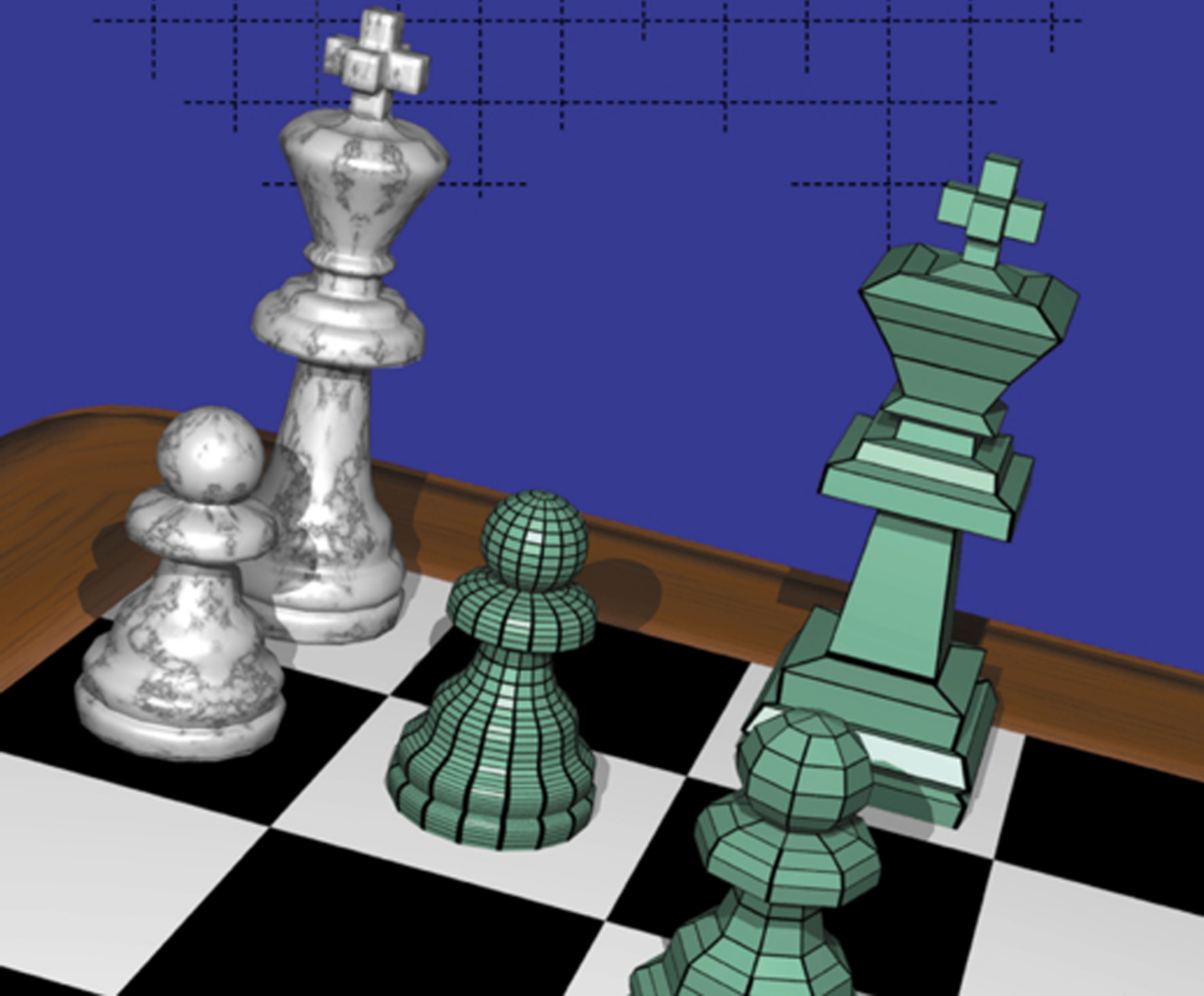
Joe Warren
Henrik Weimer

Subdivision Methods for Geometric Design



Subdivision Methods for Geometric Design: A Constructive Approach

The Morgan Kaufmann Series in Computer Graphics and Geometric Modeling
Series Editor: Brian A. Barsky,
University of California, Berkeley

Subdivision Methods for Geometric Design: A Constructive Approach
Joe Warren and Henrik Weimer

The Computer Animator's Technical Handbook
Lynn Pocock and Judson Rosebush

Computer Animation: Algorithms and Techniques
Rick Parent

Advanced RenderMan: Creating CGI for Motion Pictures
Anthony A. Apodaca and Larry Gritz

Curves and Surfaces in Geometric Modeling: Theory and Algorithms
Jean Gallier

Andrew Glassner's Notebook: Recreational Computer Graphics
Andrew S. Glassner

Warping and Morphing of Graphical Objects
Jonas Gomes, Lucia Darsa, Bruno Costa, and Luis Velho

Jim Blinn's Corner: Dirty Pixels
Jim Blinn

Rendering with Radiance: The Art and Science of Lighting Visualization
Greg Ward Larson and Rob Shakespeare

Introduction to Implicit Surfaces
Edited by Jules Bloomenthal

Jim Blinn's Corner: A Trip Down the Graphics Pipeline
Jim Blinn

Interactive Curves and Surfaces: A Multimedia Tutorial on CAGD
Alyn Rockwood and Peter Chambers

Wavelets for Computer Graphics: Theory and Applications
Eric J. Stollnitz, Tony D. DeRose, and David H. Salesin

Principles of Digital Image Synthesis
Andrew S. Glassner

Radiosity & Global Illumination
François X. Sillion and Claude Puech

Knotty: A B-Spline Visualization Program
Jonathan Yen

User Interface Management Systems: Models and Algorithms
Dan R. Olsen, Jr.

Making Them Move: Mechanics, Control, and Animation of Articulated Figures
Edited by Norman I. Badler, Brian A. Barsky, and David Zeltzer

Geometric and Solid Modeling: An Introduction
Christoph M. Hoffmann

An Introduction to Splines for Use in Computer Graphics and Geometric Modeling
Richard H. Bartels, John C. Beatty, and Brian A. Barsky

Subdivision Methods for Geometric Design: A Constructive Approach

Joe Warren
Rice University

Henrik Weimer
DaimlerChrysler AG, Berlin

MORGAN KAUFMANN PUBLISHERS
AN IMPRINT OF ACADEMIC PRESS
A Division of Harcourt, Inc.
SAN FRANCISCO SAN DIEGO NEW YORK BOSTON
LONDON SYDNEY TOKYO

Executive Editor	Diane D. Cerra
Publishing Services Manager	Scott Norton
Production Editor	Howard Severson
Assistant Editor	Belinda Breyer
Cover Design	Ross Carron Design
Text Design	Rebecca Evans & Associates
Composition and Illustration	Interactive Composition Corporation
Copyeditor	Daril Bentley
Proofreader	Mary Roybal
Indexer	Bill Meyers
Printer	Courier Corporation

Morgan Kaufmann Publishers
340 Pine Street, Sixth Floor, San Francisco, CA 94104-3205, USA
http://www.mkp.com

ACADEMIC PRESS
A Division of Harcourt, Inc.
525 B Street, Suite 1900, San Diego, CA 92101-4495, USA
http://www.academicpress.com

Academic Press
Harcourt Place, 32 Jamestown Road, London, NW1 7BY, United Kingdom
http://www.academicpress.com

06 05 04 03 02   5 4 3 2 1

Library of Congress Control Number: 2001094349
ISBN: 1-55860-446-4

This book is printed on acid-free paper.

Foreword

Tony DeRose
Pixar Animation Studios

Considerable effort has been expended over the last forty years in the area of geometric design, the goal of which is to develop efficient and flexible representations of complex shapes. Applications of geometric design are broad and varied; they include aerospace and automotive design, industrial design, and computer animation for interactive games, commercials, and full-length feature films.

Bézier curves, developed independently by Pierre Bézier at Renault and Paul de Casteljau at Citroen in the early '60s, were one of the first successful computer representations. They are still the representation of choice in some applications. Bézier curves were followed by B-splines, which were introduced into geometric design in the mid '70s by Riesenfeld and others. B-splines possess a beautiful underlying theory that leads to a wide range of highly efficient algorithms, which is why they have become the standard representation in geometric design systems. B-spline surfaces do however have a severe shortcoming. They can only represent a limited set of surface topologies—namely, topological disks, cylinders, and tori.

The search for a representation that overcomes the topological limitations of B-splines began shortly after the introduction of B-splines to geometric design. Inspired by George Chaikin's work that described curves as the limit of a sequence of subdivision steps, two teams of researchers (Ed Catmull and Jim Clark, and Malcolm Sabin and D. Doo) independently developed the notion of subdivision surfaces in 1978.

Subdivision surfaces naturally admit arbitrary surface topologies. They were initially met with considerable enthusiasm, but since they lacked the beautiful and powerful theory of B-splines, they were treated as a rather exotic representation for nearly 20 years. That began to change in the mid '90s. Advances were made on a number of fronts by researchers in numerous disciplines, including approximation theory, numerical analysis, and computer graphics. These surfaces are rapidly establishing themselves as the new representation of choice, and I predict that in coming years they will largely supplant B-splines in many application domains. In feature film production here at Pixar for instance, subdivision surfaces are already the preferred way to represent the shape of virtually everything that moves.

The authors of this book have been on the forefront of these advances, and this book represents the first comprehensive description of subdivision methods. However, the book is not simply a collection of material that can be found in the open literature. Rather, it offers a deep and unique perspective into the world of subdivision. I learned a tremendous amount from reading this book. I'm sure you will too.

Contents

Preface

Subdivision is an exciting new area in computer graphics that allows a geometric shape to be modeled as the limit of a sequence of increasingly faceted polyhedra. Companies such as Pixar and Alias/Wavefront have made subdivision surfaces the basic building block for much of their computer graphics/modeling software. Assuming a background of basic calculus and linear algebra, this book is intend to provide a self-contained introduction to the theory and practice of subdivision. The book attempts to strike a balance between providing a complete description of the basic theory of subdivision (such as convergence/smoothness analysis) while considering practical issues associated with implementing subdivision (such as representing and manipulating polyhedral meshes). Its target audience consists of graduate and advanced undergraduate students of computer graphics, as well as practitioners.

Overview

The book consists of roughly three parts: The first part, consisting of Chapters 1–3, is introductory. Chapter 1 introduces subdivision as a method for unifying functional and fractal representations. Chapter 2 presents repeated integration, a simple technique for creating subdivision schemes, and it uses this method to construct two basic examples of subdivision schemes, B-splines and box splines. Chapter 3 considers the problem of analyzing the convergence and smoothness of a given uniform subdivision scheme. The second part of the book, Chapters 4–6, focuses on a new differential method for constructing subdivision schemes. This method, developed by the authors, allows for the construction of a much wider range of subdivision schemes than the integral method, and it provides a framework for systematically generating subdivision schemes on bounded domains. The last part of the book, Chapters 7–8, focus on the current "hot" topic in modeling: subdivision on polyhedral meshes. Chapter 7 introduces various subdivision schemes for polyhedral meshes, such as the Catmull-Clark scheme and Loop's scheme, and it considers the problem of implementing these schemes in detail.

Chapter 8 considers the problems of testing and proving the smoothness of polyhedral surface schemes.

www.subdivision.org

During the course of using the book, the reader will often notice a small chess piece either embedded in the text or placed on the left-hand side of the page. This icon is designed to alert the reader that there is more related material available at the Web site *www.subdivision.org*. This Web material is stored in a sequence of Mathematica 4.1 notebooks (one per chapter) and consists of items such as the Mathematica implementations used in generating the figures in this book. Almost all the computational concepts discussed in the book are implemented in these notebooks. We highly recommend that the interested reader download and explore these notebooks during the course of using this book. *www.subdivision.org* also contains an interactive tutorial on subdivision (using Java applets) that allows the user to subdivide both curves and surfaces. The Java implementation of this tutorial is also available for download and is intended to provide a starting point for someone interested in building modeling tools based on subdivision.

Acknowledgements

Many people helped in the writing of this book. We would specifically like to thank Malcolm Sabin, Luiz Velho, Tony DeRose, Igor Guskov, Brian Barsky, Ron Goldman, and Ju Tao for their help in reviewing the manuscript. Finally, Scott Schaeffer deserves an especially hearty thanks for his help in putting together the Web site *www.subdivision.org*.

Table of Symbols

Notation

The manuscript for this book (and its associated implementation) was prepared using Mathematica. Therefore, our general approach to notation is to use the StandardForm notation of Mathematica. The main advantage of this choice is that we avoid some of the inherent ambiguities of traditional mathematical notation. The highlights of this notation are the following:

- Function application is denoted using square brackets, for example, $p[x]$. This choice allow parentheses $()$ to be reserved for grouping expressions.
- Vectors (and lists) are created by enclosing their members in curly brackets, for example $\{a, b, c\}$, which denotes the vector whose entries consist of a, b and c. Conversely, the ith entry of a vector p is denoted by $p[\![i]\!]$. This choice allows us to define a sequence of vector p_k and then index a particular element of these vectors via $p_k[\![i]\!]$. (In Mathematica, vectors are always indexed starting at one; we allow the $[\![\,]\!]$ operator to also be used on vectors whose indices range over the integers.)
- The symbol $==$ denotes equality while the symbol $=$ denotes assignment. In traditional mathematics, this distinction is typically blurred; we attempt (to the best of our ability) to maintain this distinction.
- The expression $p^{(i,j)}[x, y]$ denotes the ith derivative with respect to x and jth derivative with respect to y of the function $p[x, y]$.

We also follow several important stylistic rules when assigning variable names. Roman variable are used to denote discrete quantities while script variables are used to denote continuous quantities. Often, the same variable may have both a Roman and a script version denoting the discrete and continuous aspects of the same quantity. For example, the vector p often denotes a vector of coefficients while the function $p[x]$ denotes the continuous limit function associated with this vector p. In the discrete case, we typically use lowercase letters to denote vectors, with uppercase letters being reserved for matrices. For example, the expression Sp denotes the product of the matrix S times the column vector p.

Roman

A affine transformation, generic matrix (Chapter 1)

$c[x]$, C circulant mask, circulant matrix (Chapter 8)

$d[x]$, $d_k[x]$ discrete difference mask, discrete divided difference mask (Chapter 4, 5)

$e[x]$, E, E_k energy mask, energy matrices (Chapter 6)

h index of sector for mesh at extraordinary vertex (Chapter 8)

i, j integer indices

k level of subdivision

$l[x, y]$, $l_k[x, y]$ discrete Laplacian masks (Chapter 5)

m order of spline

M, M_k topological mesh (Chapter 7, 8)

n size of grid, valence of extraordinary vertex

$n[x]$, N, N_k interpolation mask, interpolation matrices (Chapter 6)

p, p_k, q, q_k generic vectors of coefficients, generic control polygons/polyhedra

$r[x]$ various residual masks

S, S_k subdivision matrices

s, $s[x]$, $s^m[x]$, $s^\Sigma[x, y]$ subdivision mask, subdivision mask of order m, subdivision mask for box spline with direction vectors Σ (Chapter 2, 3, 4)

T difference matrix (Chapter 3)

U, U_k upsampling matrices (Chapter 6)

$u_k[x, y]$, $v_k[x, y]$ generating functions for discrete flow (Chapter 5)

v arbitrary vertex, extraordinary vertex

V_k, W_k function spaces, complementary spaces (Chapter 6)

w_k weight vector on grid $\frac{1}{2^k}\mathbb{Z}^2$ (Chapter 6)

$w[n]$ weight function at extraordinary vertex (Chapter 7, 8)

x, y variables for generating functions

z, z_i eigenvectors of subdivision matrix (Chapter 8)

Script

$b_i^m[\mathsf{x}], \mathcal{B}^m[\mathsf{x}]$ Bernstein basis function, vector of Bernstein basis functions (Chapter 1)

$c[x], c[x, y]$ truncated powers, cone splines (Chapter 2, 4)

$\mathcal{C}^k$ space of functions with k continuous derivatives

$\mathcal{D}[x]$ differential operator (Chapter 4, 5)

$\mathcal{E}[p]$ energy functional (Chapter 6)

$g[x, y]$ source of rotational flow (Chapter 5)

$\mathcal{H}^m$ m-dimensional hypercube (Chapter 2)

$\mathcal{I}[x]$ integral operator (Chapter 4)

$\mathcal{L}[x, y]$ continuous Laplacian operator (Chapter 5)

$n[x], n^m[x], n^\Sigma[x, y]$ scaling function for a subdivision scheme (Chapter 2, 4)

$\mathcal{N}[x], \mathcal{N}_k[x]$ vector of scaling functions for a subdivision scheme (Chapter 2, 4, 6)

$p[x], p[x, y], q[x], q[x, y]$ generic limit functions

$r_k[x]$ residual functions

$\mathcal{R}, \mathcal{R}^+, (\mathcal{R}^+)^m$ set of real numbers, set of non-negative real numbers, m-dimensional cone

s, t temporary continuous variables

$\{u[x, y], v[x, y]\}^T$ continuous flow in the plane (Chapter 5)

x, y, z continuous domain variables

Greek Letters

α, β, ϵ temporary real valued constants

$\delta[x], \delta[x, y]$ Dirac delta function (Chapter 4, 5)

γ tension parameter for exponential splines (Chapter 4)

λ_i eigenvalues of a subdivision matrix (Chapter 8)

σ_k, ρ_k tension parameters for non-stationary scheme (Chapter 4, 7)

ψ characteristic map (Chapter 8)

ϕ functional representation at an extraordinary vertex (Chapter 8)

Σ set of direction vectors (Chapter 2)

ω_n nth root of unity (Chapter 8)

Ω domain of integration (Chapter 6)

Miscellaneous

$\mathbb{e}$ exponential constant

$\mathbb{i}$ square root of -1

$\mathsf{g}, \mathsf{e}, \mathsf{v}, \mathsf{f}$ genus, number of edges, vertices, faces for a topological mesh (Chapter 7)

$\mathbb{D}$ topological dual of one-dimensional grid or a two-dimensional surface mesh (Chapter 5, 7)

$\mathbb{Z}, \mathbb{Z}^2, \mathbb{Z}^+, (\mathbb{Z}^+)^2$ set of integers, set of integer pairs, set of non-negative integers, set of non-negative integer pairs

Functions

$\mathrm{area}[p]$ area enclosed by a polygon p (Chapter 1)

$\mathrm{dim}[v]$ dimension of the smallest cell containing a vertex v in a topological mesh (Chapter 7)

$\mathrm{mod}[i, n]$ the integer i taken modulo n

$\mathrm{ring}[v]$ the set of faces containing a vertex v in a topological mesh (Chapter 7)

$\mathrm{val}[v]$ the number of faces in $\mathrm{ring}[v]$ (Chapter 7)

$\mathrm{vol}_m[A]$ the m-dimensional volume of a set A (Chapter 2)

CHAPTER 1

Subdivision: Functions as Fractals

Describing a complex geometric shape is a challenging task. Artists and designers may spend weeks and even months describing a single complex shape, such as that of a new car. To specify geometric shapes using a computer, one must first settle on a language for describing shape. Fortunately, there already exists a language that can describe a wide range of shapes: mathematics. The aim of this book is to expose the reader to subdivision, a new and exciting method for mathematically describing shape. Subdivision may be viewed as the synthesis of two previously distinct approaches to modeling shape: functions and fractals. This chapter reviews some of the basic ideas behind modeling shapes using both functions and fractals. It concludes with a brief introduction to the fundamentals of subdivision.

1.1 Functions

To describe the shape of a curve, a range of mathematical tools is available. Perhaps the simplest of these tools is the function. A *function* p maps an input parameter x to a function value $p[x]$. If p is plotted versus a range of parameter values x, the resulting graph defines a geometric shape. The crux here is to choose the function p so that the resulting graph has the desired shape. For example, if p is the trigonometric function Sin, the resulting graph is a wave-like curve (shown in Figure 1.1).

Obviously, this explicit approach restricts the resulting curve to be single valued with respect to the x axis. That is, for any x there is exactly one value $p[x]$. One way to remedy this problem is to define a curve *parametrically* using a pair of functions in a common parameter x. Each of these functions describes a separate coordinate of the curve. As x varies, this pair of functions traces out a curve in the coordinate plane

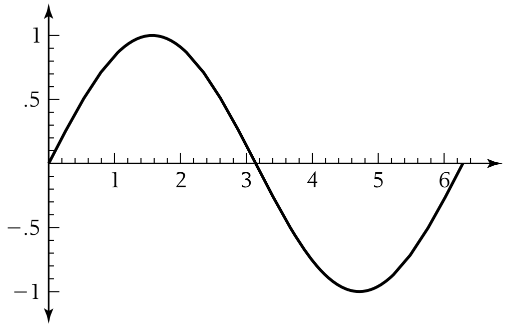


Figure 1.1 Graph of the function Sin[x] for $x \in [0, 2\pi]$.

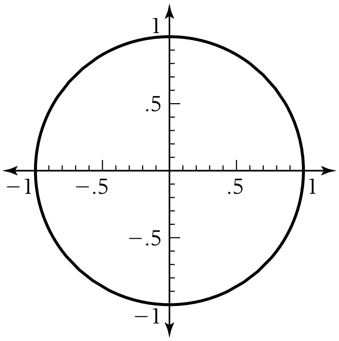


Figure 1.2 Plot of the parametric function {Cos[x], Sin[x]} for $x \in [0, 2\pi]$.

that describes the desired shape. For example, the parametric function {Cos[x], Sin[x]} defines the circle of Figure 1.2 as x varies over the range $[0, 2\pi]$.

Although parametric curves are extremely popular, they are by no means the only method available for defining curves. Another well-known method is to define a curve in the xy plane *implicitly* as the solution to the equation $p[x, y] == 0$. For example, the circle of Figure 1.2 can be represented implicitly as the solution to the equation $x^2 + y^2 - 1 == 0$. Figure 1.3 shows a darkened plot of this circle and its interior, $x^2 + y^2 - 1 \le 0$.

The difference in appearance of these two plots illustrates the fundamental difference between the two representations: The parametric curve {Cos[x], Sin[x]} describes a sequence of points lying on the circle itself. The contour plot of the

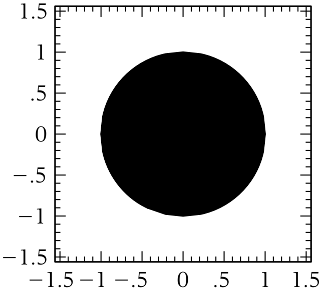


Figure 1.3 Plot of an implicitly defined circle and its interior, $x^2 + y^2 - 1 \leq 0$.

implicit region $x^2 + y^2 - 1 \leq 0$ shows two separate regions: the interior and the exterior of the circle. Each representation describes the circle precisely.

Choosing between these two representation methods is often difficult. There are applications for which one of the representations is better suited than the other. For example, a parametric representation can be evaluated at a sequence of parameter values to produce a sequence of points on a parametric curve. Generating points that lie exactly on an implicit curve is a bit more complex and usually involves finding the roots of a univariate equation. On the other hand, testing whether a point lies in the interior of a region bounded by an implicit curve reduces to evaluating $p[x, y]$ and checking its sign, which is quite easy. Answering the same question for a closed parametric curve is much more involved. Comparing and contrasting these two representations could be the subject of an entire textbook by itself. For those readers interested in exploring these questions more, we suggest the survey by Requicha and Voelker [131]. Our approach in this book will be to focus mainly on the explicitly defined functional case while occasionally examining the parametric case.

1.1.1 Piecewise Polynomials

No matter which representation one prefers, a question remains: What type of functions should be used in defining a shape p? The answer that has been most popular in computer-aided geometric design for the last few decades is to use piecewise polynomial functions. These functions describe a rich class of shapes, are computationally efficient, and have been subjected to rigorous numerical analysis. Unfortunately, controlling the shape of a piecewise polynomial function can be difficult. For example, consider the piecewise polynomial function $n[x]$ whose value has the form

$$n[x] = \begin{cases} 0 & \text{if } x \in [-\infty, 0], \\ \frac{x^3}{6} & \text{if } x \in [0, 1], \\ \frac{2}{3} - 2x + 2x^2 - \frac{x^3}{2} & \text{if } x \in [1, 2], \\ -\frac{22}{3} + 10x - 4x^2 + \frac{x^3}{2} & \text{if } x \in [2, 3], \\ -\frac{1}{6}(-4 + x)^3 & \text{if } x \in [3, 4], \\ 0 & \text{if } x \in [4, \infty]. \end{cases}$$

Surprisingly, these seemingly unrelated polynomials define a very nice smooth bump function, graphed in Figure 1.4. ($n[x]$ is the cubic B-spline basis function.)

Representing each of these polynomial functions in the monomial basis gives no clue about the shape of the graph of $n[x]$. Ideally, the definition of $n[x]$ as a piecewise polynomial should provide information about the shape of its graph. The careful choice of an appropriate basis for the piecewise polynomials provides the solution to this problem. The coefficients of this basis should give some geometric information about the shape of the graph of the resulting piecewise polynomial function. In particular, these coefficients should define a piecewise linear shape that approximates the associated function. This idea of relating a piecewise linear shape to an associated continuous shape is one of the fundamental paradigms of geometric design. As we shall see later, subdivision is one of the most successful instances of this paradigm.

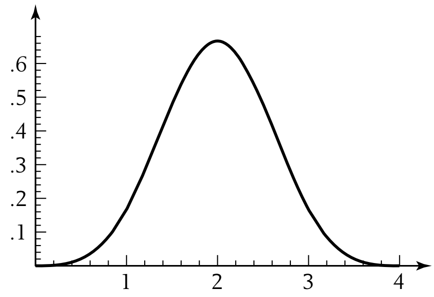


Figure 1.4 Plot of the piecewise cubic polynomial $n[x]$.

1.1.2 **Bézier Curves**

Perhaps the simplest example of a functional representation that embodies this paradigm is that of Bernstein polynomials and Bézier curves. Our goal in this section is to give a brief introduction to Bézier curves in preparation for discussing subdivision for these curves later in the chapter. For those readers interested in a more complete exploration of the properties of Bézier curves, the authors suggest Farin [59] or Bartels et al. [9].

Recall that our goal is to find a basis whose coefficients provide geometric intuition concerning the shape of the function defined by these coefficients. To be specific, let us first consider the problem of representing a single polynomial function $p[x]$ of degree m over the unit interval $[0, 1]$. For polynomials, the key idea is to construct a set of $\mathsf{m}+1$ basis functions $b_i^m[x]$ of degree m where $0 \le i \le m$. Then, every polynomial $p[x]$ of degree m can be represented as a unique linear combination of these basis functions:

$$p[x] = \sum_{i=0}^{m} \mathsf{p}[\![i]\!] b_i^m[x]. \tag{1.1}$$

Here, the $\mathsf{m}+1$ coefficients are represented as a column vector p whose ith coefficient is denoted by $\mathsf{p}[\![i]\!]$. For the sake of notational compactness, equation 1.1 can also be in vector form as $p[x] = \mathcal{B}^m[x]\,\mathsf{p}$, where $\mathcal{B}^m[x]$ is a row vector consisting of the $\mathsf{m}+1$ basis function $b_i^m[x]$.

If the basis functions $b_i^m[x]$ are chosen to be the *Bernstein basis* functions, the coefficients of p provide strong geometric intuition about the behavior of $p[x]$ on the interval $[0, 1]$. The Bernstein basis functions can be defined in a variety of manners. One simple construction is to consider the binomial expansion of the expression $((1-x)+x)^m$. The binomial expansion consists of $\mathsf{m}+1$ terms of degree m in x. For example,

$$((1-x)+x)^3 == (1-x)^3 + 3(1-x)^2 x + 3(1-x)x^2 + x^3.$$

The resulting terms of this expansion are the Bernstein basis functions of degree m. Applying the binomial theorem yields an explicit expression for the ith Bernstein basis function of degree m:

$$b_i^m[x] = \binom{m}{i} (1-x)^{m-i} x^i. \tag{1.2}$$

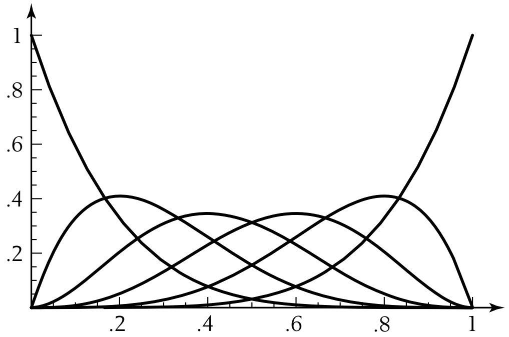


Figure 1.5 Plots of the six quintic Bernstein basis functions $b_i^5[x]$.

Figure 1.5 shows a plot of the six Bernstein basis functions $b_i^5[x]$ of degree five. These basis functions have a wide range of remarkable properties. These properties include:

- The functions $b_i^m[x]$ are non-negative polynomials of degree m on the interval [0, 1].
- $\sum_{i=0}^{m} b_i^m[x] == 1$ due to the definition of the $b_i^m[x]$ in terms of the binomial expansion.
- $b_i^m[x]$ is a bell-shaped function that reaches its maximum at $x == \frac{i}{m}$. (This property can be verified by noting that the derivative of $b_i^m[x]$ is zero at $x == \frac{i}{m}$.)

If the entries of the vector p are points (instead of scalar coefficients), $p[x]$ is a parametric curve known as a *Bézier curve*. The open polygon defined by the $m + 1$ *control points* of p is the *control polygon* associated with the curve $p[x]$. Observe that the definition of equation 1.1 involves only weighted combinations of the control points of the form $\sum_i \alpha_i \, p[\![i]\!]$. Because the Bernstein basis functions $b_i^m[x]$ sum to one, these weights α_i always sum to one (i.e., $\sum_i \alpha_i == 1$). Taking such an *affine combination* of a set of points is a frequent operation in geometric design; restricting the sum of the weights to be one guarantees that the sum $\sum_i \alpha_i \, p[\![i]\!]$ is a well-defined point whose location is independent of the particular coordinate system used in representing the points $p[\![i]\!]$. (Technically, the construction is affinely invariant.)

Due to the properties of the Bernstein polynomials, the Bézier curve $p[x]$ follows the control polygon p in a very natural manner as x varies from 0 to 1. For example,

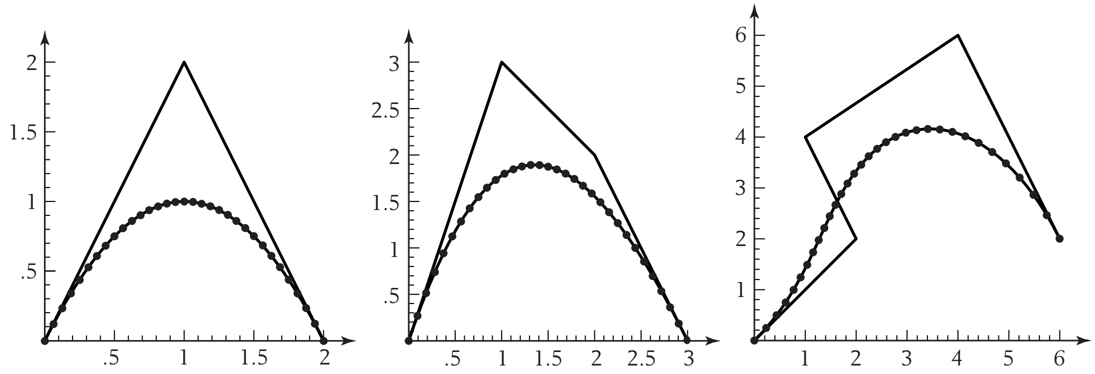


Figure 1.6 Several Bézier curves and their control polygons.

$p[x]$ interpolates the endpoints of the polygon p (i.e., $p[0] == \mathrm{p}[\![0]\!]$ and $p[1] == \mathrm{p}[\![\mathrm{m}]\!]$). Because the Bernstein basis functions $b_i^m[x]$ are non-negative on the interval $[0, 1]$, $p[x]$ is an affine combination of the points $\mathrm{p}[\![\mathrm{i}]\!]$, where the weights are always non-negative. Such a combination is called a *convex combination* because the resulting point always lies inside the convex hull of the points $\mathrm{p}[\![\mathrm{i}]\!]$. Figure 1.6 shows several control polygons and their corresponding Bézier curves $p[x]$.

This method for defining parametric curves and using them to design shapes was discovered independently by two researchers for rival French automakers in the early 1960s. One researcher, P. Bézier of Renault, was able to publish his ideas; the resulting curves became known as Bézier curves. The other researcher, P. de Casteljau of Citroën, was not allowed to publish his ideas, because they were viewed as trade secrets by Citroën. It was not until 1972 that the link between these two independent pieces of work was uncovered [76]. (Do not feel too sorry for de Casteljau; the subdivision algorithm for Bézier curves is known as the de Casteljau algorithm in honor of his contributions.)

Due to its many nice properties, a designer can control the shape of the Bézier curve $p[x]$ simply by manipulating the polygon p. Ideally, this manipulation of Bézier curves should be done interactively. The designer modifies p and the modeling system instantly updates $p[x]$. To achieve this goal, the system should produce some type of dense polygonal approximation to $p[x]$ that can be rendered easily on the screen. The simplest approach is to evaluate $p[x]$ at a sequence of $\mathrm{n} + 1$ values $x = \{0, \frac{1}{\mathrm{n}}, \frac{2}{\mathrm{n}}, \ldots, 1\}$ and to render the polygon $\{p[0], p[\frac{1}{\mathrm{n}}], p[\frac{2}{\mathrm{n}}], \ldots, p[1]\}$. As $\mathrm{n} \to \infty$, this discrete approximation converges to the true Bézier curve $p[x]$. Figure 1.6 uses such a discretization for $\mathrm{n} = 32$.

The rendering method previously described is reasonable but has some flaws. Evaluating $p[x]$ repeatedly to compute $\{p[0], p[\frac{1}{n}], p[\frac{2}{n}], \ldots, p[1]\}$ is expensive. Much more efficient methods for repeated evaluation, such as forward differencing [2, 9], exist. (Note that forward differencing is not always numerically stable.) Another drawback is the need to determine n prior to the discretization. If n proves to be too small (i.e., the discrete approximation does not follow $p[x]$ closely enough), $p[x]$ must be reevaluated using a larger n. One simple trick, which avoids this total recomputation, is to replace n by $2n$. To compute $\{p[0], p[\frac{1}{2n}], p[\frac{2}{2n}], \ldots, p[1]\}$, one need only compute the value of $p[x]$ at $x = \frac{1}{2n}, \frac{3}{2n}, \frac{5}{2n}, \ldots, \frac{2n-1}{2n}$. The remaining values have already been computed earlier. Because this algorithm determines discretizations of $p[x]$ of increasingly fine resolution, this procedure is an example of a *multiresolution* algorithm. Figure 1.7 shows an example of this approach applied to a cubic Bézier curve for $n = 4, 8, 16, 32$.

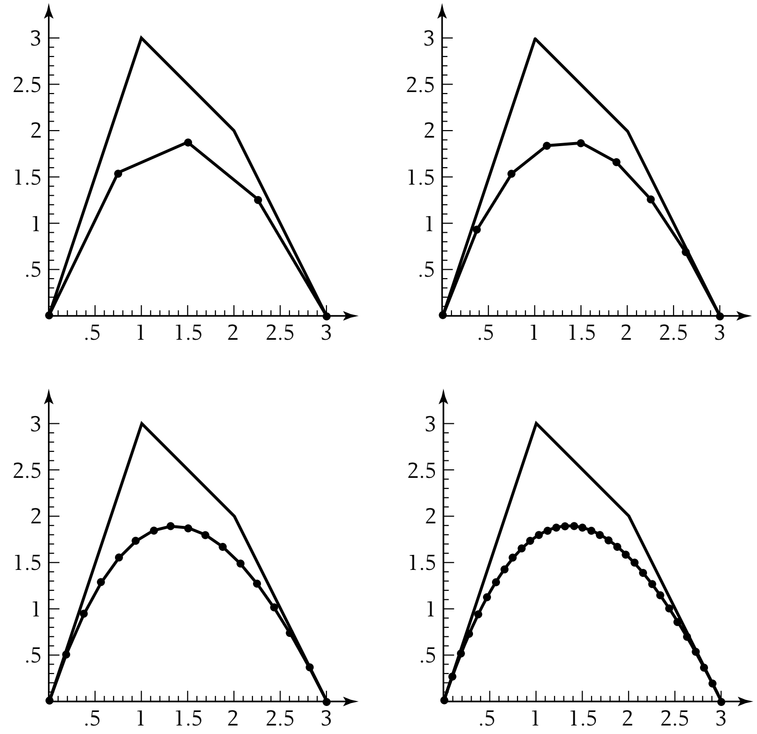


Figure 1.7 A multiresolution algorithm for drawing a cubic Bézier curve.

1.2 Fractals

The plotting algorithm described at the end of the previous section has an important property: it produces increasingly detailed plots of a function by doubling the resolution of the plot at each step. Eventually, the plots converge to the exact Bézier curve. This method is a simple example of a multiresolution algorithm. Starting with a coarse shape, multiresolution methods build increasingly finer shapes that eventually converge to some limit shape. Ideally, successive shapes are related by a simple transformation. In the case of Bézier curves, the actual transformation relating two consecutive approximations is not quite clear. As described in the previous section, each approximation was actually computed directly from the coarsest starting shape. The true multiresolution nature of Bézier curves will become clearer later in this section.

This section considers another interesting example of multiresolution shapes: *fractals*. Conceptually, building a fractal requires two ingredients: a space of geometric shapes p_k and a *transformation* S that maps a coarse shape p_{k-1} to a fine shape p_k. Given an initial shape p_0, a fractal method defines an infinite sequence of shapes by iterating S according to $p_k = Sp_{k-1}$. (The expression Sp_{k-1} denotes the application of S to p_{k-1}.) If the transformation S is chosen appropriately, the limit of p_k as $k \to \infty$ is a well-defined shape p_∞ satisfying the fixed-point property $p_\infty = Sp_\infty$. Most of the material on constructing fractals using this approach is drawn from Barnsley [8], an excellent introduction to the area of fractals.

1.2.1 Iterated Affine Transformations

One of the simplest methods for constructing fractals involves iterating a set of affine transformations. Before attempting to construct fractals using this method, we briefly review some of the properties of affine transformations and discuss an elegant method for representing affine transformations. Mathematically, an *affine transformation* is a function A that maps points to points and satisfies the relation

$$A\left[\sum_i \alpha_i \, p[\![i]\!]\right] = \sum_i \alpha_i \, A[p[\![i]\!]] \tag{1.3}$$

for all vectors of points p provided that $\sum_i \alpha_i == 1$. For points in the plane, some standard examples of affine transformations include rigid-body motions, such as rotations and translations, as well as scaling and shearing. (See Mortenson [108] for more information on affine transformations.)

One interesting feature of affine transformations in the plane is that once the value of an affine transformation A is known at three noncollinear points $p[\![0]\!]$, $p[\![1]\!]$ and $p[\![2]\!]$ the action of A on an arbitrary point v can be computed using equation 1.3. The trick is to express the point v as an affine combination of the points $p[\![0]\!]$, $p[\![1]\!]$, and $p[\![2]\!]$; that is, $v == \alpha_0\, p[\![0]\!] + \alpha_1\, p[\![1]\!] + \alpha_2\, p[\![2]\!]$, where $\alpha_0 + \alpha_1 + \alpha_2 == 1$. Given these weights α_i, the value $A[v]$ of the affine transformation A satisfies

$$A[v] = \alpha_0\, A[p[\![0]\!]] + \alpha_1\, A[p[\![1]\!]] + \alpha_2\, A[p[\![2]\!]]$$

due to equation 1.3. These affine weights are the *barycentric coordinates* of v with respect to the base triangle p and can be computed as the ratio of the areas of various triangles defined by v and the vertices of p,

$$\{\alpha_0, \alpha_1, \alpha_2\} = \frac{\{\mathrm{area}[v,\ p[\![1]\!],\ p[\![2]\!]],\ \mathrm{area}[p[\![0]\!],\ v,\ p[\![2]\!]],\ \mathrm{area}[p[\![0]\!],\ p[\![1]\!],\ v]\}}{\mathrm{area}[p[\![0]\!],\ p[\![1]\!],\ p[\![2]\!]]}. \tag{1.4}$$

Given this observation, all that remains is to specify the behavior of A at the vertices of p. One simple solution is to express the position of points $A[p[\![i]\!]]$ in terms of their barycentric coordinates with respect to the triangle p. These three sets of coordinates can be collected into a 3×3 matrix S whose rows are the barycentric coordinates for $A[p[\![0]\!]]$, $A[p[\![1]\!]]$, and $A[p[\![2]\!]]$ with respect to the triangle p. Given a point v with barycentric coordinates $\{\alpha_0, \alpha_1, \alpha_2\}$ with respect to the triangle p, the barycentric coordinates of $A[v]$ with respect to points $A[p[\![0]\!]]$, $A[p[\![1]\!]]$, and $A[p[\![2]\!]]$ are the vector/matrix product $\{\alpha_0, \alpha_1, \alpha_2\}S$.

