.3{\cdot}10^{-09}$	$5.3{\cdot}10^{-11}$	8	34 604
NC-6	1.572 916 666 66	$7.5{\cdot}10^{-10}$	$3.4{\cdot}10^{-12}$	4	12 435
NC-7	1.572 916 666 66	$4.6{\cdot}10^{-10}$	$2.1{\cdot}10^{-12}$	4	16 859
Gau-0	1.572 916 666 78	$-4.3{\cdot}10^{-10}$	$-1.1{\cdot}10^{-10}$	8 192	447 392 425
Gau-1	1.572 916 666 67	$-3.3{\cdot}10^{-10}$	$-2.1{\cdot}10^{-11}$	256	1 747 620
Gau-2	1.572 916 666 67	$-1.9{\cdot}10^{-09}$	$-3.0{\cdot}10^{-11}$	32	61 425
Gau-3	1.572 916 666 67	$-2.5{\cdot}10^{-10}$	$-1.0{\cdot}10^{-12}$	16	27 280
Gau-4	1.572 916 666 67	$-2.2{\cdot}10^{-10}$	$-2.3{\cdot}10^{-13}$	8	10 625
Gau-5	1.572 916 666 67	$-1.2{\cdot}10^{-09}$	$-3.5{\cdot}10^{-13}$	4	3 780
Gau-6	1.572 916 666 67	$-3.5{\cdot}10^{-08}$	$-3.4{\cdot}10^{-12}$	2	1 225
Gau-7	1.572 916 666 67		$-1.9{\cdot}10^{-10}$	1	320

□

15.8 Kubaturformeln für Dreieckbereiche

Analog zu der Behandlung der Integration über Rechtecke lässt sich die Integration über ein Referenzdreieck durchführen.

15.8.1 Kubaturformeln für Dreieckbereiche mit achsenparallelen Katheten

Es sei $R\Delta$

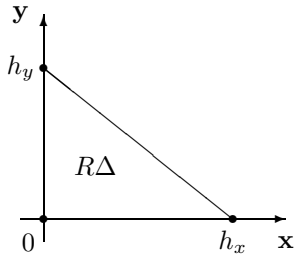


Abb. 15.7.

ein Referenz-Dreieck mit den drei Eckpunkten

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} h_x \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ h_y \end{pmatrix}.$$

Im Folgenden werden einige Kubaturformeln angegeben, die alle das Integral

$$\iint\limits_{R\Delta} f(x,y)\,\mathrm{d}x\,\mathrm{d}y$$

annähern. Je nach Wahl der Knoten (in den Ecken, auf den Kanten oder im Inneren des Referenzdreiecks) erhält man folgende Newton-Cotes- oder Gauß-Kubaturformeln (z. T. sind sie identisch):

15.8.1.1 Newton-Cotes-Kubaturformeln für Dreieckbereiche mit achsenparallelen Katheten

Es sei $R\Delta_1$

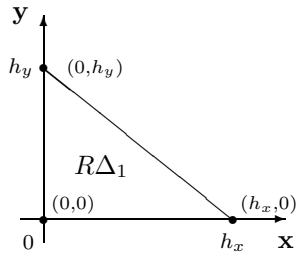


Abb. 15.8.

ein Referenzdreieck mit den drei Eckpunkten als Knoten. Dann lautet die Kubaturformel

$$C(f; R\Delta_1) = \frac{h_x h_y}{2} \cdot \frac{1}{3}\big\{f(0,0) + f(h_x,0) + f(0,h_y)\big\}, \tag{15.11}$$

sie besitzt den Genauigkeitsgrad $L = 1$ und integriert damit alle Polynome P_1 vom Grad $s + t \leq 1$ (d. h. mit den Monomen 1, x, y) exakt und hat die globale Fehlerordnung $O(h_{\max}^2)$.

Kubaturformeln vom Genauigkeitsgrad $L = 2$ ergeben sich unter Verwendung folgender Knoten des Referenzdreiecks:

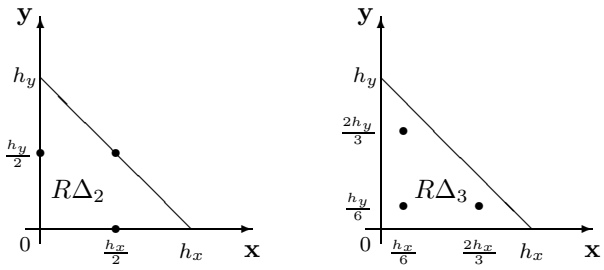


Abb. 15.9.

Die zugehörigen Formeln lauten:

$$C(f; R\Delta_2) = \frac{h_x h_y}{2}\frac{1}{3}\left[f\left(\frac{h_x}{2}, 0\right) + f\left(\frac{h_x}{2}, \frac{h_y}{2}\right), f\left(0, \frac{h_y}{2}\right)\right] \tag{15.12}$$

$$C(f; R\Delta_3) = \frac{h_x h_y}{2}\frac{1}{3}\left[f\left(\frac{h_x}{6}, \frac{h_y}{6}\right) + f\left(\frac{2h_x}{3}, \frac{h_y}{6}\right), f\left(\frac{h_x}{6}, \frac{2h_y}{3}\right)\right] \tag{15.13}$$

Es ist bemerkenswert, dass diese Formeln bei gleicher Knotenzahl wie (15.11) einen höheren Genauigkeitsgrad besitzen; sie integrieren alle Polynome P_2 vom Grad ≤ 2 exakt und besitzen die globale Fehlerordnung $O(h_{\max}^3)$.
Die entsprechenden summierten Formeln ergeben sich, wenn man einen Bereich B in Dreiecke B_j zerlegt und auf jedes Dreieck eine der angegebenen Formeln anwendet und aufsummiert.

Beispiel 15.10.

Gegeben: Das Polynom

$$f(x,y) = 4\,x^2 + 6\,x\,y - y$$

Gesucht: Das Integral

$$I = \int_{x=1}^{5} \int_{y=1}^{\frac{7}{2} - \frac{x}{2}} f(x,y)\,\mathrm{d}y\,\mathrm{d}x$$

Es handelt sich dabei um ein Doppel-Integral, und die Fläche, über die integriert wird, ist ein Dreieck, das begrenzt wird durch die Geraden

$$x = 1\,, \quad y = 1\,, \quad y = \frac{7}{2} - \frac{x}{2}\,.$$

Graphisch also:

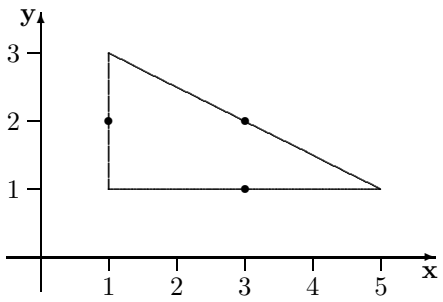


Abb. 15.10.

Lösung: Zur Berechnung von I wird hier die Formel für das Referenz-Dreieck $R\Delta_2$ verwendet, wobei natürlich noch die Koordinaten der auszuwertenden Punkte verschoben werden müssen. Mit den Größen

$$h_x = 5 - 1 = 4 \quad \text{und} \quad h_y = 3 - 1 = 2$$

ergibt die Formel den Wert

$$\begin{aligned} C(f; R\Delta_2) &= \frac{4 \cdot 2}{2}\,\frac{1}{3}\,\big(f(3,1) + f(3,2) + f(1,2)\big) \\ &= \frac{4}{3}\,(53 + 70 + 14) = \frac{548}{3} = 182.\overline{6}. \end{aligned}$$

Durch exakte Berechnung des Doppelintegrals kann man sich davon überzeugen, dass der Wert exakt ist, denn das Polynom enthält keine x- und y-Potenzen, die in der Summe größer als zwei sind.

Zur Anschauung die Fläche mit den Auswertungspunkten:

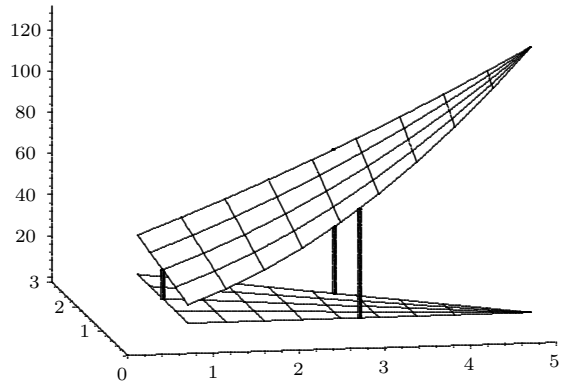


Abb. 15.11.

□

15.8.1.2 Gauß-Kubaturformeln für Dreieckbereiche mit achsenparallelen Katheten

Je nach gefordertem Genauigkeitsgrad L ergeben sich aus einem zu (15.8) äquivalenten nichtlinearen System für dreieckige Integrationsgebiete $\frac{1}{2}(L+1)(L+2)$ Gleichungen für entsprechend viele Parameter x_j, y_j, A_j.