To illustrate this method, we next construct the matrix S that represents the affine transformation mapping a base triangle p to an image triangle q. These two triangles themselves are represented as a column vector consisting of three points in the plane; that is, a 3×2 matrix, where each row contains the Cartesian coordinates of the vertices of the triangle. For our example, we let p and q be the triangles

$$p = \begin{pmatrix} 0 & 0 \\ 2 & 0 \\ 1 & 1 \end{pmatrix}, \quad q = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix}.$$

As observed, the rows of the matrix S are the barycentric coordinates of the vertices of q with respect to the base triangle p. Because $q[\![0]\!] == p[\![0]\!]$, the first row of S is simply $\{1, 0, 0\}$. Similarly, $q[\![1]\!]$ is the midpoint of the edge from $p[\![0]\!]$ to $p[\![1]\!]$. Therefore, the second row of S is $\{\frac{1}{2}, \frac{1}{2}, 0\}$. Finally, $q[\![2]\!]$ is the midpoint of the edge from p⟦0⟧ to p⟦2⟧. Therefore, the third and final row of S is $\{\frac{1}{2}, 0, \frac{1}{2}\}$. Thus, the matrix S has the form

$$S = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \end{pmatrix}.$$

Observe that the matrix S satisfies $q = Sp$. One quick way to compute S from p and q is as follows: extend both p and q to form 3 × 3 matrices, $\widetilde{p}$ and $\widetilde{q}$, by adding an extra column of 1s (i.e., converting the coordinates of p and q to their homogeneous form). Because the rows of S sum to one, these extended matrices satisfy $S\widetilde{p} = \widetilde{q}$. As long as p is a nondegenerate triangle, the matrix $\widetilde{p}$ is invertible and S is exactly $\widetilde{q}\widetilde{p}^{-1}$. In terms of our previous example,

$$\begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ \frac{1}{2} & \frac{1}{2} & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 \\ 2 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}^{-1}.$$

Observe that the matrix S actually encodes a relative transformation with respect to the base triangle p. Choosing a different base triangle defines a different transformation. This particular method of representing affine transformations is advantageous in that many of the constructions for fractals have the form "Take a shape p and transform it as follows." The matrix S encapsulates this relative transformation without encoding any particular information about the base triangle p. For example, the action of the matrix S shown previously is to shrink the triangle p by a factor of $\frac{1}{2}$ and reposition the image triangle q so that it continues to touch the first vertex of p. Figure 1.8 shows the result of applying the affine transformation associated with S to two different base triangles p (outlined). The corresponding triangles $q = Sp$ are black.

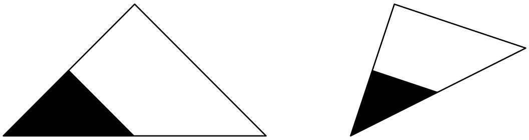


Figure 1.8 Application of the transformation S to the large outlined triangles yields the smaller solid triangles.

Given an affine transformation encoded by a matrix S, we can now consider the question of iterating this transformation via $p_k = Sp_{k-1}$. In particular, if p_0 is some base triangle, we are interested in the behavior of the triangles p_k as $k \to \infty$. To answer this question, we note that an affine transformation A is *contractive* if there exists a constant $0 \leq \beta < 1$ such that

$$\text{dist}[A[u], A[v]] \leq \beta * \text{dist}[u, v]$$

for all points u, v in the plane. ($\text{dist}[u, v]$ is the Euclidean distance between the points u and v.) If the affine transformation associated with S is contractive, the diameter of the triangles p_k decreases after each application of S and therefore the limit of this process is a point.

Of course, fractals consisting of a single point are not particularly interesting. However, if we consider a set of contractive affine transformations, iterating the composite transformation consisting of the union of these transformations yields a much more interesting limit shape. If S is a set of matrices associated with these affine transformations and p is a set of triangles, we let Sp denote the new set of triangles representing all possible combinations of applying any one transformation from S to one triangle from p. Obviously, the size of the resulting set is the product of the size of the set of transformations and the size of the set of input triangles. In this framework, we can now define the fractal process, mapping sets of triangles p_{k-1} to sets of triangles p_k, as

$$p_k = Sp_{k-1}.$$

If each of the affine transformations in S is contractive, the limit of this process is a fixed set of points p_∞ whose shape depends on S and the base triangle p_0. Instead of attempting to characterize and prove this convergence, we refer the interested reader to Chapters 2 and 3 of Barnsley [8]. Our agenda for the remainder of this section is to consider some intriguing examples of this fractal process and to show that the Bézier curves introduced in section 1.1 can be represented as the limit of this process.

1.2.2 The Sierpinski Triangle

Our first example of a fractal process is the *Sierpinski triangle*. Conceptually, the set of iterated affine transformations describing the fractal process for the Sierpinski

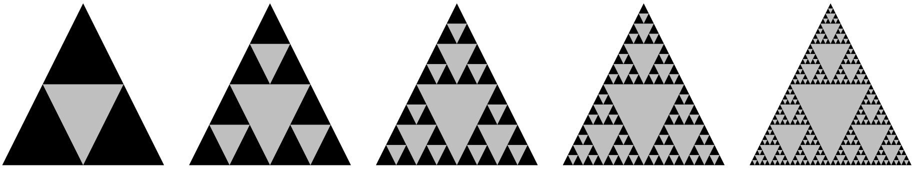


Figure 1.9 Progressive approximations to the Sierpinski triangle.

triangle maps a base triangle into three children by splitting each edge of the base triangle in half. The set S of transformations consists of the three matrices

$$\mathsf{S} = \left\{ \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 \\ \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix} \right\}.$$

Because each of these transformations is contractive (with $\beta == \frac{1}{2}$), there exists a fractal, the Sierpinski triangle, associated with the limit of the iteration $\mathsf{p}_k = \mathsf{S}\mathsf{p}_{k-1}$. To draw the Sierpinski fractal, we repeatedly apply the set of affine transformations S to a single base triangle p_0 until the size of the triangles in the set p_k falls below the desired resolution. Figure 1.9 shows the result of applying this process five times to the gray base triangle p_0.

Based on the figure, we observe that the Sierpinski triangle has a very curious property: the final shape consists of three copies of itself, placed in the three corners of the original triangle. In fact, the three shapes are exactly the result of applying the three affine transformations from S to the Sierpinski triangle itself. This property follows from the fact that the limit shape p_∞ is a fixed point of the transformation S (i.e., $\mathsf{p}_\infty = \mathsf{S}\mathsf{p}_\infty$). This self-similarity relation (often referred to as the *collage* property) is characteristic of fractals defined by iterated affine transformations.

1.2.3 The Koch Snowflake

A second example of this fractal process is the *Koch snowflake*. Here, the set of affine transformations S has the form

$$\mathsf{S} = \left\{ \begin{pmatrix} 1 & 0 & 0 \\ \frac{2}{3} & \frac{1}{3} & 0 \\ \frac{2}{3} & 0 & \frac{1}{3} \end{pmatrix}, \begin{pmatrix} \frac{2}{3} & 0 & \frac{1}{3} \\ \frac{2}{3} & \frac{1}{3} & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{pmatrix}, \begin{pmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & \frac{1}{3} & \frac{2}{3} \\ \frac{1}{3} & 0 & \frac{2}{3} \end{pmatrix}, \begin{pmatrix} \frac{1}{3} & 0 & \frac{2}{3} \\ 0 & \frac{1}{3} & \frac{2}{3} \\ 0 & 0 & 1 \end{pmatrix} \right\}.$$

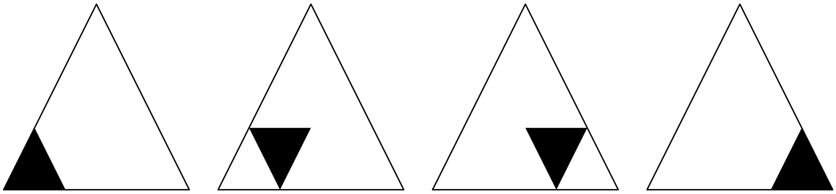


Figure 1.10 The four affine transformations used in defining the Koch snowflake.

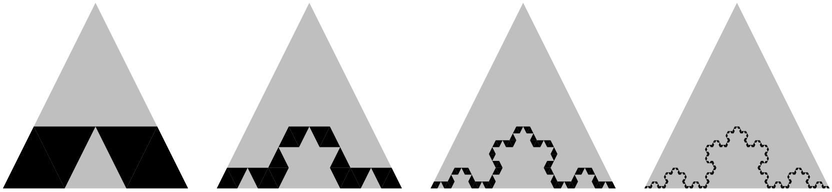


Figure 1.11 Progression approximations of the Koch curve.

Each of these four transformations maps the base triangle to one of the four image triangles shown in Figure 1.10. The two transformations on the left map the upper vertex of the base triangle to a point two-thirds of the way down the upper left edge of the base triangle. The transformations on the right map the upper vertex of the base triangle to a point two-thirds of the way down the upper right edge of the base triangle.

Because each of these affine transformations is contractive (with $\beta == \frac{1}{3}$), there exists a fractal limit to the process $p_k = Sp_{k-1}$. As before, the set of transformations is repeatedly applied to the base triangle p_0 until the size of the children reaches a desired resolution. Figure 1.11 depicts the four applications of the transformation to the gray parent triangle p_0. Again, we observe that the fractal limit is a collage of four affine copies of itself, placed along the base of the triangle p_0 according to the four transformations in S.

In the previous example, we started the fractal rendering process with just one base triangle. A neat trick can be applied to close the Koch curve into an actual snowflake. We can apply one round of the fractal process for the Sierpinski triangle to a single, initial triangle. The result is three smaller triangles we then use as the initial geometry for the fractal process associated with the Koch curve. Figure 1.12 shows the three triangles resulting from one round of the fractal process for the Sierpinski triangle. The result of applying the Koch transformations to these three

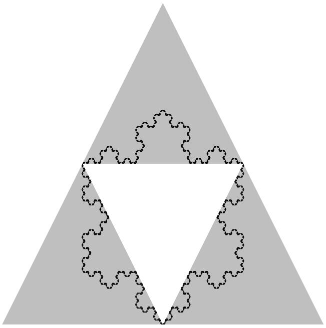


Figure 1.12 The Koch snowflake.

triangles repeatedly is shown in black in Figure 1.12 and resembles the outline of a snowflake.

1.2.4 **Bézier Curves**

As a final example, we develop a fractal process for the class of shapes first encountered in section 1.1, Bézier curves. Recall that a Bézier curve $p[x]$ was defined in equations 1.1 and 1.2 as the convex combination of a set of points p. The weights used for this combination were the values of the Bernstein basis functions $b_i^m[x]$. Our goal now is to construct two affine transformations that form a collage for this Bézier curve. These affine transformations can be derived from the subdivision algorithm for Bézier curves, the *de Casteljau algorithm*. This subdivision algorithm splits the control polygon for an entire Bézier curve $p[x]$ into two control polygons: one for the first half of the curve, corresponding to $x \in [0, \frac{1}{2}]$, and one for the second half of the curve, corresponding to $x \in [\frac{1}{2}, 1]$.

Given a vector of points p, this algorithm recursively computes a related triangular array of points $\widetilde{p}$. In the base case, the base of the triangular array $\widetilde{p}[\![i, i]\!]$ is initialized to $p[\![i]\!]$ for $0 \leq i \leq m$. For $0 \leq i < j \leq m$, the points $\widetilde{p}[\![i, j]\!]$ are defined by the recurrence

$$\widetilde{p}[\![i, j]\!] = \frac{1}{2}\widetilde{p}[\![i, j-1]\!] + \frac{1}{2}\widetilde{p}[\![i+1, j]\!]. \tag{1.5}$$

For $m == 3$, this recurrence has the tabular form shown in Figure 1.13. The beauty of this algorithm is that the control points for the two new control polygons lie on the lateral edges of Figure 1.13. In particular, the control polygon for the

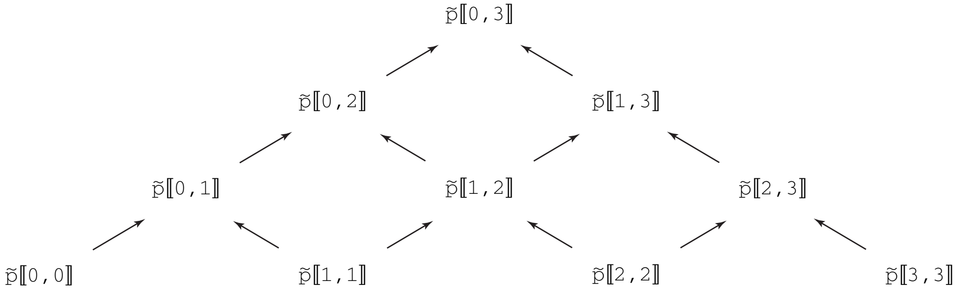


Figure 1.13 The de Casteljau algorithm for cubic Bézier curves.

restriction of $p[x]$ to the interval $[0, \frac{1}{2}]$ is $\{\widetilde{\mathrm{p}}\llbracket 0, 0\rrbracket, \widetilde{\mathrm{p}}\llbracket 0, 1\rrbracket, \ldots, \widetilde{\mathrm{p}}\llbracket 0, \mathrm{m}\rrbracket\}$. Likewise, the control polygon for the restriction of $p[x]$ to the interval $[\frac{1}{2}, 1]$ is $\{\widetilde{\mathrm{p}}\llbracket 0, \mathrm{m}\rrbracket, \widetilde{\mathrm{p}}\llbracket 1, \mathrm{m}\rrbracket, \ldots, \widetilde{\mathrm{p}}\llbracket \mathrm{m}, \mathrm{m}\rrbracket\}$. A powerful tool for understanding the behavior (and correctness) of the de Casteljau algorithm is ***blossoming***, credited to Ramshaw [127]. Seidel [140] gives a nice introduction to this topic.

Based on the de Casteljau algorithm, we can express Bézier curves as iterated affine transformations. For quadratic Bézier curves (i.e., $\mathrm{m} == 2$), one affine transformation maps the polygon p to the polygon $\{\widetilde{\mathrm{p}}\llbracket 0, 0\rrbracket, \widetilde{\mathrm{p}}\llbracket 0, 1\rrbracket, \widetilde{\mathrm{p}}\llbracket 0, 2\rrbracket\}$. The second affine transformation maps the polygon p to the polygon $\{\widetilde{\mathrm{p}}\llbracket 0, 2\rrbracket, \widetilde{\mathrm{p}}\llbracket 1, 2\rrbracket, \widetilde{\mathrm{p}}\llbracket 2, 2\rrbracket\}$. Based on equation 1.5, the two 3×3 matrices defining these transformations have the form

$$\mathrm{S} = \left\{ \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{pmatrix}, \begin{pmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 1 \end{pmatrix} \right\}.$$

One application of this set of transformations to an initial triangle yields two finer triangles built from the midpoints of the legs of the base triangle, as well as the center of these two midpoints. Figure 1.14 shows this construction applied to the gray parent triangle.

Because both of these transformations are contractive (with $\beta = \frac{3}{4}$), iteratively applying this pair of affine transformations leads to an elegant multiresolution algorithm for rendering Bézier curves. The visual appearance of the plots produced by this method can be improved by drawing only the line segment connecting the first and last vertex of each triangle in p_k. Due to the structure of the two affine transformations, the result of n iterations of $\mathrm{p}_k = \mathrm{S}\mathrm{p}_{k-1}$ is a continuous sequence of

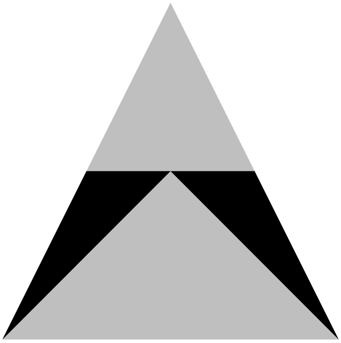


Figure 1.14 One round of an iterated affine transformation based on the quadratic de Casteljau algorithm.

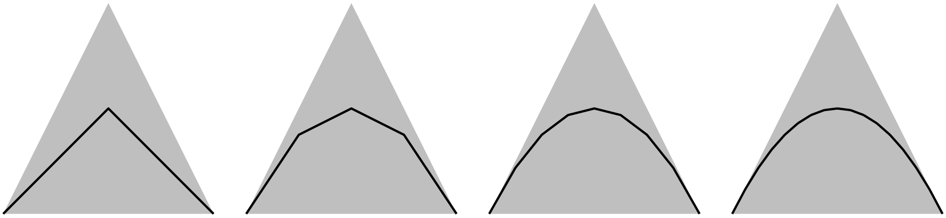


Figure 1.15 Four rounds of an iterated affine transformation converging to a quadratic Bézier curve.

2^n line segments whose $2^n + 1$ vertices lie on the Bézier curve. Figure 1.15 shows the progression of this rendering algorithm for a quadratic Bézier curve.

Higher-degree Bézier curves can also be constructed as the limit of iterated affine transformations. To generate a Bézier curve of degree m, the set of transformations S consists of a pair of $(m+1) \times (m+1)$ matrices defined via the de Casteljau algorithm. These matrices can be viewed as defining a pair of m-dimensional affine transformations. In the case of planar Bézier curves, these m-dimensional transformations are then applied to a degenerate m-dimensional simplex (i.e., an $(m+1)$-gon) that lies entirely in the plane. For example, cubic Bézier curves have a pair of associated transformation matrices of the form

$$S = \left\{ \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & 0 \\ \frac{1}{8} & \frac{3}{8} & \frac{3}{8} & \frac{1}{8} \end{pmatrix}, \begin{pmatrix} \frac{1}{8} & \frac{3}{8} & \frac{3}{8} & \frac{1}{8} \\ 0 & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix} \right\}.$$

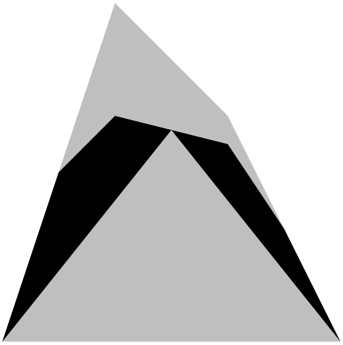


Figure 1.16 One round of an iterated affine transformation based on the cubic de Casteljau algorithm.

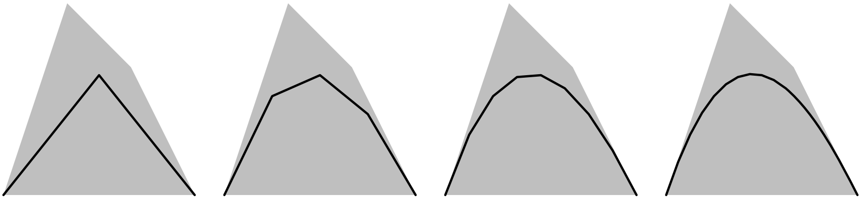


Figure 1.17 Four rounds of an iterated affine transformation converging to a cubic Bézier curve.

Figure 1.16 shows the result of applying this transformation to an initial quadrilateral (a degenerate tetrahedron). Note that the vertices of the black quadrilaterals lie at the various combinations of midpoints of the gray quadrilateral. Given the gray base quadrilateral p_0, iterating $p_k = Sp_{k-1}$ yields a continuous sequence of 2^k line segments that approximates the associated cubic Bézier curve. Figure 1.17 shows several rounds of these transformations applied to the cubic control polygon of Figure 1.7. For those readers interested in more information on extensions of this approach to curve subdivision, we recommend two papers by Micchelli and Prautzsch [104, 105].

1.3 Subdivision

The first part of this chapter described two popular techniques for representing complex shapes: explicit functions and fractal procedures. When describing shapes with functions, piecewise polynomials are a particularly common and practical choice. Piecewise polynomial functions provide a convenient and efficient method of defining smooth curved shapes. In addition, the mathematics of the resulting geometry is well understood. On the other hand, iterated affine transformations define a richer class of shapes using the very simple concept of iterative transformations. However, analyzing the behavior of the final limit shapes can be extremely challenging.

Subdivision captures the best of both methods. Using subdivision, a curve is defined as the limit of an increasingly faceted sequence of polygons. Each polygon is related to a successor by a simple linear transformation that provides an increasingly accurate approximation to the final limit curve. In most applications, a sufficiently dense polygon can be used in place of the final limit curve without any noticeable error.

Remarkably, many of the traditional shapes used in geometric design can be defined using subdivision. As the previous section illustrated, Bézier curves have a very simple subdivision method, the de Casteljau algorithm. The solutions to many variational problems and partial differential equations can also be expressed using subdivision. Moreover, many new shapes useful in geometric design are defined only in terms of subdivision. The rest of this section introduces the basic mechanics of subdivision in terms of a simple example: piecewise linear splines.

1.3.1 Piecewise Linear Splines

Our approach is to introduce piecewise linear splines in terms of their polynomial definition and later propose a functional analog of the collage property. This property leads to a simple multiresolution rendering method for piecewise linear splines. Consider the piecewise linear function of the form

$$n[x] = c[x+1] - 2c[x] + c[x-1],$$

where $c[x]$ is the truncated linear function that is x if $x \geq 0$ and zero if $x < 0$. For $x \notin [-1, 1]$, the function $n[x]$ is identically zero. At $x == 0$, $n[x]$ is exactly one. Elsewhere, $n[x]$ varies linearly. Therefore, $n[x]$ is the *unit hat function*. Figure 1.18 shows plots of $c[x]$ and $n[x]$, respectively.

Replacing the parameter x in the function $n[x]$ by $x - \mathrm{i}$ has the effect of translating the unit hat function i units to the right. Figure 1.19 plots the functions $n[x - \mathrm{i}]$ over the parameter range $x \in [-2, 2]$ for various integer values of i. Now, consider

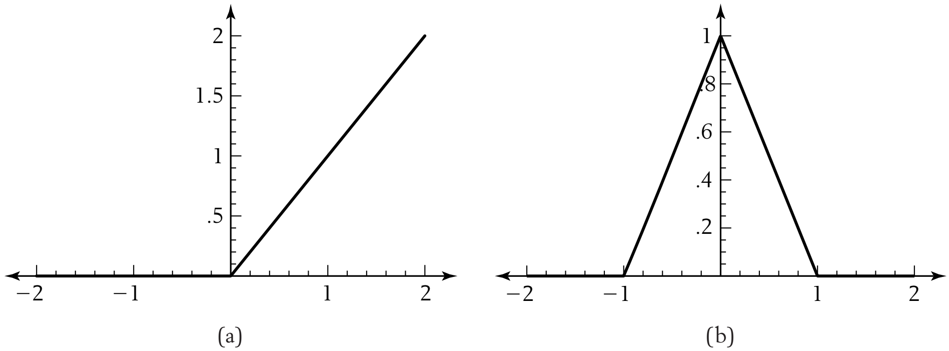


Figure 1.18 The functions $c[x]$ (a) and $n[x]$ (b).

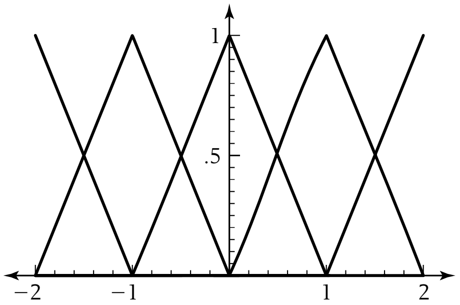


Figure 1.19 Several translates of the unit hat function.

a new function $p[x]$ constructed as a linear combination of translated hat functions, that is, $p[x]$ having the form

$$p[x] = \sum_{\mathrm{i}=-\infty}^{\infty} \mathrm{p}[\![\mathrm{i}]\!]\, n[x - \mathrm{i}], \tag{1.6}$$

where $\mathrm{p}[\![\mathrm{i}]\!]$ is the ith element of a column vector of coefficients p. Again, this expression can be written in vector form as $p[x] = \mathcal{N}[x]\mathrm{p}$, where $\mathcal{N}[x]$ is a row vector whose ith entry is $n[x - \mathrm{i}]$.

Due to the piecewise definition of $n[x]$, the breakpoints (called *knots*) between the linear pieces of the function $p[x]$ lie at the integers $\mathbb{Z}$. Due to the uniform spacing of these knots, $p[x]$ is a *uniform* spline. For uniform splines, the function $n[x]$ is often referred to as the *scaling function* associated with the spline. If the translates of the scaling function $n[x - \mathrm{i}]$ are linearly independent, these scaling functions are *basis functions* for the space of splines in equation 1.6. Much of the first part of this book considers the problem of subdivision for uniform splines.

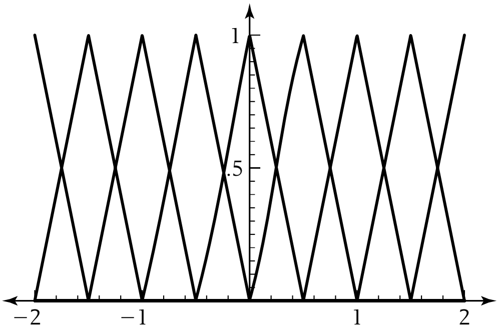


Figure 1.20 Several translates of dilates of the unit hat function.

To develop a multiresolution representation for linear splines, we next consider the relationship between splines with knots at the integers $\mathbb{Z}$ and splines with knots at the half-integers $\frac{1}{2}\mathbb{Z}$. Note that replacing the parameter x by $2x$ has the effect of *dilating* (i.e., expanding) the coordinate axis. Applying this transformation to the function $n[x]$ yields a new hat function $n[2x]$ that has been compressed by a factor of two and that is supported on the interval $[-\frac{1}{2}, \frac{1}{2}]$. Figure 1.20 displays translates of dilated scaling functions of the form $n[2x - \mathrm{i}]$, plotted over the range $x \in [-2, 2]$. Note that there are twice as many scaling functions as before. Moreover, these dilated scaling functions are centered over the half-integers $\frac{1}{2}\mathbb{Z}$, whereas the original scaling functions are centered over the integers $\mathbb{Z}$. Finally, the range in which a scaling function $n[2x - \mathrm{i}]$ is non-zero is just half as wide as for the original scaling functions $n[x - \mathrm{i}]$. These translates $n[2x - \mathrm{i}]$ of the dilated hat function form a basis for the space of piecewise linear functions with knots at the half-integers $\frac{1}{2}\mathbb{Z}$.

In the previous section on fractals, the collage property stated roughly that a fractal could be expressed as the union of a collection of affine copies of itself. In our current functional setting, the analog of this property is to express the hat functions $n[x]$ as linear combination of translated dilates $n[2x - \mathrm{i}]$. Recalling our definition of $n[x]$ in terms of $c[x]$ and the fact that $c[2\mathrm{x}] = 2c[\mathrm{x}]$, it is easy to verify that the hat function $n[x]$ satisfies the *refinement relation*

$$n[x] == \frac{1}{2}n[2x+1] + n[2x] + \frac{1}{2}n[2x-1]. \tag{1.7}$$

Recurrences of this type are sometimes referred to as *two-scale relations*, in that they relate scaling functions defined on two consecutive scales. The refinement relation for a scaling function $n[x]$ is the analog of a collage for a fractal and provides the key ingredient necessary for subdivision. Note that a variant of this refinement relation also holds for translates of the scaling function $n[2^k x]$ on the grid $\frac{1}{2^k}\mathbb{Z}$.

1.3.2 Subdivision for Piecewise Linear Splines

Given this refinement relation, we can now construct the subdivision scheme for piecewise linear splines. Consider a linear spline $p[x]$ with knots on $\mathbb{Z}$. The spline $p[x]$ can be written as a linear combination of hat functions $n[x - \mathrm{i}]$ on the coarse grid $\mathbb{Z}$. In vector form, this fact can be expressed as

$$p[x] = \mathcal{N}[x]\,\mathrm{p}_0, \tag{1.8}$$

where $\mathcal{N}[x]$ is a row vector whose ith entry is the function $n[x-\mathrm{i}]$ and p_0 is a column vector whose entries are the values of $p[x]$ on $\mathbb{Z}$. Due to equation 1.7, $p[x]$ can also be written as a linear combination of hat functions $n[2x - \mathrm{i}]$, with knots on the fine grid $\frac{1}{2}\mathbb{Z}$. In vector form, this linear combination can be expressed as

$$p[x] = \mathcal{N}[2x]\,\mathrm{p}_1, \tag{1.9}$$

where p_1 is a vector whose entries correspond to the values of $p[x]$ on $\frac{1}{2}\mathbb{Z}$. Given these two expressions, our task is to find the change of basis matrix relating p_0 and p_1. Rewriting equation 1.7 in vector form, the coarse scaling functions $\mathcal{N}[x]$ can be expressed as a linear combination of the next finer scaling functions $\mathcal{N}[2x]$ via a change of basis matrix S satisfying

$$\mathcal{N}[x] == \mathcal{N}[2x]\mathrm{S}. \tag{1.10}$$

This *subdivision matrix,* S, has entries that depend only on the coefficients of the refinement relation. In particular, the ith column of S consists of the coefficients attached to the entries of $\mathcal{N}[2x]$ that reproduce the scaling function $n[x-\mathrm{i}]$. For piecewise linear splines, this column vector has exactly three non-zero entries $\{\frac{1}{2}, 1, \frac{1}{2}\}$, with corresponding row indices $\{2\mathrm{i} - 1, 2\mathrm{i}, 2\mathrm{i} + 1\}$. Note that each time i is increased by one, the corresponding row indices of the non-zero entries are shifted by two. The resulting matrix S has the property that each of its columns is a two-shift of a single fundamental vector s. In particular, the ith entry $\mathrm{s}[\![\mathrm{i}]\!]$ is the coefficient of the function $n[2x - \mathrm{i}]$ on the right-hand side of equation 1.7. Such matrices whose columns are two-shifts of a fundamental vector are known as *two-slanted matrices*. In this example, S can be viewed as a two-slanted matrix whose columns are two-shifts of the common sequence $\{\frac{1}{2}, 1, \frac{1}{2}\}$. In particular, equation 1.10 expands into

$$\begin{pmatrix} \cdot \\ n[x+2] \\ n[x+1] \\ n[x] \\ n[x-1] \\ n[x-2] \\ \cdot \end{pmatrix}^{\mathsf{T}} == \begin{pmatrix} \cdot \\ n[2x+3] \\ n[2x+2] \\ n[2x+1] \\ n[2x] \\ n[2x-1] \\ n[2x-2] \\ n[2x-3] \\ \cdot \end{pmatrix}^{\mathsf{T}} \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & \cdot \\ \cdot & 0 & 1 & 0 & 0 & 0 & \cdot \\ \cdot & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & \cdot \\ \cdot & 0 & 0 & 1 & 0 & 0 & \cdot \\ \cdot & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & \cdot \\ \cdot & 0 & 0 & 0 & 1 & 0 & \cdot \\ \cdot & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}.$$

(Note that the dots are used to emphasize that the vectors and matrices of this relation are bi-infinite.) Substituting the right-hand side of equation 1.10 into equation 1.8 yields $p[x] == \mathcal{N}[2x](\mathsf{S}\mathsf{p}_0)$. Because the functions of $\mathcal{N}[2x]$ are linearly independent, $\mathsf{p}_1 = \mathsf{S}\mathsf{p}_0$. Given that equation 1.10 holds, the following theorem makes these observations precise.

THEOREM 1.1 Let the matrix refinement relation of equation 1.10 hold (i.e., $\mathcal{N}[x] = \mathcal{N}[2x]\mathsf{S}$). If function $p[x]$ has the form $p[x] = \mathcal{N}[x]\mathsf{p}_0$, then $p[x] == \mathcal{N}[2^{\mathsf{k}}x]\mathsf{p}_{\mathsf{k}}$, where

$$\mathsf{p}_{\mathsf{k}} = \mathsf{S}\mathsf{p}_{\mathsf{k}-1}. \tag{1.11}$$

Proof Observe that replacing x by $2^{\mathsf{k}-1}x$ in equation 1.10 yields a matrix refinement relation for basis functions on $\frac{1}{2^{\mathsf{k}-1}}\mathbb{Z}$ and $\frac{1}{2^{\mathsf{k}}}\mathbb{Z}$ (i.e., $\mathcal{N}[2^{\mathsf{k}-1}x] = \mathcal{N}[2^{\mathsf{k}}x]\mathsf{S}$). Proceeding inductively, we assume that $p[x] = \mathcal{N}[2^{\mathsf{k}-1}x]\mathsf{p}_{\mathsf{k}-1}$. Replacing $\mathcal{N}[2^{\mathsf{k}-1}x]$ by the right-hand side of this matrix refinement relation yields $p[x] = \mathcal{N}[2^{\mathsf{k}}x](\mathsf{S}\mathsf{p}_{\mathsf{k}-1})$. Given that the functions of $\mathcal{N}[2^{\mathsf{k}}x]$ are linearly independent, equation 1.11 holds.

Given an initial vector of coefficients p_0, we can apply equation 1.11 to generate a sequence of new vectors p_{k}. Because the entries of p_{k} are coefficients of hat functions taken on the grid $\frac{1}{2^{\mathsf{k}}}\mathbb{Z}$, plotting the ith entry of p_{k} at $x == \frac{\mathsf{i}}{2^{\mathsf{k}}}$ gives a good

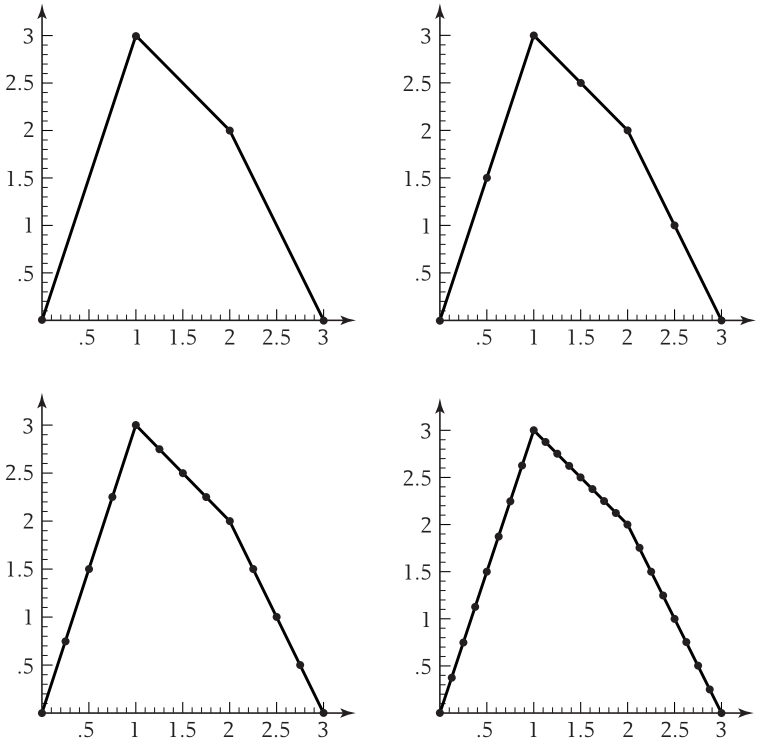


Figure 1.21 Progression of the subdivision process for a piecewise linear function.

approximation to shape of the function $p[x]$. Figure 1.21 shows three rounds of this refinement procedure starting from

$$\mathsf{p}_0 = \begin{pmatrix} 0 & 0 \\ 1 & 3 \\ 2 & 2 \\ 3 & 0 \end{pmatrix}.$$

The ith coefficient of p_k is plotted at $x == \frac{i}{2^k}$.

This process of taking a coarse approximation p_0 and computing increasingly dense approximations p_k to a final limit function $p[x]$ is referred to as *subdivision*. The beauty of subdivision is that we can now ignore the functions $\mathcal{N}[2^k x]$ entirely, and instead rely on the subdivision matrix S when generating approximations to $p[x]$. In effect, subdivision is a multiresolution rendering algorithm that generates increasingly dense plots by taking convex combinations of points used in previous plots.

Note that these convex combinations, embodied by $\mathsf{S}\mathsf{p}_{k-1}$, can be computed without constructing the entire subdivision matrix S. Because the columns of S are two-shifts of the sequence $\{\ldots, 0, \frac{1}{2}, 1, \frac{1}{2}, 0, \ldots\}$, multiplication of the matrix S by a column vector of coefficients p_{k-1} determines new coefficients in the vector p_k according to two simple rules. Coefficients in even-indexed rows of p_k correspond to coefficients from p_{k-1}, because the weights in the even-indexed rows of S are $\{\ldots, 0, 1, 0, \ldots\}$. Coefficients in odd-indexed rows of p_k are the average of two consecutive coefficients from p_{k-1}, because the odd-indexed rows of S have the form $\{\ldots, 0, \frac{1}{2}, \frac{1}{2}, 0, \ldots\}$.

As seen from this example, subdivision is potentially a very powerful method of modeling shapes. In particular, it combines the best of functional and fractal representations. As this book will demonstrate, subdivision has four principal advantages:

- Ease of use: Modeling with subdivision is easy to implement because it involves only discrete shapes.
- Expressiveness: Simple subdivision schemes can reproduce a large class of interesting shapes.
- Efficiency: Many subdivision schemes use repeated averaging, a fast method of computing discrete approximations of the limit shapes.
- Linear analysis tools: The subdivision process defines scaling functions whose properties can be analyzed using linear algebra.

1.4 **Overview**

One interesting property of the subdivision scheme for linear splines is that it makes no explicit use of the piecewise polynomial definition of the unit hat function $n[x]$. The scheme used only the subdivision matrix S. For piecewise linear splines, the subdivision matrix S was "nice" (i.e., it defined a convergent process). Thus, the scaling functions $\mathcal{N}[2^k x]$ associated with the subdivision scheme could be computed entirely in terms of the matrix S. To recover a scaling function in $\mathcal{N}[x]$, one would simply apply the subdivision process to the vector $\{\ldots, 0, 0, 1, 0, 0, \ldots\}$. This observation allows us to focus our entire attention on the subdivision matrix S and leads to two related questions:

- How can we construct a subdivision matrix S whose associated limit functions are "interesting"?
- Given a subdivision matrix S, is the associated subdivision scheme convergent? If so, are the associated limit functions smooth?

The next two chapters of this book address these two questions. Chapter 2 introduces a simple technique for generating subdivision schemes: directional integration. This integral approach is the standard technique used in constructing subdivision schemes for B-splines and box splines and leads directly to a simple repeated averaging algorithm for subdividing these splines. Chapter 3 considers the second question of convergence and smoothness. The key to answering this question is developing a method for subdividing various differences associated with the vector p_k. Taken together, the first three chapters of this book form an introduction to the basics of subdivision.

The second part of the book (Chapters 4, 5, and 6) reconsiders the first problem in more detail. Chapter 4 introduces an alternative approach, developed by the authors, for constructing "interesting" subdivision schemes. This differential method, although slightly more complicated, is capable of generating a wider range of schemes than those produced by the integral method. For example, the last section in Chapter 4 introduces a new curve subdivision scheme that reproduces splines in tension, B-splines, and trigonometric splines as special cases of a more general scheme. This subdivision scheme is the building block for a new scheme for surfaces of revolution introduced in Chapter 7. As an application of the differential method, Chapter 5 derives an infinitely supported subdivision scheme for solutions to the harmonic equation via the differential approach and discusses building finite approximations to this scheme. Chapter 6 considers variational extensions of the differential method for the problem of generating subdivision rules along the boundaries of finite domains.

The last part of the book, Chapters 7 and 8, introduces and analyzes a range of subdivision schemes for polyhedral meshes. Chapter 7 discusses an elegant method of subdividing surface meshes using repeated averaging (including an extension to the case of surfaces of revolution). This chapter also includes a simple method for handling creases and boundaries of subdivision surfaces. Chapter 8 extends the smoothness tests developed in Chapter 3 to the extraordinary vertices of these surface schemes and concludes by considering some of the promising areas for future research on subdivision.

CHAPTER 2

An Integral Approach to Uniform Subdivision

Chapter 1 provided a glimpse into the nature of subdivision. The key to building a subdivision scheme for splines with uniformly spaced knots is a scaling function $n[x]$ that possesses a refinement relation, that is, an expression of the function $n[x]$ as a linear combination of integer translates of its dilates, $n[2x - \mathrm{i}]$. For example, the scaling function for piecewise linear splines satisfies the refinement relation

$$n[x] == \frac{1}{2}n[2x + 1] + n[2x] + \frac{1}{2}n[2x - 1]. \tag{2.1}$$

The coefficients s of this refinement relation are then used to define a two-slanted matrix S that relates coefficient vectors $\mathrm{p}_{\mathrm{k}-1}$ and p_{k} defined on the grids $\frac{1}{2^{\mathrm{k}-1}}\mathbb{Z}$ and $\frac{1}{2^{\mathrm{k}}}\mathbb{Z}$, respectively, via the subdivision relation $\mathrm{p}_{\mathrm{k}} = \mathrm{S}\mathrm{p}_{\mathrm{k}-1}$.