Hier seien einige Beispiele aufgeführt:

Für $L = 1$ ergeben sich folgende Formeln von der globalen Fehlerordnung $O(h_{\max}^2)$:

i Ein-Punkt-Kubaturformel für das Referenzdreieck $R\Delta 4$, die nur einen Knoten verwendet:

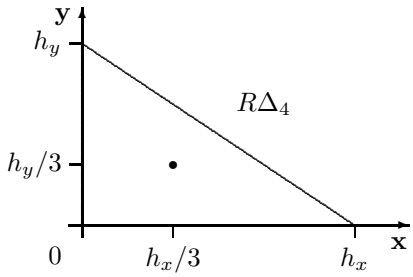


Abb. 15.12.

$$C^{G_1}(f; R\Delta_4) = \frac{h_x h_y}{2} f\left(\frac{h_x}{3}, \frac{h_y}{3}\right)$$

ii Zwei-Punkt-Kubaturformel für RΔ_5, die zwei Knoten verwendet:

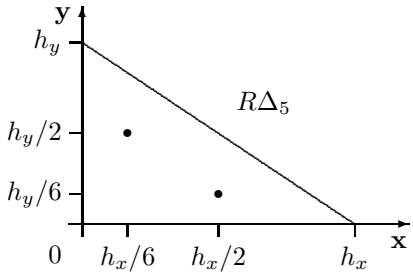


Abb. 15.13.

$$C^{G_2}(f; R\Delta_5) = \frac{h_x h_y}{4}\left\{f\left(\frac{h_x}{6}, \frac{h_y}{2}\right) + f\left(\frac{h_x}{2}, \frac{h_y}{6}\right)\right\}$$

Für $L = 2$ gelten die beiden Drei-Punkt-Formeln aus Abschnitt 15.8.1.1 für $R\Delta_2$ und $R\Delta_3$.

Für $L = 5$ ist die folgende Sieben-Punkt-Formel aus [ENGE1980] entnommen worden:

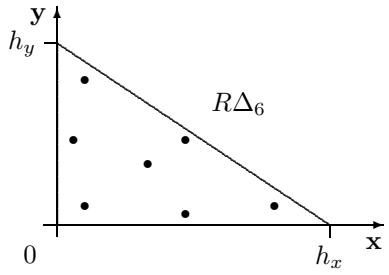


Abb. 15.14.

$$
\begin{aligned}
C^{G_7}(f; R\Delta_6) \;=\; & \tfrac{h_x h_y}{2400}\Big\{270 f\left(\tfrac{h_x}{3}, \tfrac{h_y}{3}\right) + (155+\sqrt{15})\Big[f\left(\tfrac{h_x}{21}(6+\sqrt{15}), \tfrac{h_y}{21}(6+\sqrt{15})\right) \\
& + f\left(\tfrac{h_x}{21}(9-\sqrt{15}), \tfrac{h_y}{21}(6+\sqrt{15})\right) + f\left(\tfrac{h_x}{21}(6+\sqrt{15}), \tfrac{h_y}{21}(9-\sqrt{15})\right)\Big] \\
& + (155-\sqrt{15})\Big[f\left(\tfrac{h_x}{21}(6-\sqrt{15}), \tfrac{h_y}{21}(6-\sqrt{15})\right) \\
& + f\left(\tfrac{h_x}{21}(9+2\sqrt{15}), \tfrac{h_y}{21}(6-\sqrt{15})\right) + f\left(\tfrac{h_x}{21}(6-\sqrt{15}), \tfrac{h_y}{21}(9+2\sqrt{15})\right)\Big]\Big\}
\end{aligned}
$$

Die globale Fehlerordnung ist $O(h_{\max}^6)$.

Beispiel 15.11.

Gegeben: Die Funktion

$$f(x,y) = \frac{x}{y} + 2$$

Gesucht: Das Integral

$$I = \int_{x=1}^{5} \int_{y=1}^{\frac{7}{2} - \frac{x}{2}} f(x,y)\,\mathrm{d}y\,\mathrm{d}x$$

Es handelt sich dabei um ein Doppel-Integral, und die Fläche, über die integriert wird, ist ein Dreieck, das begrenzt wird durch die Geraden

$$x = 1\,, \quad y = 1\,, \quad y = \frac{7}{2} - \frac{x}{2}\,.$$

Es werden dabei die Ein-Punkt- und die Zwei-Punkt-Kubaturformel angewendet und die Ergebnisse verglichen.

Lösung: Für beide Formeln werden die Schrittweiten

$$\begin{aligned} h_x &= 5-1 = 4\,, \\ h_y &= 3-1 = 2 \end{aligned}$$

verwendet. Bei der Auswertung der Funktion muss darauf geachtet werden, dass das Referenz-Dreieck vom Ursprung in die Ecke (1,1) verschoben wurde. Also müssen auch die Funktionsargumente angepasst werden.

Zur Ein-Punkt-Formel:

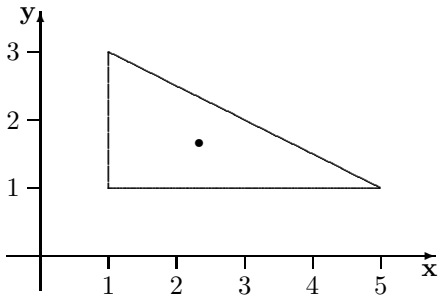


Abb. 15.15.

$$\begin{aligned} C^{G_1}(f;R\Delta) &= \frac{h_x\,h_y}{2}\,f\left(1+\frac{h_x}{3},\,1+\frac{h_y}{3}\right) \\ &= \frac{4\cdot 2}{2}\,f\left(1+\frac{4}{3},\,1+\frac{2}{3}\right) \\ &= 4\,f\left(\frac{7}{3},\,\frac{5}{3}\right) = 4\left(\frac{7/3}{5/3}+2\right) \\ &= 4\,\frac{17}{5} = \frac{68}{5} = 13.6 \end{aligned}$$

Zur Anschauung:

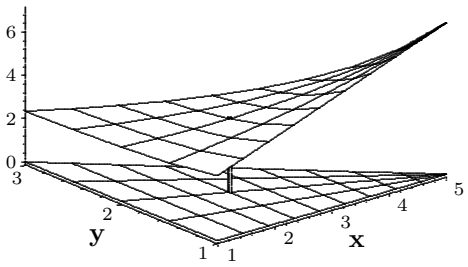


Abb. 15.16.

Zur Zwei-Punkt-Formel:

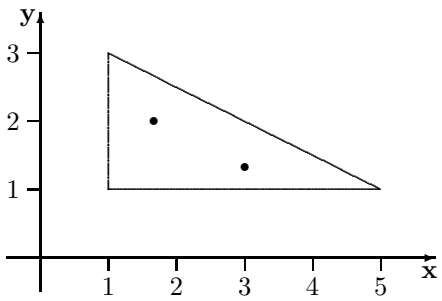


Abb. 15.17.

$$
\begin{aligned}
C^{G_2}(f;R\Delta) &= \frac{h_x\,h_y}{4}\left\{f\left(1+\frac{h_x}{6},1+\frac{h_y}{2}\right)+f\left(1+\frac{h_x}{2},1+\frac{h_y}{6}\right)\right\}\\
&= \frac{4\cdot 2}{4}\left\{f\left(1+\frac{4}{6},1+\frac{2}{2}\right)+f\left(1+\frac{4}{2},1+\frac{2}{6}\right)\right\}\\
&= 2\left\{f\left(\frac{5}{3},2\right)+f\left(3,\frac{4}{3}\right)\right\}=2\left\{\frac{5/3}{2}+2+\frac{3}{4/3}+2\right\}\\
&= 2\left(\frac{17}{6}+\frac{17}{4}\right)=2\,\frac{51+34}{12}=\frac{85}{6}=14.1\overline{6}
\end{aligned}
$$

Zur Anschauung:

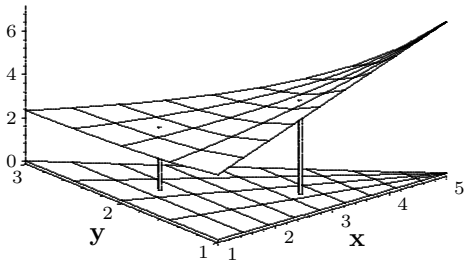


Abb. 15.18.

Die Ergebnisse sind nicht gleich, aber in der Größenordnung stimmen sie gut überein: Der Unterschied beträgt weniger als 8 Prozent. □

Beispiel 15.12.

Gegeben: Die Funktion $f(x)=\frac{x^2}{1+x^2}$
(Zur einfacheren Nachvollziehbarkeit wurde hier eine Funktion gewählt, die nur von x abhängt.)