This chapter introduces a general method for creating scaling functions that may be refined: directional integration. For example, the B-spline basis functions have a simple recursive definition in terms of integration. Based on this definition, we derive the refinement relation for B-splines and construct the associated subdivision scheme. Later in the chapter, we describe an equivalent geometric method for constructing the B-spline basis functions as the cross-sectional volumes of high-dimensional hypercubes. This construction easily generalizes from the univariate case to the bivariate case and allows creation of refineable scaling functions in two variables. The splines associated with these scaling functions, box splines, also have an elegant subdivision algorithm. The chapter concludes with a simple method for constructing explicit piecewise polynomial expressions for both B-splines and box splines. Again, these constructions have an interesting geometric interpretation in terms of high-dimensional geometry.

2.1 A Subdivision Scheme for B-splines

Chapter 1 introduced a simple method for constructing a smooth curve that followed a control polygon p with $\mathsf{m}+1$ vertices. The Bézier curve $p[x]$ associated with p is a parametric curve of degree m that approximates p as x varies from 0 to 1. For small values of m, the Bézier technique works well. However, as m grows large (say, $\mathsf{m} > 20$), Bézier curves begin to exhibit some undesirable properties. In particular, the Bernstein basis functions $\mathcal{B}^{\mathsf{m}}[x]$ are supported over the entire interval $[0, 1]$. Therefore, modifying any one of the control points in p induces a change to the entire curve $p[x]$. Moreover, subdividing $p[x]$ using the de Casteljau algorithm requires $\mathsf{O}[\mathsf{m}^2]$ computations.

One solution to this problem is to construct a collection of Bézier curves, say of degree three, that approximate the shape of p. The resulting composite curve is a piecewise polynomial curve. This modification allows local control of the composite curve in the sense that modifying one Bézier curve in the collection does not necessarily induce a change in the entire curve. Moreover, subdividing the $\mathsf{O}[\mathsf{m}]$ Bézier curves of degree three can be done in $\mathsf{O}[\mathsf{m}]$ time. Because in many applications it is desirable for this composite curve to be smooth (i.e., have a continuous derivative), any method for representing $p[x]$ as a collection of Bézier curves must ensure that the control polygons for consecutive Bézier curves meet smoothly.

An alternative approach is to construct a set of smooth piecewise polynomial basis functions that behave like the Bernstein basis functions. Instead of being globally supported, each basis function is locally supported. For example, if the associated curve $p[x]$ is defined on the interval $[0, \mathsf{m}]$, the ith basis function is supported only over a small subrange, say $[\mathsf{i}-2, \mathsf{i}+2]$ for cubic splines. Taking linear combinations of these smooth basis functions automatically yields a smooth curve $p[x]$. The advantage of this approach is that there is no need to manage each polynomial piece of $p[x]$ explicitly; smoothness of the entire curve $p[x]$ is guaranteed by construction.

The canonical example of such smooth piecewise polynomial basis functions is the B-spline basis functions. If the breakpoints between each polynomial piece (i.e., the knots) are placed at the integers, the curve formed by taking linear combinations of these basis functions is a uniform B-spline. Uniform B-splines are a fundamental tool in a number of areas, including numerical analysis, finite element methods, geometric design, and computer graphics. The success story of B-splines in geometric design starts with the landmark publications by De Boor [37, 38]. Bartels et al. [9] give an extensive introduction to the various remarkable properties and algorithms associated with B-splines.

2.1.1 B-spline Basis Functions via Repeated Integration

We begin our discussion of B-splines with a definition of B-spline basis functions in terms of repeated integration. Later, we derive the refinement relation for B-spline basis functions and conclude by constructing the corresponding subdivision scheme for uniform B-splines via Lane and Riesenfeld [93]. This subdivision scheme has a particularly beautiful interpretation in terms of repeated averaging. This averaging algorithm is simple to implement and provides the intellectual stepping-stone to the surface schemes discussed in Chapter 7. One important fact the reader should keep in mind during the rest of this chapter is that although the mathematics of constructing subdivision schemes may seem complicated the resulting schemes themselves are beautifully simple.

Our definition of B-spline basis functions is recursive. For the base case, the *B-spline basis function of order one* has the form

$$n^1[x] = \begin{cases} 1 & \text{if } 0 \le x < 1, \\ 0 & \text{otherwise.} \end{cases} \tag{2.2}$$

Figure 2.1 shows the graph of $n^1[x]$. This function is a piecewise constant function that is zero everywhere except on the interval $[0, 1)$, where it is the constant one. (In signal processing circles, this function is also referred to as the Haar scaling function.) Given the B-spline basis function $n^{m-1}[x]$, the *B-spline basis function of order* m, $n^m[x]$, is defined via the recurrence

$$n^m[x] = \int_0^1 n^{m-1}[x - t]\, \mathrm{d}t. \tag{2.3}$$

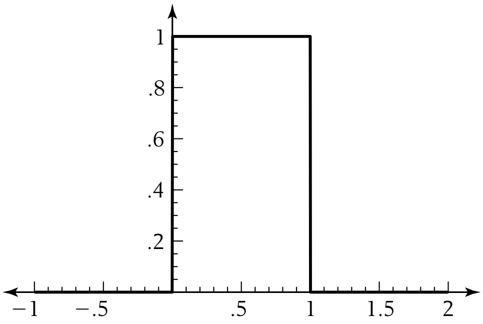


Figure 2.1 The B-spline basis function of order one.

Note that the *order* of a B-spline is by construction always one more than the degree of the B-spline. Due to their definition by repeated integration, B-spline basis functions have the following easily proven properties:

- $n^{\mathrm{m}}[x]$ is a $\mathcal{C}^{\mathrm{m}-2}$ piecewise polynomial of degree $\mathrm{m}-1$, with knots at the integers $\mathbb{Z}$. This observation follows from the fact that each round of integration via equation 2.3 increases the smoothness of a function.
- $n^{\mathrm{m}}[x]$ is non-negative and is supported on the interval $[0, \mathrm{m}]$. Again, this observation follows from the inductive definition of $n^{\mathrm{m}}[x]$.
- $n^{\mathrm{m}}[x]$ has unit integral. In fact, the sum $\sum_{\mathrm{i}\in\mathbb{Z}} n^{\mathrm{m}}[x-\mathrm{i}]$ is exactly one. This fact holds trivially for the base case $n^1[x]$ and follows by induction on m.

Together, these three properties are sufficient to establish that the function $n^{\mathrm{m}}[x]$ is unique. Let's continue by considering some examples of B-spline basis functions of low order. For example, the B-spline basis function $n^2[x]$ is a piecewise linear function of the form

$$n^2[x] = \begin{cases} x & \text{if } 0 < x \le 1, \\ 2-x & \text{if } 1 < x \le 2, \\ 0 & \text{otherwise.} \end{cases}$$

Figure 2.2 shows a plot of the piecewise linear basis function $n^2[x]$. Due to the recursive nature of the definition in equation 2.3, this hat function is supported on the interval $[0, 2]$ (as opposed to the hat function centered around the origin used

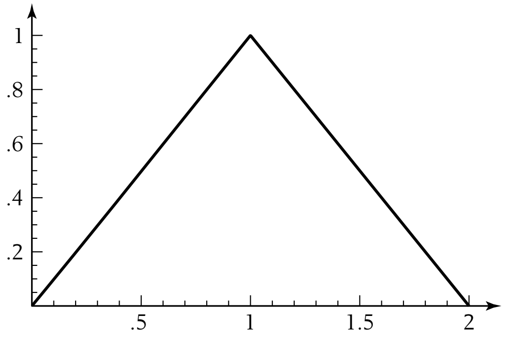


Figure 2.2 The linear B-spline basis function (order two).

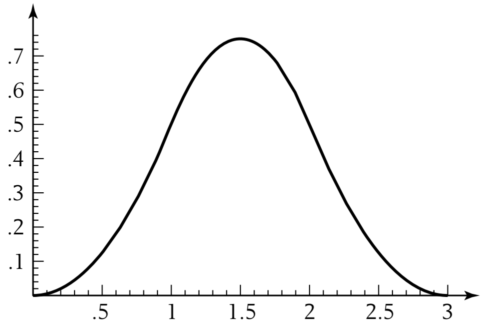


Figure 2.3 The quadratic B-spline basis function (order three).

in Chapter 1). Applying the recursive definition once again yields the piecewise quadratic basis function $n^3[x]$ (depicted in Figure 2.3):

$$n^3[x] = \begin{cases} \frac{x^2}{2} & \text{if } 0 < x \leq 1, \\ -\frac{3}{2} + 3x - x^2 & \text{if } 1 < x \leq 2, \\ \frac{1}{2}(-3+x)^2 & \text{if } 2 < x \leq 3, \\ 0 & \text{otherwise.} \end{cases}$$

An alternative method of defining the B-spline basis function $n^{\mathrm{m}}[x]$ involves repeated convolution with the basis function $n^1[x]$. The *continuous convolution* of two functions $p[x]$ and $q[x]$, denoted $p[x] \otimes q[x]$, is the integral $\int_{-\infty}^{\infty} p[t]q[x-t]\,\mathrm{d}t$. (Observe that the value of this integral is again a function of x.) Given this definition, the continuous convolution of the basis functions $n^1[x]$ and $n^{\mathrm{m}-1}[x]$ is the integral $\int_{-\infty}^{\infty} n^1[t]n^{\mathrm{m}-1}[x-t]\,\mathrm{d}t$. Because the effect of multiplying by the function $n^1[t]$ inside this integral is to simply restrict the range of integration for the variable t to the interval [0, 1], this integral reduces to the right-hand side of equation 2.3. Therefore, the basis function $n^{\mathrm{m}}[x]$ is simply the continuous convolution of two lower-order basis functions, $n^1[x]$ and $n^{\mathrm{m}-1}[x]$; that is,

$$n^{\mathrm{m}}[x] == n^1[x] \otimes n^{\mathrm{m}-1}[x].$$

Linear combinations of integer translates of the basis function $n^{\mathrm{m}}[x]$ can now be used to construct arbitrary B-splines. For example, a piecewise constant B-spline with knots at the integers $\mathbb{Z}$ can be written as a combination of translates of $n^1[x]$:

$$p[x] = \sum_{\mathrm{i} \in \mathbb{Z}} \mathrm{p}[\![\mathrm{i}]\!]\, n^1[x - \mathrm{i}].$$

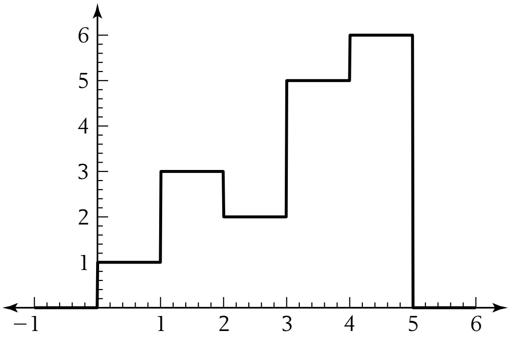


Figure 2.4 A piecewise constant B-spline.

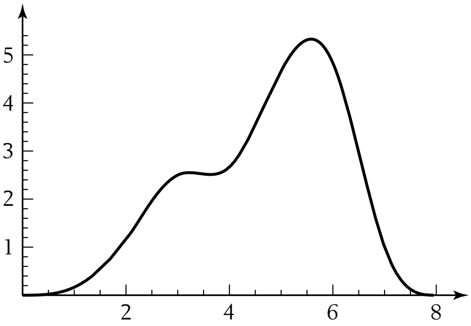


Figure 2.5 A piecewise cubic B-spline.

As in Chapter 1, this expression can be written in vector form as $p[x] = \mathcal{N}^1[x]\mathrm{p}$, where $\mathcal{N}^1[x]$ is the vector whose ith entry is $n^1[x - \mathrm{i}]$. Figure 2.4 shows a plot of a piecewise constant B-spline for which the vector p has the form $\{\ldots, 0, 1, 3, 2, 5, 6, 0, \ldots\}$. (The non-zero entries of p have indices ranging from 0 to 4.)

Higher-order B-splines $p[x]$ can be constructed by taking linear combinations of integer translates of higher-order basis functions (i.e., functions of the form $\sum_{\mathrm{i}\in\mathbb{Z}} \mathrm{p}[\![\mathrm{i}]\!] n^{\mathrm{m}}[x - \mathrm{i}]$). Figure 2.5 shows a cubic B-spline with the same coefficients as in Figure 2.4. Observe that this function is supported now over the interval $[0, 8]$, because the basis function $n^4[x]$ is supported on the interval $[0, 4]$.

2.1.2 A Refinement Relation for B-spline Basis Functions

For the piecewise linear splines of Chapter 1, the key ingredient in constructing a subdivision scheme was deriving a refinement relation for the associated scaling function. In this section, we derive a refinement relation for the B-spline basis function of order m. In general, this relation has the form

$$n^{m}[x] == \sum_{i \in \mathbb{Z}} s^{m}[\![i]\!] n^{m}[2x - i]. \tag{2.4}$$

For uniform schemes, the coefficient sequence s^m (m is a superscript here, not an exponent) is the *subdivision mask* associated with the refinement relation. Due to the recursive definition of $n^m[x]$, the construction for the mask s^m is also recursive. For the base case, we observe that $n^1[x]$ possesses the particularly simple refinement relation

$$n^{1}[x] == n^{1}[2x] + n^{1}[2x - 1]. \tag{2.5}$$

The refinement relation of equation 2.5 expresses a basis function defined on a coarse integer grid $\mathbb{Z}$ as the sum of two basis functions defined on a finer, half-integer grid $\frac{1}{2}\mathbb{Z}$. This relation is depicted in Figure 2.6.

A very useful technique for manipulating the subdivision mask s associated with a refinement relation is to construct the *generating function* s[x] associated with the mask s. By definition, s[x] is the sum of terms of the form $s[\![i]\!] x^i$ where $i \in \mathbb{Z}$ (i.e., $s[x] = \sum s[\![i]\!] x^i$). Be aware that the generating function s[x] may possess negative powers of x and thus is not necessarily a polynomial in x. Returning to our initial example, the refinement coefficients for $n^1[x]$ define the subdivision mask $s^1[x] = 1 + x$. (We use the term *mask* to refer to s and its associated generating function s[x] interchangeably.) For a B-spline basis function of order m, the subdivision mask $s^m[x]$ obeys the recurrence relation of Theorem 2.1, which follows.

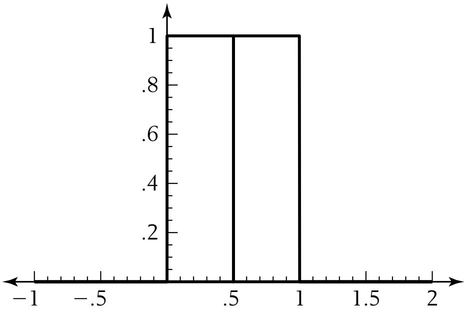


Figure 2.6 The B-spline basis function of order one as the sum of two translates of its dilates.

THEOREM 2.1 For all $\mathsf{m} > 1$, the subdivision mask $\mathsf{s}^{\mathsf{m}}[\mathsf{x}]$ for the B-spline basis function $n^{\mathsf{m}}[x]$ of order m satisfies the recurrence

$$\mathsf{s}^{\mathsf{m}}[\mathsf{x}] = \frac{1}{2}(1+\mathsf{x})\mathsf{s}^{\mathsf{m}-1}[\mathsf{x}]. \tag{2.6}$$

Proof The proof is inductive. Assume that the subdivision mask $\mathsf{s}^{\mathsf{m}-1}[\mathsf{x}]$ encodes the coefficients of the refinement relation for $n^{\mathsf{m}-1}[x]$. Our task is to show that the mask $\frac{1}{2}(1+\mathsf{x})\mathsf{s}^{\mathsf{m}-1}[\mathsf{x}]$ encodes the coefficients of the refinement relation for $n^{\mathsf{m}}[x]$. We begin with the inductive definition of $n^{\mathsf{m}}[x]$ as $\int_0^1 n^{\mathsf{m}-1}[x-t]\,\mathbb{d}t$. Refining $n^{\mathsf{m}-1}[x-t]$ via equation 2.4 yields the new integral

$$\int_0^1 \left(\sum_{\mathsf{i}} \mathsf{s}^{\mathsf{m}-1}[\![\mathsf{i}]\!]\, n^{\mathsf{m}-1}[2(x-t)-\mathsf{i}]\right) \mathbb{d}t.$$

Reparameterizing via $t \to \frac{1}{2}t$ yields a new integral on the interval $[0, 2]$ scaled by a factor of $\frac{1}{2}$. Splitting this integral into two integrals on the interval $[0, 1]$ and reindexing yields the equivalent integral

$$\frac{1}{2}\int_0^1 \left(\sum_{\mathsf{i}} (\mathsf{s}^{\mathsf{m}-1}[\![\mathsf{i}]\!] + \mathsf{s}^{\mathsf{m}-1}[\![\mathsf{i}-1]\!])\, n^{\mathsf{m}-1}[2x-t-\mathsf{i}]\right) \mathbb{d}t.$$

Moving the integral inside the sum yields

$$\frac{1}{2}\sum_{\mathsf{i}} (\mathsf{s}^{\mathsf{m}-1}[\![\mathsf{i}]\!] + \mathsf{s}^{\mathsf{m}-1}[\![\mathsf{i}-1]\!]) \int_0^1 n^{\mathsf{m}-1}[2x-t-\mathsf{i}]\,\mathbb{d}t.$$

Finally, applying the definition of $n^{\mathsf{m}}[2x-\mathsf{i}]$ yields the desired scaling relation for $n^{\mathsf{m}}[x]$ as

$$n^{\mathsf{m}}[x] == \frac{1}{2}\sum_{\mathsf{i}} (\mathsf{s}^{\mathsf{m}-1}[\![\mathsf{i}]\!] + \mathsf{s}^{\mathsf{m}-1}[\![\mathsf{i}-1]\!])n^{\mathsf{m}}[2x-\mathsf{i}].$$

The subdivision mask associated with this refinement relation is exactly $\frac{1}{2}(1+\mathsf{x})\mathsf{s}^{\mathsf{m}-1}[\mathsf{x}]$.

Starting from the base case $\mathsf{s}^1[\mathsf{x}] = 1+\mathsf{x}$, we can now iterate this theorem to obtain the subdivision masks for B-splines of higher order. In particular, the subdivision mask for B-splines of order m has the form

$$s^m[x] = \frac{1}{2^{m-1}}(1+x)^m. \tag{2.7}$$

Due to this formula, the coefficients of the refinement relation for $n^m[x]$ are simply the coefficients of the binomial expansion of $(1+x)^m$ scaled by a factor of $\frac{1}{2^{m-1}}$. Another technique for deriving the refinement relation for uniform B-spline basis functions uses the definition of $n^m[x]$ in terms of repeated continuous convolution. Unrolling the recurrence $n^m[x] = n^1[x] \otimes n^{m-1}[x]$ leads to the formula

$$n^m[x] = \otimes_{i=1}^m n^1[x] = \otimes_{i=1}^m (n^1[2x] + n^1[2x-1]).$$

Based on the linearity of convolution, this convolution of $m+1$ functions can be expanded using the binomial theorem into the sum of terms consisting of the convolution of $m+1$ functions of the form $n^1[2x]$ or $n^1[2x-1]$. Because these dilated functions satisfy the recurrence

$$n^m[2x] = \frac{1}{2}(n^1[2x] \otimes n^{m-1}[2x]),$$

these terms have the form $\frac{1}{2^{m-1}} n^m[2x-i]$, where $0 \leq i \leq m$. Accumulating these terms according to the binomial theorem yields the mask of equation 2.7.

2.1.3 The Associated Subdivision Scheme

Having computed the subdivision mask $s^m[x]$ for the B-spline basis function $n^m[x]$, we can use the coefficients of s^m to construct a matrix refinement relation that expresses translates of the basis function $n^m[x]$ defined on the coarse grid $\mathbb{Z}$ in terms of translates of the dilated function $n^m[2x]$ on the finer grid $\frac{1}{2}\mathbb{Z}$. This matrix relation has the form $\mathcal{N}^m[x] = \mathcal{N}^m[2x]S$, where $\mathcal{N}^m[x]$ is a row vector whose ith entry is $n^m[x-i]$. Just as in the linear case, this subdivision matrix S is a bi-infinite matrix whose columns are two-shifts of the subdivision mask s^m. This two-shifting arises from the fact that translating a refineable scaling function $n^m[x]$ of the form $\sum_i s^m[\![i]\!] n^m[2x-i]$ by j units on $\mathbb{Z}$ induces a shift of 2j units in the translates of the dilated scaling function $n^m[2x]$ on $\frac{1}{2}\mathbb{Z}$; that is,

$$n^m[x-j] == \sum_i s^m[\![i]\!] n^m[2x-i-2j]. \tag{2.8}$$

Based on this formula, we observe that the ijth entry of the matrix S is $s^m[\![i-2j]\!]$. For example, the subdivision mask for cubic B-splines is $\frac{1}{8}+\frac{1}{2}x+\frac{3}{4}x^2+\frac{1}{2}x^3+\frac{1}{8}x^4$. The corresponding subdivision matrix S has the form

$$\begin{pmatrix} . & . & . & . & . & . & . \\ . & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & . \\ . & \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & 0 & 0 & . \\ . & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & . \\ . & 0 & \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & 0 & . \\ . & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & . \\ . & 0 & 0 & \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & . \\ . & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & . \\ . & . & . & . & . & . & . \end{pmatrix}.$$

Given a B-spline $p[x]$ expressed in vector form as $\mathcal{N}^m[x]p_0$, this matrix refinement relation implies that $p[x]$ can also be expressed as $\mathcal{N}^m[2x](Sp_0)$. If we define a subdivision process of the form

$$p_k = Sp_{k-1},$$

the resulting vectors p_k are the coefficients for the function $p[x]$ defined on the grid $\frac{1}{2^k}\mathbb{Z}$ (i.e., $p[x]=\mathcal{N}^m[2^kx]p_k$). Of course, in practice one should never actually construct the subdivision matrix S, because it is very sparse. Instead, the ith coefficient of p_k can be computed as a linear combination of the coefficients in p_{k-1} (using equations 2.4 and 2.8):

$$p_k[\![i]\!] = \sum_{j\in\mathbb{Z}} s^m[\![i-2j]\!]\,p_{k-1}[\![j]\!]. \tag{2.9}$$

If $p_{k-1}[x]$ and $s^m[x]$ are the generating functions associated with p_{k-1} and s^m, the summation of pairwise products in equation 2.9 looks suspiciously like an expression for the coefficients of the product $s^m[x]p_{k-1}[x]$. In fact, if the expression $i-2j$ were replaced by $i-j$, equation 2.9 would indeed compute the coefficients of $s^m[x]p_{k-1}[x]$. To model equation 2.9 using generating functions, we instead *upsample* the coefficient vector p_{k-1} by introducing a new zero coefficient between each coefficient of p_{k-1}. The entries of the upsampled vector are exactly the coefficients of the generating function $p_{k-1}[x^2]$. Because the jth coefficient of $p_{k-1}[x]$ corresponds to the $(2j)$th coefficient of $p_{k-1}[x^2]$, equation 2.9 can now be viewed as defining the i th coefficient of the product $s^m[x]p_{k-1}[x^2]$. In particular, the analog in terms of generating functions for the subdivision relation $p_k = Sp_{k-1}$ is

$$p_k[x] = s^m[x]p_{k-1}[x^2]. \tag{2.10}$$

To illustrate this subdivision process for uniform B-splines, we next consider two examples, one linear and one cubic. For both examples, our initial coefficient vector p_0 will have the form $\{\ldots, 0, 1, 3, 2, 5, 6, 0, \ldots\}$, where the non-zero entries of this vector are indexed from 0 to 4. The corresponding generating function $p_0[x]$ has the form $1 + 3x + 2x^2 + 5x^3 + 6x^4$. In the piecewise linear case, the subdivision mask $s^2[x]$ has the form $\frac{1}{2}(1+x)^2$. Applying equation 2.10 yields $p_1[x]$ of the form

$$\begin{aligned} p_1[x] &= s^2[x]p_0[x^2], \\ &= \frac{1}{2}(1+x)^2(1+3x^2+2x^4+5x^6+6x^8), \\ &= \frac{1}{2} + x + 2x^2 + 3x^3 + \frac{5x^4}{2} + 2x^5 + \frac{7x^6}{2} + 5x^7 + \frac{11x^8}{2} + 6x^9 + 3x^{10}. \end{aligned}$$

Subdividing a second time yields $p_2[x] = s^2[x]p_1[x^2]$, where $p_2[x]$ has the form

$$\frac{1}{2}(1+x)^2\left(\frac{1}{2} + x^2 + 2x^4 + 3x^6 + \frac{5x^8}{2} + 2x^{10} + \frac{7x^{12}}{2} + 5x^{14} + \frac{11x^{16}}{2} + 6x^{18} + 3x^{20}\right).$$

Using the technique of Chapter 1, we plot the i th coefficient of the generating function $p_k[x]$ at the grid point $x == \frac{i}{2^k}$. Figure 2.7 shows plots of the coefficients of $p_0[x]$, $p_1[x]$, $p_2[x]$, and $p_3[x]$ versus the grids $\mathbb{Z}$, $\frac{1}{2}\mathbb{Z}$, $\frac{1}{4}\mathbb{Z}$, and $\frac{1}{8}\mathbb{Z}$, respectively. We next repeat this process for cubic B-splines using the subdivision mask $s^4[x] = \frac{1}{8}(1+x)^4$. Figure 2.8 shows plots of the coefficients of $p_0[x]$, $p_1[x]$, $p_2[x]$, and $p_3[x]$ versus the grids $\mathbb{Z}$, $\frac{1}{2}\mathbb{Z}$, $\frac{1}{4}\mathbb{Z}$, and $\frac{1}{8}\mathbb{Z}$, respectively, where $p_k[x] = s^4[x]p_{k-1}[x^2]$.

The cubic B-spline $p[x]$ of Figure 2.5 was supported over the interval $[0, 8]$. In Figure 2.8, the polygons defined by the vector p_k appear to be converging to this function $p[x]$ as $k \to \infty$. Note that the polygons in this figure shift slightly to the right during each round of subdivision. This behavior is due to the fact that the subdivision mask $\frac{1}{8}(1+x)^4$ involves only non-negative powers of x; each multiplication by this mask shifts the resulting coefficient sequence to the right. To avoid this shifting, we can translate the basis function $n^m[x]$ such that it is supported on the interval $[-\frac{m}{2}, \frac{m}{2}]$. The effect of the subdivision mask $s^m[x]$ is to multiply it by a factor of $x^{-\frac{m}{2}}$. The resulting subdivision mask is now *centered* (as in equation 2.1). As a result, the polygons corresponding to the vectors p_k no longer shift during subdivision.

For linear B-splines that are centered, it is clear that the coefficients of p_k interpolate the values of $p[x]$ on the grid $\frac{1}{2^k}\mathbb{Z}$, in that the basis function $n^2[x]$ has

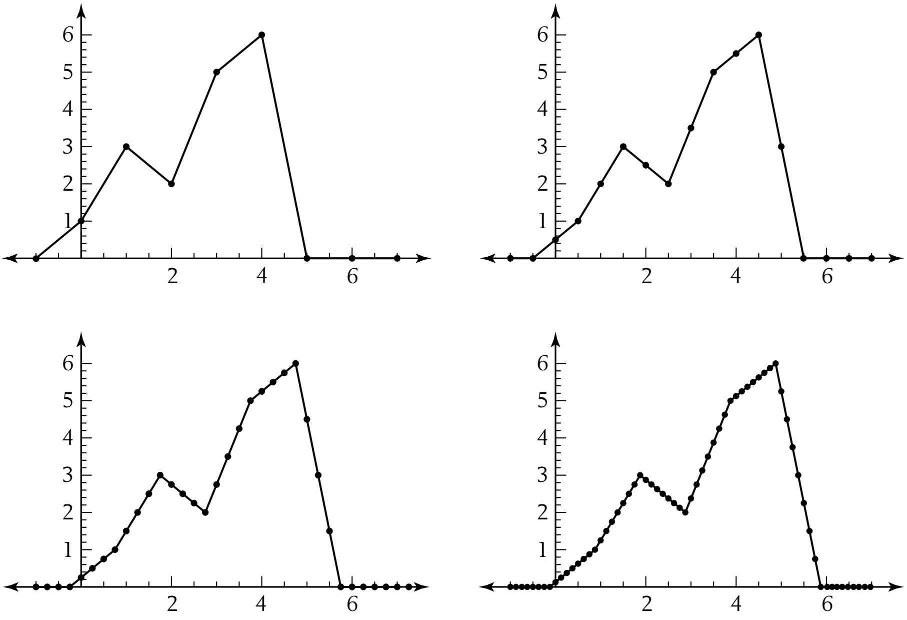


Figure 2.7 Three rounds of subdivision for a linear B-spline (m == 2).

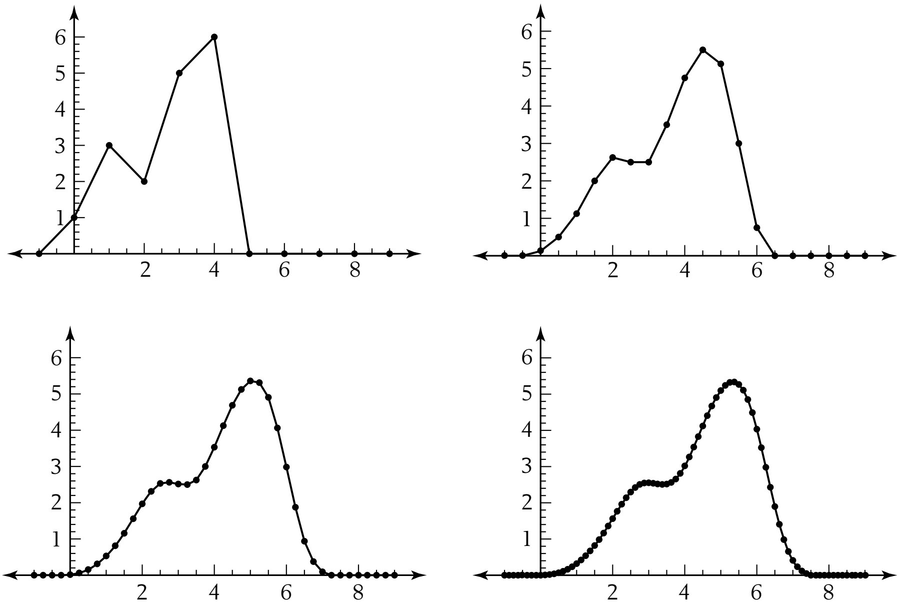


Figure 2.8 Three rounds of subdivision for a cubic B-spline (m == 4).

value 1 at $x == 0$ and is zero at the remaining integer knots. However, higher-order B-splines are not interpolatory (i.e., their limit curves do not pass through the coefficients of p_k plotted on the grid $\frac{1}{2^k}\mathbb{Z}$). For now, the convergence of the vectors p_k to the actual values of $p[x]$ as $k \to \infty$ is simply assumed. For those interested in a proof of this fact, we suggest skipping ahead to Chapter 3, where this question is considered in detail.

This subdivision scheme for B-splines was first documented by Chaikin [18] (for the quadratic case) and by Lane and Riesenfeld [93] (for the general case). Lane and Riesenfeld described a particularly elegant implementation of this subdivision method based on the factorization of $s^m[x]$ in Theorem 2.1. The subdivision mask $s^m[x]$ can be written as $2(\frac{1+x}{2})^m$. Thus, given a set of coefficients p_{k-1} defined on a coarse grid $\frac{1}{2^{k-1}}\mathbb{Z}$, we can compute a set of coefficients p_k defined on the fine grid $\frac{1}{2^k}\mathbb{Z}$ as follows:

- Construct the generating function $p_{k-1}[x]$ from p_{k-1}. Upsample $p_{k-1}[x]$ to obtain $p_{k-1}[x^2]$. Set $p_k[x] = 2\,p_{k-1}[x^2]$.
- Update $p_k[x]$ m times via the recurrence $p_k[x] = (\frac{1+x}{2})\,p_k[x]$.
- Extract the coefficients p_k of the resulting generating function $p_k[x]$.

In geometric terms, each multiplication of $p_k[x]$ by $\frac{(1+x)}{2}$ corresponds to computing the midpoints of adjacent vertices of the control polygon p_k. Thus, a single round of subdivision for a B-spline of order m can be expressed as upsampling followed by m rounds of midpoint averaging. Figure 2.9 shows an example of a single round of subdivision for cubic B-splines implemented in this manner.

In practice, using generating functions to implement subdivision is inefficient. A better method is to observe that the coefficient sequence for the product $s^m[x]p_{k-1}[x^2]$ is simply the discrete convolution of the sequence s^m and the upsampled sequence p_{k-1}. Using an algorithm such as the Fast Fourier Transform (FFT), the discrete convolution of these two coefficient sequences can be computed very quickly. (See Cormen, Leiserson, and Rivest [27] for more information on the FFT.) More generally, upsampling and midpoint averaging can be viewed as simple filters. Used in conjunction with wiring diagrams, these filters are a useful tool for creating and analyzing uniform subdivision schemes. An added benefit of the filter approach is that it is compatible with the filter bank technology used in constructing wavelets and in multiresolution analysis. Strang and Nguyen [150] give a nice introduction to the terminology and theory of filters. Guskov et al. [69], Kobbelt and Schröder [87], and Kobbelt [84] give several examples of using filter technology to construct subdivision schemes.

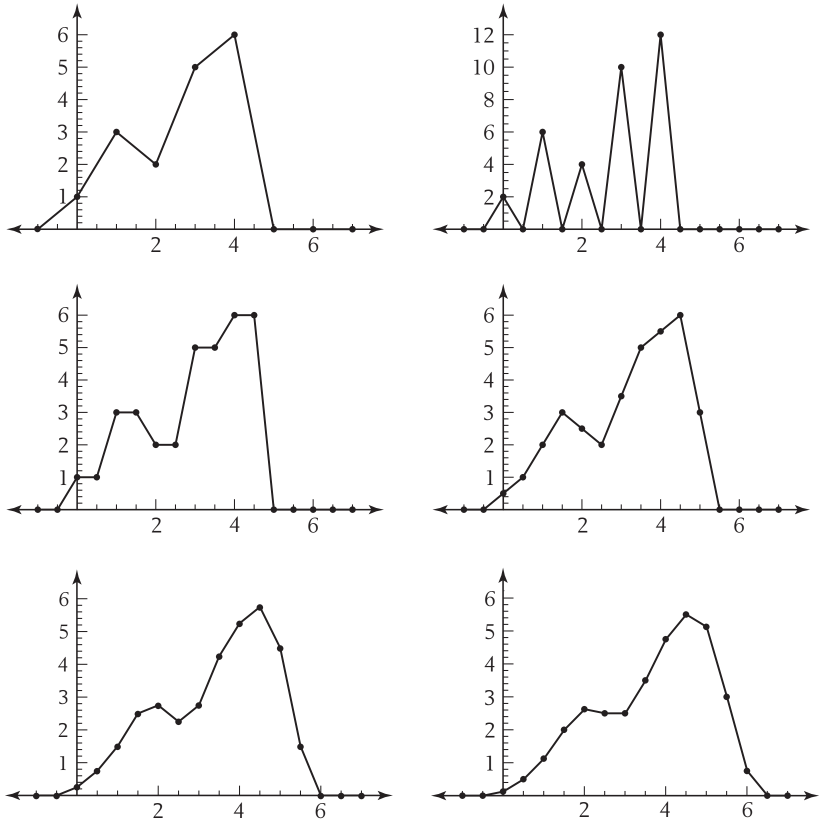


Figure 2.9 One round of cubic B-spline subdivision expressed as upsampling (top right) followed by four rounds of midpoint averaging.

2.2 A Subdivision Scheme for Box Splines

In the previous section, B-spline basis functions were defined via repeated integration. Although this definition has the advantage that it is mathematically succinct, it fails to deliver much geometric intuition about why B-spline basis functions possess such a simple refinement relation. In this section, we give a more intuitive geometric construction for these basis functions. Using this geometric construction, the possibility of building analogs to B-splines in higher dimensions becomes immediately clear. These analogs, the box splines, are simply cross-sectional volumes of high-dimensional hypercubes. For the sake of notational simplicity, we restrict our investigation of box splines to two dimensions. However, all of the results that appear in this section extend to higher dimensions without any difficulty. A good source for most of the material that follows is the book by De Boor et al. [40].

2.2.1 B-spline Basis Functions as Cross-sectional Volumes

We begin with a simple geometric construction for the B-spline basis function of order m as the cross-sectional volume of an m-dimensional hypercube. Given an interval $\mathcal{H} = [0, 1)$, we let $\mathcal{H}^{\mathsf{m}}$ denote the m-dimensional hypercube consisting of all points $\{t_1, \ldots, t_{\mathsf{m}}\}$ such that $0 \le t_{\mathsf{i}} < 1$ for $1 \le \mathsf{i} \le \mathsf{m}$. Now, consider a function $n[x]$ satisfying the equation

$$n[x] = \frac{1}{\sqrt{\mathsf{m}}} \mathsf{vol}_{\mathsf{m}-1}\left[\left\{\{t_1, \ldots, t_{\mathsf{m}}\} \in \mathcal{H}^{\mathsf{m}} \;\middle|\; \sum_{\mathsf{i}=1}^{\mathsf{m}} t_{\mathsf{i}} == x\right\}\right], \tag{2.11}$$

where $\mathsf{vol}_{\mathsf{m}-1}[\mathsf{B}]$ is the $(\mathsf{m}-1)$-dimensional volume of the set B. For a particular value of x, $n[x]$ returns the $(\mathsf{m}-1)$-dimensional volume of the intersection of the m-dimensional hypercube $\mathcal{H}^{\mathsf{m}}$ and the $(\mathsf{m}-1)$-dimensional hyperplane $\sum_{\mathsf{i}=1}^{\mathsf{m}} t_{\mathsf{i}} == x$. (The extra factor of $\frac{1}{\sqrt{\mathsf{m}}}$ normalizes $n(x)$ to have unit integral.)

The left-hand portion of Figure 2.10 illustrates this construction for $\mathsf{m} == 2$. At the top of the figure is the unit square $\mathcal{H}^2$. At the bottom of the figure is a plot of $n[x]$, the length of the line segment formed by intersecting the vertical line $t_1 + t_2 == x$ with the square $\mathcal{H}^2$. Observe that $n[x]$ is exactly the piecewise linear hat function. The right-hand portion of Figure 2.10 illustrates this construction for $\mathsf{m} = 3$. At the top of the figure is the unit cube $\mathcal{H}^3$. At the bottom of the figure is a plot of $n[x]$, the area of the polygon formed by intersecting the plane $t_1 + t_2 + t_3 == x$ (orthogonal to the x axis) with the cube $\mathcal{H}^3$. Again, $n[x]$ looks remarkably like the quadratic B-spline basis function.

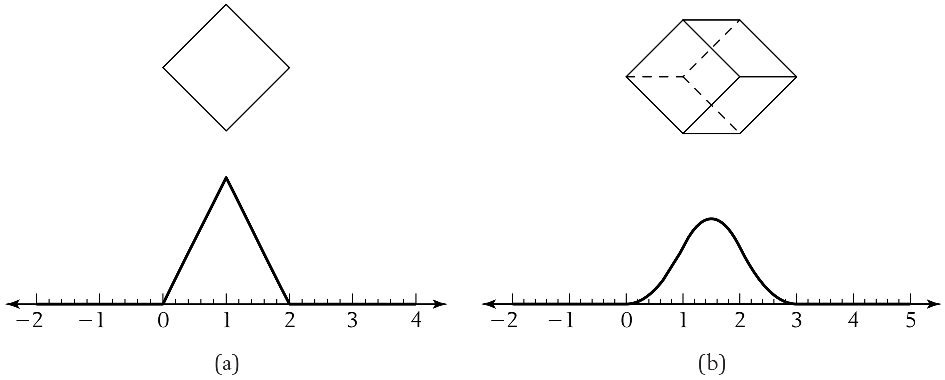


Figure 2.10 (a) Linear and (b) quadratic B-spline basis functions as cross-sectional volumes of the hypercubes $\mathcal{H}^2$ and $\mathcal{H}^3$.

Because the vertices of $\mathcal{H}^{\mathrm{m}}$ project onto integer points on the x axis, the function $n[x]$ is a piecewise polynomial function of degree $\mathrm{m}-1$, with knots at $\mathbb{Z}$. In fact, the function $n[x]$ generated by the cross-sectional volume of $\mathcal{H}^{\mathrm{m}}$ is exactly the B-spline basis function $n^{\mathrm{m}}[x]$. The following theorem shows that cross-sectional volumes used in defining $n[x]$ satisfy a recurrence identical to that of equation 2.3.

THEOREM 2.2 The B-spline basis function $n^{\mathrm{m}}[x]$ of equation 2.3 satisfies equation 2.11; that is,

$$n^{\mathrm{m}}[x] == \frac{1}{\sqrt{\mathrm{m}}}\,\mathrm{vol}_{\mathrm{m}-1}\left[\left\{\{t_1,\ldots,t_{\mathrm{m}}\}\in\mathcal{H}^{\mathrm{m}}\;\middle|\;\sum_{\mathrm{i}=1}^{\mathrm{m}} t_{\mathrm{i}} == x\right\}\right].$$

Proof The proof proceeds by induction on m. The base case of $\mathrm{m}==1$ holds by inspection. If we assume that $n^{\mathrm{m}-1}[x]$ satisfies the volume definition of equation 2.11,

$$n^{\mathrm{m}-1}[x] = \frac{1}{\sqrt{\mathrm{m}-1}}\,\mathrm{vol}_{\mathrm{m}-2}\left[\left\{\{t_1,\ldots,t_{\mathrm{m}-1}\}\in\mathcal{H}^{\mathrm{m}-1}\;\middle|\;\sum_{\mathrm{i}=1}^{\mathrm{m}-1} t_{\mathrm{i}} == x\right\}\right],$$

our task is to show that $n^{\mathrm{m}}[x]$ also satisfies equation 2.11. To complete this induction, we observe that the $(\mathrm{m}-1)$-dimensional volume on the right-hand side of equation 2.11 satisfies the recurrence:

$$\begin{aligned}&\frac{1}{\sqrt{\mathrm{m}}}\,\mathrm{vol}_{\mathrm{m}-1}\left[\left\{\{t_1,\ldots,t_{\mathrm{m}}\}\in\mathcal{H}^{\mathrm{m}}\;\middle|\;\sum_{\mathrm{i}=1}^{\mathrm{m}} t_{\mathrm{i}} == x\right\}\right] \qquad (2.12)\\ &== \int_0^1 \frac{1}{\sqrt{\mathrm{m}-1}}\,\mathrm{vol}_{\mathrm{m}-2}\left[\left\{\{t_1,\ldots,t_{\mathrm{m}-1}\}\in\mathcal{H}^{\mathrm{m}-1}\;\middle|\;\sum_{\mathrm{i}=1}^{\mathrm{m}-1} t_{\mathrm{i}} == x-t_{\mathrm{m}}\right\}\right] \mathrm{d}t_{\mathrm{m}}.\end{aligned}$$

The various square roots in this expression normalize the integral of the $(\mathrm{m}-2)$-dimensional volumes to agree with the $(\mathrm{m}-1)$-dimensional volume. Applying our inductive hypothesis to the right-hand side of this equation yields

$$\frac{1}{\sqrt{\mathrm{m}}}\,\mathrm{vol}_{\mathrm{m}-1}\left[\left\{\{t_1,\ldots,t_{\mathrm{m}}\}\in\mathcal{H}^{\mathrm{m}}\;\middle|\;\sum_{\mathrm{i}=1}^{\mathrm{m}} t_{\mathrm{i}} == x\right\}\right] == \int_0^1 n^{\mathrm{m}-1}[x-t_{\mathrm{m}}]\,\mathrm{d}t_{\mathrm{m}}. \qquad (2.13)$$

Applying the inductive definition of $n^{\mathrm{m}}[x]$ from equation 2.3 completes the proof.

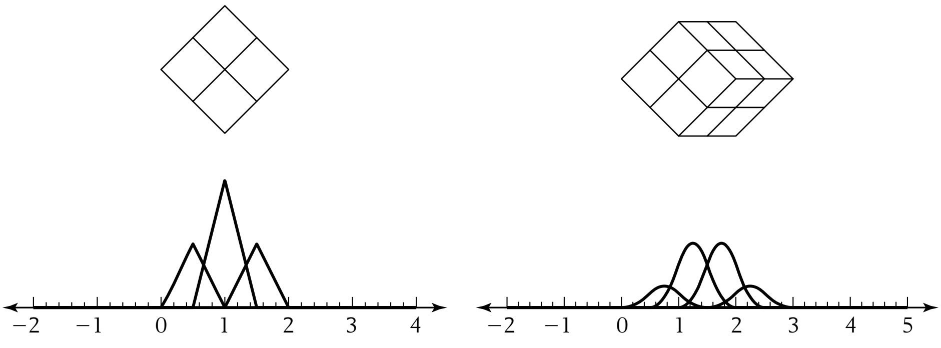
This idea of constructing B-spline basis functions as the cross-sectional volumes of a higher-dimensional polytope appeared as early as 1903 in Sommerfeld [142] and was popularized by Curry and Schönberg in [28]. (Curry and Schönberg actually suggested using a simplex in place of $\mathcal{H}^{\mathrm{m}}$.)

Given this geometric construction for $n^{\mathrm{m}}[x]$, the refinement relation for $n^{\mathrm{m}}[x]$ has an elegant interpretation. Consider the effect of splitting the hypercube $\mathcal{H}^{\mathrm{m}}$ into 2^{m} subcubes by splitting the interval $\mathcal{H}$ into the subintervals $[0, \frac{1}{2})$ and $[\frac{1}{2}, 1)$. Because all of these new subcubes are scaled translates of the original hypercube $\mathcal{H}^{\mathrm{m}}$, their cross-sectional volumes are multiples of integer translates of the dilated cross-sectional volume $n^{\mathrm{m}}[2x]$.

For example, the left-hand side of Figure 2.11 illustrates the case of $\mathrm{m} == 2$. At the top, the square $\mathcal{H}^2$ has been split into four subsquares. At the bottom is a plot of the cross-sectional lengths of each of these subsquares. (Note that the cross-sectional lengths for subsquares with the same projection onto x have been summed.) Because cross-sectional length is halved when the square $\mathcal{H}^2$ is split, the cross-sectional lengths satisfy the recurrence

$$n^2[x] == \frac{1}{2}n^2[2x] + n^2[2x-1] + \frac{1}{2}n^2[2x-2].$$

This equation is exactly the refinement relation for linear B-splines. The right-hand side of Figure 2.11 illustrates the case of $\mathrm{m} == 3$. At the top, the cube $\mathcal{H}^3$ has been split into eight subcubes. At the bottom is a plot of the cross-sectional areas of each of these subcubes. (Note that the cross-sectional areas for subcubes with the



(a)

(b)

Figure 2.11 Subdividing (a) linear and (b) quadratic basis functions by splitting the hypercubes $\mathcal{H}^2$ and $\mathcal{H}^3$.

same projection onto x have been summed.) Because cross-sectional area is scaled by a factor of $\frac{1}{4}$ when the cube $\mathcal{H}^3$ is split, these cross-sectional areas satisfy the recurrence

$$n^3[x] == \frac{1}{4}n^3[2x] + \frac{3}{4}n^3[2x-1] + \frac{3}{4}n^3[2x-2] + \frac{1}{4}n^3[2x-3].$$

Again, this equation is exactly the refinement relation for a quadratic B-spline.

2.2.2 Box-spline Scaling Functions as Cross-sectional Volumes

The beauty of this geometric construction for B-spline basis functions is that it immediately implies the existence of a refinement relation for the basis functions. De Boor and Höllig [39] observed that this construction can be generalized to produce smooth multivariate scaling functions that are also refineable. (Note that integer translates of these scaling functions are not necessarily linearly independent, and hence not necessarily basis functions.) The key idea is to define a set of *direction vectors* Σ of the form $\{\{\mathrm{a}_i, \mathrm{b}_i\} \in \mathbb{Z}^2 \mid i = 1, \ldots, \mathrm{m}\}$ and then consider the cross-sectional volume of the form

$$\mathrm{vol}_{\mathrm{m}-2}\left[\left\{\{t_1, \ldots, t_{\mathrm{m}}\} \in \mathcal{H}^{\mathrm{m}} \;\middle|\; \sum_{i=1}^{\mathrm{m}} \{\mathrm{a}_i, \mathrm{b}_i\} t_i == \{x, y\}\right\}\right]. \tag{2.14}$$

If this function is normalized to have unit integral, the resulting function $n^{\Sigma}[x, y]$ is the *box-spline scaling function* associated with the direction vectors Σ. Note that instead of computing the $(\mathrm{m}-1)$-dimensional volume of the intersection of $\mathcal{H}^{\mathrm{m}}$ and an $(\mathrm{m}-1)$-dimensional hyperplane this definition computes the $(\mathrm{m}-2)$-dimensional volume of the intersection of $\mathcal{H}^{\mathrm{m}}$ and two $(\mathrm{m}-1)$-dimensional hyperplanes of the form $\sum_{i=1}^{\mathrm{m}} \mathrm{a}_i t_i == x$ and $\sum_{i=1}^{\mathrm{m}} \mathrm{b}_i t_i == y$. One way to visualize the relation between the hypercube $\mathcal{H}^{\mathrm{m}}$ and the direction vectors $\{\mathrm{a}_i, \mathrm{b}_i\}$ is to orient $\mathcal{H}^{\mathrm{m}}$ with respect to the xy plane such that the origin of $\mathcal{H}^{\mathrm{m}}$ projects onto the origin of the xy plane and the coordinate axes of $\mathcal{H}^{\mathrm{m}}$ project onto the line segments defined by placing the direction vectors $\{\mathrm{a}_i, \mathrm{b}_i\}$ at the origin. Based on this definition, the following properties follow (almost) immediately:

- $n^{\Sigma}[x, y]$ is a piecewise polynomial function of degree $\mathrm{m}-2$ with $\mathcal{C}^{\alpha-2}$ continuity, where α is the size of the smallest subset $\mathrm{A} \subset \Sigma$ such that the complement of A in Σ does not span $\mathbb{R}^2$.
- $n^{\Sigma}[x, y]$ is non-negative and is supported on the domain $\{\sum_{i=1}^{\mathrm{m}} \{\mathrm{a}_i, \mathrm{b}_i\} t_i \mid 0 \le t_i < 1\}$.

- $n^{\Sigma}[x, y]$ has unit integral by construction. As in the case of B-splines, this property can be strengthened to $\sum_{\mathrm{i},\mathrm{j}} n^{\Sigma}[x - \mathrm{i}, y - \mathrm{j}] == 1$.

The only one of these properties that requires substantial analysis is the smoothness bound of $\mathcal{C}^{\alpha-2}$. In the case of B-splines, the basis functions $n^{\mathrm{m}}[x]$ always have maximal smoothness (i.e., $\mathcal{C}^{\mathrm{m}-2}$). However, box-spline scaling functions $n^{\Sigma}[x, y]$ do not always achieve the maximum possible smoothness of $\mathcal{C}^{\mathrm{m}-3}$. If the direction vectors in Σ are independent (i.e., no two vectors are constant multiples of each other), all but one of the vectors must be removed from Σ for the space of the remaining vectors to collapse to a line. In this case, the box-spline scaling function has maximal smoothness $\mathcal{C}^{\mathrm{m}-3}$, in that $\alpha == \mathrm{m} - 1$. However, if one of the direction vectors in Σ is repeated several times, removing all vectors except for this repeated vector leads to $\alpha < \mathrm{m} - 1$. In this case, the box-spline scaling function does not have maximal smoothness. See section 2.2.4 for examples of computing the smoothness of various box splines.

Geometrically, the effect of repeated direction vectors on the resulting projection of $\mathcal{H}^{\mathrm{m}}$ is to cause a higher-dimensional face of $\mathcal{H}^{\mathrm{m}}$ to project onto a line segment in the xy plane. As a result, the cross-sectional volume of the hypercube $\mathcal{H}^{\mathrm{m}}$ taken across this edge no longer has the maximal smoothness of $\mathcal{C}^{\mathrm{m}-3}$. For a more detailed analysis of this and other properties of box splines, we direct the interested reader to [40].

2.2.3 Subdivision for Box Splines

Our next task is to derive a refinement relation (and the corresponding subdivision mask) for the box-spline scaling functions $n^{\Sigma}[x, y]$. Geometrically, this refinement relation is a consequence of splitting the hypercube $\mathcal{H}^{\mathrm{m}}$ into 2^{m} subcubes. To compute the actual coefficients of this refinement relation, we instead reformulate box splines in terms of an inductive definition based on repeated integration (as done for B-splines in equation 2.3). The subdivision mask for box splines then follows from a variant of Theorem 2.1.

To simplify this recursive definition, we always choose the set Σ of direction vectors to consist of the two standard unit vectors, $\{1, 0\}$ and $\{0, 1\}$ in the base case. In this case, the box-spline scaling function $n^{\Sigma}[x, y]$ is simply the step function defined over the unit square:

$$n^{\Sigma}[x, y] = \begin{cases} 1 & \text{if } 0 \le x < 1 \text{ and } 0 \le y < 1, \\ 0 & \text{otherwise.} \end{cases} \tag{2.15}$$

Larger sets of direction vectors are formed by inductively adding new direction vectors to this initial set. Given an existing set of direction vectors $\tilde{\Sigma}$ and associated box-spline scaling function $n^{\tilde{\Sigma}}[x, y]$, the box-spline scaling function associated with the set $\Sigma = \tilde{\Sigma} \cup \{\{a, b\}\}$ has the form

$$n^{\Sigma}[x, y] = \int_0^1 n^{\tilde{\Sigma}}[x - at, y - bt]\, \mathrm{d}t.$$

A proof of the equivalence of this recursive definition of $n^{\Sigma}[x, y]$ to the definition of equation 2.14 is almost identical to the proof of Theorem 2.2. The main difference is that the $(m - 2)$-dimensional volume of equation 2.14 is reduced to the integral of $(m - 3)$-dimensional volumes in a manner similar to that of equation 2.12 for B-splines.

Given this definition (in terms of repeated directional integration), we next establish the link between repeated directional integration and subdivision. For the base case of $\Sigma = \{\{1, 0\}, \{0, 1\}\}$, the two-direction constant box splines of equation 2.15 satisfy the following refinement relation:

$$n^{\Sigma}[x, y] == n^{\Sigma}[2x, 2y] + n^{\Sigma}[2x - 1, 2y] + n^{\Sigma}[2x, 2y - 1] + n^{\Sigma}[2x - 1, 2y - 1].$$

This relation simply states that a constant function defined over the unit square can be expressed as the sum of four constant functions, each defined over one of four subsquares that constitute the original square. The coefficients of this relation define the subdivision mask for this scaling function, $s^{\Sigma}[x, y] = (1 + x)(1 + y)$. Next, based on the recursive definition of $n^{\Sigma}[x, y]$, we recursively construct the subdivision mask $s^{\Sigma}[x, y]$ for box splines. This mask satisfies a refinement relation of the form

$$n^{\Sigma}[x, y] == \sum_{i} s^{\Sigma}[\![i, j]\!]\, n^{\Sigma}[2x - i, 2y - j].$$

Theorem 2.3, which follows, generalizes Theorem 2.1 and establishes the effect of integration in the direction $\{a, b\}$ on the subdivision mask for a scaling function.

THEOREM 2.3 Let $\tilde{\Sigma} = \Sigma \cup \{\{a, b\}\}$. If $s^{\Sigma}[x, y]$ is the subdivision mask for $n^{\Sigma}[x, y]$, the subdivision mask $s^{\tilde{\Sigma}}[x, y]$ for $n^{\tilde{\Sigma}}[x, y]$ has the form

$$s^{\tilde{\Sigma}}[x, y] = \frac{1}{2}(1 + x^{a} y^{b})\, s^{\Sigma}[x, y].$$

Proof The proof is entirely analogous to that of Theorem 2.1.

Recursively applying Theorem 2.3 to a set of direction vectors Σ, we note that the subdivision mask $s^{\Sigma}[x, y]$ has the explicit form

$$s^{\Sigma}[x, y] = 4 \prod_{\{a,b\}\in\Sigma} \frac{1 + x^a y^b}{2}. \tag{2.16}$$

The factor of four here arises from rewriting the mask $s^{\Sigma}[x, y]$ for the base case $\Sigma = \{\{1, 0\}, \{0, 1\}\}$ as $4\frac{(1+x)}{2}\frac{(1+y)}{2}$. If the vector $\{a, b\}$ appears multiple times in Σ, the corresponding factor $\frac{1+x^a y^b}{2}$ should appear to the appropriate power.

A box spline $p[x, y]$ is simply a linear combination of integer translates of the box-spline scaling function $n^{\Sigma}[x - i, y - j]$; that is, a sum of the form $\sum_{i,j} p_0[\![i, j]\!]$ $n^{\Sigma}[x-i, y-j]$, where $p_0[\![i, j]\!]$ is the entry of the vector p_0 attached to the point $\{i, j\}$ in the grid $\mathbb{Z}^2$. As in the univariate case, we write this sum in vector form as $\mathcal{N}^{\Sigma}[x, y]\,p_0$. Subdividing the entries of basis vector $\mathcal{N}^{\Sigma}[x, y]$ using the subdivision mask s^{Σ} allows the function $p[x, y]$ to be expressed as a sum of the form $\mathcal{N}^{\Sigma}[2^k x, 2^k y]\,p_k$. If $p_{k-1}[x, y]$ and $p_k[x, y]$ are the generating functions associated with successive coefficient vectors p_{k-1} and p_k, these generating functions are related via the expression

$$p_k[x, y] = s^{\Sigma}[x, y]\,p_{k-1}[x^2, y^2]. \tag{2.17}$$

Note that Theorem 2.3 makes clear the need to restrict the direction vector $\{a, b\}$ to the integer grid $\mathbb{Z}^2$. Allowing nonintegral values for a and b would give rise to a refinement relation for $\mathcal{N}^{\Sigma}[x, y]$ that does not involve members of the vector $\mathcal{N}^{\Sigma}[2x, 2y]$ and would therefore make this type of subdivision process impossible.

For box splines, the subdivision relation of equation 2.17 gives rise to four types of subdivision rules (as opposed to the two rules of equation 2.9 in the univariate case). These four rules take convex combination of p_{k-1} using the weights of the form $s^{\Sigma}[\![2i, 2j]\!]$, $s^{\Sigma}[\![2i + 1, 2j]\!]$, $s^{\Sigma}[\![2i, 2j + 1]\!]$, and $s^{\Sigma}[\![2i + 1, 2j + 1]\!]$, where $\{i, j\} \in \mathbb{Z}^2$.

In the one-dimensional case, the ith coefficient of p_k was plotted over $x == \frac{i}{2^k}$ to form a discrete approximation to the B-spline $p[x]$. In the case of box splines, the ijth coefficient of p_k is plotted at $\{x, y\} == \{\frac{i}{2^k}, \frac{j}{2^k}\}$. As long as the set of direction vectors Σ includes the initial directions $\{1, 0\}$ and $\{0, 1\}$, the polyhedra formed by the p_k converge to the box-spline function $p[x, y]$. For those readers interested in a proof of this convergence result, we recommend Cohen et al. [25] and Dahmen and Micchelli [33]. Chapter 3 also provides tools for addressing this question.

In the univariate case, Theorem 2.1 led to a repeated averaging algorithm for B-spline subdivision. For box splines, Theorem 2.3 yields a similar repeated averaging algorithm for box-spline subdivision. Given a set of coefficients p_{k-1} on the coarse grid $\frac{1}{2^{k-1}}\mathbb{Z}^2$, the algorithm computes a set of coefficients p_k on the refined grid $\frac{1}{2^k}\mathbb{Z}^2$ as follows:

- Construct the generating function $p_{k-1}[x, y]$ from p_{k-1}. Upsample $p_{k-1}[x, y]$ to yield $p_{k-1}[x^2, y^2]$. Set $p_k[x, y] = 4p_{k-1}[x^2, y^2]$.
- For each direction vector $\{a, b\} \in \Sigma$, update $p_k[x, y]$ via the recurrence $p_k[x, y] = \frac{(1+x^a y^b)}{2} p_k[x, y]$. Each multiplication by $\frac{(1+x^a y^b)}{2}$ corresponds to midpoint averaging on p_k in the direction $\{a, b\}$.
- Extract the coefficients p_k of the generating function $p_k[x, y]$.

2.2.4 Examples

We conclude this section by considering three examples of well-known box splines.

Three-direction Linear Splines

In Chapter 1, we considered a subdivision scheme for piecewise linear hat functions. These hat functions can be generalized to a triangular grid in two dimensions. If each triangle in the grid is subdivided into four subtriangles, the hat functions can be refined in a manner similar to that of the univariate case. For uniform triangular grids, this subdivision process can be expressed in terms of three-direction box splines. Let Σ consist of the set of three direction vectors $\{\{1, 0\}, \{0, 1\}, \{1, 1\}\}$. For $k == 0$, the resulting grid consists of triangles bounded by lines of the form $x == i$, $y == j$, and $x - y == h$, where $h, i, j \in \mathbb{Z}$. Applying equation 2.16, we obtain the subdivision mask for this scheme:

$$\begin{aligned} s^{\Sigma}[x, y] &= 4\frac{1 + x^1 y^0}{2}\,\frac{1 + x^0 y^1}{2}\,\frac{1 + x^1 y^1}{2} \\ &= \frac{(1 + x)(1 + y)(1 + xy)}{2}. \end{aligned}$$

The coefficients of this subdivision mask s^{Σ} can be visualized as a two-dimensional array of the form

$$s^{\Sigma}[x, y] == \begin{pmatrix} 1 & x & x^2 \end{pmatrix} \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & 1 & \frac{1}{2} \\ 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} 1 \\ y \\ y^2 \end{pmatrix}.$$

Note that this 3×3 matrix is not the subdivision matrix S but a plot of a single column of S as a two-dimensional array. The four subdivision rules for this scheme

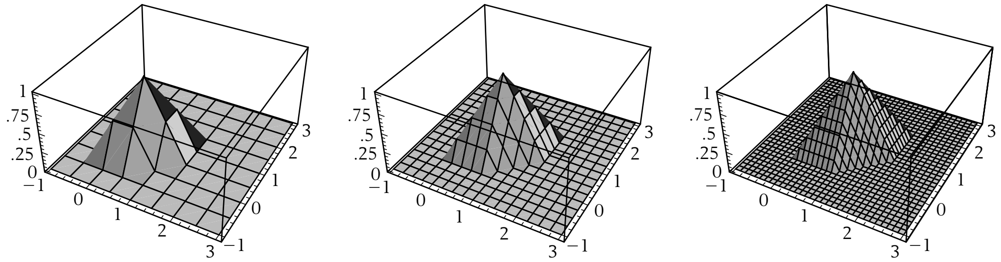


Figure 2.12 Three rounds of subdivision for the linear box-spline basis function.

correspond to taking convex combinations of the entries of the vector p_{k-1} using the weights

$$\begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix}, \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \end{pmatrix}, \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \end{pmatrix}, (1).$$

The basis function $n^{\Sigma}[x, y]$ associated with this scheme is piecewise linear and is supported over the hexagon consisting of all points of the form $\{1, 0\}t_1 + \{0, 1\}t_2 + \{1, 1\}t_3$, where $0 \leq t_i < 1$. The basis function is $\mathcal{C}^0$ (i.e., $\alpha == 2$) because removing any two vectors from Σ causes the remaining vector to fail to span the plane $\mathcal{R}^2$.

Figure 2.12 shows the results of three rounds of subdivision using this scheme, applied to the initial generating function $p_0[x, y] = 1$. In each plot, the coefficients of p_k are taken on the grid $\frac{1}{2^k}\mathbb{Z}$. Observe that these plots are converging to the piecewise linear hat function $n^{\Sigma}[x, y]$ on the three-direction grid defined by Σ, with the shifting of the hat function due to the use of an uncentered subdivision mask.

Four-direction Quadratic Box Splines

The piecewise linear hat function in the previous example is a continuous (i.e., $\mathcal{C}^0$) function. Adding an extra copy of the direction vector $\{1, 1\}$ to the three direction vectors $\{1, 0\}$, $\{0, 1\}$, and $\{1, 1\}$ for linear box splines yields a new set Σ of four direction vectors that defines a quadratic box spline. However, observe that $\alpha == 2$ for this case, in that removing $\{1, 0\}$ and $\{0, 1\}$ from Σ leaves the two copies of the vector $\{1, 1\}$ that together do not span the plane $\mathcal{R}^2$. Therefore, this quadratic box spline is only $\mathcal{C}^0$.

On the other hand, if we add a new vector in an independent direction—say $\{-1, 1\}$—to the direction vectors $\{1, 0\}$, $\{0, 1\}$, and $\{1, 1\}$, the resulting set Σ of direction vectors $\{\{1, 0\}, \{0, 1\}, \{1, 1\}, \{-1, 1\}\}$ defines a smooth box-spline scaling function, because $\alpha == 3$. Via equation 2.16, the subdivision mask for this box spline has the form

$$s^{\Sigma}[x, y] = 4 \prod_{(a,b)\in\Sigma} \frac{1 + x^a y^b}{2} == \frac{1}{4}(1 + x)(1 + y)(1 + xy)(1 + x^{-1}y).$$

Plotting the coefficients of $s^{\Sigma}[x, y]$ as a two-dimensional array yields

$$s^{\Sigma}[x, y] == \begin{pmatrix} x^{-1} & 1 & x & x^2 \end{pmatrix} \begin{pmatrix} 0 & \frac{1}{4} & \frac{1}{4} & 0 \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{2} & \frac{1}{4} \\ 0 & \frac{1}{4} & \frac{1}{4} & 0 \end{pmatrix} \begin{pmatrix} 1 \\ y \\ y^2 \\ y^3 \end{pmatrix}.$$

The four subdivision rules for this scheme correspond to taking convex combinations of the entries of the vector p_{k-1} using the weights

$$\begin{pmatrix} 0 & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{2} \end{pmatrix}, \begin{pmatrix} \frac{1}{4} & \frac{1}{2} \\ 0 & \frac{1}{4} \end{pmatrix}, \begin{pmatrix} \frac{1}{4} & 0 \\ \frac{1}{2} & \frac{1}{4} \end{pmatrix}, \begin{pmatrix} \frac{1}{2} & \frac{1}{4} \\ \frac{1}{4} & 0 \end{pmatrix}.$$

Figure 2.13 shows the results of three rounds of subdivision under this scheme, applied to the generating function $p_0[x, y] = 1$. The resulting polyhedra are converging to the scaling function $n^{\Sigma}[x, y]$. This function, $n^{\Sigma}[x, y]$, is a piecewise quadratic function supported over an octagon. Note that the integer translates of the scaling function $n^{\Sigma}[x, y]$ for this box spline are linearly dependent. In particular, choosing the initial coefficients p_0 for this box spline to satisfy $p_0[\![i, j]\!] = (-1)^{i+j}$ yields the function zero.

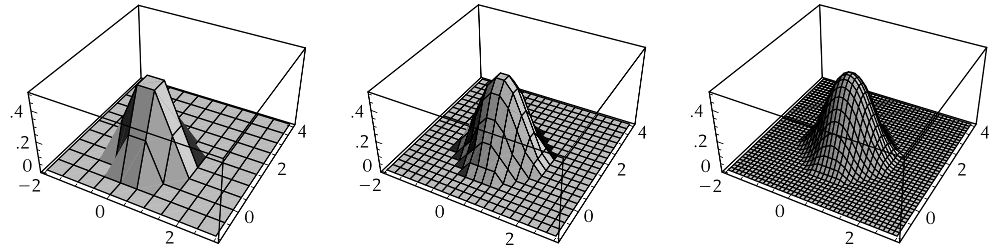


Figure 2.13 Three rounds of subdivision for the four-direction quadratic box-spline scaling function.

Three-direction Quartic Box Splines

We conclude with an example of a box spline whose scaling function has continuous second derivatives (i.e., is $\mathcal{C}^2$). This box spline, the three-direction quartic, is defined in terms of six direction vectors: the three direction vectors used in defining the linear hat function, each repeated. Specifically, the set of direction vectors Σ has the form $\{\{1, 0\}, \{1, 0\}, \{0, 1\}, \{0, 1\}, \{1, 1\}, \{1, 1\}\}$. For this example, $\alpha == 4$, and therefore the resulting spline is $\mathcal{C}^2$. According to equation 2.16, the subdivision mask for the scheme has the form

$$s^{\Sigma}[x, y] \; == \; 4 \prod_{(a,b)\in\Sigma} \frac{1 + x^a y^b}{2} = \frac{1}{16}(1+x)^2(1+y)^2(1+xy)^2.$$

Plotted in two dimensions, the coefficients of this subdivision mask have the form

$$s^{\Sigma}[x, y] == \begin{pmatrix} 1 & x & x^2 & x^3 & x^4 & x^5 \end{pmatrix} \begin{pmatrix} \frac{1}{16} & \frac{1}{8} & \frac{1}{16} & 0 & 0 \\ \frac{1}{8} & \frac{3}{8} & \frac{3}{8} & \frac{1}{8} & 0 \\ \frac{1}{16} & \frac{3}{8} & \frac{5}{8} & \frac{3}{8} & \frac{1}{16} \\ 0 & \frac{1}{8} & \frac{3}{8} & \frac{3}{8} & \frac{1}{8} \\ 0 & 0 & \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \end{pmatrix} \begin{pmatrix} 1 \\ y \\ y^2 \\ y^3 \\ y^4 \\ y^5 \end{pmatrix}.$$

The four subdivision rules for this scheme correspond to taking convex combinations of the entries of the vector p_{k-1} using the weights

$$\begin{pmatrix} \frac{1}{16} & \frac{1}{16} & 0 \\ \frac{1}{16} & \frac{5}{8} & \frac{1}{16} \\ 0 & \frac{1}{16} & \frac{1}{16} \end{pmatrix}, \begin{pmatrix} \frac{1}{8} & \frac{3}{8} & 0 \\ 0 & \frac{3}{8} & \frac{1}{8} \end{pmatrix}, \begin{pmatrix} \frac{1}{8} & 0 \\ \frac{3}{8} & \frac{3}{8} \\ 0 & \frac{1}{8} \end{pmatrix}, \begin{pmatrix} \frac{3}{8} & \frac{1}{8} \\ \frac{1}{8} & \frac{3}{8} \end{pmatrix}.$$

Figure 2.14 depicts three rounds of subdivision based on this scheme, starting from the initial generating function $p_0[x, y] = 1$. The resulting plot is an approximation of the basis function $n^{\Sigma}[x, y]$, a piecewise quartic function defined over a hexagon, with a support that is twice the size of the support hexagon for the hat function. Figure 2.15 shows an example of a toroidal surface constructed using a parametric version of the three-direction quartic box spline in which one coordinate is treated as being periodic.

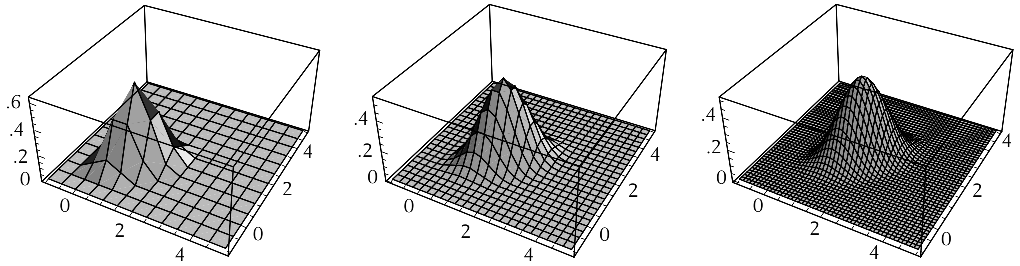


Figure 2.14 Three rounds of subdivision for the three-direction quartic box-spline basis function.

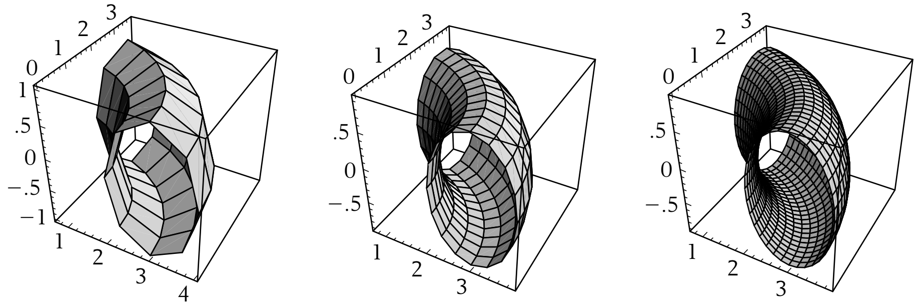


Figure 2.15 A parametric surface constructed using subdivision for a three-direction quartic box spline.

2.3 B-splines and Box Splines as Piecewise Polynomials

In the preceding examples, the plots of the box-spline scaling functions $n^{\Sigma}[x, y]$ were constructed via subdivision. Starting with the initial generating function $p_0[x, y] = 1$, we applied several rounds of subdivision to compute $p_k[x, y]$ and plotted the resulting coefficients of p_k on the grid $\frac{1}{2^k}\mathbb{Z}^2$. In practice, this approach of computing discrete approximations is an extremely effective method for manipulating box splines. However, occasionally it is useful to have access to the piecewise polynomial representation of the box spline. For example, in finite element analysis, knowing the exact value of the finite element basis functions at certain quadrature points is very useful in constructing the inner product matrix associated with the basis functions. Using the piecewise polynomial representation, the box spline can be evaluated exactly at those quadrature points. Further, all of the results from classical functional analysis can be applied to the piecewise polynomial representation to investigate the properties of a given box spline.

Attempting to construct the piecewise polynomial representation of a box-spline scaling function based on the support and smoothness conditions for the box spline is quite difficult. The number of polynomial pieces is large, and the linear relationships between coefficients for adjacent polynomial pieces are tedious to construct. In this final section, we discuss an elegant technique for constructing the piecewise polynomial representation for B-splines and box splines. For B-splines, this technique expresses the basis functions $n^{\mathrm{m}}[x]$ as a linear combination of truncated powers. For box splines, the scaling functions $n^{\Sigma}[x, y]$ are expressed as a linear combination of cone splines, the cross-sectional volumes of high-dimensional cones. As we will see later, the differential approach to subdivision, introduced in Chapter 4, is closely tied to these results.

2.3.1 B-splines as Combinations of Truncated Powers

In section 2.2.1, we expressed the B-spline basis function $n^{\mathrm{m}}[x]$ as the cross-sectional volume of the hypercube $\mathcal{H}^{\mathrm{m}}$. In this section, we consider the cross-sectional volume $c^{\mathrm{m}}[x]$ of a related cone and show that $n^{\mathrm{m}}[x]$ can be expressed as a linear combination of translates $c^{\mathrm{m}}[x - \mathrm{i}]$. If $\mathcal{R}^{+}$ denotes the half-interval consisting of all non-negative real numbers, the *cone* $(\mathcal{R}^{+})^{\mathrm{m}}$ consists of those points $\{t_1, \ldots, t_{\mathrm{m}}\}$ such that $0 \le t_{\mathrm{i}}$ for $1 \le \mathrm{i} \le \mathrm{m}$. Just as was done for hypercubes, we consider the cross-sectional volumes of the form

$$c^{\mathrm{m}}[x] = \frac{1}{\sqrt{\mathrm{m}}}\,\mathrm{vol}_{\mathrm{m}-1}\left[\left\{\{t_1, \ldots, t_{\mathrm{m}}\} \in (\mathcal{R}^{+})^{\mathrm{m}} \;\middle|\; \sum_{\mathrm{i}=1}^{\mathrm{m}} t_{\mathrm{i}} == x\right\}\right]. \tag{2.18}$$

Because the interval $\mathcal{H}$ can be expressed as the difference of the half-interval $\mathcal{R}^{+}$ and its integer translate in the positive direction—that is, the interval $[0, 1)$ is the difference of the intervals $[0, \infty)$ and $[1, \infty)$—the hypercube $\mathcal{H}^{\mathrm{m}}$ can be decomposed into an alternating sum of 2^{m} integer translates of the cone $(\mathcal{R}^{+})^{\mathrm{m}}$. This decomposition induces an interesting relation between cross-sectional volumes of these objects: the B-spline basis function $n^{\mathrm{m}}[x]$ and integer translates of the function $c^{\mathrm{m}}[x]$.

For example, in Figure 2.10, a piecewise linear hat function could be viewed as the cross-sectional length of a square $\mathcal{H}^2$. In Figure 2.16, the square $\mathcal{H}^2$ has been expressed as the alternating sum of four translates of the cone $(\mathcal{R}^{+})^2$, each with a vertex at the corner of the square and opening toward the right. The leftmost cone contains the square. The top and bottom cones are subtracted from the leftmost cone, leaving the square. The top and bottom cones overlap, forming the rightmost cone. Adding this rightmost cone back accounts for this overlap and leaves the original square.

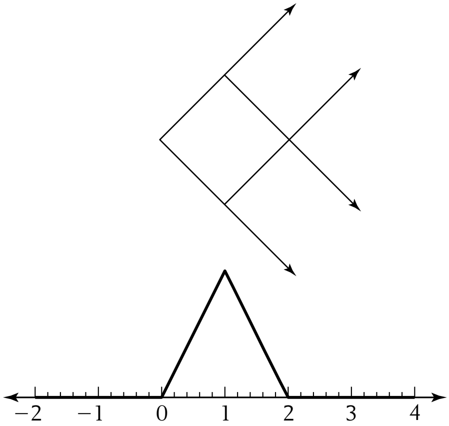


Figure 2.16 A square expressed as the alternating sum of translated cones.

In a similar manner, the cross-sectional length of the square can be expressed as a linear combination of the cross-sectional lengths of these cones. For this particular example, this relation can be expressed as

$$n^2[x] = c^2[x] - 2c^2[x-1] + c^2[x-2],$$

where $c^2[x]$ is the cross-sectional length of the cone $(\mathcal{R}^+)^2$. Our goal in this section is to precisely capture the linear relationship between the functions $n^{\mathsf{m}}[x]$ and the integer translates $c^{\mathsf{m}}[x - \mathsf{i}]$ for arbitrary order m.

As was the case for B-splines, we redefine the function $c^{\mathsf{m}}[x]$ inductively using repeated integration. The advantage of this inductive approach is that the linear relationship between $n^{\mathsf{m}}[x]$ and the integer translates $c^{\mathsf{m}}[x-\mathsf{i}]$ can also be computed recursively. The base case for our inductive definition of $c^{\mathsf{m}}[x]$ starts at $\mathsf{m} == 1$. In particular, we define $c^1[x]$ to be 1 if $x \geq 0$, and zero otherwise. Given the function $c^{\mathsf{m}-1}[x]$, we define $c^{\mathsf{m}}[x]$ to be the integral

$$c^{\mathsf{m}}[x] = \int_0^\infty c^{\mathsf{m}-1}[x-t]\, \mathbb{d}t. \tag{2.19}$$

Again, the equivalence of this recursive definition to the definition of equation 2.18 follows in a manner almost identical to the proof of Theorem 2.2. Based on this inductive definition, we can show that $c^{\mathsf{m}}[x] = \frac{1}{(\mathsf{m}-1)!}x^{\mathsf{m}-1}$ if $x \geq 0$, and zero otherwise. These functions are the *truncated powers* of x, often written as $\frac{1}{(\mathsf{m}-1)!}x_+^{\mathsf{m}-1}$ in standard B-spline notation. Figure 2.17 shows plots of the functions $c^{\mathsf{m}}[x]$ for small values of m.

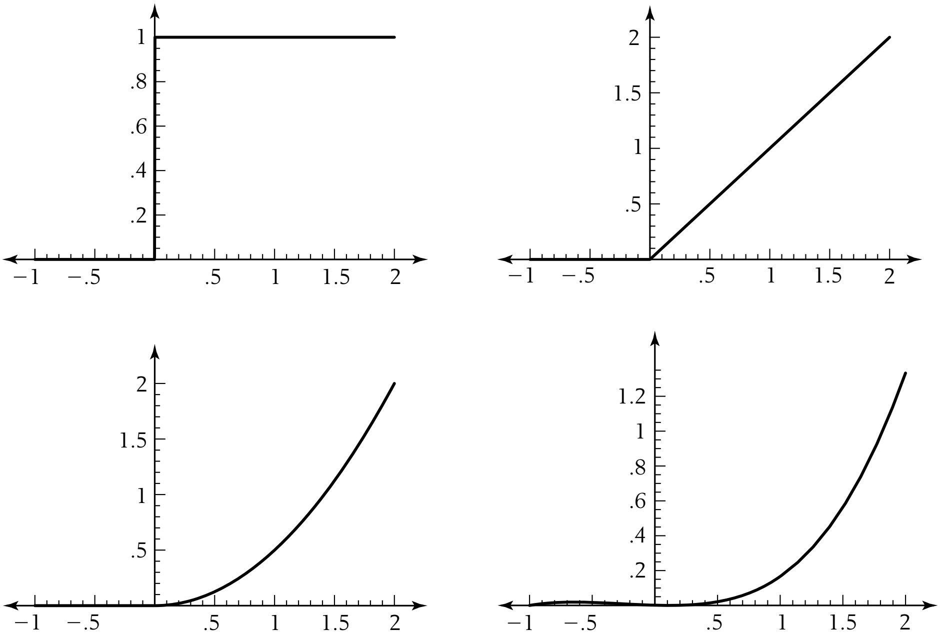


Figure 2.17 Plots of truncated powers $c^{\mathsf{m}}[x]$ for $\mathsf{m} = 1, \ldots, 4$.