Gesucht: Näherungen für das Integral

$$\int_0^1\int_0^{1-x}\frac{x^2}{1+x^2}\,\mathrm{d}x\,\mathrm{d}y=0.061\,175\,426\,884\ldots$$

unter verschiedenen Genauigkeitsanforderungen

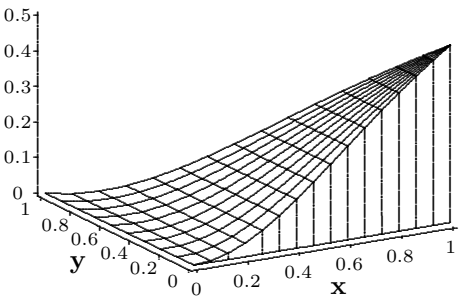


Abb. 15.19. Graph der Funktion $f(x)$

Lösung: (Gau-n sind die summierten Gauß-Formeln C^{G_n})

1. Geforderte Genauigkeit = $5{\cdot}10^{-7}$

Verfahren	ermittelte Näherung	relativer Schätzfehler	Funktions-Auswertungen
Gau-1	0.061 174 579 17	$2.1{\cdot}10^{-7}$	87 381
Gau-2	0.061 173 731 51	$4.2{\cdot}10^{-7}$	43 690
Gau-3	0.061 175 494 93	$-2.0{\cdot}10^{-8}$	255
Gau-7	0.061 175 477 82	$-1.7{\cdot}10^{-8}$	147

2. Geforderte Genauigkeit = $5{\cdot}10^{-8}$

Verfahren	ermittelte Näherung	relativer Schätzfehler	Funktions-Auswertungen
Gau-1	0.061 175 373 90	$1.3{\cdot}10^{-8}$	1 398 101
Gau-2	0.061 175 320 92	$2.6{\cdot}10^{-8}$	699 050
Gau-3	0.061 175 494 93	$-2.0{\cdot}10^{-8}$	255
Gau-7	0.061 175 477 82	$-1.7{\cdot}10^{-8}$	147

□

15.8.2 Kubaturformeln für Dreieckbereiche allgemeiner Lage

Analog zu der Behandlung der Integration über Rechtecke und über Dreiecke mit achsenparallelen Katheten lässt sich die Integration auch über Dreiecke allgemeiner Lage durchführen.

Es sei $R\Delta$

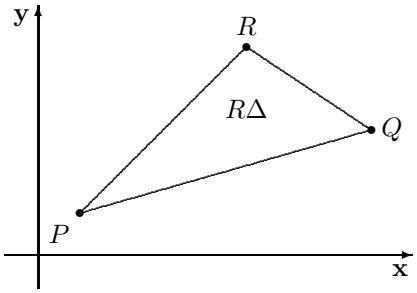


Abb. 15.20.

ein Referenz-Dreieck mit den drei Eckpunkten

$$P = \begin{pmatrix} p_1 \\ p_2 \end{pmatrix}, \quad Q = \begin{pmatrix} q_1 \\ q_2 \end{pmatrix}, \quad R = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}.$$

Im Folgenden werden einige Kubaturformeln angegeben, die alle das Integral

$$\iint\limits_{R\Delta} f(x,y)\,\mathrm{d}x\,\mathrm{d}y$$

annähern. Praktischerweise fasst man dazu die Argumente x und y der Funktion $f(x,y)$ zusammen zu einem Vektor $\boldsymbol{x}^\mathsf{T} = (x,y)$, schreibt die Funktion um zu

$$f(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathbf{R}^2,$$

und bildet die beiden Kantenvektoren

$$\begin{array}{rcccccc} \boldsymbol{a} & := & \overrightarrow{PQ} & = & Q-P & = & \begin{pmatrix} q_1-p_1 \\ q_2-p_2 \end{pmatrix} \\ \boldsymbol{b} & := & \overrightarrow{PR} & = & R-P & = & \begin{pmatrix} r_1-p_1 \\ r_2-p_2 \end{pmatrix}. \end{array}$$

Nun können alle in Abschnitt 15.8.1 aufgestellten Formeln einfach umgeschrieben werden. Die dort enthaltene Referenz-Dreiecks-Fläche $\frac{h_x\, h_y}{2}$ wird dabei ersetzt durch

$$A_{PQR} = \frac{1}{2}\big((q_1-p_1)(r_2-p_2) - (q_2-p_2)(r_1-p_1)\big)$$

und in der Angabe der jeweiligen Fehlerordnung $O(h_{\max}^n)$ entspricht $h_{\max}$ dem Maximum der Längen von $\boldsymbol{a}$ und $\boldsymbol{b}$.

15.8.2.1 Newton-Cotes-Kubaturformeln für Dreieckbereiche allgemeiner Lage

Es sei $R\Delta_1$

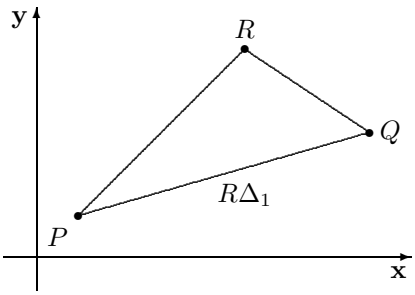


Abb. 15.21.

ein Referenz-Dreieck mit den drei Eckpunkten als Knoten. Dann lautet die Kubaturformel

$$C(f; R\Delta_1) = A_{PQR}\,\frac{1}{3}\left\{f(P) + f(Q) + f(R)\right\},$$

sie besitzt den Genauigkeitsgrad $L = 1$ und integriert alle Polynome P_1 vom Grad $s + t \le 1$ (d. h. Linearkombinationen der Monome 1, x und y) und hat die Fehlerordnung $O(h_{\max}^2)$.

Kubaturformeln vom Genauigkeitsgrad $L = 2$ ergeben sich unter Verwendung der Knoten der Referenz-Dreiecke $R\Delta_2$

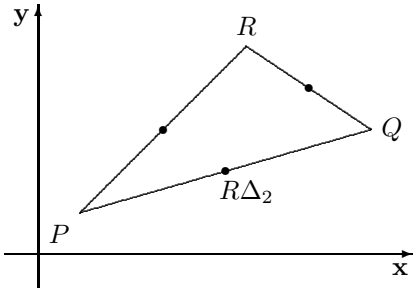


Abb. 15.22.

mit der Formel

$$C(f; R\Delta_2) = A_{PQR}\,\frac{1}{3}\left\{f\left(\frac{P+Q}{2}\right) + f\left(\frac{Q+R}{2}\right) + f\left(\frac{R+P}{2}\right)\right\}$$

oder $R\Delta_3$

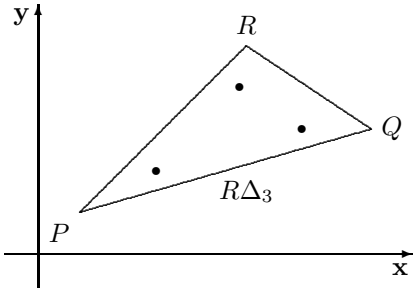


Abb. 15.23.

mit der Formel

$$\begin{aligned} C(f; R\Delta_3) \quad = \quad A_{PQR}\,\tfrac{1}{3}\Big\{ & \quad f\left(P + \tfrac{1}{6}\,\boldsymbol{a} + \tfrac{1}{6}\,\boldsymbol{b}\right) \\ & + f\left(P + \tfrac{2}{3}\,\boldsymbol{a} + \tfrac{1}{6}\,\boldsymbol{b}\right) \\ & + f\left(P + \tfrac{1}{6}\,\boldsymbol{a} + \tfrac{2}{3}\,\boldsymbol{b}\right)\Big\} \end{aligned}$$

bzw. (ohne Verwendung der Kantenvektoren)

$$\begin{aligned} C(f; R\Delta_3) \quad = \quad A_{PQR}\,\tfrac{1}{3}\Big\{ & \quad f\left(\tfrac{2}{3}\,P + \tfrac{1}{6}\,Q + \tfrac{1}{6}\,R\right) \\ & + f\left(\tfrac{1}{6}\,P + \tfrac{2}{3}\,Q + \tfrac{1}{6}\,R\right) \\ & + f\left(\tfrac{1}{6}\,P + \tfrac{1}{6}\,Q + \tfrac{2}{3}\,R\right)\Big\}. \end{aligned}$$

Beide Formeln integrieren alle Polynome vom Grad $s+t \leq 2$ (also Linearkombinationen der Monome 1, x, y, xy, x^2 und y^2) und besitzen die Fehlerordnung $O(h_{\max}^3)$.

Die entsprechenden summierten Formeln ergeben sich, wenn man einen Bereich B in Dreiecke B_j zerlegt und auf jedes Dreieck eine der angegebenen Formeln anwendet und aufsummiert.

Beispiel 15.13.

Gegeben: Die Funktion

$$f(x,y) = \frac{1}{4}\,x^2 - xy - \frac{1}{4}\,y^2 + 30$$

Gesucht: Ein Näherungswert für das Integral

$$I = \int_{x=0}^{10} \int_{y=1}^{\frac{2}{5}-x+2} f(x,y)\,\mathrm{d}x\,\mathrm{d}y$$

Graphisch veranschaulicht ist dies das Volumen zwischen den eingezeichneten Grund- und Deckflächen:

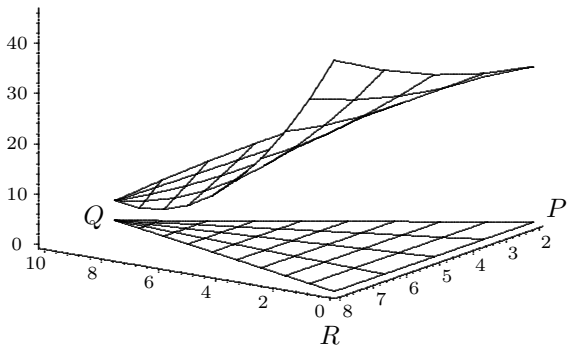


Abb. 15.24.