Given this recursive definition for the truncated powers $c^{\mathsf{m}}[x]$, the following theorem captures the exact relationship between the B-spline basis function $n^{\mathsf{m}}[x]$ and the integer translates of truncated powers $c^{\mathsf{m}}[x]$.

THEOREM 2.4 If $n^{\mathsf{m}}[x]$ and $c^{\mathsf{m}}[x]$ are, respectively, the B-spline basis functions and the truncated powers of order m, then

$$n^{\mathsf{m}}[x] = \sum_{\mathsf{i}} \mathsf{d}^{\mathsf{m}}\llbracket \mathsf{i} \rrbracket \, c^{\mathsf{m}}[x - \mathsf{i}],$$

where $\mathsf{d}^{\mathsf{m}}[\mathsf{x}]$ is the generating function $(1 - \mathsf{x})^{\mathsf{m}}$.

Proof The proof proceeds by induction on m. For $\mathsf{m} == 1$, the theorem holds by observation, in that $n^1[x] = c^1[x] - c^1[x - 1]$, where $\mathsf{d}^1[\mathsf{x}] = 1 - \mathsf{x}$. Next, assume that the theorem holds for splines of order $\mathsf{m} - 1$; that is,

$$n^{\mathsf{m}-1}[x] = \sum_{\mathsf{i}} \mathsf{d}^{\mathsf{m}-1}\llbracket \mathsf{i} \rrbracket \, c^{\mathsf{m}-1}[x - \mathsf{i}],$$

where $\mathsf{d}^{\mathsf{m}-1}[\mathsf{x}] = (1-\mathsf{x})^{\mathsf{m}-1}$. We now show that the theorem holds for splines of order m. By definition, $n^{\mathsf{m}}[x]$ is exactly $\int_0^1 n^{\mathsf{m}-1}[x-t]\,dt$. Substituting the inductive hypothesis for $n^{\mathsf{m}-1}[x]$ yields

$$\begin{aligned} n^{\mathsf{m}}[x] &= \int_0^1 \sum_{\mathsf{i}} \mathsf{d}^{\mathsf{m}-1}[\![\mathsf{i}]\!]\, c^{\mathsf{m}-1}[x-t-\mathsf{i}]\,dt, \\ &= \sum_{\mathsf{i}} \mathsf{d}^{\mathsf{m}-1}[\![\mathsf{i}]\!] \int_0^1 c^{\mathsf{m}-1}[x-t-\mathsf{i}]\,dt. \end{aligned}$$

The inner integral $\int_0^1 c^{\mathsf{m}-1}[x-t-\mathsf{i}]\,dt$ can be rewritten as $\int_0^\infty c^{\mathsf{m}-1}[x-t-\mathsf{i}]\,dt - \int_1^\infty c^{\mathsf{m}-1}[x-t-\mathsf{i}]\,dt$, which is exactly $c^{\mathsf{m}}[x-\mathsf{i}] - c^{\mathsf{m}}[x-\mathsf{i}-1]$. Substituting this expression into the previous equation, we obtain

$$\begin{aligned} n^{\mathsf{m}}[x] &= \sum_{\mathsf{i}} \mathsf{d}^{\mathsf{m}-1}[\![\mathsf{i}]\!](c^{\mathsf{m}}[x-\mathsf{i}] - c^{\mathsf{m}}[x-\mathsf{i}-1]), \\ &= \sum_{\mathsf{i}} (\mathsf{d}^{\mathsf{m}-1}[\![\mathsf{i}]\!] - \mathsf{d}^{\mathsf{m}-1}[\![\mathsf{i}-1]\!])c^{\mathsf{m}}[x-\mathsf{i}]. \end{aligned}$$

The coefficients $\mathsf{d}^{\mathsf{m}-1}[\![\mathsf{i}]\!] - \mathsf{d}^{\mathsf{m}-1}[\![\mathsf{i}-1]\!]$ of this new relation are exactly the coefficients of $(1-\mathsf{x})\mathsf{d}^{\mathsf{m}-1}[\mathsf{x}] == \mathsf{d}^{\mathsf{m}}[\mathsf{x}]$.

One particularly useful application of this theorem lies in evaluating $n^{\mathsf{m}}[x]$. To evaluate this B-spline basis function at a particular value of x, we simply evaluate the corresponding truncated power $c^{\mathsf{m}}[x]$ at $\mathsf{m}+1$ values and multiply by various binomial coefficients. Evaluating $c^{\mathsf{m}}[x]$ is easy, given its simple definition. However, one word of caution: although the value of this expression should always be zero outside of the range $[0, \mathsf{m}]$, numerical errors can cause this evaluation method to be unstable and return non-zero values. We suggest that any evaluation method using this expression explicitly test whether $x \in [0, \mathsf{m}]$ and return zero if $x \notin [0, \mathsf{m}]$.

Note here that the coefficients of $\mathsf{d}^{\mathsf{m}}[\mathsf{x}]$ are the mth discrete differences with respect to an integer grid. For those readers familiar with the theory of B-splines, Theorem 2.4 may be viewed as expressing the B-spline basis function of order m as the mth divided difference of the function $x_+^{\mathsf{m}-1}$ with respect to an integer grid. (See De Boor [38] and Schumaker [137] for a complete exposition on divided differences and B-splines.) One advantage of this view is that this relation also holds for nonuniform grids. In fact, it is often used as the initial definition for B-spline basis functions.

2.3.2 Box Splines as Combinations of Cone Splines

Section 2.2.2 defined a box-spline scaling function $n^{\Sigma}[x, y]$ as the cross-sectional volume of a hypercube $\mathcal{H}^{\mathrm{m}}$ based on a set of direction vectors Σ. By replacing the hypercube $\mathcal{H}^{\mathrm{m}}$ with the cone $(\mathcal{R}^{+})^{\mathrm{m}}$ in this definition, we can define a related spline, known as a *cone spline*, $c^{\Sigma}[x, y]$. As before, viewing the hypercube $\mathcal{H}^{\mathrm{m}}$ as an alternating sum of integer translates of the cone $(\mathcal{R}^{+})^{\mathrm{m}}$ leads to a linear relationship between the box-spline scaling function $n^{\Sigma}[x, y]$ and integer translates of the cone spline $c^{\Sigma}[x, y]$. Our task in this section is to capture this relationship precisely.

Just as for box splines, our approach is to characterize the *cone splines* recursively using repeated integration. For the base case, the set Σ of direction vectors consists of the two standard unit vectors $\{1, 0\}$ and $\{0, 1\}$. In this case, the cone spline $c^{\Sigma}[x, y]$ has the form

$$c^{\Sigma}[x, y] = \begin{cases} 1 & \text{if } 0 \le x \text{ and } 0 \le y, \\ 0 & \text{otherwise.} \end{cases} \tag{2.20}$$

Larger sets of direction vectors are formed by inductively adding new direction vectors to this initial set. Given an existing set of direction vectors Σ and its associated cone spline $c^{\Sigma}[x, y]$, the cone spline associated with the set $\widetilde{\Sigma} = \Sigma \cup \{\{\mathrm{a}, \mathrm{b}\}\}$ has the form

$$c^{\widetilde{\Sigma}}[x, y] = \int_{0}^{\infty} c^{\Sigma}[x - \mathrm{a}t, y - \mathrm{b}t]\, dt. \tag{2.21}$$

Cone splines, sometimes also referred to as multivariate truncated powers, were introduced by Dahmen in [30]. Dahmen and others have further analyzed the behavior of these splines in [29], [26], and [103]. If Σ contains m direction vectors, the basic properties of the cone spline $c^{\Sigma}[x, y]$ are summarized as follows:

- $c^{\Sigma}[x, y]$ is a piecewise polynomial of degree $\mathrm{m} - 2$, with $\mathcal{C}^{\alpha-2}$ continuity, where α is the size of the smallest subset $\mathrm{A} \subset \Sigma$ such that the complement of A in Σ does not span $\mathbb{R}^2$.
- Each polynomial piece of $c^{\Sigma}[x, y]$ is supported on a cone with a vertex at the origin, bounded by two vectors in Σ. Inside each of these regions in $c^{\Sigma}[x, y]$ is a homogeneous polynomial of degree $\mathrm{m} - 2$ in x and y.

Due to this homogeneous structure, the cone spline $c^{\Sigma}[x, y]$ satisfies a *scaling relation* of the form $c[x, y] == 2^{2-\mathrm{m}} c[2x, 2y]$, where m is the number of direction vectors in Σ. Before considering several examples of common cone splines, we conclude this section by proving that the box-spline scaling function $n^{\Sigma}[x, y]$ can be expressed as a linear combination of integer translates of the cone spline $c^{\Sigma}[x, y]$. In this theorem, due to Dahmen and Micchelli [32], the coefficients used in this linear combination encode directional differences taken with respect to Σ.

THEOREM 2.5 If $n^{\Sigma}[x, y]$ and $c^{\Sigma}[x, y]$ are, respectively, the box-spline scaling function and cone spline associated with the set of direction vectors Σ, then

$$n^{\Sigma}[x, y] = \sum_{\mathrm{i,j}} \mathrm{d}^{\Sigma}[\![\mathrm{i, j}]\!] c^{\Sigma}[x - \mathrm{i}, y - \mathrm{j}],$$

where $\mathrm{d}^{\Sigma}[\mathrm{x, y}] = \prod_{\{\mathrm{a,b}\} \in \Sigma} (1 - \mathrm{x}^{\mathrm{a}}\mathrm{y}^{\mathrm{b}})$.

Proof The proof proceeds by induction on the size of Σ. The base case is $\Sigma = \{\{1, 0\}, \{0, 1\}\}$. In this case, $\mathrm{d}^{\Sigma}[\mathrm{x, y}] = (1 - \mathrm{x})(1 - \mathrm{y})$. One can explicitly verify that

$$n^{\Sigma}[x, y] == c^{\Sigma}[x, y] - c^{\Sigma}[x - 1, y] - c^{\Sigma}[x, y - 1] + c^{\Sigma}[x - 1, y - 1].$$

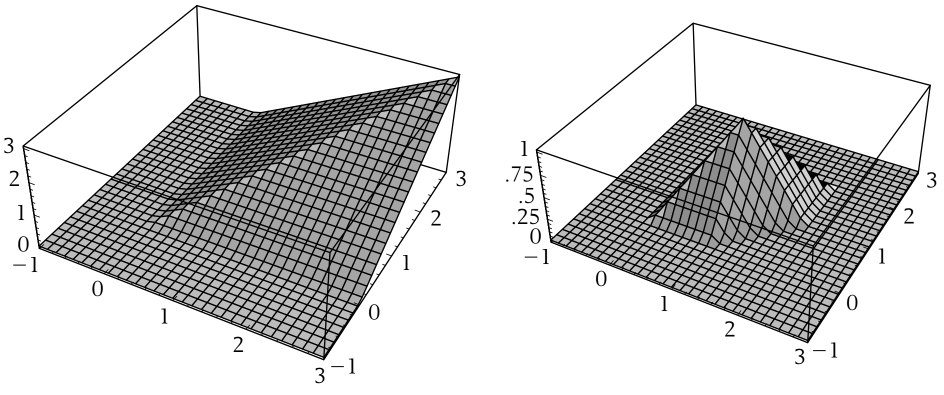
For larger sets of direction vectors Σ, the inductive portion of the proof continues in complete analogy to the univariate proof of Theorem 2.4.

2.3.3 Bivariate Examples

Section 2.2 considered three examples of box splines: the three-direction linear, the four-direction quadratic, and the three-direction quartic. We conclude this chapter by constructing the cone splines corresponding to these box splines. By deriving the difference mask $\mathrm{d}^{\Sigma}[\mathrm{x, y}]$ and applying Theorem 2.5 to the cone splines, we construct a relatively simple piecewise polynomial representation of the corresponding box splines.

Three-direction Linear Cone Splines

Our first example is the three-direction linear cone spline. As for the three-direction linear box spline, the three directions are $\Sigma = \{\{1, 0\}, \{0, 1\}, \{1, 1\}\}$. The cone spline



(a)

(b)

Figure 2.18 The (a) linear cone spline and (b) linear box-spline basis function obtained as a difference of integer translates of the cone spline.

$c^{\Sigma}[x, y]$ consists of two wedge-shaped linear pieces of the form

$$c^{\Sigma}[x, y] = \begin{cases} y & \text{if } y \geq 0 \text{ and } x \geq y, \\ x & \text{if } x \geq 0 \text{ and } x \leq y, \\ 0 & \text{otherwise.} \end{cases}$$

The left-hand portion of Figure 2.18 shows a plot of the cone spline $c^{\Sigma}[x, y]$ on the square $[-1, 3]^2$. Via Theorem 2.5, the difference mask $d^{\Sigma}[x, y]$ has the form $(1 - x)(1 - y)(1 - xy)$. Plotting the coefficients of this generating function as a two-dimensional array yields

$$d^{\Sigma}[x, y] == \begin{pmatrix} 1 & x & x^2 \end{pmatrix} \begin{pmatrix} 1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ y \\ y^2 \end{pmatrix}.$$

The right-hand portion of Figure 2.18 shows a plot of the box-spline scaling function $n^{\Sigma}[x, y]$ computed using Theorem 2.5.

Four-direction Quadratic Cone Splines

The next example is the four-direction quadratic cone spline. For this spline, Σ has the form $\{\{1, 0\}, \{0, 1\}, \{1, 1\}, \{-1, 1\}\}$. Applying the recursive definition for cone

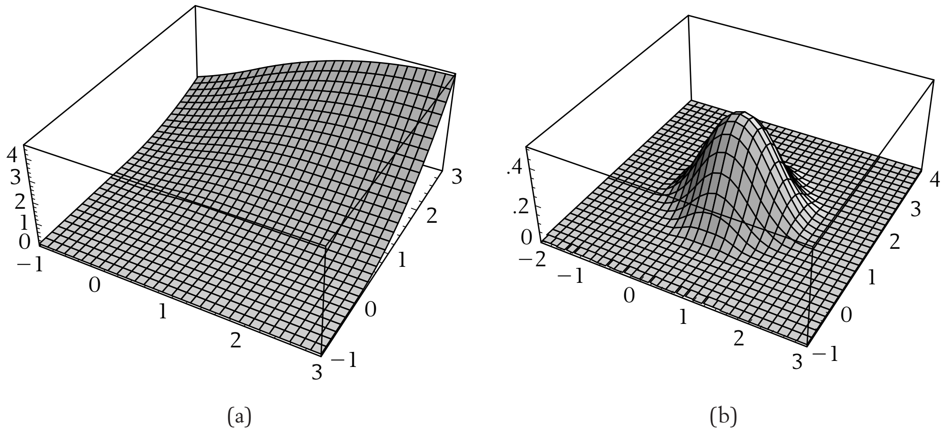


Figure 2.19 The (a) four-direction quadratic cone spline and (b) four-direction quadratic box-spline scaling function obtained as a difference of integer translates of the cone spline.

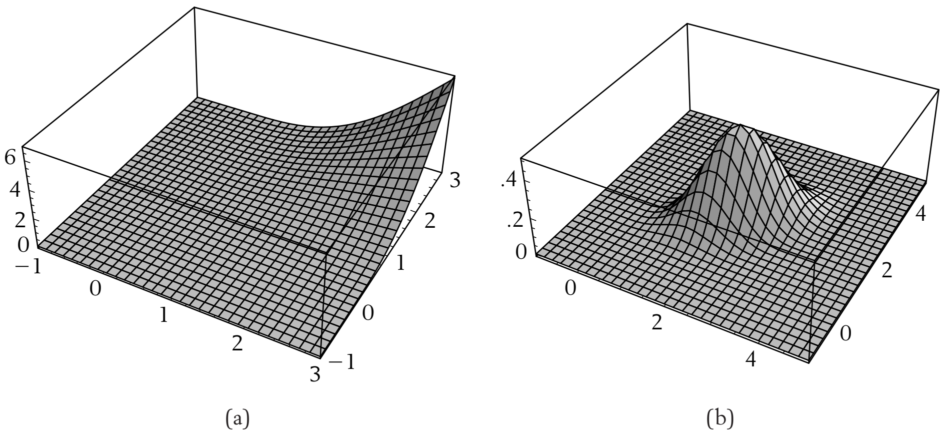
splines to the previous example of the three-direction linear cone spline yields the cone spline $c^{\Sigma}[x, y]$, with three quadratic, wedge-shaped pieces. These pieces satisfy

$$c^{\Sigma}[x,y] = \begin{cases} \frac{y^2}{2} & \text{if } y \geq 0 \text{ and } x \geq y, \\ \frac{1}{4}(-x^2 + 2xy + y^2) & \text{if } x \geq 0 \text{ and } x \leq y, \\ \frac{1}{4}(x+y)^2 & \text{if } x \leq 0 \text{ and } x \geq -y, \\ 0 & \text{otherwise.} \end{cases}$$

The left-hand portion of Figure 2.19 shows a plot of $c^{\Sigma}[x, y]$ on the square $[-1, 3]^2$. Via Theorem 2.5, the difference mask $d^{\Sigma}[x, y]$ has the form $(1 - x)(1 - y)(1 - xy)(1 - x^{-1}y)$. Plotting the coefficients of this generating function as a two-dimensional array yields

$$d^{\Sigma}[x, y] == (x^{-1} \quad 1 \quad x \quad x^2) \begin{pmatrix} 0 & -1 & 1 & 0 \\ 1 & 0 & 0 & -1 \\ -1 & 0 & 0 & 1 \\ 0 & 1 & -1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ y \\ y^2 \\ y^3 \end{pmatrix}.$$

The right-hand portion of Figure 2.19 shows a plot of the box-spline scaling function $n^{\Sigma}[x, y]$ computed using Theorem 2.5.



(a)

(b)

Figure 2.20 The (a) three-direction quartic cone spline and (b) three-direction quartic box-spline basis function as a difference of integer translates of the cone spline.

Three-direction Quartic Cone Splines

Our final example is the three-direction quartic cone spline. Here, Σ has the form {{1, 0}, {1, 0}, {0, 1}, {0, 1}, {1, 1}, {1, 1}}. This cone spline, $c^{\Sigma}[x, y]$, consists of two quartic, wedge-shaped pieces. These pieces satisfy

$$c^{\Sigma}[x, y] = \begin{cases} \frac{1}{12}y^3(2x - y) & \text{if } y \geq 0 \text{ and } x \geq y, \\ \frac{1}{12}x^3(2y - x) & \text{if } x \geq 0 \text{ and } x \leq y, \\ 0 & \text{otherwise.} \end{cases}$$

The left-hand portion of Figure 2.20 shows a plot of $c^{\Sigma}[x, y]$ on the interval $[-1, 3]^2$. Via Theorem 2.5, the difference mask $d^{\Sigma}[x, y]$ has the form $(1 - x)^2(1 - y)^2(1 - xy)^2$. Plotting the coefficients of this generating function as a two-dimensional array yields

$$d^{\Sigma}[x, y] == (1 \quad x \quad x^2 \quad x^3 \quad x^4) \begin{pmatrix} 1 & -2 & 1 & 0 & 0 \\ -2 & 2 & 2 & -2 & 0 \\ 1 & 2 & -6 & 2 & 1 \\ 0 & -2 & 2 & 2 & -2 \\ 0 & 0 & 1 & -2 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ y \\ y^2 \\ y^3 \\ y^4 \end{pmatrix}.$$

The right-hand portion of Figure 2.20 shows a plot of the box-spline basis function $n^{\Sigma}[x, y]$ computed using Theorem 2.5.

CHAPTER 3

Convergence Analysis for Uniform Subdivision Schemes

Chapter 2 introduced a simple tool, repeated integration, for creating interesting subdivision schemes. In the univariate case, repeated integration led to subdivision masks of the form $s[x] = 2(\frac{1+x}{2})^m$, whose coefficients defined the refinement relation for the B-spline basis function of order m. In the bivariate case, repeated integration led to a subdivision mask $s[x, y]$ of the form $4\prod_{\{a,b\}\in\Sigma} \frac{1+x^a y^b}{2}$ for box splines. The coefficients of this subdivision mask defined the refinement relation for the box-spline scaling function associated with the set of direction vectors Σ. Essentially, the previous chapter gave us tools for constructing subdivision masks $s[x]$ (or $s[x, y]$) whose subdivision schemes are "nice." In the context of subdivision, "nice" usually means that the limit of the subdivision process always exists (i.e., the scheme is convergent) and that the limit function is smooth.

This chapter considers the converse problem: Given a subdivision mask $s[x]$ (or $s[x, y]$), does the associated subdivision scheme define a "nice" curve (or surface)? For the schemes of the previous chapter, this question may not seem particularly important, in that the limiting splines had a definition in terms of repeated integration. However, for many subdivision schemes (especially interpolatory ones), there is no known piecewise representation; the only information known about the scheme is the subdivision mask $s[x]$. In this case, we would like to understand the behavior of the associated subdivision process in terms of the structure of the subdivision mask $s[x]$ alone.

Given a subdivision mask $s[x]$ and an initial set of coefficients p_0, the subdivision process $p_k[x] = s[x]p_{k-1}[x^2]$ defines an infinite sequence of generating functions. The basic approach of this chapter is to associate a piecewise linear function $p_k[x]$ with the coefficients of the generating function $p_k[x]$ and to analyze the behavior of this sequence of functions. The first section reviews some mathematical basics necessary to understand convergence for a sequence of functions. The next section derives sufficient conditions on the mask $s[x]$ for the functions $p_k[x]$ to converge to a continuous function $p_\infty[x]$. The final section considers the bivariate version of this question and derives sufficient conditions on the mask $s[x, y]$ for the functions $p_k[x, y]$ to converge to a continuous function $p_\infty[x, y]$.

3.1 Convergence of a Sequence of Functions

The previous chapter considered subdivision schemes that took as input a coarse vector p_0 and produced a sequence of increasingly dense vectors p_k. In plotting these vectors, we constructed a sequence of piecewise linear functions $p_k[x]$ that interpolated the ith coefficient of p_k at $x == \frac{i}{2^k}$; that is,

$$p_k\left[\frac{i}{2^k}\right] = p_k[\![i]\!].$$

For B-splines, we observed that these functions $p_k[x]$ appear to converge to a limit curve that is the B-spline curve associated with the initial control points p_0. This section reviews the basic mathematical ideas necessary to make this construction precise and to prove some associated theorems. For those readers interested in more background material, we suggest Chapter 18 of Taylor [151], the source of most of the material in this section. For less mathematically inclined readers, we suggest skipping this section and proceeding to section 3.2. There, we derive a simple computational test to determine whether a subdivision scheme is convergent.

3.1.1 Pointwise Convergence

Before considering sequences of functions, we first review the definition of convergence for a sequence of real numbers, $x_0, x_1, x_2, \ldots$. This sequence *converges* to a limit x_∞ if for all $\epsilon > 0$ there exists n such that $|x_\infty - x_k| < \epsilon$ for all $k \geq n$. This limit is denoted by $\lim_{k\to\infty} x_k = x_\infty$. For example, consider the limit of the sequence $x_k = \frac{2^{-k}}{e^{2^{-k}}-1}$. As $k \to \infty$, 2^{-k} goes to zero, and the function $e^{2^{-k}} - 1$ also goes to zero. What is the value of their ratio as $k \to \infty$? Recall that the exponential function e^α has a power series expansion of the form $\sum_{i=0}^{\infty} \frac{1}{i!}\alpha^i$. Substituting $\alpha = 2^{-k}$ yields x_k of the form

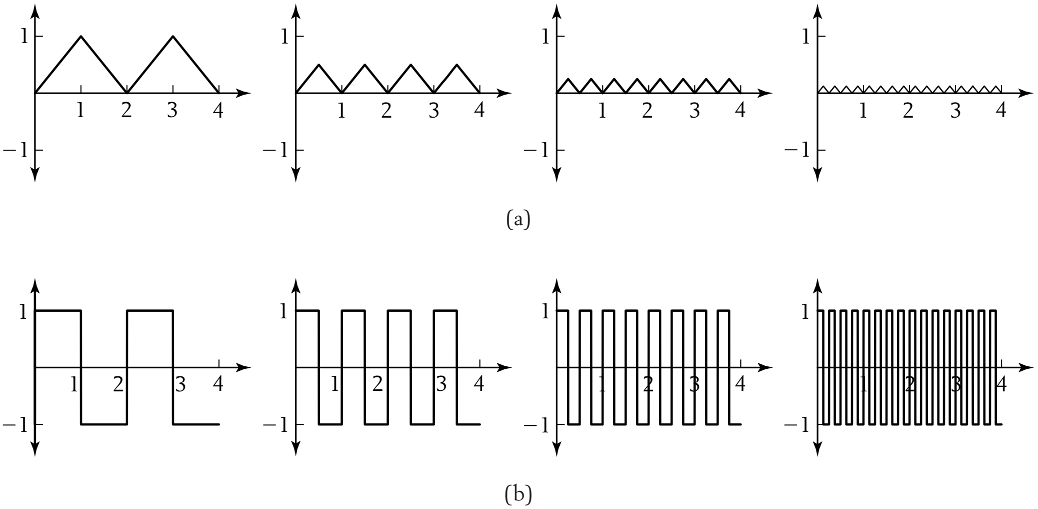
$$x_k = \frac{2^{-k}}{\left(1 + 2^{-k} + \frac{1}{2}(2^{-k})^2 + \cdots\right) - 1} == \frac{2^{-k}}{2^{-k}\left(1 + \frac{1}{2}(2^{-k}) + \cdots\right)} == \frac{1}{1 + \frac{1}{2}(2^{-k}) + \cdots}.$$

Clearly, as $k \to \infty$, the terms 2^{-k} go to zero and the limit x_∞ is exactly 1.

Given a sequence of functions $p_0[x]$, $p_1[x]$, $p_2[x], \ldots$, there are several ways to define the limit of this sequence of functions. The simplest definition, *pointwise convergence*, defines the limit $\lim_{k\to\infty} p_k[x] = p_\infty[x]$ independently for each possible value of x.

The main drawback of pointwise convergence is that properties that are true for a sequence of functions $p_k[x]$ may not be true for their pointwise limit $p_\infty[x]$. For example, consider the sequence of functions $p_k[x] = x^k$. On the interval $0 \leq x \leq 1$, the limit function $p_\infty[x]$ is zero if $x < 1$ and one if $x == 1$. However, this function $p_\infty[x]$ is discontinuous, while for any k the monomial $p_k[x]$ is continuous. Thus, the pointwise limit of a sequence of continuous functions is not necessarily a continuous function.

Another drawback is that the derivatives of a sequence of functions $p_k[x]$ do not necessarily converge to the derivative of their pointwise limit. For example, consider the piecewise linear function $p_k[x]$ whose value at $\frac{i}{2^k}$ is 2^{-k} if i is odd and zero if i is even. The top portion of Figure 3.1 shows plots of these sawtooth functions for $k == 0, 1, 2, 3$. Clearly, the pointwise limit of the functions $p_k[x]$ as $k \to \infty$ is simply the function zero. However, note that the derivative $p'_k[x]$ of these functions has value ± 1 for any value of k and does not converge to zero (i.e., the derivative of the function zero) in any sense. The lower portion of Figure 3.1 shows plots of these derivatives.



(a)

(b)

Figure 3.1 A sequence of functions converging to zero (a) whose derivatives (b) do not converge to zero.

3.1.2 Uniform Convergence

The difficulties alluded to in the previous section lie in the definition of pointwise convergence. Given an ϵ, each value of x can have a distinct n associated with it in the definition of pointwise convergence. A stronger definition would require one value of n that holds for all x simultaneously. To make this approach work, we need a method for computing the difference of two functions over their entire domain. To this end, we define the *norm* of a function $p[x]$, denoted by $\|p[x]\|$, to be the maximum of the absolute of $p[x]$ over all possible x; that is,

$$\|p[x]\| = \underset{x}{\mathsf{Max}}\,|p[x]|.$$

In standard numerical analysis terminology, this norm is the infinity norm of $p[x]$. (Observe that one needs to use care in applying this norm to functions defined on infinite domains, because Max is not always well defined for some functions, such as $p[x] = x$.) Based on this norm, we define a new, stronger type of convergence for sequences of functions. A sequence of functions $p_{\mathsf{k}}[x]$ *uniformly* converges to a limit function $p_\infty[x]$ if for all $\epsilon > 0$ there exists an n such that for all $\mathsf{k} \geq \mathsf{n}$

$$\|p_\infty[x] - p_{\mathsf{k}}[x]\| < \epsilon.$$

Figure 3.2 illustrates this definition. For $\mathsf{k} \geq \mathsf{n}$, each function $p_{\mathsf{k}}[x]$ must lie in the ribbon bounded above by $p_\infty[x] + \epsilon$ and bounded below by $p_\infty[x] - \epsilon$.

When proving that a sequence of functions $p_{\mathsf{k}}[x]$ defined by subdivision converges uniformly to a limit function $p_\infty[x]$, we usually do not have access to the limit

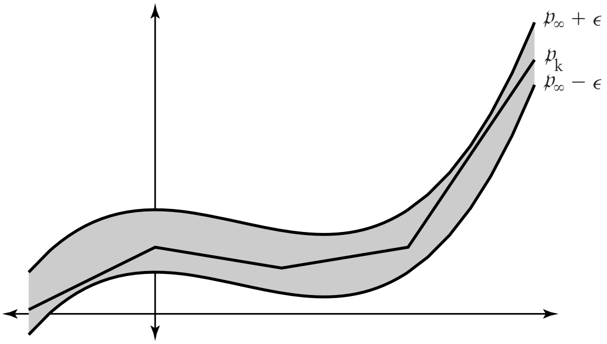


Figure 3.2 Two functions $p_\infty[x] \pm \epsilon$ bounding the function $p_{\mathsf{k}}[x]$ in the definition of uniform convergence.

function $p_\infty[x]$ beforehand. Thus, it is difficult to bound the norm $\|p_\infty[x] - p_k[x]\|$ directly. Instead, we examine the norm of differences between successive approximations $\|p_k[x] - p_{k-1}[x]\|$. The following theorem establishes a useful condition on these differences that is sufficient to guarantee uniform convergence of the sequence of functions $p_k[x]$.

THEOREM 3.1 Given a sequence of functions $p_0[x], p_1[x], p_2[x], \ldots$, if there exist constants $0 < \alpha < 1$ and $\beta > 0$ such that

$$\|p_k[x] - p_{k-1}[x]\| < \beta\alpha^{k-1}$$

for all $k > 0$, the sequence converges uniformly to a limit function $p_\infty[x]$.

Proof To begin, we prove that there exists a pointwise limit $p_\infty[x]$ for the functions $p_k[x]$. For a fixed value of x, we observe that the sum of the absolute value of the difference of successive functions satisfies the inequality

$$\sum_{j=1}^{k} |p_j[x] - p_{j-1}[x]| < \sum_{j=1}^{k} \beta\alpha^{j-1} == \beta\frac{1-\alpha^k}{1-\alpha}.$$

Because $0 < \alpha < 1$, the expression $\beta\frac{1-\alpha^k}{1-\alpha}$ converges to $\frac{\beta}{1-\alpha}$ as $k \to \infty$. Therefore, by the comparison test (Theorem II, page 548 in [151]), the sum $\sum_{j=1}^{\infty} |p_j[x] - p_{j-1}[x]|$ is also convergent. Moreover, because this sum of absolute values is convergent, the sum $\sum_{j=1}^{\infty} (p_j[x] - p_{j-1}[x])$ is also convergent (Theorem IX, page 557 in [151]). Given this convergence, we define the pointwise limit of the subdivision process to be the infinite sum

$$p_\infty[x] == p_0[x] + \sum_{j=1}^{\infty} (p_j[x] - p_{j-1}[x]).$$

To conclude the proof, we prove that the functions $p_k[x]$ uniformly converge to $p_\infty[x]$. To this end, we observe that the difference $p_\infty[x] - p_k[x]$ can be expressed as an infinite sum of the form

$$p_\infty[x] - p_k[x] == \sum_{j=k+1}^{\infty} (p_j[x] - p_{j-1}[x]).$$

Taking the norm of both sides of this expression and applying the triangle inequality yields

$$\|p_\infty[x] - p_k[x]\| \le \sum_{j=k+1}^{\infty} \|p_j[x] - p_{j-1}[x]\|.$$

Now, by hypothesis, the norm $\|p_j[x] - p_{j-1}[x]\|$ is bounded by $\beta\alpha^{k-1}$. Substituting this bound into the infinite sum above yields

$$\|p_\infty[x] - p_k[x]\| \le \beta\frac{\alpha^k}{1-\alpha}.$$

Given an $\epsilon > 0$ from the definition of uniform continuity, we simply choose k sufficiently large such that the right-hand side of this equation is less than ϵ. This condition ensures that the functions $p_k[x]$ converge uniformly to $p_\infty[x]$.

3.1.3 **Uniform Convergence for Continuous Functions**

One advantage of uniform convergence over pointwise convergence is that it guarantees that many important properties of the functions $p_k[x]$ also hold for the limit function $p_\infty[x]$. In the case of subdivision, the most important property is continuity. By construction, the piecewise linear approximations $p_k[x]$ are continuous. Thus, a natural question to ask is whether their limit $p_\infty[x]$ is also continuous. If the convergence is uniform, the answer is yes. The following theorem establishes this fact.

THEOREM 3.2 Let $p_k[x]$ be a sequence of continuous functions. If the sequence $p_k[x]$ converges uniformly to a limit function $p_\infty[x]$, then $p_\infty[x]$ is continuous.

Proof We begin by recalling the definition of continuity for a function. The function $p_\infty[x]$ is *continuous* at $x = x_0$ if given an $\epsilon > 0$ there exists a neighborhood of x_0 such that $|p_\infty[x] - p_\infty[x_0]| < \epsilon$ for all x in this neighborhood. To arrive at this inequality, we first note that due to the uniform convergence of the $p_k[x]$, for any $\epsilon > 0$ there exists an n independent of x such that, for all $k \geq n$, $|p_k[x] - p_\infty[x]| < \frac{\epsilon}{3}$ for all x. At this point, we can expand the expression $p_\infty[x] - p_\infty[x_0]$ in the following manner:

$$\begin{aligned} p_\infty[x] - p_\infty[x_0] &== (p_\infty[x] - p_k[x]) + (p_k[x] - p_k[x_0]) + (p_k[x_0] - p_\infty[x_0]), \\ |p_\infty[x] - p_\infty[x_0]| &\leq |p_\infty[x] - p_k[x]| + |p_k[x] - p_k[x_0]| + |p_k[x_0] - p_\infty[x_0]|. \end{aligned}$$

Combining these two inequalities yields a new inequality of the form

$$|p_\infty[x] - p_\infty[x_0]| < |p_k[x] - p_k[x_0]| + \frac{2\epsilon}{3}.$$

Because $p_k[x]$ is continuous at x_0, there exists a neighborhood of x_0 for which $|p_k[x] - p_k[x_0]| < \frac{\epsilon}{3}$ holds. Thus, $|p_\infty[x] - p_\infty[x_0]| < \epsilon$ for all x in this neighborhood of x_0.

One immediate consequence of this theorem is that anytime we prove that the piecewise linear approximations $p_k[x]$ for a subdivision scheme converge uniformly the resulting limit function $p_\infty[x]$ is automatically continuous.

3.1.4 Uniform Convergence for Smooth Functions

As we saw earlier in this section, one drawback of pointwise convergence is that it does not ensure that the derivative of the functions $p_k[x]$ converge to the derivative of the limit function $p_\infty[x]$. Because the derivatives $p'_k[x]$ of the piecewise linear functions $p_k[x]$ associated with a subdivision scheme are piecewise constant functions, we arrive at another natural question: Do these piecewise constant functions $p'_k[x]$ converge to the derivative of $p_\infty[x]$ (i.e., of the limit of the functions $p_k[x]$)? The answer is yes if the functions $p'_k[x]$ converge uniformly to a continuous function. The following theorem is a slight variant of Theorem V, page 602 in [151].

THEOREM 3.3 Let the functions $p_k[x]$ converge uniformly to $p_\infty[x]$. If the derivative functions $p'_k[x]$ converge uniformly to a function $q[x]$, then $q[x]$ is the derivative of $p_\infty[x]$ with respect to x.

Proof We start by showing that $\int_0^x p'_k[t]\, dt$ is pointwise convergent to $\int_0^x q[t]\, dt$ for all x. Given that the functions $p'_k[x]$ converge uniformly to $q[x]$ as $k \to \infty$, for all $\epsilon > 0$ there exists n such that for all $k \geq n$

$$|p'_k[t] - q[t]| < \epsilon$$

for all t. For a fixed x, integrating the left-hand side of this expression on the range $[0, x]$ yields

$$\left|\int_0^x (p'_k[t] - q[t])\, \mathrm{d}t\right| \leq \int_0^x |p'_k[t] - q[t]|\, \mathrm{d}t \leq \epsilon\, x$$

for all $k \geq n$. Recalling that $p'_k[t]$ is by definition the derivative of $p_k[t]$, the left-hand side of this expression reduces to

$$\left|p_k[x] - p_k[0] - \int_0^x q[t]\, \mathrm{d}t\right| \leq \epsilon\, x.$$

Given a fixed x, this inequality implies that the limit of $p_k[x] - p_k[0]$ as $k \to \infty$ is exactly $\int_0^x q[t]\, \mathrm{d}t$. However, by assumption, $p_k[x] - p_k[0]$ also converges to $p_\infty[x] - p_\infty[0]$. Therefore,

$$p_\infty[x] == \int_0^x q[t]\, \mathrm{d}t + p_\infty[0]$$

for any choice of x. Thus, $q[x]$ is the derivative of $p_\infty[x]$ with respect to x.

3.2 Analysis of Univariate Schemes

The univariate subdivision schemes considered in the previous chapter generated a sequence of vectors p_k via the subdivision relation $\mathrm{p}_k[x] = s[x]\mathrm{p}_{k-1}[x^2]$. Associating each vector p_k with a grid $\frac{1}{2^k}\mathbb{Z}$ yields a sequence of piecewise linear functions $p_k[x]$ whose value at the ith knot, $p_k\left[\frac{i}{2^k}\right]$, is the ith entry of the vector p_k; that is,

$$p\left[\frac{i}{2^k}\right] = \mathrm{p}_k[\![i]\!].$$

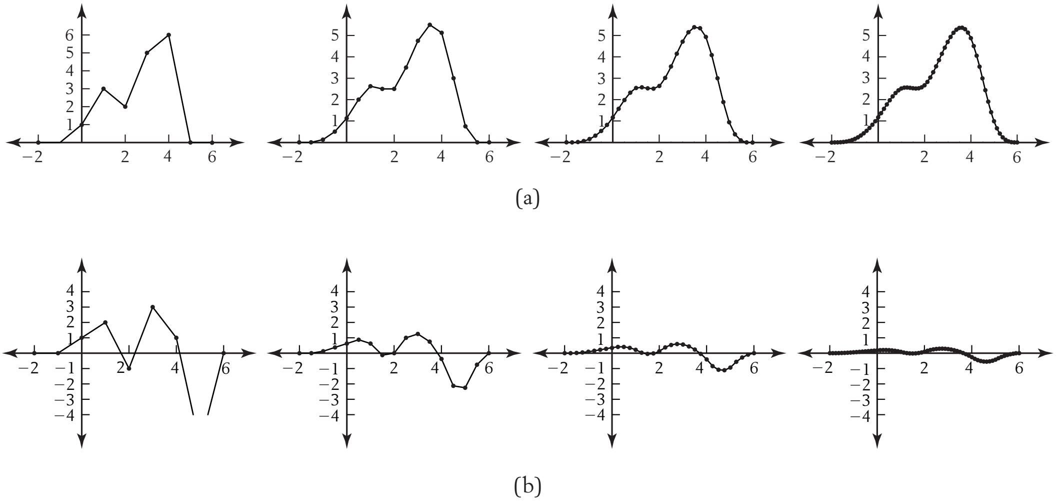
This section addresses two main questions concerning the sequence of functions produced by this subdivision scheme: Does this sequence of functions $p_k[x]$ converge uniformly to a limit function $p_\infty[x]$? If so, is this limit function smooth? The answers to these questions have been addressed in a number of papers, including those by Dahmen, Caveretta, and Micchelli in [17] and Dyn, Gregory, and Levin in [51]. An excellent summary of these results appears in Dyn [49].

3.2.1 A Subdivision Scheme for Differences

To answer the first question, we begin with the following observation: If the maximal difference between consecutive coefficients of p_k goes to zero as $k \to \infty$, the functions $p_k[x]$ converge to a continuous function. If $\mathsf{d}[x]$ is the difference mask $(1 - x)$, the differences between consecutive coefficients of p_k are simply the coefficients of the generating function $\mathsf{d}[x]\mathsf{p}_k[x]$. Intuitively, as the coefficients of $\mathsf{d}[x]\mathsf{p}_k[x]$ converge to zero, any jumps in the associated linear functions $p_k[x]$ become smaller and smaller. Therefore, the functions $p_k[x]$ converge to a continuous function $p[x]$. Figure 3.3 shows plots of these differences for three rounds of cubic B-spline subdivision. Note that the maximum value of these differences is decreasing geometrically.

Our task in this and the next section is to develop a method for explicitly bounding the decay of these differences in terms of the subdivision mask $\mathsf{s}[x]$. At the heart of this method is the observation that if two consecutive approximations $\mathsf{p}_{k-1}[x]$ and $\mathsf{p}_k[x]$ are related by the subdivision mask $\mathsf{s}[x]$ (by $\mathsf{p}_k[x] = \mathsf{s}[x]\mathsf{p}_{k-1}[x^2]$) then the differences at consecutive levels, $\mathsf{d}[x]\mathsf{p}_{k-1}[x]$ and $\mathsf{d}[x]\mathsf{p}_k[x]$, are themselves also related by a subdivision mask $\mathsf{t}[x]$:

$$\mathsf{d}[x]\mathsf{p}_k[x] == \mathsf{t}[x](\mathsf{d}[x^2]\mathsf{p}_{k-1}[x^2]). \tag{3.1}$$



(a)

(b)

Figure 3.3 Three rounds of cubic B-spline subdivision (a) and corresponding differences (b).

Under a relatively mild assumption concerning the structure of $s[x]$, this new subdivision mask $t[x]$ can be constructed directly from $s[x]$. If we assume that the subdivision process defined by $s[x]$ is invariant under affine transformations, the rows of the associated subdivision matrix S must sum to one. For uniform subdivision schemes, the matrix S has two types of rows: those corresponding to the odd-indexed coefficients $s[\![2i+1]\!]$ of $s[x]$ and those corresponding to even-indexed coefficients $s[\![2i]\!]$ of $s[x]$. Requiring the rows of S to sum to one is equivalent to requiring each of these coefficient sequences to sum to one. Using simple linear algebra, this condition can be expressed directly in terms of the mask $s[x]$ as $s[1] == 2$ and $s[-1] == 0$. Under the assumption that $s[-1] == 0$, the following theorem (credited to Dyn, Gregory, and Levin [51]) relates the masks $s[x]$ and $t[x]$.

THEOREM 3.4 Given a subdivision mask $s[x]$ satisfying $s[-1] == 0$, there exists a subdivision mask $t[x]$ relating the differences $d[x]p_{k-1}[x]$ and $d[x]p_k[x]$ of the form

$$t[x] = \frac{s[x]}{1+x}.$$

Proof Recall that the approximations $p_{k-1}[x]$ and $p_k[x]$ are related via $p_k[x] = s[x]p_{k-1}[x^2]$. Multiplying both sides of this expression by the difference mask $d[x]$ yields

$$d[x]p_k[x] == d[x]s[x]p_{k-1}[x^2].$$

Because $s[x]$ has a root at $x == -1$, the polynomial $1+x$ exactly divides $s[x]$. Therefore, there exists a mask $t[x]$ satisfying the relation

$$d[x]s[x] == t[x]d[x^2]. \tag{3.2}$$

Substituting the $t[x]d[x^2]$ for $d[x]s[x]$ in the previous equation yields equation 3.1, the subdivision relation for the differences $d[x]\,p_{k-1}[x]$ and $d[x]\,p_k[x]$.

Note that equation 3.2 is at the heart of this theorem. One way to better understand this equation is to construct the associated matrix version of it. The matrix analog D of the difference mask $d[x] = 1 - x$ is the bi-infinite two-banded matrix with -1 and 1 along the bands. If S and T are the bi-infinite two-slanted subdivision matrices whose $\{i, j\}$-th entries are $s[\![i - 2j]\!]$ and $t[\![i - 2j]\!]$, respectively, the matrix analog of equation 3.2 has the form

$$DS == TD. \tag{3.3}$$

Note that the action of the difference mask $d[x]$ in equation 3.2 is modeled by multiplying the two-slanted matrix S on the left by D, whereas the action of the mask $d[x^2]$ is modeled by multiplying the two-slanted matrix T on the right by D.

For example, if $s[x] = \frac{(1+x)^2}{2x}$, then $s[x]$ is the subdivision mask for linear subdivision. The subdivision mask $t[x]$ for the difference scheme is simply $\frac{(1+x)}{2x}$. In matrix form (showing only a small portion of the bi-infinite matrices), equation 3.3 reduces to

$$\begin{pmatrix}
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdot \\
\cdot & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \cdot \\
\cdot & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & \cdot \\
\cdot & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & \cdot \\
\cdot & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & \cdot \\
\cdot & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & \cdot \\
\cdot & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & \cdot \\
\cdot & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot
\end{pmatrix}
\begin{pmatrix}
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & 1 & 0 & 0 & 0 & 0 & \cdot \\
\cdot & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & \cdot \\
\cdot & 0 & 1 & 0 & 0 & 0 & \cdot \\
\cdot & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & \cdot \\
\cdot & 0 & 0 & 1 & 0 & 0 & \cdot \\
\cdot & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & \cdot \\
\cdot & 0 & 0 & 0 & 1 & 0 & \cdot \\
\cdot & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & \cdot \\
\cdot & 0 & 0 & 0 & 0 & 1 & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot
\end{pmatrix}$$

$$== \begin{pmatrix}
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \frac{1}{2} & 0 & 0 & 0 & \cdot \\
\cdot & \frac{1}{2} & 0 & 0 & 0 & \cdot \\
\cdot & 0 & \frac{1}{2} & 0 & 0 & \cdot \\
\cdot & 0 & \frac{1}{2} & 0 & 0 & \cdot \\
\cdot & 0 & 0 & \frac{1}{2} & 0 & \cdot \\
\cdot & 0 & 0 & \frac{1}{2} & 0 & \cdot \\
\cdot & 0 & 0 & 0 & \frac{1}{2} & \cdot \\
\cdot & 0 & 0 & 0 & \frac{1}{2} & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot
\end{pmatrix}
\begin{pmatrix}
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & -1 & 1 & 0 & 0 & 0 & \cdot \\
\cdot & 0 & -1 & 1 & 0 & 0 & \cdot \\
\cdot & 0 & 0 & -1 & 1 & 0 & \cdot \\
\cdot & 0 & 0 & 0 & -1 & 1 & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot
\end{pmatrix}.$$

3.2.2 A Condition for Uniform Convergence

Given Theorem 3.4, we are now ready to state a condition on the subdivision mask $t[x]$ for the differences that is sufficient to guarantee that the associated piecewise linear functions $p_k[x]$ converge uniformly. This condition involves the size of the various coefficients $t[\![i]\!]$ of this mask. To measure the size of this sequence of coefficients, we first define a vector norm that is analogous to the previous functional norm. The *norm* of a vector p has the form

$$\|p\| = \operatorname*{Max}_i |p[\![i]\!]|.$$

(Again, this norm is the infinity norm in standard numerical analysis terminology.) Given this vector norm, we can define a similar norm for a matrix A of the form

$$\|A\| = \operatorname*{Max}_{p \neq 0} \frac{\|Ap\|}{\|p\|}.$$

Observe that if the matrix A is infinite it is possible that the infinite sum denoted by the product Ap is unbounded. However, as long as the matrix A has rows with only a finite number of non-zero entries, this norm is well defined. One immediate consequence of this definition of $\|A\|$ is the inequality $\|Ap\| \leq \|A\| \|p\|$ for all column vectors p. We leave it as a simple exercise for the reader to show that the norm of a matrix A is the maximum over all rows in A of the sums of the absolute values in that row (i.e., $\|A\| = \operatorname{Max}_i \sum_j |A[\![i, j]\!]|$). Using these norms, the following theorem (again credited to Dyn, Gregory, and Levin [51]) characterizes those subdivision schemes that are uniformly convergent.

THEOREM 3.5 Given an affinely invariant subdivision scheme with associated subdivision matrix S, let the matrix T be the subdivision matrix for the differences (i.e., T satisfies $DS == TD$). If $\|T\| < 1$, the associated functions $p_k[x]$ converge uniformly as $k \to \infty$ for all initial vectors p_0 with bounded norm.

Proof Our approach is to bound the difference between the functions $p_k[x]$ and $p_{k-1}[x]$ and then apply Theorem 3.1. Because $p_{k-1}[x]$ is defined using piecewise linear interpolation, it interpolates the coefficients of $\widetilde{S}p_{k-1}$ plotted on $\frac{1}{2^k}\mathbb{Z}$, where $\widetilde{S}$ is the subdivision matrix for linear subdivision. Therefore,

$$\|p_k[x] - p_{k-1}[x]\| == \|p_k - \widetilde{S}p_{k-1}\|. \tag{3.4}$$

Because $p_k = Sp_{k-1}$, the right-hand side of equation 3.4 can be rewritten as $\|(S - \widetilde{S})p_{k-1}\|$. Recalling that the subdivision mask for linear interpolation has the form $\widetilde{s}[x] = \frac{(1+x)^2}{2x}$, the mask for $S - \widetilde{S}$ has the form $s[x] - \widetilde{s}[x]$. Because both subdivision schemes are affinely invariant, the mask $s[x] - \widetilde{s}[x]$ has roots at $x == \pm 1$. Thus, $s[x] - \widetilde{s}[x]$ is divisible by $d[x^2]$ and can be rewritten in the form $a[x]d[x^2]$. Converting this expression back to matrix form, $S - \widetilde{S}$ can be expressed as the matrix product AD, where D is the difference matrix. Thus, equation 3.4 can be rewritten in the form

$$\|p_k[x] - p_{k-1}[x]\| == \|ADp_{k-1}\| \leq \|A\| * \|Dp_{k-1}\|. \tag{3.5}$$

Now, we observe that Dp_{k-1} can be written as $DS^{k-1}p_0$, where S is the subdivision matrix. This expression can be reduced to $T^{k-1}Dp_0$ by applying the substitution $DS == TD$ exactly $k - 1$ times. Substituting this expression into equation 3.5 yields

$$\|Dp_{k-1}\| == \|T^{k-1}Dp_0\| \leq \|T\| * \|T^{k-2}Dp_0\| \leq \ldots \leq \|T\|^{k-1} * \|Dp_0\|. \tag{3.6}$$

Finally, setting $\beta == \|A\| * \|Dp_0\|$ and $\alpha == \|T\|$ and applying Theorem 3.1 completes the proof.

Theorem 3.5 provides us with a constructive test for whether the subdivision scheme associated with the subdivision mask $s[x]$ is uniformly convergent: Given $s[x]$, we first compute $t[x] = \frac{s[x]}{1+x}$ and then compute $\|T\|$. Because T is a matrix whose columns are two-shifts of the sequence t, this norm is simply $\text{Max}[\sum_i |t[\![2i]\!]|, \sum_i |t[\![2i+1]\!]|]$. If this norm is less than one, the scheme defined by $s[x]$ is uniformly convergent.

This test, based on Theorem 3.5, is not conclusive in determining whether the difference scheme associated with T converges to zero. It is possible to construct examples of subdivision matrices T such that $\|T\| > 1$ but for which $\lim_{n\to\infty}\|T^n\| \to 0$. Luckily, this matrix analysis gives direct insight into a more exact condition for uniform convergence. Dyn et al. [51] observe that a necessary and sufficient condition for uniform convergence is the existence of $n > 0$ such that $\|T^n\| < 1$. If such an n fails to exist, the differences do not converge. If such an n exists, the differences converge to zero and the subdivision scheme is convergent. The matrix norm $\|T^n\|$

can be expressed in terms of the initial subdivision mask $\mathsf{t}[\mathsf{x}]$ as follows (♚):

- Subdivide n times using the mask $\mathsf{t}[\mathsf{x}]$ to form the generating function $\widetilde{\mathsf{t}}[\mathsf{x}] = \prod_{\mathsf{i}=0}^{\mathsf{n}-1} \mathsf{t}[\mathsf{x}^{2^{\mathsf{i}}}]$. The 2^{n} shifts of the coefficients $\widetilde{\mathsf{t}}$ of this generating function form the columns of the matrix T^{n}.
- Partition $\widetilde{\mathsf{t}}$ into 2^{n} subsequences corresponding to distinct rows of T^{n} and compute $\mathsf{max}_{\mathsf{i}} \sum_{\mathsf{j}} |\widetilde{\mathsf{t}}[\![2^{\mathsf{n}}\mathsf{j} + \mathsf{i}]\!]|$ for $\mathsf{i} == 0 \ldots 2^{\mathsf{n}} - 1$.

Determining whether there exists an n such that the norm $\|\mathsf{T}^{\mathsf{n}}\|$ is less than one is a difficult problem, for which the authors are aware of no easy solution. In practice, we simply suggest computing $\|\mathsf{T}^{\mathsf{n}}\|$ for a small number of values of n.

3.2.3 A Subdivision Scheme for Divided Differences

The previous section derived a sufficient condition on the subdivision mask $\mathsf{t}[\mathsf{x}] = \frac{\mathsf{s}[\mathsf{x}]}{1+\mathsf{x}}$ for the corresponding functions $p_{\mathsf{k}}[x]$ to converge uniformly. This subdivision mask $\mathsf{t}[\mathsf{x}]$ provides a scheme for computing differences $\mathsf{d}[\mathsf{x}]\mathsf{p}_{\mathsf{k}}[\mathsf{x}]$ on increasingly fine grids. This section modifies this scheme slightly to yield a subdivision mask for the divided differences of the original scheme, providing a tighter grip on the derivative of the limit curve. To derive this new scheme, we recall that the first derivative $p'[x]$ of a smooth function $p[x]$ is the limit

$$p'[x] = \lim_{t \to 0} \frac{p[x] - p[x-t]}{t}.$$

By the definition of the derivative, any possible sequence of values for t is guaranteed to converge to the respective derivative of the smooth function $p[x]$, as long as we can guarantee that $t \to 0$. Consequently, we can pick a sequence of values for t that fits our discretization on the grid $\frac{1}{2^{\mathsf{k}}}\mathbb{Z}$ exceptionally well: substituting $t = \frac{1}{2^{\mathsf{k}}}$ leads to

$$p'[x] = \lim_{\mathsf{k} \to \infty} 2^{\mathsf{k}} \left(p[x] - p\left[x - \frac{1}{2^{\mathsf{k}}}\right] \right). \tag{3.7}$$

If we multiply the difference operator $\mathsf{d}[\mathsf{x}]$ by the constant 2^{k}, the resulting difference mask $\mathsf{d}_{\mathsf{k}}[\mathsf{x}] = 2^{\mathsf{k}}\mathsf{d}[\mathsf{x}] == 2^{\mathsf{k}}(1 - \mathsf{x})$ computes the first divided differences on the grid $\frac{1}{2^{\mathsf{k}}}\mathbb{Z}$. Higher-order divided differences can be computed using powers of the first difference mask $\mathsf{d}_{\mathsf{k}}[\mathsf{x}]$.

Observe that these difference masks $\mathsf{d}_{\mathsf{k}}[\mathsf{x}]^{\mathsf{m}+1}$ have the property that they annihilate samples of polynomial functions of degree m taken on the grid $\frac{1}{2^{\mathsf{k}}}\mathbb{Z}$. For example, the second difference mask $4^k(1-x)^2$ annihilates generating functions whose coefficients are samples of linear functions. As the next theorem shows, these difference masks play a key role in determining the behavior of a subdivision scheme on polynomial functions.

THEOREM 3.6 Consider a subdivision scheme with mask $s[x]$, for which integer translates of its scaling function are linearly independent and capable of reproducing all polynomials of degree m. A limit function $p_\infty[x]$ produced by this scheme is a polynomial function of degree m if and only if its associated generating functions $p_k[x]$ satisfy the difference relation

$$(1-x)^{m+1}\, p_k[x] == 0. \tag{3.8}$$

Proof Consider a generating function $p_k[x]$ whose limit under subdivision by the mask $s[x]$ is the function $p_\infty[x]$. Due to the linearity of subdivision, multiplying $p_k[x]$ by the difference mask $(1-x)^{m+1}$ corresponds to taking an $(m+1)$st difference of translates of $p_\infty[x]$ on the grid $\frac{1}{2^k}\mathbb{Z}$. If $p_\infty[x]$ is a polynomial of degree m, this difference is identically zero. Because the integer translates of the scaling function for this scheme are linearly independent, the generating function $(1-x)^{m+1}\, p_k[x]$ must also be identically zero.

Because the coefficients of the generating function $p_k[x]$ satisfying equation 3.8 are samples of polynomial functions of degree m, the space of solutions $p_k[x]$ to equation 3.8 has dimension $m+1$, the same dimension as the space of polynomials of degree m. Because the integer translates of the scaling function for the scheme are linearly independent, any limit function $p_\infty[x]$ corresponding to a generating function $p_k[x]$ satisfying equation 3.8 must be a polynomial of degree m; otherwise, there would exist a polynomial limit function whose initial coefficients do not satisfy equation 3.8.

Our next goal is to construct the subdivision mask $t[x]$ that relates the first divided differences $d_{k-1}[x]\,p_{k-1}[x]$ at level $k-1$ to the first divided differences $d_k[x]\,p_k[x]$ at level k. Making a slight modification to equation 3.2 yields a new relation between $s[x]$ and $t[x]$ of the form

$$d_k[x]\,s[x] == t[x]\,d_{k-1}[x^2]. \tag{3.9}$$

Note that the various powers of 2 attached to $\mathrm{d}_k[\mathrm{x}]$ and $\mathrm{d}_{k-1}[\mathrm{x}^2]$ cancel, leaving a single factor of 2 on the left-hand side. Solving for $\mathrm{t}[\mathrm{x}]$ yields a subdivision mask for the first divided difference of the form $\frac{2\mathrm{s}[\mathrm{x}]}{1+\mathrm{x}}$.

Given this subdivision scheme for the first divided difference, we can now test whether the limit function $p_\infty[x]$ associated with the original subdivision scheme is smooth (i.e., $p_\infty[x] \in \mathcal{C}^1$). The coefficients of $\mathrm{d}_k[\mathrm{x}]\mathrm{p}_k[\mathrm{x}]$ correspond to the first derivatives of the piecewise linear function $p_k[x]$, that is, the piecewise constant function $p'_k[x]$. Now, by Theorem 3.3, if these functions $p'_k[x]$ uniformly converge to a continuous limit $q[x]$, then $q[x]$ is the derivative of $p_\infty[x]$. Given these observations, the test for smoothness of the original scheme is as follows: Construct the mask $\frac{2\mathrm{s}[\mathrm{x}]}{1+\mathrm{x}}$ for the divided difference scheme, and then apply the convergence test of the previous section to this mask. If differences of this divided difference scheme converge uniformly to zero, the first divided difference scheme converges to continuous limit functions, whereas the original scheme converges to smooth limit functions.

An iterated version of this test can be used to determine whether a subdivision scheme defines limit functions $p_\infty[x]$ that have m continuous derivatives (i.e., $p_\infty[x] \in \mathcal{C}^{\mathrm{m}}$). Specifically, the test must determine whether the difference of the mth divided difference of $p_k[x]$ is converging to zero. Given a subdivision mask $\mathrm{s}[\mathrm{x}]$, the subdivision mask for this difference scheme has the form $\mathrm{t}[\mathrm{x}] = \frac{2^{\mathrm{m}}\mathrm{s}[\mathrm{x}]}{(1+\mathrm{x})^{\mathrm{m}+1}}$. If $\mathrm{s}[\mathrm{x}]$ is divisible by $(1+\mathrm{x})^{\mathrm{m}+1}$, the original scheme with mask $\mathrm{s}[\mathrm{x}]$ produces $\mathcal{C}^{\mathrm{m}}$ functions if and only if there exists $\mathrm{n} > 0$ such that $\|\mathrm{T}^{\mathrm{n}}\| < 1$.

Of course, to apply this test the mask $\mathrm{s}[\mathrm{x}]$ must be divisible by $(1+\mathrm{x})^{\mathrm{m}+1}$. This condition is equivalent to requiring that the mask $\mathrm{s}[\mathrm{x}]$ has a *zero of order* $\mathrm{m}+1$ at $\mathrm{x} == -1$; that is, $\mathrm{s}^{(\mathrm{i})}[-1] == 0$ for $\mathrm{i} == 0 \ldots \mathrm{m}$. As the next theorem shows, both of these conditions have a nice interpretation in terms of the space of functions generated by the subdivision scheme.

THEOREM 3.7 If $\mathrm{s}[\mathrm{x}]$ defines a subdivision scheme for which the integer translates of its scaling function are linearly independent and capable of reproducing all polynomials up to degree m, the mask $\mathrm{s}[\mathrm{x}]$ satisfies $\mathrm{s}^{(\mathrm{i})}[-1] == 0$ for $\mathrm{i} == 0 \ldots \mathrm{m}$.

Proof Consider the function $p[x] = (1+2x)^{\mathrm{m}}$. Because this function is a global polynomial of degree m, it can be represented as a uniform B-spline of order $\mathrm{m}+1$ on the grids $\mathbb{Z}$ and $\frac{1}{2}\mathbb{Z}$, respectively. If p_0 and p_1 are coefficient vectors for $p[x]$ on these grids, their generating functions $p_0[x]$ and $p_1[x]$ are related by

$$p_1[x] == \frac{1}{2^m}(1+x)^{m+1}\,p_0[x^2].$$

Note that $p_0[x]$ satisfies the difference relation $(1-x)^{m+1}p_0[x] == 0$ because its limit function is the polynomial $p[x]$. If we replace x by $-x$ in this equation and multiply both sides of the equation by the subdivision mask $s[x]$, the resulting equation has the form

$$s[x]\,p_1[-x] == \frac{1}{2^m}(1-x)^{m+1}s[x]\,p_0[x^2]. \tag{3.10}$$

Because $(1-x)^{m+1}p_0[x] == 0$, the limit of the subdivision process defined by the mask $s[x]$ is also a polynomial of degree m, due to Theorem 3.6. Therefore, the order $(m+1)$st difference of $s[x]p_0[x^2]$ must be zero, and the right-hand side of equation 3.10 is identically zero (i.e., $s[x]\,p_1[-x] == 0$).

Now, our care in choosing the function $p[x]$ pays off. Using blossoming, we can show that the coefficients of the vector p_1 have the form $p_1[\![i]\!] = \prod_{j=i}^{i+m-1} j$. Thus, the constant coefficient of $s[x]\,p_1[-x]$ has the form

$$\sum_{i=-\infty}^{\infty}(-1)^i s[\![i]\!]\,p_1[\![-i]\!] == \sum_{i=-\infty}^{\infty}\left((-1)^i s[\![i]\!]\prod_{j=-i}^{-i+m-1} j\right).$$

Finally, we observe that the right-hand side of this expression can be reduced to $s^{(m)}[-1]$ by distributing a factor $(-1)^m$ inside the product and reversing the order of multiplication of the product. Because the constant coefficient of $s[x]\,p_1[x^2]$ is zero, $s^{(m)}[-1]$ is also zero.

This theorem guarantees that if a scheme reproduces all polynomials of up to degree m then its mask $s[x]$ is divisible by $(1+x)^{m+1}$ and thus there exists a subdivision mask $t[x]$ for the difference of the mth divided differences. Assuming polynomial reproduction of up to degree m is not a particularly restrictive condition because it is known that under fairly general conditions this condition is necessary for schemes that produce C^m limit functions (see section 8.2.4 or [123]).

3.2.4 **Example: The Four-point Scheme**

The previous section described a simple method of determining whether a subdivision scheme produces limit functions that are smooth. In this section, we use this method to analyze the smoothness of a simple uniform scheme. A univariate subdivision scheme is *interpolatory* if its mask $s[x]$ has the property that $s[\![0]\!] == 1$ and $s[\![2i]\!] == 0$ for all $i \neq 0$. (The odd entries of the mask are unconstrained.) The effect of this condition on the corresponding subdivision matrix S is to force the even-indexed rows of S to be simply unit vectors, that is, vectors of the form $\{\ldots, 0, 0, 1, 0, 0, \ldots\}$. Geometrically, interpolatory subdivision schemes have the property that control points in p_{k-1} are left unperturbed during subdivision, with only new points being inserted into p_k.

Figure 3.4 illustrates an example of such an interpolatory scheme, the *four-point scheme* of Deslauries and Dubic [43, 50]. The subdivision mask $s[x]$ for this scheme has the form

$$s[x] = -\frac{1}{16}x^{-3} + \frac{9}{16}x^{-1} + 1 + \frac{9}{16}x - \frac{1}{16}x^{3}.$$

The corresponding subdivision matrix S has the form

$$S = \begin{pmatrix} . & . & . & . & . & . & . \\ . & 1 & 0 & 0 & 0 & 0 & . \\ . & \frac{9}{16} & \frac{9}{16} & -\frac{1}{16} & 0 & 0 & . \\ . & 0 & 1 & 0 & 0 & 0 & . \\ . & -\frac{1}{16} & \frac{9}{16} & \frac{9}{16} & -\frac{1}{16} & 0 & . \\ . & 0 & 0 & 1 & 0 & 0 & . \\ . & 0 & -\frac{1}{16} & \frac{9}{16} & \frac{9}{16} & -\frac{1}{16} & . \\ . & 0 & 0 & 0 & 1 & 0 & . \\ . & 0 & 0 & -\frac{1}{16} & \frac{9}{16} & \frac{9}{16} & . \\ . & . & . & . & . & . & . \end{pmatrix}.$$

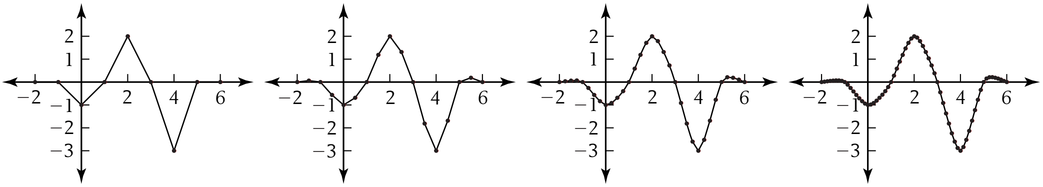


Figure 3.4 Three rounds of subdivision using the four-point scheme.

This particular subdivision mask is the interpolatory mask of minimal support, with the property that the associated subdivision scheme reproduces cubic functions. Specifically, if the initial coefficients p_0 are samples of a cubic function $p[x]$ (i.e., $p_0\llbracket i\rrbracket = p[i]$), the associated limit function is exactly $p[x]$. For example, if the initial coefficients p_0 satisfy $p_0\llbracket i\rrbracket = i^3$, applying the row mask $(-\frac{1}{16}, \frac{9}{16}, \frac{9}{16}, -\frac{1}{16})$ to four consecutive coefficients $(i^3, (i+1)^3, (i+2)^3, (i+3)^3)$ yields a new coefficient centered at $x = i + \frac{3}{2}$ of the form $(i + \frac{3}{2})^3$. Subdividing a uniformly sampled cubic function with this mask yields a denser sampling of the same cubic function.

This method of choosing subdivision masks based on reproducing polynomial functions represents a distinct approach from the integral method of the previous chapter. However, interpolatory schemes that approximate solutions to variational problems can be generated using the linear programming techniques described in Chapter 5.

For the remainder of this section, we analyze the smoothness of the limit functions produced by this four-point scheme. We begin by computing the subdivision mask $t_1[x] = \frac{s[x]}{1+x}$ for the associated difference scheme. This mask has the form

$$t_1[x] = -\frac{1}{16}x^{-3} + \frac{1}{16}x^{-2} + \frac{1}{2}x^{-1} + \frac{1}{2} + \frac{1}{16}x - \frac{1}{16}x^2.$$

If T_1 is the subdivision matrix associated with this mask t_1, then $\|T_1\| = \frac{5}{8}$. Therefore, via Theorem 3.5, this difference scheme converges to zero, and thus the four-point scheme produces continuous limit functions. Figure 3.5 depicts the application of the subdivision scheme for the divided differences $2t_1[x]$ to the differences of the coefficients of the previous example.

To determine whether the four-point scheme produces smooth limit functions, we next compute the difference scheme $t_2[x] = \frac{2t_1[x]}{1+x}$ for the divided difference scheme $2t_1[x]$. The mask $t_2[x]$ has the form

$$t_2[x] = -\frac{1}{8}x^{-3} + \frac{1}{4}x^{-2} + \frac{3}{4}x^{-1} + \frac{1}{4} - \frac{1}{8}x.$$

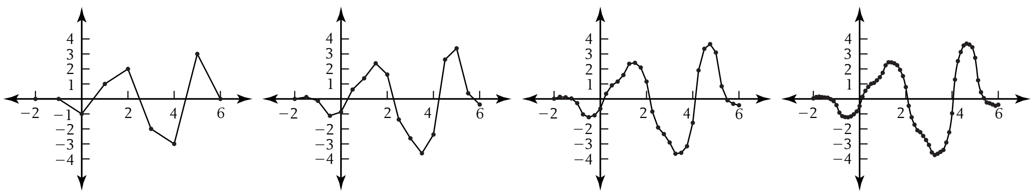


Figure 3.5 Three rounds of subdivision for the first divided differences from Figure 3.4.

At this point, we observe that $\|T_2\| = 1$. Thus, Theorem 3.5 is not sufficient to conclude that the scheme associated with $t_2[x]$ converges to zero. However, because $\|(T_2)^2\| = \frac{3}{4}$, the difference scheme $t_2[x]$ converges to zero. Therefore, the divided difference scheme $2t_1[x]$ converges to continuous functions, and thus the four-point scheme converges to smooth functions.

To conclude, we can compute the difference scheme $t_3[x] = \frac{2t_2[x]}{1+x}$ associated with the second divided difference scheme $2t_2[x]$. The mask $t_3[x]$ has the form

$$t_3[x] = -\frac{1}{4}x^{-3} + \frac{3}{4}x^{-2} + \frac{3}{4}x^{-1} - \frac{1}{4}.$$

Unfortunately, $\|(T_3)^n\|$ is one for all $n > 0$. Therefore, the four-point scheme does not produce $\mathcal{C}^2$ limit functions.

3.3 Analysis of Bivariate Schemes

We conclude this chapter by developing tools for analyzing the convergence and smoothness of uniform, bivariate subdivision schemes. As in the univariate case, the key is to develop a subdivision scheme for various differences associated with the original scheme. However, in the bivariate case, the number of differences that need to be considered is larger than in the univariate case. As a result, the subdivision schemes for these differences, instead of being scalar subdivision schemes, are subdivision schemes involving matrices of generating functions. This technique involving matrices of generating functions (pioneered by Dyn, Hed, and Levin [52]) generalizes to higher dimensions without difficulty. Again, Dyn [49] provides an excellent overview of this technique.

Given an initial vector of coefficients p_0 whose entries are indexed by points on the integer grid $\mathbb{Z}^2$, bivariate subdivision schemes generate a sequence of vectors p_k via the subdivision relation $p_k[x, y] = s[x, y]\,p_{k-1}[x^2, y^2]$. As in the univariate case, our approach to analyzing the behavior of these schemes is to associate a piecewise bilinear function $p_k[x, y]$ with the entries of the vector p_k plotted on the grid $\frac{1}{2^k}\mathbb{Z}^2$. In particular, the value of this function $p_k[x, y]$ at the grid point $\{\frac{i}{2^k}, \frac{j}{2^k}\}$ is simply the ijth coefficient of the vector p_k; that is,

$$p_k\left[\frac{i}{2^k}, \frac{j}{2^k}\right] = p_k[\![i, j]\!].$$

Given this sequence of functions $p_k[x, y]$, this section attempts to answer the same two questions posed in the univariate case: Does this sequence of functions $p_k[x, y]$

converge to a limit? If so, is the limit function smooth? The rest of this section develops the necessary tools to answer these questions.

3.3.1 A Subdivision Scheme for Differences

As in the univariate case, the key observation is to consider the behavior of various differences associated with the scheme. If the differences of $p_k[x, y]$ in both the x and y directions simultaneously converge to zero, the functions $p_k[x, y]$ converge to a continuous function. To measure both of these differences at once, we consider a column vector of differences of the form $\begin{pmatrix} d[x] \\ d[y] \end{pmatrix} p_k[x, y]$. As we shall see, the decay of these differences is governed by a theorem similar to that of Theorem 3.5.

Following the same strategy as in the univariate case, our next step is to derive a subdivision scheme that relates these differences at consecutive levels. To derive the subdivision mask for this difference scheme in the univariate case, we assumed that the subdivision mask s[x] associated with the subdivision scheme was affinely invariant. In the bivariate case, we make a similar assumption. For a bivariate subdivision mask s[x, y] to be affinely invariant, the rows of its associated subdivision matrix S must sum to one. Following a line of reasoning similar to that of the univariate case, this condition can be expressed directly in terms of the subdivision mask s[x, y] as

$$\begin{aligned} s[1,1] &= 4, \\ s[-1,1] &= 0, \\ s[1,-1] &= 0, \\ s[-1,-1] &= 0. \end{aligned} \tag{3.11}$$

However, at this point, a fundamental distinction between the univariate and the bivariate case arises. In the univariate case, the subdivision scheme for the differences had the form $t[x] = \frac{s[x]}{1+x}$, where s[x] was the subdivision mask for the original scheme. In the bivariate case, there are two distinct differences d[x] and d[y] to consider. Attempting to derive two independent subdivision schemes, one for d[x] of the form $\frac{s[x,y]}{1+x}$ and one for d[y] of the form $\frac{s[x,y]}{1+y}$, is doomed to failure. The crucial difference is that in the univariate case affine invariance of the univariate mask s[x] guaranteed that $(1 + x)$ divided s[x]. However, in the bivariate case, affine invariance is not sufficient to ensure that both $(1 + x)$ and $(1 + y)$ divide s[x, y]. Luckily, if both differences are considered simultaneously, there exists a matrix of generating functions that relates the differences at level $k - 1$, $\begin{pmatrix} d[x] \\ d[y] \end{pmatrix} p_{k-1}[x, y]$, to the differences at level k, $\binom{d[x]}{d[y]} p_k[x, y]$. This matrix of generating functions satisfies the following theorem, credited to Dyn et al. [49, 52].

THEOREM 3.8 Given a subdivision mask $s[x, y]$ that is affinely invariant (i.e., satisfies equation 3.11), there exists a matrix of generating functions $t_{ij}[x, y]$ satisfying

$$\begin{pmatrix} d[x] \\ d[y] \end{pmatrix} s[x, y] == \begin{pmatrix} t_{00}[x, y] & t_{01}[x, y] \\ t_{10}[x, y] & t_{11}[x, y] \end{pmatrix} \begin{pmatrix} d[x^2] \\ d[y^2] \end{pmatrix}. \tag{3.12}$$

Proof Because $s[x, y]$ satisfies equation 3.11, the polynomial $d[x]s[x, y]$ vanishes at the intersection of the curves $d[x^2] == 0$ and $d[y^2] == 0$, that is, the four points $\{x, y\} == \{\pm 1, \pm 1\}$. Because these two curves intersect transversally (i.e., are not tangent) at these four points, the polynomial $d[x]s[x, y]$ can be written as a polynomial combination of $d[x^2]$ and $d[y^2]$. (See section 6.1 of Kunz [90] or lemma 3.1 in Warren [156] for a proof.) The masks $t_{00}[x, y]$ and $t_{01}[x, y]$ are the coefficients used in this combination. A similar argument for $d[y]s[x, y]$ yields the masks $t_{10}[x, y]$ and $t_{11}[x, y]$.

For example, consider the subdivision mask for the four-direction quadratic box splines: $s[x, y] = \frac{1}{4}(1 + x)(1 + y)(1 + xy)(x + y)$. The matrix of subdivision masks $(t_{ij}[x, y])$ for the differences $\binom{d[x]}{d[y]} s[x, y]$ has the form

$$\frac{1}{4}\begin{pmatrix} (1 + y)(x + y)(1 + xy) & 0 \\ 0 & (1 + x)(x + y)(1 + xy) \end{pmatrix}.$$

Unfortunately, these matrix masks $t_{ij}[x, y]$ are not unique. For example, adding the term $d[y^2]$ to $t_{00}[x, y]$ can be canceled by subtracting the term $d[x^2]$ from $t_{01}[x, y]$. However, as we shall shortly see, there is a good method for choosing specific generating functions $t_{ij}[x, y]$ based on minimizing the norm for this matrix.

3.3.2 A Condition for Uniform Convergence

In the univariate case, Theorem 3.5 states a sufficient condition on a subdivision mask $s[x]$ for its associated subdivision scheme to be convergent. The key to the proof of the theorem was converting equation 3.2 to the equivalent matrix (non-generating function) form $DS = TD$. Convergence of the scheme associated with S could then be characterized in terms of the norm $\|\mathsf{T}\|$. In the bivariate case, a similar conversion for equation 3.12 into block matrix form is useful. In particular, the matrix of masks $(\mathsf{t}_{ij}[x, y])$ has a block matrix analog of the form

$$\mathsf{T} == \begin{pmatrix} \mathsf{T}_{00} & \mathsf{T}_{01} \\ \mathsf{T}_{10} & \mathsf{T}_{11} \end{pmatrix}, \tag{3.13}$$

where T_{ij} is the matrix version of the subdivision mask $\mathsf{t}_{ij}[x, y]$. Note that this block matrix T formed by the T_{ij} has eight types of rows: four rows corresponding to the first row $(\mathsf{T}_{00}\ \ \mathsf{T}_{01})$ and four rows corresponding to the second row $(\mathsf{T}_{10}\ \ \mathsf{T}_{11})$. The norm of the matrix T is the maximum, over each of these rows, of the sum of the absolute values of the entries in the row. If the t_{ij} are the coefficient vectors for the generating function $\mathsf{t}_{ij}[x, y]$, the norm of the block matrix T can be expressed in terms of the t_{ij} as

$$\|\mathsf{T}\| = \underset{i}{\mathrm{Max}} \Bigg[\sum_{j,k,l} |\mathsf{t}_{ij}[\![2k, 2l]\!]|, \sum_{j,k,l} |\mathsf{t}_{ij}[\![2k+1, 2l]\!]|, \sum_{j,k,l} |\mathsf{t}_{ij}[\![2k, 2l+1]\!]|, \sum_{j,k,l} |\mathsf{t}_{ij}[\![2k+1, 2l+1]\!]| \Bigg]. \tag{3.14}$$

Note that this equation allows us to compute the norm of T without explicitly constructing T. (In fact, T simply serves as a notational convenience for manipulating the norms of various quantities associated with the subdivision scheme.) Convergence of the subdivision scheme associated with $\mathsf{s}[x, y]$ can be characterized in terms of the norm $\|\mathsf{T}\|$.

THEOREM 3.9 Let $\mathsf{s}[x, y]$ be the subdivision mask for an affinely invariant subdivision scheme. If the block matrix subdivision mask T for the difference scheme (given in equation 3.13) has norm $\|\mathsf{T}\| < 1$, the associated functions $p_k[x, y]$ converge uniformly as $k \to \infty$ for all initial vectors p_0 with bounded norm.

Proof The proof has essentially the same structure as the proof of Theorem 3.5. Again, our goal is to show that the difference $\|p_k[x, y] - p_{k-1}[x, y]\|$ decays geometrically as $k \to \infty$ and to apply Theorem 3.1. If $\widetilde{\mathsf{S}}$ is the subdivision matrix for bilinear subdivision, then

$$\|p_k[x, y] - p_{k-1}[x, y]\| == \|\mathsf{p}_k - \widetilde{\mathsf{S}}\mathsf{p}_{k-1}\| == \|(\mathsf{S} - \widetilde{\mathsf{S}})\mathsf{p}_{k-1}\|.$$

If the mask for bilinear subdivision is $\tilde{s}[x, y]$, the mask for the expression $(S - \tilde{S})$ has the form $s[x, y] - \tilde{s}[x, y]$. Because this mask vanishes at $\{d[x^2] == 0, d[y^2] == 0\}$, $s[x, y] - \tilde{s}[x, y]$ can be written as a polynomial combination of $d[x^2]$ and $d[y^2]$ (as argued in the proof of Theorem 3.8):

$$\begin{aligned} s[x, y] - \tilde{s}[x, y] &== a_0[x, y]\, d[x^2] + a_1[x, y]\, d[y^2] \\ &== \begin{pmatrix} a_0[x, y] & a_1[x, y] \end{pmatrix} \begin{pmatrix} d[x^2] \\ d[y^2] \end{pmatrix}. \end{aligned}$$

After converting this expression back into matrix form, the proof concludes in complete analogy with the univariate case via equation 3.6.