Verwendet werden soll die Drei-Punkt-Formel $C(f; R\Delta_3)$ für Dreiecke allgemeiner Lage.

Lösung: Die drei Eckpunkte des verwendeten Dreiecks sind

$$P = \begin{pmatrix} 0 \\ 2 \end{pmatrix},\ Q = \begin{pmatrix} 10 \\ 6 \end{pmatrix} \text{ und } R = \begin{pmatrix} 0 \\ 8 \end{pmatrix}.$$

Damit berechnet sich die Dreiecksfläche zu

$$A_{PQR} = \frac{1}{2}\det(Q-P, R-P) = \frac{1}{2}\begin{vmatrix} 10 & 0 \\ 4 & 6 \end{vmatrix} = 30$$

und somit erhält man als Näherung

$$\begin{aligned} C(f;R\Delta_3) &= A_{PQR}\cdot\tfrac{1}{3}\cdot \\ &\quad \Big\{ f\left(\tfrac{2}{3}\,0+\tfrac{1}{6}\,10+\tfrac{1}{6}\,0,\ \ \tfrac{2}{3}\,2+\tfrac{1}{6}\,6+\tfrac{1}{6}\,8\right) \\ &\quad +f\left(\tfrac{1}{6}\,0+\tfrac{2}{3}\,10+\tfrac{1}{6}\,0,\ \ \tfrac{1}{6}\,2+\tfrac{2}{3}\,6+\tfrac{1}{6}\,8\right) \\ &\quad +f\left(\tfrac{1}{6}\,0+\tfrac{1}{6}\,10+\tfrac{2}{3}\,0,\ \ \tfrac{1}{6}\,2+\tfrac{1}{6}\,6+\tfrac{2}{3}\,8\right)\Big\} \\ &= \tfrac{30}{3}\Big\{ f\left(\tfrac{5}{3},\ \tfrac{11}{3}\right)+f\left(\tfrac{20}{3},\ \tfrac{17}{3}\right)+f\left(\tfrac{5}{3},\ \tfrac{20}{3}\right)\Big\} \\ &= 10\left(\tfrac{503}{18}+\tfrac{409}{36}+\tfrac{1105}{36}\right)=700\,. \end{aligned}$$

Hier zur Anschauung die Punkte, an denen die Funktion ausgewertet wurde:

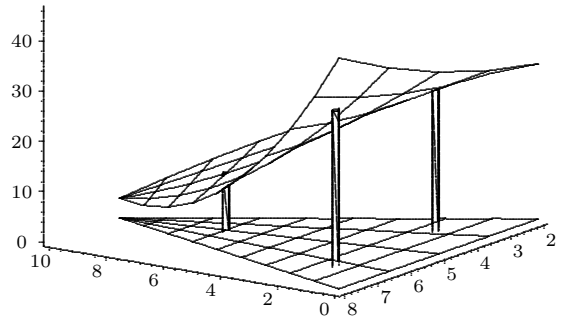


Abb. 15.25.

Man kann sich davon überzeugen, dass dieser Wert mit dem exakten Integralwert I übereinstimmt. □

15.8.2.2 Gauß-Kubaturformeln für Dreieckbereiche allgemeiner Lage

Analog zu den Gauß-Kubaturformeln für Rechteck-Bereiche und für Dreiecke mit achsenparallelen Katheten können (ähnlich wie bei den Newton-Cotes-Formeln) auch solche für Dreiecke allgemeiner Lage aufgestellt werden, jeweils abhängig vom Genauigkeitsgrad L.

Für $L=1$ beispielsweise ergeben sich die beiden folgenden Formeln, die zwar unterschiedliche Anzahlen von Funktionsauswertungen benötigen, aber beide von der Fehlerordnung $O(h_{\max}^2)$ sind:

1. Ein-Punkt-Kubaturformel für das Referenz-Dreieck $R\Delta_4$,

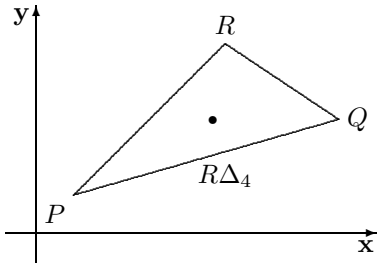


Abb. 15.26.

die nur einen einzigen Knoten verwendet:

$$C(f;R\Delta_4) = A_{PQR}\, f\left(P + \frac{1}{3}\,\boldsymbol{a} + \frac{1}{3}\,\boldsymbol{b}\right).$$

2. Zwei-Punkt-Kubaturformel für das Referenz-Dreieck $R\Delta_5$,

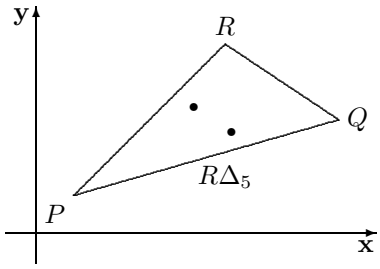


Abb. 15.27.

die zwei Knoten verwendet:

$$C(f;R\Delta_4) = A_{PQR}\,\frac{1}{2}\left\{f\left(P + \frac{1}{6}\,\boldsymbol{a} + \frac{1}{2}\,\boldsymbol{b}\right) + f\left(P + \frac{1}{2}\,\boldsymbol{a} + \frac{1}{6}\,\boldsymbol{b}\right)\right\}.$$

Für $L = 2$ gelten dieselben Drei-Punkt-Formeln für die Referenz-Dreiecke $R\Delta_2$ und $R\Delta_3$, die in Abschnitt 15.8.2.1 aufgestellt wurden.

Schließlich sei noch die Formel für $L = 5$ mit dem Referenz-Dreieck $R\Delta_6$

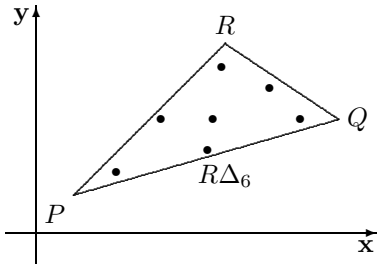


Abb. 15.28.

genannt, eine Sieben-Punkt-Formel, die aus [ENGE1980] entnommen wurde:

$$
\begin{aligned}
C(f;R\Delta_6) \;=\; & \tfrac{A_{PQR}}{1200}\Bigg\{ 270 \quad f\left(P+\tfrac{1}{3}\,\boldsymbol{a}+\tfrac{1}{3}\,\boldsymbol{b}\right) \\
& +(155+\sqrt{15})\Bigg[f\left(P+\tfrac{6+\sqrt{15}}{21}\,\boldsymbol{a}+\tfrac{6+\sqrt{15}}{21}\,\boldsymbol{b}\right) \\
& \qquad +f\left(P+\tfrac{9-2\sqrt{15}}{21}\,\boldsymbol{a}+\tfrac{6+\sqrt{15}}{21}\,\boldsymbol{b}\right) \\
& \qquad +f\left(P+\tfrac{6+\sqrt{15}}{21}\,\boldsymbol{a}+\tfrac{9-2\sqrt{15}}{21}\,\boldsymbol{b}\right)\Bigg] \\
& +(155-\sqrt{15})\Bigg[f\left(P+\tfrac{6-\sqrt{15}}{21}\,\boldsymbol{a}+\tfrac{6-\sqrt{15}}{21}\,\boldsymbol{b}\right) \\
& \qquad +f\left(P+\tfrac{9+2\sqrt{15}}{21}\,\boldsymbol{a}+\tfrac{6-\sqrt{15}}{21}\,\boldsymbol{b}\right) \\
& \qquad +f\left(P+\tfrac{6-\sqrt{15}}{21}\,\boldsymbol{a}+\tfrac{9+2\sqrt{15}}{21}\,\boldsymbol{b}\right)\Bigg]\Bigg\}
\end{aligned}
$$

mit der Fehlerordnung $O(h_{\max}^6)$.

Beispiel 15.14.

Gegeben: Die Funktion

$$f(x,y)=\frac{11}{10}-\mathrm{e}^{-3\,(x-4)^2-\frac{1}{2}\left(y-\frac{14}{3}\right)^2}$$

Gesucht: Ein Näherungswert für das Integral über die Funktion f mit der Sieben-Punkt-Gauß-Formel über dem Dreieck PQR mit

$$P=\begin{pmatrix}1\\2\end{pmatrix},\quad Q=\begin{pmatrix}7\\5\end{pmatrix},\quad R=\begin{pmatrix}4\\7\end{pmatrix}.$$

Graphisch veranschaulicht ist dies das Volumen zwischen den eingezeichneten Grund- und Deckflächen:

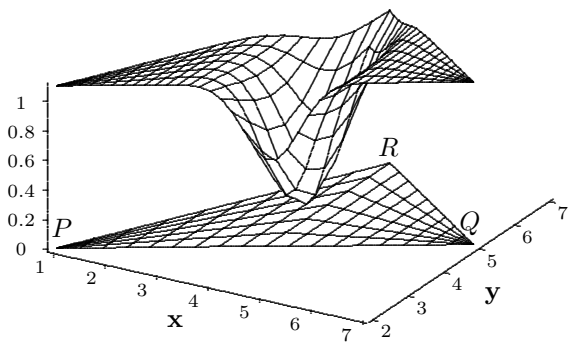


Abb. 15.29.

Lösung: Die Punkte, an denen die Funktion ausgewertet werden muss, sind (hier auf zwei Dezimalen gerundet):

(4.00, 4.67),
(5.23, 5.76), (2.77, 4.53), (4.00, 3.71),
(1.91, 2.81), (6.08, 4.90), (4.00, 6.29).

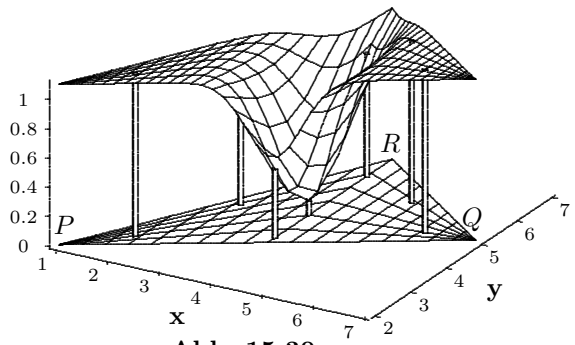


Abb. 15.30.

Damit und mit $A_{PQR} = 21/2 = 10.5$ ergibt sich der Näherungswert

$$\begin{aligned} C(f; R\Delta_6) &= 7.945\,384\,282 - 0.333\,509\,406\,2\sqrt{15} \\ &= 7.932\,467\,518. \end{aligned}$$

□

15.9 Entscheidungshilfen

Hier sind nur Kubaturformeln für Rechteckbereiche und Bereiche, die sich aus Dreiecken zusammensetzen lassen, behandelt. Zu allen Formeln sind auf der CD-ROM Programme zu finden. Die dort verwendete Fehlerschätzung verläuft analog zur Fehlerschätzung bei der Quadratur (vgl. Abschnitt 14.11).
Wie bei der Quadratur sind bei gleicher Genauigkeitsforderung die optimalen Gauß-Formeln wegen der weitaus geringeren Anzahl erforderlicher Funktionsauswertungen allen anderen Formeln überlegen; der Integrand muss an beliebiger Stelle des Integrationsbereiches auswertbar sein. Sind nur an diskreten Knoten Funktionswerte gegeben, so sind die Newton-Cotes-Formeln einsetzbar.
Das Romberg-Verfahren ist ebenfalls empfehlenswert, es benötigt jedoch gegenüber den Gauß-Formeln bei gleicher Genauigkeit weitaus mehr Funktionsauswertungen. Hier sollte mit nicht zu kleiner Anfangsschrittweite gearbeitet werden (Tests dazu in [KRAU1990]).

Ergänzende Literatur zu Kapitel 15

[ISAA1973]; [KROM1994], Kap.5; [QUAR2002], Kap15; [STRO1971].

Literaturverzeichnis

[ABRA1986] ABRAMOWITZ, M.; STEGUN, I.A. (ed.): Handbook of Mathematical Functions, Dover Publications, New York 1965, 10^{th} printing 1986.

[AHLB1986] AHLBERG, J.H.; NILSON, E.N.; WALSH, J.L.: The theory of splines and their applications, Academic Press, New York, London 1986.

[AKIM1970] AKIMA, H.: A New Method of Interpolation and Smooth Curve Fitting Based on Local Procedures, Journal of the Association for Computing Machinery, Vol. 17, No. 4, Oktober 1970, S. 589-602.

[ANDE1973] ANDERSON, N.; BJÖRCK, A.: A new high order method of regula falsi type for computing a root of an equation, BIT 13 (1973), S.253-264.

[BART2001] BARTSCH, H.-J.: Taschenbuch mathematischer Formeln. Fachbuchverlag, Hauser, 19., neu bearb. Aufl. München u.a., Fachbuchverl. Leipzig im Carl-Hanser-Verl., 2001.

[BART2004] BARTSCH, H.-J.: Taschenbuch mathematischer Formeln. Fachbuchverlag, Hauser, 20., neu bearb. Aufl. München u.a., Fachbuchverl. Leipzig im Carl-Hanser-Verl., 2004.

[BAUE1965] BAUER, F.L.; HEINHOLD, J.; SAMELSON, K.; SAUER, R.: Moderne Rechenanlagen, Teubner, Stuttgart 1965.

[BAUH1970] BAUHUBER, F.: Diskrete Verfahren zur Berechnung der Nullstellen von Polynomen, Computing 5 (1970), S.97-118.

[BEHR1975] BEHR, W.: Approximation und Interpolation stetiger Funktionen sowie Kurvenscharen aus Karosserieplänen zur Konzipierung von Heckleuchten, Staatsarbeit für das Lehramt an berufsbildenden Schulen, RWTH Aachen, 1975.

[BERE1971] BERESIN, I.S.; SHIDKOW, N.P.: Numerische Methoden, Bd. 1 und 2, VEB Deutscher Verlag der Wissenschaften, Berlin 1970, 1971.

[BEZI1972] BÉZIER, P.: Numerical Control, Mathematics and Applications, New York-London-Toronto 1972.

[BJOR1979] BJÖRCK, A.; DAHLQUIST, G.: Numerische Methoden, Oldenbourg, München, Wien (Originaltitel: "Numeriska methoder", Lund (Schweden) 1972), aus dem Schwedischen übersetzt von N. Krier u. P. Spellucci, 2. unveränd. Aufl., Studienausg. 1979, ISBN: 3-486-33852-8.

[BOHM1984] BÖHM, W.; FARIN, G.; KOHMANN, J.: A survey of curve and surface methods in CAGD. Computer Aided Geometric Design 1, (1984) 1-60.

[BOHM1985] BÖHM, W.; GOSE, G.; KAHMANN, J.: Methoden der numerischen Mathematik, Vieweg, Braunschweig-Wiesbaden 1985.

[BOOR1962] BOOR, de C.: Bicubic Spline Interpolation, J. Math. Phy. 41 (1962), S. 215.

[BOOR2001] BOOR, de C.: A Practical Guide to Splines, Springer, New York-Heidelberg-Berlin 1979, 1991, 2001, ISBN: 0-387-95366-3.

[BREN1971] BRENT, R.B.: An algorithm with guaranteed convergence for finding a zero of a function, Computer Journal, Vol. 14-4, 1971, pp 422-425.

[BRIG1997] BRIGHAM, E.O.: FFT-Anwendungen (Einführung in die Nachrichtentechnik) mit 6 Tabellen, 41 Beispielen und 188 Aufgaben sowie Programmen in BASIC, München, Wien, Oldenbourg 1997, ISBN 3-486-21567-1

[BRON1991] BRONSTEIN, I.N.; SEMENDJAJEW, K.A.: Taschenbuch der Mathematik, Teubner, Leipzig 1969, 14. Aufl. 1988 Harry Deutsch; 25. Auflage 1991.

[BRON2001] BRONSTEIN, I.N.; Semendjajew, K.A.; Musiol, G.: Taschenbuch der Mathematik, mit CD-ROM, 5., überarb. und erw. Aufl. der Neubearbeitung 2001 Frankfurt am Main [u.a.]: Deutsch, XXXVII, 1191 S., ISBN: 3-8171-2015-X, KNO-NR: 07 75 85 96.

[BROW1971] BROWN, K.M.; DENNIS, J.E., Jr.: On the second order convergence of Brown's derivative-free method for solving simultaneous nonlinear equations, Yale University, Comp.Sci.Dept., Techn. Report, 1971, S. 71-77 .

[BUNS1995] BUNSE, W.; BUNSE-GERSTNER, A.: Numerische lineare Algebra, Teubner Studienbücher, Stuttgart 1995.

[BUTZ2003] BUTZ, T.: Fouriertransformation für Fußgänger, 3., durchges. und erw. Aufl. B.G.Teubner Stuttgart - Leipzig 2003, ISBN: 3-519-20202-6.

[CARN1990] CARNAHAN, B.; LUTHER, H.A.; WILKES, J.O.: Applied Numerical Methods, John Wiley, New York 1990.

[CLIN1979] CLINE, A.K.; MOLER, C.B.; STEWART, G.W.; WILKINSON, J.H.: An estimate for the condition number of a matrix. SIAM J. Numer. Anal. 16 (1979) S.368-375.

[COLL1968] COLLATZ, L.: Funktionalanalysis und numerische Mathematik, Springer, Berlin-Heidelberg-New York 1968.

[COLL1973] COLLATZ, L.; ALBRECHT, J.: Aufgaben aus der Angewandten Mathematik I und II, Vieweg, Braunschweig 1972, 1973.