Due to this theorem, if $\|T\| < 1$, the bivariate subdivision mask $s[x, y]$ produces continuous limit functions. This observation leads us to our method for uniquely determining the $t_{ij}[x, y]$. Given $s[x, y]$, we choose the generating function $t_{ij}[x, y]$ so as to satisfy Theorem 3.8 while minimizing $\|T\|$. This minimization can be done using linear programming in a manner similar to that of section 5.2.3. As in the univariate case, having $\|T\| < 1$ is not necessary for a bivariate scheme to converge. In general, a necessary and sufficient condition for a bivariate scheme to converge is the existence of an $n > 0$ such that $\|T^n\| < 1$. (See Dyn [49] for more details.)

3.3.3 Convergence to a Smooth Function

In the univariate case, testing whether a limit function is $\mathcal{C}^m$ continuous involves taking $m + 1$ successive differences of the coefficient vector p_k. In terms of generating functions, this test for $\mathcal{C}^m$ continuity involves determining whether the coefficients of $d[x]\, d_k[x]^m p_k[x]$ converge to zero as $k \to \infty$ for any initial choice of p_0. Given a subdivision mask $s[x]$ for the original scheme, we constructed a new subdivision mask of the form $t[x] = \frac{2^m s[x]}{(1+x)^{m+1}}$ for these order $m + 1$ differences. The original scheme produces $\mathcal{C}^m$ functions if there exists an $n > 0$ such that $\|T^n\| < 1$.

In the bivariate case, deriving a test on the subdivision mask $s[x, y]$ that is sufficient to determine whether the subdivision scheme converges to $\mathcal{C}^m$ limit functions is much more complicated due to the fact that we must consider the differences $d[x]$ and $d[y]$ of $m + 1$ distinct divided differences of order m:

$$\{d_k[x]^m, d_k[x]^{m-1} d_k[y], \ldots, d_k[y]^m\}.$$

Due to the growth in the number of differences as m increases, attempting to iterate Theorem 3.8 is very awkward. Instead, we propose an alternate approach based on computing an $(\mathsf{m}+2) \times (\mathsf{m}+2)$ matrix of generating functions $(\mathsf{t}_{ij}[\mathsf{x},\mathsf{y}])$ that directly relates the various differences of order $\mathsf{m}+1$ at successive levels. As in the univariate case, the existence of this matrix depends on whether the scheme is capable of reproducing polynomials of degree m.

Via Theorem 3.7, if a univariate scheme reproduces polynomials up to degree m, its mask $\mathsf{s}[\mathsf{x}]$ has a zero of order $\mathsf{m}+1$ at $\mathsf{x} == -1$ and consequently the subdivision mask $\mathsf{t}[\mathsf{x}]$ for the $(\mathsf{m}+1)$st difference scheme is guaranteed to exist. In the bivariate case, a similar result holds. If integer translates of the scaling function for the scheme are linearly independent and reproduce all polynomials of up to degree m, its mask $\mathsf{s}[\mathsf{x},\mathsf{y}]$ has a *zero of order* $\mathsf{m}+1$ at the points $\{\{-1,1\},\{1,-1\},\{-1,-1\}\}$; that is,

$$\begin{aligned} s^{(i,j)}[-1,1] &== 0,\\ s^{(i,j)}[1,-1] &== 0,\\ s^{(i,j)}[-1,-1] &== 0 \end{aligned} \tag{3.15}$$

for all $\mathsf{i},\mathsf{j} \geq 0$ and $\mathsf{i}+\mathsf{j} \leq \mathsf{m}$ (♚). (The proof of this fact is nearly identical to that of Theorem 3.7.)

As the next theorem shows, any subdivision scheme whose mask $\mathsf{s}[\mathsf{x},\mathsf{y}]$ satisfies equation 3.15 has an associated subdivision scheme for the order $(\mathsf{m}+1)$st differences whose matrix mask $(\mathsf{t}_{ij}[\mathsf{x},\mathsf{y}])$ satisfies the following relation.

THEOREM 3.10 If the subdivision mask $\mathsf{s}[\mathsf{x},\mathsf{y}]$ satisfies equation 3.15, there exists a matrix of generating functions $\mathsf{t}_{ij}[\mathsf{x},\mathsf{y}]$ satisfying

$$2^m \begin{pmatrix} d[x]^{m+1} \\ d[x]^m d[y] \\ . \\ d[x]\, d[y]^m \\ d[y]^{m+1} \end{pmatrix} s[x,y] ==$$

$$\begin{pmatrix} t_{00}[x,y] & t_{01}[x,y] & . & t_{0m}[x,y] & t_{0(m+1)}[x,y] \\ t_{10}[x,y] & t_{11}[x,y] & . & t_{1m}[x,y] & t_{1(m+1)}[x,y] \\ . & . & . & . & . \\ t_{m0}[x,y] & t_{m1}[x,y] & . & t_{mm}[x,y] & t_{m(m+1)}[x,y] \\ t_{(m+1)0}[x,y] & t_{(m+1)1}[x,y] & . & t_{(m+1)m}[x,y] & t_{(m+1)(m+1)}[x,y] \end{pmatrix} \begin{pmatrix} d[x^2]^{m+1} \\ d[x^2]^m d[y^2] \\ . \\ d[x^2]\, d[y^2]^m \\ d[y^2]^{m+1} \end{pmatrix}.$$

Proof The polynomial span of $d[x^2]^{m+1}$, $d[x^2]^m d[y^2]$, …, $d[y^2]^{m+1}$ is the space of all polynomials with order $m + 1$ zeros at each of $\{x, y\} = \{\pm 1, \pm 1\}$. (See lemma 3.1 of Warren [156] for a proof of this fact.) Now, note that the polynomial $d[x]^i d[y]^j s[x, y]$ has zeros of order $m+1$ at $\{x, y\} = \{\pm 1, \pm 1\}$ for all $i + j == m + 1$. Therefore, $d[x]^i d[y]^j s[x, y]$ can be expressed as a polynomial combination of these functions with the ith row of the matrix consisting of the polynomial coefficients of the functions $d[x^2]^{m+1}$, $d[x^2]^m d[y^2], \ldots,$ $d[y^2]^{m+1}$.

For example, the subdivision mask $s[x, y]$ for the four-direction quadratic box spline has the form $\frac{1}{4}(1 + x)(1 + y)(1 + xy)(x + y)$. This mask has zeros of order one at $\{\{1, -1\}, \{-1, 1\}, \{-1, -1\}\}$. Therefore, there exists a matrix subdivision scheme for the second differences $\begin{pmatrix} d[x]^2 \\ d[x]d[y] \\ d[y]^2 \end{pmatrix}$ of the form

$$(t_{ij}[x, y]) == \begin{pmatrix} \frac{1}{2} + \frac{xy}{2} + \frac{xy^2}{2} + \frac{y^3}{2} & -\frac{1}{2} + \frac{x}{2} + \frac{y}{2} - \frac{xy}{2} & 0 \\ 0 & \frac{x}{2} + \frac{y}{2} + \frac{x^2y}{2} + \frac{xy^2}{2} & 0 \\ 0 & -\frac{1}{2} + \frac{x}{2} - \frac{xy}{2} + \frac{y^3}{2} & \frac{1}{2} + \frac{x^3}{2} + \frac{y}{2} + \frac{xy}{2} \end{pmatrix}. \qquad (3.16)$$

Given this matrix of generating functions $t_{ij}[x, y]$, we can now apply an analysis similar to that for the case of $m == 0$ to prove that this scheme is $\mathcal{C}^1$. The key is to convert the matrix of generating functions $t_{ij}[x, y]$ into an equivalent block matrix $T = (T_{ij})$ and compute its norm using equation 3.14. If there exists an $n > 0$ such that $\|T^n\| < 1$, the differences of the order m divided differences converge to zero as $k \to \infty$. Thus, by Theorem 3.9, the subdivision produces limit functions that are $\mathcal{C}^m$ continuous.

THEOREM 3.11 Let $s[x, y]$ be a subdivision mask $s[x, y]$ satisfying equation 3.15 that produces a limit function $p_\infty[x, y] \in \mathcal{C}^{m-1}$. If $\|T\| < 1$, where $T = (T_{ij})$ is the block matrix version of the matrix $(t_{ij}[x, y])$ in Theorem 3.10, then $p_\infty[x, y] \in \mathcal{C}^m$ for all initial vectors p_0 with bounded norm.

Proof Removing the first two entries of the difference vector $2^{mk} \begin{pmatrix} d[x]^{m+1} \\ d[x]^m d[y] \\ . \\ d[x]\, d[y]^m \\ d[y]^{m+1} \end{pmatrix}$ and factoring these two expressions reveals that the differences of the mth divided difference with respect to x, $\binom{d[x]}{d[y]} d_k[x]^m p_k[x, y]$, converge uniformly to zero, in that $\|T\| < 1$. Therefore, via Theorem 3.9, the limit of $d_k[x]^m p_k[x, y]$

as $k \to \infty$ is a continuous function. Because the original scheme was assumed to converge uniformly to a $\mathcal{C}^{m-1}$ continuous function, these divided differences $d_k[x]^m p_k[x, y]$ converge to $p_\infty^{(m,0)}[x, y]$ as $k \to \infty$, with the proof being completely analogous to that of Theorem 3.3. Because a similar argument holds for the remaining partial derivatives of $p_\infty[x, y]$ of order m, $p_\infty[x, y]$ is a $\mathcal{C}^m$ function.

Returning to our example of the four-direction quartic box splines, if $T = (T_{ij})$ is the block matrix version of the subdivision mask of equation 3.16, then $\|T\| = 1$, $\|T^2\| = 1$, and $\|T^3\| = \frac{1}{2}$ (♔). Consequently, four-direction quadratic box splines are at least $\mathcal{C}^1$. We conclude the chapter by analyzing the smoothness of two bivariate interpolatory subdivision schemes.

3.3.4 Example: Bivariate Interpolatory Schemes

Finally, we consider the smoothness of two bivariate subdivision schemes. A bivariate subdivision scheme is *interpolatory* if its mask $s[x, y]$ has the form $s[\![0, 0]\!] == 1$ and $s[\![2i, 2j]\!] == 0$ for all $i, j \neq 0$. (The remaining entries can be chosen arbitrarily.) As in the univariate case, the effect of this condition on the associated subdivision matrix S is to force one of the four types of rows in S to be shifts of the unit vector. As a result, the action of an interpolatory scheme is to leave the coefficients p_{k-1} at $\frac{1}{2^{k-1}}\mathbb{Z}^2$ unperturbed and to simply insert new coefficients into p_k at the midpoints of edges and the center of faces in $\frac{1}{2^{k-1}}\mathbb{Z}^2$. We conclude this chapter by considering two examples of bivariate interpolatory schemes: one for three-direction triangular grids and one for tensor product grids.

The Butterfly scheme of Dyn et al. [55] is a bivariate interpolatory subdivision scheme defined on a triangular grid. This triangular grid can be embedded in $\mathbb{Z}^2$ via the three direction vectors $\{\{1, 0\}, \{0, 1\}, \{1, 1\}\}$. Its subdivision mask $s[x, y]$ has the form

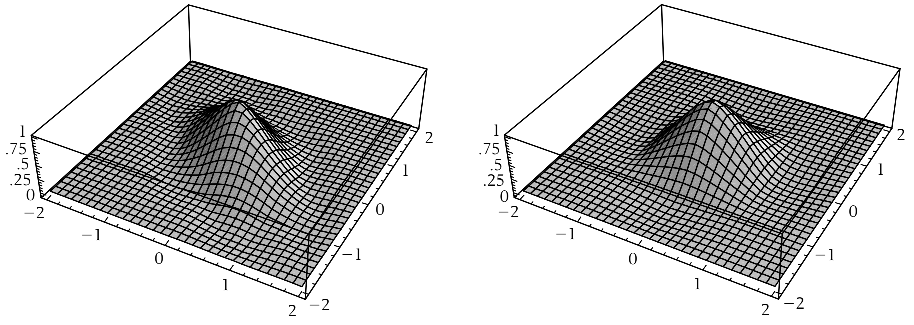
$$s[x, y] = \frac{1}{16}\begin{pmatrix} x^{-3} & x^{-2} & x^{-1} & x^0 & x^1 & x^2 & x^3 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & -1 & 0 & 2 & 0 & -1 \\ 0 & -1 & 2 & 8 & 8 & 2 & -1 \\ 0 & 0 & 8 & 16 & 8 & 0 & 0 \\ -1 & 2 & 8 & 8 & 2 & -1 & 0 \\ -1 & 0 & 2 & 0 & -1 & 0 & 0 \\ 0 & -1 & -1 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} y^{-3} \\ y^{-2} \\ y^{-1} \\ y^0 \\ y^1 \\ y^2 \\ y^3 \end{pmatrix}.$$

Applying basic calculus, it is easy to show that $s[x, y]$ has zeros of order four at $(x, y) = \{\{-1, 1\}, \{1, -1\}, \{-1, -1\}\}$. This observation agrees with the fact that this scheme reproduces polynomials of up to degree three. To analyze the behavior of this scheme, we next apply Theorem 3.8 with $m == 0$ and compute the subdivision matrix T for the first differences. (See the implementation for the actual matrix of masks (♚).) This matrix T has a norm of $\frac{7}{8}$. Therefore, the Butterfly scheme converges to a continuous function.

To continue our smoothness analysis, we next compute the subdivision matrix T for the difference of divided differences (i.e., $m == 1$). In this case, $\|T\| == \frac{3}{2}$. However, computing the norm of higher powers of T, $\|T^n\|$, yields values 1.5, 2.06658, 1.8632, 1.47043, 1.16169, and finally 0.879105 for T^6. Therefore, the Butterfly scheme produces $\mathcal{C}^1$ limit functions on uniform grids.

Finally, we note that the Butterfly scheme cannot produce $\mathcal{C}^2$ limit functions. For initial coefficient vectors p_0 that are cylinders with respect to the x axis (i.e., $p_0[\![i, j]\!] == p_0[\![i, k]\!]$ for all i, j, k), the action of the subdivision mask $s[x, y]$ on p_0 is equivalent to the action of the mask $s[x, 1]$ on p_0. Because this mask replicates the action of the four-point scheme, which is known to be exactly $\mathcal{C}^1$, the Butterfly scheme is also exactly $\mathcal{C}^1$. The left-hand portion of Figure 3.6 shows a scaling function for the Butterfly scheme.

We conclude by analyzing the smoothness of a second type of interpolatory scheme for two-direction (tensor product) meshes. The scheme has a bivariate



(a) (b)

Figure 3.6 Scaling functions for (a) the Butterfly subdivision scheme and (b) an interpolating scheme on quads.

mask s[x, y] of the form

$$s[x, y] = \frac{1}{32}\begin{pmatrix} x^{-3} & x^{-2} & x^{-1} & x^{0} & x^{1} & x^{2} & x^{3}\end{pmatrix}\begin{pmatrix} 0 & 0 & -1 & -2 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 10 & 18 & 10 & 0 & -1 \\ -2 & 0 & 18 & 32 & 18 & 0 & -2 \\ -1 & 0 & 10 & 18 & 10 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -2 & -1 & 0 & 0 \end{pmatrix}\begin{pmatrix} y^{-3} \\ y^{-2} \\ y^{-1} \\ y^{0} \\ y^{1} \\ y^{2} \\ y^{3} \end{pmatrix}.$$

Note that this subdivision mask is *not* the tensor product of the four-point mask with itself (as used in Kobbelt [81]). In particular, s[x, y] has a smaller support than the tensor product mask because the coefficients corresponding to $x^{\pm 3}y^{\pm 3}$ are zero in s[x, y].

The mask s[x, y] has zeros of order three at (x, y) = {{−1, 1}, {1, −1}, {−1, −1}}, corresponding to the fact that the scheme reproduces polynomials of up to degree two. To analyze the behavior of this scheme, we first apply Theorem 3.8 with m == 0 and compute the subdivision matrix T for the first differences. (See the implementation for the actual matrix of masks (♔).) This matrix T has a norm of $\frac{3}{4}$. Therefore, the scheme converges to a continuous function. To continue the analysis, we compute the subdivision matrix T for the differences of first divided differences (i.e., m == 1). The norm of this matrix has ‖T‖ == $\frac{5}{4}$. However, as n increases, $\|T^n\|$ takes on the values 1.25, 1.30078, 1.01002, 0.715133. Therefore, this scheme produces $\mathcal{C}^1$ limit functions on uniform grids. An argument similar to that used for the Butterfly scheme limits the smoothness of this scheme to being $\mathcal{C}^1$. The right-hand side of Figure 3.6 shows an example of a scaling function for this scheme.

CHAPTER 4

A Differential Approach to Uniform Subdivision

At its heart, much of the theory of splines revolves around a simple idea. Given a space of functions and a set of knots, splines are smooth, piecewise functions whose segments (the parts between two adjacent knots) lie in the given function space. Unfortunately, this simplicity often gets lost in the details of attempting to piece these segments of a spline together smoothly. One of the main benefits of the integral approach to subdivision described in Chapter 2 is that these smoothness conditions used in connecting adjacent segments of the spline are never considered explicitly. Instead, the smoothness of the splines arises directly from the integral definition of the scaling functions for B-splines and box splines.

This chapter introduces a differential method for constructing subdivision schemes. Given a differential equation whose solutions form the segments of the spline, this method involves deriving a finite difference equation that characterizes a sequence of solutions to the differential equation on increasingly fine grids. If the right-hand side of this difference equation is chosen appropriately, successive solutions to this finite difference equation, $p_{k-1}[x]$ and $p_k[x]$, can be related via a subdivision recurrence of the form $p_k[x] = s_{k-1}[x]p_{k-1}[x^2]$. This differential method is essentially a new technique, introduced by the authors in [160]. The primary advantage of this approach is that it can reproduce all of the schemes resulting from the integral approach (such as B-splines and box splines) while yielding new schemes (such as those for thin plates [162] and slow flows [163]) that do not possess any equivalent integral formulation. (These new subdivision schemes are the topic of Chapter 5.)

The first two sections of this chapter reconsider the two examples of Chapter 2, B-splines and box splines, from a differential point of view. In each case, the differential approach yields the same subdivision mask as the integral approach. By taking a slightly more abstract, operator-based view, the relation between these two approaches can be captured with beautiful precision. Starting from the differential approach, the third section derives the subdivision scheme for exponential B-splines. The chapter concludes with a section devoted to a particularly useful variant of exponential splines.

4.1 Subdivision for B-splines

As its first example, Chapter 2 considered the case of polynomial splines. Specifically, we constructed the refinement relation for the B-spline basis function from a recursive definition based on integration. In this chapter, we revisit the case of polynomial splines and rederive their associated subdivision scheme using a differential approach. The starting point for the differential approach is a differential equation that governs segments of the splines. In the case of polynomial splines of order m, because the segments are polynomials of degree $\mathsf{m}-1$, they satisfy a homogeneous differential equation of order m of the form

$$p^{(\mathsf{m})}[x] == 0.$$

Given an initial vector of control points p_0, our goal is to construct a subdivision scheme that converges to a function $p[x]$ that approximates p_0 on the grid $\mathbb{Z}$ and whose pieces satisfy this differential equation (i.e., are polynomials of degree $\mathsf{m}-1$). At the integer knots $\mathbb{Z}$, $p[x]$ should have maximal smoothness (i.e., be $\mathcal{C}^{\mathsf{m}-2}$ continuous).

Our approach is to derive a right-hand side for this differential equation that characterizes the B-spline curve $p[x]$ as a function of the initial control points p_0. Given this differential equation, we then construct a sequence of associated finite difference equations. If we have chosen the right-hand side of this differential equation correctly, these difference equations possess a sequence of discrete solutions $\mathsf{p}_\mathsf{k}[\mathsf{x}]$ whose coefficients converge to the B-spline $p[x]$. Finally, comparing the finite difference equations for two successive levels yields a subdivision mask $\mathsf{s}_{\mathsf{k}-1}[\mathsf{x}]$ that relates two successive solutions $\mathsf{p}_{\mathsf{k}-1}[\mathsf{x}]$ and $\mathsf{p}_\mathsf{k}[\mathsf{x}]$. This construction yields a remarkably simple expression for $\mathsf{s}_{\mathsf{k}-1}[\mathsf{x}]$ in terms of the divided difference masks $\mathsf{d}_{\mathsf{k}-1}[\mathsf{x}]$ and $\mathsf{d}_\mathsf{k}[\mathsf{x}]$ associated with the differential equation.

4.1.1 A Differential Equation for B-splines

By definition, a B-spline curve $p[x]$ is a linear combination of integer translates of the B-spline basis function $n^{\mathsf{m}}[x]$. In section 2.3.1, this basis function itself is expressed as the difference of integer translates of the truncated power $c^{\mathrm{m}}[x]$. Thus, if this truncated power can be expressed as the solution to a differential equation, this relation can be used to express a B-spline curve $p[x]$ as the solution to a related differential equation. To this end, recall that the truncated power $c^{\mathrm{m}}[x]$ satisfies the integral recurrence of equation 2.19 as

$$c^{\mathrm{m}}[x] = \int_0^{\infty} c^{\mathrm{m}-1}[x-t]\, \mathrm{d}t,$$

where $c^1[x]$ is the unit step function. At this point, we introduce two operators that play a crucial role throughout this chapter. These operators are complementary in nature and help to illuminate the link between the integral and the differential approach to subdivision. The first operator $\mathcal{I}[x]$ is an integral operator of the form

$$\mathcal{I}[x]\, p[x] = \int_0^{\infty} p[x-t]\, \mathrm{d}t.$$

(Note that the limit of $p[x]$ as $x \rightarrow -\infty$ must be zero for the integral to converge.) Repeated applications of $\mathcal{I}[x]$ can be written in operator form as $\mathcal{I}[x]^{\mathrm{m}}$.

For example, the integral recurrence for the truncated powers $c^{\mathrm{m}}[x]$ can be written in operator form as $c^{\mathrm{m}}[x] = \mathcal{I}[x] c^{\mathrm{m}-1}[x]$. The base case for this recurrence can be lowered to $\mathrm{m} == 0$ by observing that $c^1[x] == \mathcal{I}[x]\delta[x]$, where $\delta[x]$ is the Dirac delta function. This function, $\delta[x]$, is a distribution that is zero everywhere except at the origin. At the origin, $\delta[x]$ has a unit "impulse" whose integral is one. (See pages 69–97 in Bracewell [12] and pages 79–82 in Spanier et al. [143] for more details on distributions and the Dirac delta.) Unwinding this recurrence to the base case of $c^0[x] == \delta[x]$ yields an explicit integral equation for $c^{\mathrm{m}}[x]$ of the form

$$c^{\mathrm{m}}[x] = \mathcal{I}[x]^{\mathrm{m}} \delta[x]. \tag{4.1}$$

Similarly, the integral definition of the B-spline basis functions, equation 2.3, can be written as an integral recurrence of the form $n^{\mathrm{m}}[x] = \mathcal{I}[x](n^{\mathrm{m}-1}[x] - n^{\mathrm{m}-1}[x-1])$. Again, this recurrence can be extended to have a new base case of the form $n^0[x] = \delta[x]$. In fact, unwinding this recurrence and replacing $n^0[x]$ by $c^0[x]$ in the base case yields an alternative proof for Theorem 2.4.

The second important operator is $\mathcal{D}[x]$, the derivative operator. Application of this operator takes the form $\mathcal{D}[x]\, p[x]$, which by definition is simply the first derivative of $p[x]$ with respect to x (i.e., $p^{(1)}[x]$). Again, repeated application of the operator $\mathcal{D}[x]$ is written as $\mathcal{D}[x]^{\mathrm{m}}$ and corresponds to taking the mth derivative of a function with respect to x. As one might expect, these operators are closely related. In particular, $\mathcal{I}[x]$ and $\mathcal{D}[x]$ are inverses in Theorem 4.1.

THEOREM 4.1 Let $p[x]$ be a function whose integral $\mathcal{I}[x]\,p[x]$ is a continuous function. Then

$$\mathcal{D}[x]\,\mathcal{I}[x]\,p[x] == p[x].$$

Proof By definition, $\mathcal{I}[x]\,p[x]$ is the integral $\int_0^\infty p[x-t]\,\mathbb{d}t$. By a simple change of variables, the integral can be rewritten as $\int_{-\infty}^{x} p[t]\,\mathbb{d}t$. Taking the derivative of this integral with respect to x corresponds to taking the limit of the expression

$$\frac{1}{\epsilon}\left(\int_{-\infty}^{x} p[t]\,\mathbb{d}t - \int_{-\infty}^{x-\epsilon} p[t]\,\mathbb{d}t\right)$$

as $\epsilon \to 0$. Combining the two integrals yields a new expression of the form $\frac{1}{\epsilon}\int_{x-\epsilon}^{x} p[t]\,\mathbb{d}t$. Because the value of this integral is a continuous function (by assumption), the limit as $\epsilon \to \infty$ is exactly $p[x]$.

(A similar theorem holds in the case when $p[x]$ is the Dirac delta function and the integral $\mathcal{I}[x]\,p[x]$ is the step function $c^1[x]$.) Because $\mathcal{D}[x]$ is the left inverse of $\mathcal{I}[x]$, multiplying both sides of equation 4.1 by $\mathcal{D}[x]^{\mathrm{m}}$ yields a differential equation for the truncated powers of the form

$$\mathcal{D}[x]^{\mathrm{m}} c^{\mathrm{m}}[x] == \delta[x]. \tag{4.2}$$

Thus, the truncated powers $c^{\mathrm{m}}[x]$ are the solutions to the differential equation $p^{(\mathrm{m})}[x] == \delta[x]$. Solutions to differential equations whose right-hand side is the Dirac delta $\delta[x]$ are often referred to as a *Green's function,* associated with the differential operator. Note that the Green's functions for this differential operator are unique only up to addition of any polynomial of degree $\mathrm{m}-1$.

Having derived a differential equation for the truncated powers, we next construct the differential equation for a basis function $n^{\mathrm{m}}[x]$ using Theorem 2.4. In particular, if $\mathrm{d}^{\mathrm{m}}[\mathrm{x}]$ is the generating function $(1-\mathrm{x})^{\mathrm{m}}$, the B-spline basis function $n^{\mathrm{m}}[x]$ has the form $\sum_{\mathrm{i}=0}^{\mathrm{m}} \mathrm{d}^{\mathrm{m}}[\![\mathrm{i}]\!]\, c^{\mathrm{m}}[x-\mathrm{i}]$, where $\mathrm{d}^{\mathrm{m}}[\![\mathrm{i}]\!]$ is the coefficient of x^{i} in $\mathrm{d}^{\mathrm{m}}[\mathrm{x}]$. Substituting this relation into equation 4.2 reveals that $n^{\mathrm{m}}[x]$ satisfies the differential equation

$$\mathcal{D}[x]^{\mathrm{m}} n^{\mathrm{m}}[x] == \sum_{\mathrm{i}=0}^{\mathrm{m}} \mathrm{d}^{\mathrm{m}}[\![\mathrm{i}]\!]\,\delta[x-\mathrm{i}]. \tag{4.3}$$

An equivalent method of arriving at this differential equation starts from the integral definition of B-spline basis functions expressed in terms of the integral operator $\mathcal{I}[x]$:

$$n^{\mathrm{m}}[x] = \mathcal{I}[x](n^{\mathrm{m}-1}[x] - n^{\mathrm{m}-1}[x-1]). \tag{4.4}$$

Multiplying both sides of this equation by $\mathcal{D}[x]$ yields a differential recurrence for the B-spline basis functions of the form

$$\mathcal{D}[x]n^{\mathrm{m}}[x] == n^{\mathrm{m}-1}[x] - n^{\mathrm{m}-1}[x-1].$$

Iterating this recurrence m times and substituting $\delta[x]$ for $n^{0}[x]$ in the base case yields equation 4.3 as desired.

To derive the differential equation for an arbitrary B-spline $p[x]$, we recall that $p[x]$ can be written as a linear combination of translated basis functions: $\sum_{\mathrm{i}} \mathrm{p}_0[\![\mathrm{i}]\!] n^{\mathrm{m}}[x-\mathrm{i}]$, where p_0 is the initial vector of control points. Substituting this definition into the previous equation reveals that $p[x]$ satisfies a differential equation of the form

$$\mathcal{D}[x]^{\mathrm{m}} p[x] == \sum_{\mathrm{i}} \mathrm{dp}_0[\![\mathrm{i}]\!]\, \delta[x-\mathrm{i}], \tag{4.5}$$

where dp_0 is the vector of coefficients of the generating function $\mathrm{d}^{\mathrm{m}}[\mathrm{x}]\mathrm{p}_0[\mathrm{x}]$. Note that these coefficients are simply the mth differences of successive control points in p_0 on the grid $\mathbb{Z}$. Because the right-hand side of this equation is zero between integer knots, the segments of $p[x]$ are polynomials of degree $\mathrm{m}-1$. Due to the placement of the Dirac delta functions at the integer knots, the function $p[x]$ has at least $\mathcal{C}^{\mathrm{m}-2}$ continuity at these knots.

4.1.2 A Finite Difference Equation for B-splines

Our next task is to develop a finite difference equation that models equation 4.5. Although finite difference equations and their associated finite difference schemes are the subject of entire books (see [96, 144] for examples), the basic idea behind finite differencing is relatively simple: develop a set of discrete linear equations that approximates the behavior of a continuous differential equation at a set of sample points. If these sample points are chosen to lie on a uniform grid, these linear equations can be expressed succinctly using generating functions. The most important generating function used in this approach is the finite difference mask $1-\mathrm{x}$. The coefficients of this mask are the discrete analogs of the left-hand side of the differential equation (i.e., of the differential operator).

To discretize equation 4.5, our first task is to construct a difference mask $\mathsf{d}_k^m[x]$ that models the action of the differential operator $\mathcal{D}[x]^m$ on the $\frac{1}{2^k}$-integer grid. Luckily, we can rely on our observation from equation 3.7:

$$p^{(1)}[x] = \lim_{k\to\infty} 2^k \left(p[x] - p\left[x - \frac{1}{2^k}\right]\right). \tag{4.6}$$

Given a generating function $\mathsf{p}_k[x]$ whose coefficients are samples of a continuous function $p[x]$ on the knot sequence $\frac{1}{2^k}\mathbb{Z}$, we can compute a new generating function whose coefficients approximate $\mathcal{D}[x]p[x]$ on $\frac{1}{2^k}\mathbb{Z}$ by multiplying $\mathsf{p}_k[x]$ with the generating function $2^k(1 - x)$. The coefficients of $2^k(1 - x)\mathsf{p}_k[x]$ can be simply viewed as discrete divided differences of p_k taken on the grid $\frac{1}{2^k}\mathbb{Z}$. Following a similar line of reasoning, the discrete analog of the mth derivative operator $\mathcal{D}[x]^m$ on $\frac{1}{2^k}\mathbb{Z}$ is a generating function $\mathsf{d}_k^m[x]$ of the form

$$\mathsf{d}_k^m[x] = (2^k(1 - x))^m == 2^{mk}(1 - x)^m.$$

Note that multiplying the difference mask $\mathsf{d}_0^m[x]$ by a generating function $\mathsf{p}[x]$ yields a generating function whose coefficients are the mth differences of the coefficients of $\mathsf{p}[x]$. For example, the generating function $\mathsf{d}_0^2[x]\mathsf{p}[x]$ has coefficients that are second differences of the coefficients $\mathsf{p}\llbracket i \rrbracket$ of $\mathsf{p}[x]$ and that have the form $\mathsf{p}\llbracket i \rrbracket - 2\mathsf{p}\llbracket i - 1 \rrbracket + \mathsf{p}\llbracket i - 2 \rrbracket$. The corresponding coefficients of $\mathsf{d}_k^m[x]\mathsf{p}[x]$ are mth-divided differences of the coefficients of $\mathsf{p}[x]$ with respect to the grid $\frac{1}{2^k}\mathbb{Z}$.

The difference mask $\mathsf{d}_k^m[x]$ has the property that if the coefficients of $\mathsf{p}_k[x]$ are samples of a polynomial of degree $m - 1$ on the grid $\frac{1}{2^k}\mathbb{Z}$ then $\mathsf{d}_k^m[x]\mathsf{p}_k[x]$ is exactly zero. This observation is simply a rephrasing of the fact that the mth divided difference of the sample of a polynomial function of degree $m - 1$ on a uniform grid is exactly zero. Because m remains constant through the entire construction that follows, we drop the superscript m in $\mathsf{d}_k^m[x]$ and instead use $\mathsf{d}_k[x]$.

To complete our discretization of equation 4.5, we must settle on a discrete analog of the Dirac delta $\delta[x]$. Recall that by definition this function is zero everywhere but the origin; there it has an impulse with unit integral. The discrete analog of $\delta[x]$ on the grid $\frac{1}{2^k}\mathbb{Z}$ is a vector that is zero everywhere except at the origin, where the vector has value 2^k. (The factor of 2^k is chosen so that this discrete approximation has unit integral with respect to the grid $\frac{1}{2^k}\mathbb{Z}$.) The coefficient of this vector can be represented as the generating function 2^k because all terms of this generating function of the form x^i have coefficient zero for $i \neq 0$.

Finally, integer translates of the Dirac delta $\delta[x - i]$ can be modeled on the grid $\frac{1}{2^k}\mathbb{Z}$ by multiplying the generating function for $\delta[x]$, 2^k, by the monomial $x^{2^k i}$.

Absorbing these powers into the generating function $\mathrm{dp}[\mathrm{x}] == \mathrm{d}_0[\mathrm{x}]\mathrm{p}_0[\mathrm{x}]$ yields a finite difference equation of the form

$$\mathrm{d}_k[\mathrm{x}]\mathrm{p}_k[\mathrm{x}] == 2^k \mathrm{d}_0\left[\mathrm{x}^{2^k}\right]\mathrm{p}_0\left[\mathrm{x}^{2^k}\right]. \tag{4.7}$$

Given $\mathrm{p}_0[\mathrm{x}]$, this finite difference equation would ideally have a single solution $\mathrm{p}_k[\mathrm{x}]$. Unfortunately, the solution $\mathrm{p}_k[\mathrm{x}]$ to this equation is not unique because the difference equation $\mathrm{d}_k[\mathrm{x}]\mathrm{p}_k[\mathrm{x}] == 0$ has m solutions corresponding to the samples of polynomials of degree $\mathrm{m} - 1$. However, there does exist one sequence of the solutions $\mathrm{p}_k[\mathrm{x}]$ that converges to the B-spline $p[x]$. The next section demonstrates a simple method of constructing this sequence of solutions.

4.1.3 The Associated Subdivision Scheme

For each value of k, equation 4.7 characterizes a solution p_k on the grid $\frac{1}{2^k}\mathbb{Z}$ that is independent of the solution p_{k-1} on the coarser grid $\frac{1}{2^{k-1}}\mathbb{Z}$. However, the beauty of this particular finite difference equation is that there exists a simple recurrence for constructing an entire sequence of solutions p_k to equation 4.7. In particular, these solutions can be constructed using a subdivision scheme.

THEOREM 4.2 Given an initial set of control points p_0, consider the function $\mathrm{p}_k[\mathrm{x}]$ satisfying the recurrence relation $\mathrm{p}_k[\mathrm{x}] = \mathrm{s}_{k-1}[\mathrm{x}]\mathrm{p}_{k-1}[\mathrm{x}^2]$, where

$$\mathrm{s}_{k-1}[\mathrm{x}] = \frac{2\mathrm{d}_{k-1}[\mathrm{x}^2]}{\mathrm{d}_k[\mathrm{x}]} \tag{4.8}$$

for all $\mathrm{k} > 0$. Then $\mathrm{p}_k[\mathrm{x}]$ satisfies the finite difference relation of equation 4.7 for all $\mathrm{k} \geq 0$.

Proof The proof is by induction on k. For $\mathrm{k} == 0$, equation 4.7 follows reflexively. Next, assume that $\mathrm{p}_{k-1}[\mathrm{x}]$ satisfies equation 4.7; that is,

$$\mathrm{d}_{k-1}[\mathrm{x}]\mathrm{p}_{k-1}[\mathrm{x}] == 2^{k-1}\mathrm{d}_0\left[\mathrm{x}^{2^{k-1}}\right]\mathrm{p}_0\left[\mathrm{x}^{2^{k-1}}\right].$$

Replacing x with x^2 in these generating functions and multiplying each side by 2 yields a new equation of the form

$$2\mathrm{d}_{k-1}[\mathrm{x}^2]\mathrm{p}_{k-1}[\mathrm{x}^2] == 2^k\mathrm{d}_0\left[\mathrm{x}^{2^k}\right]\mathrm{p}_0\left[\mathrm{x}^{2^k}\right].$$

Substituting the right-hand side of equation 4.8 into $p_k[x] == s_{k-1}[x]p_{k-1}[x^2]$ and multiplying by $d_k[x]$ yields a finite difference equation of the form

$$d_k[x]p_k[x] == 2d_{k-1}[x^2]p_{k-1}[x^2].$$

Combining the last two equations completes the induction and proves the theorem.

For polynomial splines, the difference mask $d_k[x]$ in equation 4.8 has the form $(2^k(1-x))^m$. Therefore, the subdivision mask $s_{k-1}[x]$ has the form

$$s_{k-1}[x] = 2\frac{d_{k-1}[x^2]}{d_k[x]} == 2\frac{2^{m(k-1)}(1-x^2)^m}{2^{mk}(1-x)^m} == 2\left(\frac{1+x}{2}\right)^m.$$

This mask, $s_{k-1}[x]$, is exactly the Lane-Riesenfeld mask for B-splines derived in Chapter 2. However, independent of our previous knowledge concerning B-splines, we can still deduce several important properties of this scheme directly from the structure of the subdivision mask $s_{k-1}[x]$. For example, the convergence of the coefficients of the solution $p_k[x]$ to a limit function $p_\infty[x]$ can be proved directly from the structure of the subdivision mask $s_{k-1}[x]$ using the techniques of Chapter 3. Further, the function $p_\infty[x]$ possesses the following properties:

- $p_\infty[x]$ is a C^{m-2} piecewise polynomial function of degree $m-1$, with knots at the integers $\mathbb{Z}$. This observation follows from the fact that $p_\infty[x]$ is a linear combination of integer translates of the truncated power $c^m[x]$.
- The associated basis function $n[x]$ for this scheme is non-negative and is supported on the interval $[0, m]$. This observation follows from the fact that the subdivision mask $s_{k-1}[x]$ has only $m+1$ non-zero coefficients, all of which are positive.
- The basis function $n[x]$ has unit integral. This observation follows from the fact that the coefficients of the generating function $\prod_{i=1}^{k} s_{i-1}[x^{2^{k-i}}]$ are converging to the basis function $n[x]$ when plotted on the grid $\frac{1}{2^k}\mathbb{Z}$. Because the masks $s_{k-1}[x]$ satisfy $s_{k-1}[1] == 2$ for all $k > 0$, the sum of the coefficients in this product of generating functions is 2^k. Therefore, the integrals of these bounded approximations to $n[x]$ are converging to one, and thus the integral of $n[x]$ is also one.

The beauty of this differential approach is that all that is needed to construct the subdivision mask $s_{k-1}[x]$ is the original differential operator governing the spline.

Given this operator, we simply discretized it over the grid $\frac{1}{2^k}\mathbb{Z}$ and constructed the discrete difference mask $\mathrm{d}_k[\mathrm{x}]$. The corresponding subdivision mask $\mathrm{s}_{k-1}[\mathrm{x}]$ had the form $\frac{2\mathrm{d}_{k-1}[\mathrm{x}^2]}{\mathrm{d}_k[\mathrm{x}]}$. The rest of this chapter and the next demonstrate that this approach can be applied to a wide range of differential operators, including those in more than one variable.

4.2 Subdivision for Box Splines

Chapter 2 constructed box-spline scaling functions in terms of an integral recurrence on a set of direction vectors. In this section, we recast box splines from a differential viewpoint. As in the univariate case, our approach is to construct explicitly the integral operator used in defining the box-spline scaling functions. Using the differential operator that is the inverse of the integral operator, we next show that the cone splines are Green's functions associated with this operator. Because the box splines can be written as linear combinations of cone splines, the partial differential equation for box splines follows immediately. We conclude the section by deriving the finite difference equation associated with this differential equation and an associated subdivision mask that relates solutions to this difference equation on successive grids.