[CONT1987] CONTE, S.D.; de BOOR, C.: Elementary Numerical Analysis. An algorithmic approach, New York-Sidney-Toronto 1965, Third Edition, McGraw-Hill, Auckland, 1987, ISBN: 0-07-012447-7.

[CUTH1969] CUTHILL, E.; MCKEE, J.: Reducing the bandwith of sparse symmetric matrices, ACM, New York 1969, S.157-172.

[DEKK1969] DEKKER, T.J.: Finding a zero by means of successive linear interpolation in: Constructive aspects of the fundamental theorem of algebra, herausgegeben von B. Dejon und P. Henrici, Wiley-Interscience, New York 1969.

[DEMI1968] DEMIDOWITSCH, B.P.; MARON, I.A.; SCHUWALOWA, E.S.: Numerische Methoden der Analysis, VEB Deutscher Verlag der Wissenschaften, Berlin 1968.

[DENN1996] DENNIS, J.E.; SCHNABEL, R.B.: Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Prentice Hall, Englewood Cliffs, New York, SIAM Philadelphia, 1996.

[DEUF2002] DEUFLHARD, P.: Numerische Mathematik, Bd.1: Eine algorithmisch orientierte Einführung: Unter Mitarb. v. Andreas Hohmann. Gruyter Lehrbuch. 3., überarb. u. erw. Aufl. 2002, ISBN: 3-11-017182-1, KNO-NR: 04 42 36 08; Bd.2: Gewöhnliche Differentialgleichungen: Unter Mitarb. v. Folkmar Bornemann. Gruyter Lehrbuch. 2., überarb. u. erw. Aufl. 2002, ISBN: 3-11-017181-3, KNO-NR: 05 61 70 68.

[DONG1993] DONGARRA, J.J.: LINPACK Users' Guide, SIAM Philadelphia 1979, 10^{th} printing 1993.

[DOWE1971] DOWELL, M.; JARRATT, P.: A Modified Regula Falsi Method for Computing the Root of an Equation, BIT 11 (1971), S.168-174.

[DOWE1972] DOWELL, M.; JARRATT, P.: The "Pegasus" Method for Computing the Root of an Equation, BIT 12 (1972), S.503-508.

[ENGE1980] ENGELS, H.: Numerical Quadrature and Cubature, Academic Press, London-New York-Toronto-Sydney-San Francisco 1980.

[ENGE1996] ENGELN-MÜLLGES, G.; REUTTER, F.: Numerik-Algorithmen, VDI-Verlag, 8. neubearbeitete und erweiterte Auflage, 1996.

[FADD1979] FADDEJEW, D.K.; FADDEJEWA, W.N.: Numerische Methoden der linearen Algebra, Oldenbourg, Berlin 1970, München, Wien, 5. Aufl. 1979.

[FICH1987] FICHTENHOLZ, G.M.: Differential- und Integralrechnung Bd. III, 11. Auflage, VEB Deutscher Verlag der Wissenschaften, Berlin 1987; 12. Auflage 1992.

[FORD1977] FORD, J.A.: A Generalization of the Jenkins-Traub Method, Mathematics of Computation, Vol. 31 (1977), S.193-203.

[FORS1971] FORSYTHE, G.E.; MOLER, C.B.: Computer-Verfahren für lineare algebraische Systeme, Oldenbourg, München-Wien 1971.

[FORS1977] FORSYTHE, G.E.; MALCOM, M.M.; MOLER, C.B.: Computer methods for mathematical computations, Prentice-Hall, Englewood-Cliffs, New Jersey 1977.

[FRAN1982] FRANKE, R.: Scatterend data Interpolation: Tests of some methods. Mathematics of Computation 38 (1982), S.181-200.

[GOLU1996] GOLUB, G.H.; van LOAN, C.F.: Matrix Computations, Baltimore, Maryland, The John Hopkins University Press, 3. Auflage 1996, ISBN/ISSN: 0801854148 (pbk. : alk. paper) 080185413X (hc : alk. paper).

[GOOS1988] GOOS, G.; HARTMANIS, J.; LEEUWEN, J. VAN (Hrsg.): Lectures Notes in Computer Science, Vol. 6 Matrix Eigensystem Routines - EISPACK Guide, Springer, Berlin-Heidelberg-New York 1974, 2. Aufl. 1976; 4. Nachdruck 1988.

[GRAM2000] GRAMLICH, G.; WERNER, W.: Numerische Mathematik mit Matlab, Eine Einführung für Naturwissenschaftler und Ingenieure, dpunkt/PRO, ISBN 3-932588-55-X, Broschiert, 03/2000

[HAMM1978] HÄMMERLIN, G.: Numerische Mathematik I, BI-Hskrpt. 498/498a, BI-Wissenschaftsverlag, Mannheim-Wien-Zürich 1970, 2. überarbeitete Aufl. 1978.

[HAMM1994] HÄMMERLIN, G.; HOFFMANN, K.H. Numerische Mathematik, Springer, Berlin-Heidelberg 1994; 4. DURCHGES: Auflage 1994, ISBN: 3-540-58033-6, KNO-NR: 03 55 20 72.

[HALL1968] HALL, C.A.: On Error Bounds for Spline Interpolation, J. Approx. Theory, 1, 1968, 209-218.

[HENR1972] HENRICI, P.: Elemente der Numerischen Analysis, Bd. 1 und 2, BI-Htb. 551 und 562, BI-Wissenschaftsverlag, Mannheim-Wien-Zürich 1972.

[HERM2001] HERMANN, M.: Numerische Mathematik, 2001, Oldenbourg, ISBN: 3-486-25558-4, KNO-NR: 09 33 08 74

[HILD1987] HILDEBRAND, F.B.: Introduction to numerical Analysis, 2. Auflage, McGraw-Hill, New York 1987.

[HOSC1989] HOSCHEK, J.; LASSER, D.: Grundlagen der geometrischen Datenverarbeitung, Teubner, Stuttgart 1989; 2. Auflage 1992.

[IGAR1985] IGARASHI, M.: Practical stopping rule for finding roots of nonlinear equations, J. of Computational and Applied Mathematics 12, 13 (1985), S.371-380 North Holland.

[ISAA1973] ISAACSON, E.; KELLER, H.B.: Analyse numerischer Verfahren, Harri Deutsch, Zürich und Frankfurt 1973.

[JENK1970] JENKINS, M.A.; TRAUB, J.F.: A Three-Stage-Algorithm for Real Polynomials using Quadratic Iteration, SIAM J. Num. Anal. Vol.7 (1970), S.545-566 (vgl. a. Numer. Math. 14 (1970) S.252-263).

[KAHA1983] KAHANER, D.K.; STOER, J.: Extrapolated adaptive quadrature, SIAM J. Sci. Stat. Comput., Vol. 4, No. 1, (1983).

[KELL1990] KELLER, H.B.: Numerical Solution of Two Point Boundary Value Problems, SIAM Philadelphia, 1976, 4. print. 1990

[KIEL1988] KIELBASINSKI, A.; SCHWETLICK, H.: Numerische lineare Algebra, VEB Deutscher Verlag der Wissenschaften, Berlin 1988.

[KING1973] KING, R.F.: An Improved Pegasus-Method for Root Finding, BIT 13 (1973), S.423-427.

[KIOU1978] KIOUSTELIDIS, J.B.: Algorithmic Error Estimation for Approximate Solution of Nonlinear Systems of Equations, Computing 19 (1978), S.313-320.

[KIOU1979] KIOUSTELIDIS, J.B.: A Derivative-Free Transformation Preserving the Order of Convergence of Iteration Methods in Case of Multiple Zeros, Num. Math. 33 (1979), S.385-389.

[KNOR2003] KNORRENSCHILD, M.: Numerische Mathematik, Eine beispielorientierte Einführung, Mit 77 Beispielen und 69 Aufgaben, Mathematik-Studienhilfen, 2003. ISBN: 3-446-22169-7, KNO-NR: 11 30 03 88, Hanser Fachbuchverlag, Fachbuchverlag Leipzig.

[KOCK1990] KÖCKLER, N.: Numerische Algorithmen in Softwaresystemen, Teubner, Stuttgart 1990.

[KRAB1975] KRABS, W.: Optimierung und Approximation, Teubner, Suttgart 1975.

[KRAU1990] KRAUSE, B.: Tests zu einigen Kubaturformeln für Rechteckgebiete, Studienarbeit FH Aachen 1990, Aufgabensteller und Betreuer G. Engeln-Müllges.

J. Ass. for Comp. Mach., Vol. 13 (1966), S.374-385.

[KROM1994] KROMMER, A.R.; Überhuber, C.W.: Numerical Integration: On Advanced Computer Systems, Lecture Notes in Computational Science and Engineering Vol.848. 1994. ISBN: 3-540-58410-2, KNO-NR: 09 22 70 54, Springer, Berlin.

[KRYL1991] KRYLOV, V.I.: Approximate Calculation of Integrals, Macmillan, New York-London 1962 (Translated by STROUD, A.H.) Prentice Hall 1991.

[KUHN1990] KÜHN: Entwicklung eines interaktiven Programms zur graphischen Darstellung glatter Kurven und Flächen unter Verwendung der Shepard-Interpolation sowie des Graphik-Paketes DISSPLA, Diplomarbeit FH Aachen 1990, Referent und Betreuer: G. Engeln-Müllges.

[LAUX1988] LAUX, M.: Automatische Herleitung und Verifikation von Quadraturformeln, Institutsbericht 88-5, Institut für Aerodynamik und Gasdynamik der Universität Stuttgart, 1988

[LOAN1992] VAN LOAN, C.: Matrix Framework for the Fast Fourier Transform, SIAM, Philadelphia, 1992.

[LOUI1998] LOUIS, A.K.; MAAß, P.; RIEDER, A.: Wavelets, Theorie und Anwendungen, 2. überarbeitete und erweiterte Auflage, B. G. Teubner Stuttgart, 1998

[MAES1985] MAESS, G.: Vorlesungen über Numerische Mathematik I, Akademie-Verlag, Berlin 1984; Birkhäuser, Basel, Stuttgart 1985.

[MAES1988] MAESS, G.: Vorlesungen über Numerische Mathematik II, Akademie-Verlag, Berlin 1988.

[MART1968] MARTIN, R.S.; J.H. WILKINSON: Similarity Reduction of a General Matrix to Hessenberg Form, Num. Math. 12 (1968), S.349-368.

[MCCA1967] McCALLA, Th. R.: Introduction to Numerical Methods and FORTRAN Programming, John Wiley, New York 1967.

[MCCR1987] McCRACKEN, D.D.; DORN, W.S.: Numerical Methods with FORTRAN IV Case Studies, John Wiley, New York 1987.

[MEIN1967] MEINARDUS, G.: Approximation von Funktionen und ihre numerische Behandlung, Springer, Berlin-Heidelberg-New York 1964, engl. Ausgabe 1967.

[MEING1979] MEINGUET, B.J.: Multivariate Interpolation at Arbitrary Points Made Simple, Z.A.M.P, Vol. 30 (1979), S. 292-304.

[MOOR1980] MOORE, R.E.; KIOUSTELIDIS, J.B.: A Simple Test for Accuracy of Approximate Solutions to Nonlinear (or linear) Systems, SIAM J. Num. Anal. 17 (1980) 4, S. 521-529.

[MUEL1995] MÜLLER, M.W.; FELTEN, M.; MACHE, D.H.: Approximationstheorie Akademie Verlag, Berlin, 1995, ISBN 3-05-501673-4.

[MULL1956] MULLER, D.E.: A Method for Solving Algebraic Equations using an Automatic Computer, Math. Tables Aids Comp. 10 (1956), S.208-215.

[MUWI1999] MÜLLER-WICHARDS, D.: Transformationen und Signale, Stuttgart, Leipzig, Teubner 1999, ISBN 3-519-02742-9

[NIED1984] NIEDERDRENK, K.: Die endliche Fourier- und Walsh-Transformation mit einer Einführung in die Bildverarbeitung, Hrsg. G. Engeln-Müllges, 2. Auflage, Vieweg-Verlag, Wiesbaden 1984; 3. Auflage 1993.

[NIED1987] NIEDERDRENK, K.; YSERENTANT, H.: Funktionen einer Veränderlichen, Reihe: Rechnerorientierte Ingenieurmathematik, Hrsg. G. Engeln-Müllges, Vieweg, Braunschweig-Wiesbaden 1987.

[NIEM1987] NIEMEYER, H.: Lineare Algebra, Reihe: Rechnerorientierte Ingenieurmathematik, Hrsg. G. Engeln-Müllges, Vieweg-Verlag, Braunschweig-Wiesbaden 1987.

[NIEM1991] NIEMEYER, H.: Funktionen von mehreren Veränderlichen, analytische und numerische Behandlung, Reihe: Rechnerorientierte Ingenieurmathematik, Hrsg. G. Engeln-Müllges, Vieweg-Verlag, Braunschweig-Wiesbaden 1991.

[NIET1970] NIETHAMMER, W.: Über- und Unterrelaxation bei linearen Gleichungssystemen, Computing 5 (1970), S.303-311.

[NITS1968] NITSCHE, J.: Praktische Mathematik, BI-Hskprt. 812, BI-Wissenschaftsverlag, Mannheim-Zürich 1968.

[OPFE2002] OPFER, G.: Numerische Mathematik für Anfänger, Eine Einführung für Mathematiker, Ingenieure und Informatiker mit zahlreichen Beispielen und Programmen, Vieweg, F/VVA, 4. durchges. Auflage 2002 ISBN: 3-528-37265-6.

[OPPE1992] OPPENHEIM, A.V.; WILLSKY, A.S.: Signale und Systeme, Lehrbuch, 2., durchges. Aufl., unter Mitarbeit von Jan T. Young, VCH Weinheim, Basel, Cambridge, New York, 1992, ISBN 3-527-28433-8.

[OVER2001] OVERTON, M.L.: Numerical Computing with IEEE Floating Point Arithmetic, Including One Theorem, One Rule of Thumb, and One Hundred and One Exercises, Courant Institute of Mathematical Sciences, New York, 2001, ISBN 0-89871-482-6.

[PALM1988] PALM, G.: Programme zur Berechnung polynomialer Ausgleichssplines dritten Grades, Studienarbeit FH Aachen 1988, Aufgabensteller und Betreuer: G. Engeln-Müllges.

[PARL1969] PARLETT, B.N.; REINSCH, C.: Balancing a Matrix for Calculation of Eigenvalues and Eigenvectors, Num. Math. 13 (1969), S.293-304.

[PETE1970] PETERS, G.; WILKINSON, J.H.: Eigenvectors of Real and Complex Matrices by LR and QR triangularizations, Num. Math. 16 (1970), S.181-204.

[PIEG1997] PIEGL, L.; TILLER, W.: The Nurbs Book, Springer, New York-Heidelberg-Berlin, 2. Auflage 1997.

[POLO1964] POLOSHI, G.N.: Mathematisches Praktikum, Teubner, Leipzig 1964.

[PLAT2000] PLATO, R.: Numerische Mathematik kompakt. Grundlagenwissen für Studium und Praxis. VIEWEG, mit Online-Service z. Buch. 2000. ISBN: 3-528-03153-0, KNO-NR: 08 92 82 35.

[PREU2001] PREUSS, W.; WENISCH, G.: Lehr- und Übungsbuch Numerische Mathematik Hanser, C/VM, ISBN 3-446-21375-9, Gebunden, 2001

[QUAD1983] QUADPACK, A Subrative Package for Automatic Integration, Springer 1983.

[QUAR2001] QUARTERONI, A.; SACCO, R.; SALERI, F.: Numerische Mathematik, Bd.2, Springer-Lehrbuch, 2001.

[QUAR2002] QUARTERONI, A.; SACCO, R.; SALERI, F.: Numerische Mathematik, Bd.2, Springer-Lehrbuch, 2002, ISBN: 3-540-43616-2, KNO-NR: 10 93 51 03.

[RALS2001] RALSTON, A.; RABINOWITZ, P.: A First Course in Numerical Analysis, International Student Edition, McGraw-Hill, Kogokusha, 2. Aufl. 1978, Mineola: Dover, 2001, ISBN - 048641454X.

[RALS1979] RALSTON, A.; WILF, H.S.: Mathematische Methoden für Digitalrechner I, Oldenbourg, München-Wien 1967, 2. Aufl. 1972, II: München-Wien 1969, 2. Aufl. 1979.

[REIN1971] REINSCH, C.: Smoothing by Spline Functions I, Num. Math. 10 (1967), S.177-183; II.: Num. Math. 16 (1971), S.451-454.

[RENN1981] RENNER, G.; POCHOP, V.: A New Method for Local Smooth Interpolation, Eurographics 81, J.L. Encarnacao (ed.), S.137-147.

[RENN1982] RENNER, G.: A method of shape description for mechanical engineering practice, Computers in Industry 3 (1982), S.137-142.

[RICE1993] RICE, John R.: Numerical Methods, Software and Analysis, McGraw-Hill, New York 1983, Boston Acad. Pr. 1993.

[SAUE1969] SAUER, R.; SZABO, I.: Mathematische Hilfsmittel des Ingenieurs, Springer, Berlin-Heidelberg-New York, Teil II, 1969, Teil III, 1968.

[SCHE1989] SCHENDEL, U.: Sparse matrices: numerical aspects with applications for scientists and engineers, Ellis Horwood, 1989, ISBN/ISSN: 0470214066 (Halsted) 074580635X.

[SCHM1963] SCHMIDT, J.W.: Eine Übertragung der Regula Falsi auf Gleichungen in Banachräumen, ZAMM 43 (1963), S.1-8 und S.97-110.

[SCHU1977] SCHUMACHER, H.: Möglichkeiten des Einsatzes von Splines bei der Lösung von Problemen aus der Hochtemperatur-Gasdynamik. Staatsarbeit für das Lehramt an berufsbildenden Schulen, RWTH Aachen, 1977.

[SCHU1992] SCHUMAKER, L.L.: Fitting surfaces to scattered data, Approximation Theory II, Hrsg.: G.G. Lorentz, C.K. Chui u. L.L. Schumaker, Academic Press, New York 1976, 4. vollst. überarb. Aufl. 1992

[SCHW1972] SCHWARZ, H.R.; STIEFEL, E.; RUTISHAUSER, H.: Numerik symmetrischer Matrizen, Teubner, Stuttgart 1968, 2., durchges. und erw. Auflage 1972.

[SCHW1988] SCHWARZ, H.R.: FORTRAN-Programme zur Methode der finiten Elemente, Teubner, Stuttgart 1988; 3. Auflage 1991.

[SCHW1991] SCHWARZ, H.R.: Methoden der finiten Elemente, Eine Einführung unter besonderer Berücksichtigung der Rechenpraxis, Teubner, Stuttgart, 2. Auflage 1984; 3. Auflage 1991.

[SCHW1997] SCHWARZ, H.R.: Numerische Mathematik, Teubner, Stuttgart 1986; 3. Auflage 1993; 4., überarb. und erw. Auflage 1997.

[SCHWE1979] SCHWETLICK, H.: Numerische Lösung nichtlinearer Gleichungssysteme, VEB Verlag der Wissenschaften, Berlin 1979.

[SELD1979] SELDER, H.: Einführung in die Numerische Mathematik für Ingenieure, Hanser, München 1973, 2. durchges. und ergänzte Aufl. 1979.

[SHAM1973] SHAMPINE, L.F.; ALLEN, R.C. Jr.: Numerical Computing: An Introductuion, Saunders, Philadelphia, London, Toronto 1973.

[SHEP1968] SHEPARD, D.: A two dimensional interpolation function for irregularly-spaced data, ACM National Conference (1968), S.517-524.

[SPAT1973] SPÄTH, H.: Algorithmen für elementare Ausgleichsmodelle, Oldenbourg, München-Wien 1973.

[SPAT1974] SPÄTH, H.: Algorithmen für multivariable Ausgleichsmodelle, Oldenbourg, München-Wien 1974.

[SPAT1986] SPÄTH, H.: Spline Algorithmen zur Konstruktion glatter Kurven und Flächen, Oldenbourg, München-Wien 1973, 4. Aufl. 1986.

[SPEL1985] SPELLUCCI, P.; TÖRNIG, W.: Eigenwertberechnung in den Ingenieurwissenschaften, Teubner, Stuttgart 1985.

[STEP1979] STEPLEMAN, R.S.; WINARSKY, N.D.: Adaptive Numerical Differentiation, Mathematics of Computations, Vol. 33, No. 148, 1979, p. 1257-1264.

[STEU1979] STEUTEN, G.: Realisierung von Splinemethoden zur Konstruktion glatter Kurven und Flächen auf dem Bildschirm, Diplomarbeit RWTH Aachen 1979, Referenten: G. Engeln-Müllges, H. Petersen.

[STEW1973] STEWART, G.W.: Introduction to Matrix Computations, Academic Press, New York 1973.

[STIE1976] STIEFEL, E.: Einführung in die Numerische Mathematik, Teubner, Stuttgart 1970, 5. Aufl. 1976.

[STOE1989] STOER, J.: Numerische Mathematik 1, früher: Einführung in die Numerische Mathematik I, Springer, Berlin-Heidelberg-New York 1970, 3. Aufl. 1979, 4. Aufl. 1983, 5. Aufl. 1989.

[STOE1990] STOER, J.; BULIRSCH, R.: Numerische Mathematik 2, früher: Einführung in die Numerische Mathematik II, Springer, Berlin-Heidelberg-New York 1973, 2. neu bearb. Aufl. 1978, 3., verb. Aufl. 1990.

[STOE1999] STOER, J.: Numerische Mathematik. Eine Einführung. Unter Berücksichtigung von Vorlesungen v. F. L. Bauer. Bd.1 Springer-Lehrbuch. 8., neubearb. u. erw. Aufl. 1999. ISBN: 3-540-66154-9, KNO-NR: 00 54 61 43 -SPRINGER, BERLIN-, Bd.2 Unter Mitarb. v. Roland Bulirsch. Springer-Lehrbuch. 4., neubearb. u. erw. Aufl. 2000. ISBN: 3-540-67644-9, KNO-NR: 00 21 11 56 -SPRINGER, BERLIN-

[STOE2002] STOER, J.: Numerische Mathematik 1, Springer, Berlin-Heidelberg-New York 1970, 5. Aufl. 1989, 7. Aufl. 1994, 8. Auflage 2002.

[STRO1966] STROUD, A.H.; SECREST, D.: Gaussian Quadrature Formulas, Prentice-Hall, Englewood Cliffs, New Jersey, 1966.

[STRO1971] STROUD, A.H.: Approximate Calculation of Multiple Integrals, Prentice-Hall, Englewood Cliffs, New Jersey, 1971.

[STUM1982] STUMMEL, F.; HAINER, K.: Praktische Mathematik, Teubner, Stuttgart 1970, 2. überarb. u. erw. Aufl. 1982, ISBN 3-519-12040-2.

[TORN1990] TÖRNIG, W.; SPELLUCCI, P.: Numerische Mathematik für Ingenieure und Physiker, Springer, Berlin-Heidelberg-New York 1979, Bd. 1: Numerische Methoden der Algebra, 2. Aufl. 1988; Bd. 2: Numerische Methoden der Analysis, 2. Aufl. 1990.

[TRAU1966] TRAUB, J.F.: A Class of Globally Convergent Iteration Functions for the Solution of Polynomal Equations, Math. of Comp. 20 (1966), S.113-138.

[TRAU1984] TRAUB, J.F.: Iterative Methods for the Solution of Equations, Prentice-Hall, Englewood Cliffs, New Jersey 1964, 2. Aufl. 1984.

[UBER1995] ÜBERHUBER; C.W.: Computer-Numerik. Tl.1 Neuaufl. 2002. ISBN: 3-540-59151-6, KNO-NR: 05 95 50 32 -SPRINGER, BERLIN- Tl.2 1995. ISBN: 3-540-59152-4, KNO-NR: 05 95 50 54 -SPRINGER, BERLIN-.

[WEIS1984] WEISSINGER, J.: Numerische Mathematik auf Personal-Computern, Teil 1, 2, BI-Wissenschaftsverlag, Mannheim 1984.

[WEIS1990] WEISSINGER, J.: Spärlich besetzte Gleichungssysteme. Eine Einführung mit BASIC- und PASCAL-Programmen, BI-Wissenschaftsverlag, Mannheim 1990.

[WERN1980] WERNER, H.; JANSSEN, J.P.; ARNDT, H.: Probleme der praktischen Mathematik. Eine Einführung, 2. Aufl., Bd. I/II, BI-Htb. 134/135, BI-Wissenschaftsverlag, Mannheim, Wien, Zürich 1980.

[WERN1993] WERNER, H.; SCHABACK, R.: Numerische Mathematik I, Springer, Berlin-Heidelberg-New York 1972, 4., vollst. überarb. Aufl. 1993 (siehe auch [SCHA1993]).

[WILK1961] WILKINSON, J.H.: Error Analysis of Direct Methods of Matrix Inversion, J. Assoc. Comput. March 8, S. 281-330, 1961.

[WILK1969] WILKINSON, J.H.: Rundungsfehler, Springer, Berlin-Heidelberg-New York 1969.

[WILK1996] WILKINSON, J.H.: The Algebraic Eigenvalue Problem, Clarendon Press, Oxford 1965; Reprint 1996.

[WILL1971] WILLERS, F.A.: Methoden der praktischen Analysis, de Gruyter, Berlin 1957, 4. verb. Aufl. 1971.

[WODI1991] WODICKA, R.: Ergänzungen zu Akima's Steigungsformel, Mitteilungen aus dem Mathem. Seminar Giessen, Heft 203, Selbstverlag des Mathematischen Instituts, Giessen 1991.

[YOUN2003] YOUNG, D.M.: Iterative Solution of Large Linear Systems, Mineola, N.Y. : Dover Publ., 2003, ISBN 0-486-42548-7.

[ZIEL1974] ZIELKE, G.: Testmatrizen mit maximaler Konditionszahl, Computing 13 (1974), S.33-54.

[ZIEL1975] ZIELKE, G.: Testmatrizen mit freien Parametern, Computing 15 (1975), S.87-103.

[ZIEL1986] ZIELKE, G.: Report on Test Matrices for Generalized Inverses, Computing 36 (1986), S.105-162.

[ZURM1965] ZURMÜHL, R.: Praktische Mathematik für Ingenieure und Physiker, Springer, Berlin-Heidelberg-New York, 5. neubearb. Aufl. 1965; Nachdruck 1984.

[ZURM1997] ZURMÜHL, R.; FALK, S.: Matrizen und ihre Anwendungen für Ingenieure, Physiker und Angewandte Mathematiker, Teil 1: Grundlagen, 6. vollst. neu bearb. Aufl., Springer, Berlin-Heidelberg-New York 1992; Teil 2: Numerische Methoden, 5. überarb. und erw. Aufl., Springer, Berlin-Heidelberg-New York 1984/1986; 7. Auflage, Berlin Springer 1997.

Sachwortverzeichnis