4.2.1 A Differential Equation for Box Splines

The integral operator $\mathcal{I}[x]$ was the crux of our initial integral construction for univariate B-splines. Using the inverse differential operator $\mathcal{D}[x]$, we were able to construct the differential equation for B-splines from the original integral definition. Two-dimensional versions of these operators play analogous roles for box splines. Given a direction $\{\mathrm{a}, \mathrm{b}\}$, the integral operator $\mathcal{I}[x^{\mathrm{a}}, y^{\mathrm{b}}]$ computes the *directional integral* of a function $p[x, y]$ via the definition

$$\mathcal{I}[x^{\mathrm{a}}, y^{\mathrm{b}}]\, p[x, y] = \int_0^\infty p[x - \mathrm{a}t, y - \mathrm{b}t]\, \mathbb{d}t.$$

One simple application of this operator is the recursive definition for cone splines given in equation 2.21. If the set of direction vectors $\widetilde{\Sigma}$ can be decomposed into $\Sigma \cup \{\{\mathrm{a}, \mathrm{b}\}\}$, the cone spline $c^{\widetilde{\Sigma}}[x, y]$ can be expressed in terms of the cone spline $c^{\Sigma}[x, y]$ via

$$\begin{aligned} c^{\widetilde{\Sigma}}[x, y] &= \int_0^\infty c^{\Sigma}[x - \mathrm{a}t, y - \mathrm{b}t]\, \mathbb{d}t \\ &= \mathcal{I}[x^{\mathrm{a}}, y^{\mathrm{b}}]\, c^{\Sigma}[x, y]. \end{aligned}$$

As was the case for B-splines, the base case for this recurrence can be reduced to the case when Σ is empty. In this case, $c^{\{\}}[x, y]$ is taken to be the Dirac delta function in two dimensions, $\delta[x, y]$. Unwinding the previous recurrence to this base case yields an explicit integral expression for the cone spline $c^{\Sigma}[x, y]$ of the form

$$c^{\Sigma}[x, y] = \left(\prod_{\{a,b\} \in \Sigma} \mathcal{I}[x^{a}, y^{b}] \right) \delta[x, y]. \tag{4.9}$$

Note that if the direction vector $\{a, b\}$ appears in Σ with a particular multiplicity the operator $\mathcal{I}[x^{a}, y^{b}]$ appears in this product with the corresponding multiplicity.

Given the integral representation for cone splines, our next task is to construct a differential equation that characterizes cone splines. As in the univariate case, the key is to construct the inverse operator for $\mathcal{I}[x^{a}, y^{b}]$. In two dimensions, this operator is simply the directional derivative. Given a direction vector $\{a, b\}$, the *directional derivative* of the function $p[x, y]$ along the direction $\{a, b\}$ is given by

$$\mathcal{D}[x^{a}, y^{b}]\, p[x, y] = \lim_{t \to 0} \frac{p[x, y] - p[x - at, y - bt]}{t}.$$

Just as in the univariate case, the operators $\mathcal{I}[x^{a}, y^{b}]$ and $\mathcal{D}[x^{a}, y^{b}]$ are inverses of each other in Theorem 4.3.

THEOREM 4.3 Given a direction $\{a, b\}$, let $p[x, y]$ be a function such that its directional integral $\mathcal{I}[x^{a}, y^{b}]\, p[x, y]$ is continuous. Then

$$\mathcal{D}[x^{a}, y^{b}]\, \mathcal{I}[x^{a}, y^{b}]\, p[x, y] == p[x, y]. \tag{4.10}$$

Proof The key observation is to restrict $p[x, y]$ to lines parallel to the vector $\{a, b\}$. For example, consider the line $\{x = x_0 + at,\ y = y_0 + bt\}$. The restriction of the function $p[x, y]$ to this line can be viewed as a univariate function:

$$\widetilde{p}[t] = p[x_0 + at,\ y_0 + bt].$$

On this line, the left-hand side of equation 4.10 reduces to $\mathcal{D}[t]\mathcal{I}[t]\widetilde{p}[t]$. Because $\mathcal{I}[x^{a}, y^{b}]\, p[x, y]$ is continuous, $\mathcal{I}[t]\widetilde{p}[t]$ must also be continuous. Therefore, due to Theorem 4.1, $\mathcal{D}[t]\mathcal{I}[t]\widetilde{p}[t]$ is exactly $\widetilde{p}[t]$. Because this construction holds independent of the choice of $\{x_0, y_0\}$, the theorem is proved.

Based on this theorem, we can derive the partial differential equation governing cone splines. Multiplying both sides of equation 4.9 by the directional derivative $\mathcal{D}[x^{\mathrm{a}}, y^{\mathrm{b}}]$ for each vector $\{\mathrm{a}, \mathrm{b}\} \in \Sigma$ yields the differential equation for cone splines:

$$\left(\prod_{\{\mathrm{a},\mathrm{b}\}\in\Sigma} \mathcal{D}[x^{\mathrm{a}}, y^{\mathrm{b}}]\right) c^{\Sigma}[x, y] == \delta[x, y]. \tag{4.11}$$

(Again, the differential operator $\mathcal{D}[x^{\mathrm{a}}, y^{\mathrm{b}}]$ should appear with the correct multiplicity.) Per Theorem 2.5, the box-spline scaling function $n^{\Sigma}[x, y]$ can be expressed as a linear combination of translates of the cone spline $c^{\Sigma}[x, y]$, where the coefficients of this linear combination are based on the discrete difference mask $\mathrm{d}[\mathrm{x}, \mathrm{y}]$ of the form $\prod_{\{\mathrm{a},\mathrm{b}\}\in\Sigma}(1 - \mathrm{x}^{\mathrm{a}}\mathrm{y}^{\mathrm{b}})$. Applying this theorem to equation 4.11 yields a differential equation for the box-spline scaling functions:

$$\left(\prod_{\{\mathrm{a},\mathrm{b}\}\in\Sigma} \mathcal{D}[x^{\mathrm{a}}, y^{\mathrm{b}}]\right) n^{\Sigma}[x, y] == \sum_{\mathrm{i},\mathrm{j}} \mathrm{d}[\![\mathrm{i}, \mathrm{j}]\!]\, \delta[x - \mathrm{i}, y - \mathrm{j}]. \tag{4.12}$$

To complete our derivation, a box spline $p[x]$ can be written as a linear combination of translated scaling functions: $\sum_{\mathrm{i},\mathrm{j}} \mathrm{p}_0[\![\mathrm{i}, \mathrm{j}]\!] n^{\Sigma}[x - \mathrm{i}, y - \mathrm{j}]$, where p_0 is the initial vector of control points. Substituting this definition into equation 4.12 reveals that $p[x, y]$ satisfies a partial differential equation of the form

$$\left(\prod_{\{\mathrm{a},\mathrm{b}\}\in\Sigma} \mathcal{D}[x^{\mathrm{a}}, y^{\mathrm{b}}]\right) p[x, y] == \sum_{\mathrm{i},\mathrm{j}} \mathrm{dp}_0[\![\mathrm{i}, \mathrm{j}]\!]\, \delta[x - \mathrm{i}, y - \mathrm{j}], \tag{4.13}$$

where dp_0 is the vector of coefficients of the generating function $\mathrm{d}[\mathrm{x}, \mathrm{y}]\mathrm{p}_0[\mathrm{x}, \mathrm{y}]$. As we shall see, these coefficients dp_0 are the discrete directional difference of the control points p_0 analogous to the directional derivative $\prod_{\{\mathrm{a},\mathrm{b}\}\in\Sigma} \mathcal{D}[x^{\mathrm{a}}, y^{\mathrm{b}}]\, p[x, y]$.

4.2.2 The Subdivision Scheme for Box Splines

In the univariate case, our approach is to derive a finite difference equation that models the differential equation for B-splines. In the bivariate case, we follow a similar approach. The key to constructing this finite difference scheme is building the discrete analog of $\mathcal{D}[x^{\mathrm{a}}, y^{\mathrm{b}}]$ on the grid $\frac{1}{2^{\mathrm{k}}}\mathbb{Z}^2$. If we replace t with $\frac{1}{2^{\mathrm{k}}}$ in the definition of the directional derivative, we obtain

$$\mathcal{D}[x^{\mathrm{a}}, y^{\mathrm{b}}]\, p[x, y] = \lim_{\mathrm{k}\to\infty} 2^{\mathrm{k}} \left(p[x, y] - p\left[x - \frac{1}{2^{\mathrm{k}}}\mathrm{a},\ y - \frac{1}{2^{\mathrm{k}}}\mathrm{b}\right]\right). \tag{4.14}$$

Now, consider a vector p_k whose entries are samples of a function $p[x, y]$ on the grid $\frac{1}{2^k}\mathbb{Z}$. The discrete analog of the directional derivative $\mathcal{D}[x^a, y^b]p[x, y]$ is a new vector whose entries are coefficients of the generating function $2^k(1 - x^a y^b)\mathrm{p}_k[x, y]$. In particular, the coefficients of the generating function $2^k(1 - x^a y^b)$ exactly encode the differences required by equation 4.14. To model the product of several differential operators, we simply take the product of their corresponding difference masks. Specifically, the discrete analog of the differential operator $\prod_{\{a,b\}\in\Sigma} \mathcal{D}[x^a, y^b]$ has the form

$$d_k[x, y] = \prod_{\{a,b\}\in\Sigma} 2^k(1 - x^a y^b). \tag{4.15}$$

To complete our discretization of equation 4.13, we note that the analog of $\delta[x, y]$ on the grid $\frac{1}{2^k}\mathbb{Z}^2$ is the constant 4^k. Here, the factor of four arises from the move to two dimensions. (In n dimensions, the constant is $(2^n)^k$.) Given these discretizations, the finite difference equation corresponding to equation 4.13 for polynomials has the form

$$d_k[x, y]\,p_k[x, y] == 4^k d_0\left[x^{2^k}, y^{2^k}\right] p_0\left[x^{2^k}, y^{2^k}\right].$$

Just as in the univariate case, this equation has many solutions. However, there exists a sequence of solutions $p_k[x]$ for this difference equation that are related by a simple subdivision scheme. Given an initial set of control points p_0, these successive solutions $p_k[x, y]$ can be constructed via the subdivision relation $p_k[x, y] = s_{k-1}[x, y]\,p_{k-1}[x^2, y^2]$, where

$$s_{k-1}[x, y] = \frac{4\,d_{k-1}[x^2, y^2]}{d_k[x, y]}.$$

As in the univariate case, the generating function $d_k[x, y]$ divides $d_{k-1}[x^2, y^2]$, yielding a finite subdivision mask. In particular, the difference mask $2^k(1 - x^a y^b)$ divides $2^{k-1}(1 - x^{2a} y^{2b})$, yielding $\frac{1}{2}(1 + x^a y^b)$. Based on this observation, the subdivision mask $s_{k-1}[x, y]$ has the form

$$s_{k-1}[x, y] = 4 \prod_{\{a,b\}\in\Sigma} \left(\frac{1 + x^a y^b}{2}\right). \tag{4.16}$$

This subdivision mask $s_{k-1}[x, y]$ is exactly the subdivision mask for box splines derived in Chapter 2. Therefore, the discrete solutions p_k produced by this scheme converge to the box spline $p[x, y]$ that satisfies equation 4.13.

4.3 Subdivision for Exponential B-splines

The first section in this chapter outlined a method of deriving the subdivision scheme for B-splines directly from the homogeneous differential equation $p^{(m)}[x] == 0$. The crux of this construction is that the subdivision mask $s_{k-1}[x]$ for B-splines has the form $\frac{2d_{k-1}[x^2]}{d_k[x]}$, where $d_k[x]$ is the discretization of the differential operator $\mathcal{D}[x]^m$ on the grid $\frac{1}{2^k}\mathbb{Z}$. Remarkably, this construction can be extended to inhomogeneous differential equations of the form

$$\sum_{i=0}^{m} \beta_i p^{(i)}[x] == 0, \tag{4.17}$$

where the β_i are real constants. In this case, the resulting subdivision mask $s_{k-1}[x]$ defines $\mathcal{C}^{m-2}$ splines whose pieces are exponential functions (the solutions to the differential equation 4.17). These splines, the *exponential B-splines*, have been studied in a number of papers (e.g., by Dyn and Ron [57], Koch and Lyche [88], and Zhang [168]).

In this section, we apply our differential approach to the problem of constructing the subdivision scheme for exponential splines. Instead of attempting to construct the differential equation for exponential B-splines from an integral definition, we simply construct a finite difference scheme of the form

$$d_k[x]p_k[x] == 2^k d_0\left[x^{2^k}\right]p_0\left[x^{2^k}\right], \tag{4.18}$$

where $d_k[x]$ is a difference mask that annihilates samples of the appropriate exponential functions. Given $p_0[x]$, successive solutions $p_k[x]$ to this difference equation are related by a subdivision mask $s_{k-1}[x]$ of the form $\frac{2d_{k-1}[x^2]}{d_k[x]}$. The coefficients of these solutions $p_k[x]$ converge to exponential splines. The remarkable feature of this construction is its simplicity. All that needs to be done is to develop a discrete version $d_k[x]$ of the differential operator in equation 4.17. Given this difference mask, the subdivision mask for the scheme follows immediately.

4.3.1 Discretization of the Differential Equation

Before discretizing equation 4.17, we first rewrite this equation in terms of the differential operator $\mathcal{D}[x]$. Due to the linearity of differentiation, linear combinations of various powers of $\mathcal{D}[x]$ can be used to model differential operators of higher order. In particular, equation 4.17 can be rewritten as $(\sum_{i=0}^{m} \beta_i \mathcal{D}[x]^i)p[x] == 0$. If we assume, without loss of generality, that the leading coefficient β_m is one, the differential operator for this equation can be factored into linear terms:

$$\sum_{i=0}^{m} \beta_i \mathcal{D}[x]^i == \prod_{i=1}^{m} (\mathcal{D}[x] - \alpha_i), \tag{4.19}$$

where the α_i are roots of the polynomial equation $\sum_{i=0}^{m} \beta_i x^i == 0$. Observe that if the coefficients β_i are real then these roots α_i are either real or complex conjugates.

We now proceed to our main challenge: constructing the difference mask $d_k[x]$ for the differential operator $\prod_{i=1}^{m} (\mathcal{D}[x] - \alpha_i)$ on the grid $\frac{1}{2^k}\mathbb{Z}$. As in the previous cases, our approach is to factor the differential operator and then develop difference masks for each of these factors. The product of these difference masks provides the desired discrete operator.

Focusing on a single factor of the form $(\mathcal{D}[x] - \alpha)$, we observe that the differential equation $(\mathcal{D}[x] - \alpha)p[x] == 0$ has a solution of the form $p[x] = e^{\alpha x}$. As in the polynomial case, the difference mask associated with $(\mathcal{D}[x] - \alpha)$ should annihilate samples of this solution taken on the grid $\frac{1}{2^k}\mathbb{Z}$. Given that samples of $e^{\alpha x}$ taken on the grid $\frac{1}{2^k}\mathbb{Z}$ form the geometric sequence $\{\ldots, \eta[k, \alpha]^{-2}, \eta[k, \alpha]^{-1}, 1, \eta[k, \alpha], \eta[k, \alpha]^2, \ldots\}$ where

$$\eta[k, \alpha] = e^{2^{-k} * \alpha},$$

the difference mask $(1 - \eta[k, \alpha]x)$ yields the desired cancellation.

All that remains is to choose an appropriate normalization for this difference mask to account for the grid size. In the polynomial case, this normalization for the difference mask $(1 - x)$ was the factor 2^k. On the grid $\frac{1}{2^k}\mathbb{Z}$, discrete approximations $p_k[x]$ to the function x have the form $\sum_i \frac{i}{2^k} x^i$. Now, the product $2^k(1 - x)p_k[x]$ is the generating function $\sum_i x^i$, which is exactly the discretization for the constant function 1. Thus, the difference mask $2^k(1 - x)$ acts analogously to the differential operator $\mathcal{D}[x]$: it annihilates constants and maps the function x to the function 1.

In the exponential case, the correct normalizing constant is $\frac{\alpha}{\eta[k,\alpha]-1}$. The reader may verify this choice by observing that if $p[x]$ is the constant function 1 then $(\mathcal{D}[x] - \alpha)p[x] == -\alpha$. Likewise, multiplying the difference mask $(\frac{\alpha}{\eta[k,\alpha]-1})(1 - \eta[k, \alpha]x)$ by the generating function $\sum_i x^i$ yields the generating function $-\alpha \sum_i x^i$. This normalization is compatible with the polynomial case, $\alpha == 0$, because the limit as $\alpha \to 0$ of the expression $\frac{\alpha}{\eta[k,\alpha]-1}$ is exactly the constant 2^k of the polynomial case.

If the root α has multiplicity n, solutions to the differential equation $(\mathcal{D}[x] - \alpha)^n p[x] == 0$ also satisfy equation 4.17. For this differential equation, the reader may verify that the space of solutions is simply the span of the functions $p[x] = x^j e^{\alpha x}$ where $j = 0 \ldots n-1$. Again, the difference mask $(1 - \eta[k, \alpha]x)^n$ annihilates samples of the exponential functions $x^{j}\,e^{\alpha x}$ on $\frac{1}{2^k}\mathbb{Z}$ for $j == 0 \ldots n-1$. Thus, the product mask $\prod_{i=1}^{m}(1-\eta[k,\alpha_i]x)$ annihilates samples of solutions to equation 4.17 taken on $\frac{1}{2^k}\mathbb{Z}$. If each term $(1-\eta[k,\alpha_i]x)$ in this product is normalized by the constant $\frac{\alpha_i}{\eta[k,\alpha_i]-1}$, the appropriate difference mask $d_k[x]$ has the form

$$d_k[x] = \prod_{i=1}^{m}\left(\frac{\alpha_i}{\eta[k,\alpha_i]-1}\right)\left(1-\eta[k,\alpha_i]x\right). \tag{4.20}$$

4.3.2 A Subdivision Scheme for Exponential Splines

As shown in Theorem 4.2, successive solutions $p_{k-1}[x]$ and $p_k[x]$ to the finite difference equation $d_k[x]p_k[x] == 2^k d_0[x^{2^k}]p_0[x^{2^k}]$ are related via the recurrence $p_k[x] = s_{k-1}[x]p_{k-1}[x^2]$, where the subdivision mask $s_{k-1}[x]$ has the form

$$s_{k-1}[x] = \frac{2d_{k-1}[x^2]}{d_k[x]}.$$

At this point, our care in defining the difference mask $d_k[x]$ in equation 4.20 pays off. Just as in the polynomial case, the difference mask $d_k[x]$ divides $d_{k-1}[x^2]$. Specifically, each term in the numerator, $d_{k-1}[x^2]$, has the form $(1-(\eta[k,\alpha]x)^2)$. Likewise, each term in the denominator, $d_k[x]$, has the form $(1-\eta[k,\alpha]x)$. The latter term divides the former, leaving $(1+\eta[k,\alpha]x)$. After suitable simplification, $s_{k-1}[x]$ is a finite mask of the form

$$s_{k-1}[x] = 2\prod_{i=1}^{m}\left(\frac{1+\eta[k,\alpha_i]x}{1+\eta[k,\alpha_i]}\right). \tag{4.21}$$

For polynomial B-splines, the subdivision masks $s_{k-1}[x]$ have the property that their coefficients are independent of k. Subdivision schemes for which the subdivision rules are independent of the level of subdivision are known as *stationary schemes*. The subdivision rules for box splines are also stationary. On the other hand, the current subdivision scheme is *nonstationary:* the coefficients of the subdivision mask $s_{k-1}[x]$ in equation 4.21 depend explicitly on the level of subdivision k.

Given an initial set of control points p_0, the subdivision masks $s_{k-1}[x]$ define a sequence of solutions $p_k[x]$ whose coefficients converge to a function $p_\infty[x]$. Based on the structure of the finite difference equation for this scheme, and based on the form of the subdivision mask $s_{k-1}[x]$, this function $p_\infty[x]$ has the following properties:

- $p_\infty[x]$ is a C^{m-2} piecewise exponential function with knots at the integers $\mathbb{Z}$. The smoothness of $p_\infty[x]$ follows directly from the structure of the subdivision mask $s_{k-1}[x]$. Section 4.4.4 discusses techniques for proving this convergence directly from the mask $s_{k-1}[x]$ in the nonstationary case. The convergence of segments of $p_\infty[x]$ to exponentials follows from the fact that the difference mask $d_k[x]$ in equation 4.18 annihilates exactly those generating functions whose coefficients are samples of exponentials on $\frac{1}{2^k}\mathbb{Z}$.

- $p_\infty[x]$ is the sum of integer translates of a scaling function $n[x]$. This scaling function is non-negative and is supported on the interval $[0, m]$. This fact follows from the observation that the subdivision mask $s_{k-1}[x]$ has only $m+1$ non-zero coefficients, all of which are positive.
- The scaling function $n[x]$ has unit integral. This observation follows from the fact that the coefficients of the generating function $\prod_{i=1}^{k} s_{i-1}[x^{2^{k-i}}]$ are converging to the scaling function $n[x]$ when plotted on the grid $\frac{1}{2^k}\mathbb{Z}$. Because the masks $s_{k-1}[x]$ satisfy $s_{k-1}[1] == 2$ for all $k > 0$, the sum of the coefficients in this product of generating functions is 2^k. Therefore, the integrals of these bounded approximations to $n[x]$ are converging to one, and thus the integral of $n[x]$ is also one.

Together, these three properties are sufficient to establish that the function $n[x]$ is unique. In fact, $n[x]$ is the *exponential B-spline basis function* of order m. See Hoschek and Lasser (section 3.6.1 of [76]) for a more general discussion of some of the properties of such splines.

4.3.3 Exponential B-splines as Piecewise Analytic Functions

All things being equal, the natural tendency is to prefer stationary schemes over nonstationary schemes. For example, applying a nonstationary subdivision mask requires keeping track of the level of subdivision. Another difficulty is that the refinement relation for scaling functions associated with nonstationary schemes is more complex. In particular, the scaling functions at various levels of subdivision for a nonstationary scheme are not dilates of a single scaling function $n[x]$. Although this difficulty is not particularly crippling, it does make constructing an analytic representation for the underlying scaling functions more complex. We suggest using nonstationary schemes only when there is a compelling advantage to their use. (See the next section for such an example.)

This section concludes by considering the problem of constructing a piecewise analytic representation for exponential B-splines. In particular, we derive an integral operator that serves as the inverse to the differential operator $(\mathcal{D}[x] - \alpha)$, and use this operator to construct the Green's function associated with the original differential equation. As was the case for B-splines and box splines, the basis functions for exponential splines can then be expressed as a linear combination of these Green's functions.

In the polynomial case, the associated Green's functions are the truncated powers $c^{\mathrm{m}}[x]$ that satisfy the differential equation $\mathcal{D}[x]^{\mathrm{m}} c^{\mathrm{m}}[x] == \delta[x]$. These truncated powers are explicitly defined in a related integral equation of the form $c^{\mathrm{m}}[x] = \mathcal{I}[x]^{\mathrm{m}} \delta[x]$. In the exponential case, we started from a differential definition of the Green's function instead of an equivalent integral definition. In particular, the Green's function $c[x]$ associated with equation 4.17 satisfies the differential equation

$$\prod_{\mathrm{i}=1}^{\mathrm{m}} (\mathcal{D}[x] - \alpha_{\mathrm{i}})\, c[x] == \delta[x]. \tag{4.22}$$

To construct an explicit integral representation for $c[x]$, we now must find the integral operator that serves as an inverse to the differential operator $(\mathcal{D}[x] - \alpha)$. This inverse operator $\mathcal{I}[x, \alpha]$ has the form

$$\mathcal{I}[x, \alpha]\, p[x] = \int_0^{\infty} \mathbb{e}^{\alpha t} p[x - t]\, \mathbb{d}t.$$

De Boor and Lynch [41] give an example of this integral operator for splines in tension. Note that if $\alpha == 0$, then this operator reduces to the integral operator $\mathcal{I}[x]$ of the polynomial case. (For those readers familiar with the theory of differential equations, the operator $\mathcal{I}[x, -\alpha]$ is simply the Laplace transform.)

This integral operator $\mathcal{I}[x, \alpha]$ is the right inverse of $(\mathcal{D}[x] - \alpha)$; that is $(\mathcal{D}[x] - \alpha)\ \mathcal{I}[x, \alpha]\ p[x] == p[x]$. (The proof of the relation involves integration by parts on the intermediate expression $\int \mathbb{e}^{\alpha t} p'[x - t]\, \mathbb{d}\mathrm{t}$.) Therefore, as done in the case of polynomial splines and box splines, we can construct a Green's function $c[x]$ satisfying equation 4.22 via an integral construction of the form

$$c[x] = \prod_{\mathrm{i}=1}^{\mathrm{m}} \mathcal{I}[x, \alpha_{\mathrm{i}}]\, \delta[x].$$

In the following we consider several examples of Green's functions constructed using this operator. In general, $c[x]$ is a $\mathcal{C}^{\mathrm{m}-2}$ piecewise exponential function that is zero for $x < 0$, with a single knot at the origin. If $\mathrm{m} == 1$, then $c[x]$ is the exponential step function of the form $\mathbb{e}^{\alpha_1 x}$ for $x \geq 0$ and zero otherwise. The left-hand portion of Figure 4.1 shows a plot of this function for $\alpha_1 == 1$. If $\mathrm{m} == 2$, two cases are possible: If $\alpha_1 == \alpha_2$, then $c[x] = x\, \mathbb{e}^{x \alpha_1}$ if $x \geq 0$ and zero otherwise. The middle

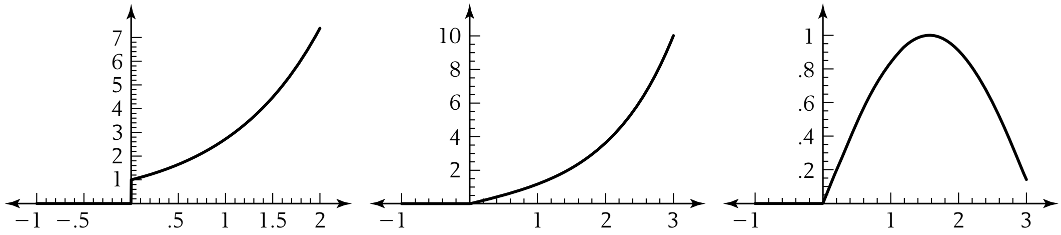


Figure 4.1 Several Green's functions for exponential splines of low order.

portion of Figure 4.1 shows a plot of this function for $\alpha_1 == \alpha_2 == 1$. If $\alpha_1 \neq \alpha_2$, then $c[x] = \frac{1}{\alpha_1 - \alpha_2}(\mathbb{e}^{\alpha_1 x} - \mathbb{e}^{\alpha_2 x})$ if $x \geq 0$ and zero otherwise. Note that $c[x]$ is a real-valued function even if the roots α_i are complex conjugates. For example, if $\alpha_1 == \mathbb{i}$ and $\alpha_2 == -\mathbb{i}$, the function $\frac{1}{\alpha_1 - \alpha_2}(\mathbb{e}^{\alpha_1 x} - \mathbb{e}^{\alpha_2 x})$ reduces to the function $\mathsf{Sin}[x]$. The right-hand portion of Figure 4.1 shows a plot of this function.

At this point, we can construct an explicit representation of the basis functions for exponential B-splines in terms of the Green's functions $c[x]$. If $\mathsf{p}_0[\mathsf{x}]$ is initialized to be 1, the coefficients of the resulting generating functions $\mathsf{p}_k[\mathsf{x}]$ converge to a basis function $n_0[x]$ defined on the integer grid $\mathbb{Z}$. Via equation 4.18, these solutions satisfy the finite difference equation $\mathsf{d}_k[\mathsf{x}]\mathsf{p}_k[\mathsf{x}] == 2^k \mathsf{d}_0[\mathsf{x}^{2^k}]$. Now, observe that the finite difference equation for the Green's function $c[x]$ (i.e., analogous to equation 4.22) is $\mathsf{d}_k[\mathsf{x}]\mathsf{p}_k[\mathsf{x}] == 2^k$. Therefore, the basis function $n_0[x]$ can be expressed as a linear combination of translated Green's functions in the form

$$n_0[x] = \sum_{\mathsf{i}=0}^{\mathsf{m}} \mathsf{d}_0[\![\mathsf{i}]\!]\, c[x - \mathsf{i}].$$

Figure 4.2 shows plots of the scaling functions $n_0[x]$ corresponding to the Green's functions of Figure 4.1. The rightmost plot is the exponential B-spline basis function of order one corresponding to $\alpha_1 == 1$, the middle plot corresponds to an exponential basis function of order two for $\alpha_1 == 1$ and $\alpha_2 == 1$, and the leftmost plot corresponds to a similar basis function of order two for $\alpha_1 == \mathbb{i}$ and $\alpha_2 == -\mathbb{i}$. Note that this last basis function is real valued even though the roots α_1 and α_2 are complex conjugates.

One potential problem with this construction is that the Green's function as defined in equation 4.22 is not unique. Adding any exponential solution $p[x]$ of equation 4.17 to $c[x]$ yields another solution $c[x] + p[x]$ of equation 4.22. However, observe that $\sum_{\mathsf{i}=0}^{\mathsf{m}} \mathsf{d}_0[\![\mathsf{i}]\!]\, p[x - \mathsf{i}]$ is identically zero due to the definition of the

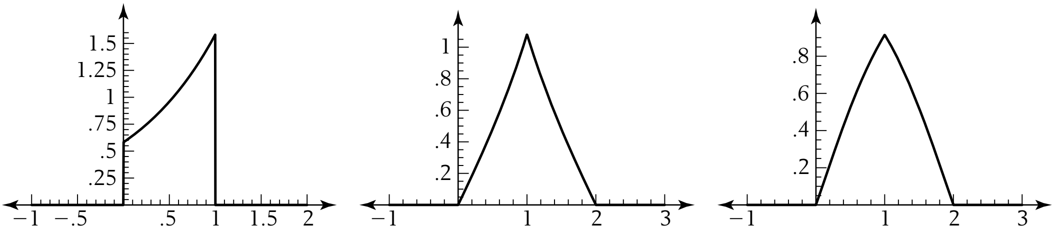


Figure 4.2 Several basis functions for exponential B-splines of low order.

difference mask $d_0[x]$. In other words, exponential solutions of equation 4.17 are annihilated by the difference mask $d_0[x]$. Therefore, this construction for $n_0[x]$ yields a unique scaling function independent of the particular Green's function used.

More generally, we can start the subdivision process on the grid $\frac{1}{2^k}\mathbb{Z}$ by initializing $p_k[x]$ to be the generating function 1. The resulting solutions $p_{k+j}[x]$ satisfy the finite difference equation

$$d_{k+j}[x]p_{k+j}[x] == 2^{k+j}\left(2^{-k}d_k\left[x^{2^j}\right]\right) \tag{4.23}$$

for all $j > 0$. Because the finite difference equation $d_{k+j}[x]p_{k+j}[x] == 2^{k+j}$ models the Green's function $c[x]$, the coefficients of the solution $p_{k+j}[x]$ to equation 4.23, plotted on the grid $\frac{1}{2^{k+j}}\mathbb{Z}$, converge to a basis function $n_k[x]$ of the form

$$n_k[x] == 2^{-k}\sum_{i=0}^{m} d_k[\![i]\!]\, c\left[x-\frac{i}{2^k}\right]. \tag{4.24}$$

(Note that this basis function is dilated so as to be supported on the interval $[0, \frac{m}{2^k}]$.) Given this definition of $n_k[x]$, the reader may verify that these basis functions satisfy the refinement relation defined by the subdivision mask $s_{k-1}[x]$; that is,

$$n_{k-1}[x] == \sum_{i=0}^{m} s_{k-1}[\![i]\!]\, n_k\left[x-\frac{i}{2}\right].$$

Just as in the polynomial case, exponential B-splines also have an equivalent definition in terms of an integral recurrence based on $\mathcal{I}[x,\alpha]$ that is analogous to equation 4.4. This recurrence can also be used to compute an analytic representation for the resulting exponential B-splines. The associated implementation contains more details on this recurrence and an implementation in *Mathematica* of this recurrence (♔).

4.4 A Smooth Subdivision Scheme with Circular Precision

The subdivision scheme for exponential splines has a nonstationary mask $s_{k-1}[x]$ (i.e., the entries in the mask depend on k). At first glance, it might seem that there is no reason to prefer this nonstationary scheme over the stationary scheme for polynomial splines. However, there are good reasons to prefer nonstationary schemes in some applications. For example, consider the problem of constructing a subdivision scheme whose associated limit curves $p[x]$ include circles. A circle is a fundamental geometric shape that arises in a variety of modeling operations, such as offsetting and blending. Due to the fact that circles do not possess a polynomial parameterization, most modeling approaches based on subdivision either approximate a circle to some given tolerance or use a rational parameterization for the circle. (See Farin [60] for more details.)

Unfortunately, any rational parameterization for a circle is nonuniform. For example, the rational parameterization $\{\frac{1-x^2}{1+x^2}, \frac{2x}{1+x^2}\}$ traces out the first quadrant of the unit circle as x varies from 0 to 1. On the other hand, as x varies from 1 to ∞, the parameterization traces out the second quadrant of the circle. Instead, we desire a subdivision scheme capable of representing circles in their arc-length parameterization $\{\mathrm{Cos}[x], \mathrm{Sin}[x]\}$. Now, as x varies from 0 to 2π, the parameterization traces out the entire circle uniformly. Luckily, exponential B-splines are capable of reproducing such trigonometric functions.

In the two sections that follow, we consider two related instances of exponential splines. The first instance, splines in tension, is a standard generalization of polynomial splines that allows the introduction of a tension parameter. The second instance considers a scheme that converges to a mixture of polynomial and trigonometric functions. This scheme is smooth and capable of reproducing circles. The final section unifies these two schemes and cubic B-splines as a single nonstationary scheme. This scheme is the basis for a surface scheme, discussed in Chapter 7, that can generate surfaces of revolution.

4.4.1 Splines in Tension

Our first example of an exponential spline is splines in tension. These splines are a standard generalization of polynomial splines that allow the introduction of a tension parameter into the scheme. Varying the tension parameter results in a curve $p[x]$ that follows the initial polygon p_0 with varying degrees of "tightness." Splines in tension have been the subject of numerous papers [23, 41, 138]. The subdivision mask for splines in tension can be derived directly from their defining differential equation. In particular, segments of splines in tension satisfy the differential equation

$$p^{(4)}[x] - \gamma^2 p^{(2)}[x] == 0,$$

where γ is a real constant. (The constant γ^2 serves as the "tension" parameter in most constructions for splines in tension.) To construct the subdivision mask $s_{k-1}[x]$ for splines in tension, we first compute the roots of the polynomial equation $x^4 - \gamma^2 x^2 == 0$. Because these roots have values $\{0, 0, -\gamma, \gamma\}$, the segments of a spline in tension are linear combinations of 1, x, $e^{-\gamma x}$, and $e^{\gamma x}$. Next, recalling the definition of the hyperbolic sine, $\text{Sinh}[x] = \frac{1}{2}(e^x - e^{-x})$, and of the hyperbolic cosine, $\text{Cosh}[x] = \frac{1}{2}(e^{-x} + e^x)$, we observe that the segments of $p[x]$ are linear combinations of 1, x, $\text{Sinh}[\gamma x]$, and $\text{Cosh}[\gamma x]$. Plugging these values for the roots into equation 4.21 and simplifying the resulting expression using these hyperbolic identities yields

$$\begin{aligned} s_{k-1}[x] &= \frac{1}{2}(1+x)^2\left(\frac{1 + \left(e^{\gamma 2^{-k}} + e^{-\gamma 2^{-k}}\right)x + x^2}{e^{-\gamma 2^{-k}} + 2 + e^{\gamma 2^{-k}}}\right) \\ &== \frac{1}{2}(1+x)^2\left(\frac{1 + 2\,\text{Cosh}[2^{-k}\gamma]x + x^2}{2 + 2\,\text{Cosh}[2^{-k}\gamma]}\right). \end{aligned}$$

Applying the subdivision mask $s_{k-1}[x]$ requires evaluating $\text{Cosh}[2^{-k}\gamma]$ for successive integral values of k. Although this computation is not particularly onerous, the trigonometric identity $\text{Cosh}[x] == 2\,\text{Cosh}[\frac{x}{2}]^2 - 1$ yields a pleasing recurrence for these values of the form

$$\text{Cosh}[2^{-k}\gamma] = \sqrt{\frac{1 + \text{Cosh}[2^{-(k-1)}\gamma]}{2}}.$$

Given a control polygon $p_0 = \{\{1, 0\}, \{0, 1\}, \{-1, 0\}, \{0, -1\}\}$, the three curves on the right-hand side of Figure 4.3 are examples of splines in tension for this initial

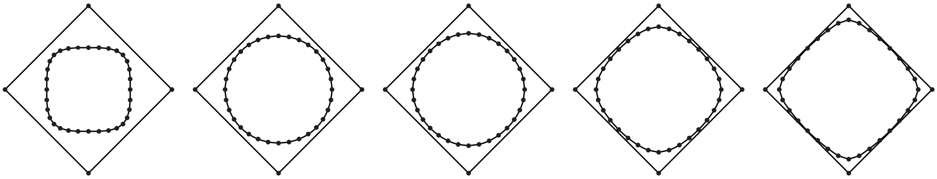


Figure 4.3 A family of curves approximating an initial diamond shape.

diamond. (The two curves on the left are discussed in the next section.) The middle curve has tension $\gamma == 0$ and is a cubic B-spline curve. The next two curves to the right are splines in tension with $\gamma = \mathsf{Cosh}^{-1}[10]$ and $\gamma = \mathsf{Cosh}^{-1}[100]$, respectively. By construction, all of these curves are C^2 continuous. Note that as γ increases the resulting splines bend more sharply and behave increasingly like linear B-splines.

4.4.2 Mixed Trigonometric Splines

In our first example, we constructed a subdivision scheme whose limit curves consist of polynomial and hyperbolic segments. In this example, we construct a scheme whose segments are combinations of polynomials and trigonometric functions. These "mixed" trigonometric curves, capable of reproducing circles, have been studied by several authors. For example, Dyn and Levin [53] consider a scheme of order three whose segments are combinations of 1, $\mathsf{Sin}[x]$, and $\mathsf{Cos}[x]$. Zhang [168] considers a scheme of order four that generates limit curves, which he refers to as "C curves." Pottmann and Wagner [119, 154] consider a similar space of splines, which they refer to as "helix" splines. In both cases, the segments of the spline are solutions to the differential equation

$$p^{(4)}[x] + \gamma^2 p^{(2)}[x] == 0,$$

where γ is a real constant. Because the roots of the polynomial $x^4 + \gamma^2 x^2 == 0$ have the form $\{0, 0, -\gamma\mathbb{i}, \gamma\mathbb{i}\}$, the solutions to this differential equation are linear combinations of 1, x, $\mathbb{e}^{\gamma \mathbb{i} x}$, and $\mathbb{e}^{-\gamma \mathbb{i} x}$. Due to the identities $\mathbb{i}\,\mathsf{Sin}[x] == \frac{1}{2}(\mathbb{e}^{\mathbb{i}x} - \mathbb{e}^{-\mathbb{i}x})$ and $\mathsf{Cos}[x] == \frac{1}{2}(\mathbb{e}^{-\mathbb{i}x} + \mathbb{e}^{\mathbb{i}x})$, the solutions can also be expressed as linear combinations of the functions 1, x, $\mathsf{Sin}[\gamma x]$, and $\mathsf{Cos}[\gamma x]$.

Recall that the discrete difference masks $\mathsf{d_k[x]}$ associated with a differential equation were constructed so as to annihilate samples of solutions to this equation on the grid $\frac{1}{2^{\mathsf{k}}}\mathbb{Z}$. Because Sin and Cos are periodic, the constant γ is typically restricted to lie in the range $0 < \gamma \leq \pi$ to ensure that these difference masks are unique. Plugging the roots $\{0, 0, -\gamma\mathbb{i}, \gamma\mathbb{i}\}$ into equation 4.21 and simplifying the resulting expression using the previously given trigonometric identities yields

$$\begin{aligned} \mathsf{s_{k-1}[x]} &= \frac{1}{2}(1+\mathsf{x})^2 \left(\frac{1 + \left(\mathbb{e}^{\gamma \mathbb{i} 2^{-\mathsf{k}}} + \mathbb{e}^{-\gamma \mathbb{i} 2^{-\mathsf{k}}}\right)\mathsf{x} + \mathsf{x}^2}{\mathbb{e}^{-\gamma \mathbb{i} 2^{-\mathsf{k}}} + 2 + \mathbb{e}^{\gamma \mathbb{i} 2^{-\mathsf{k}}}} \right) \\ &== \frac{1}{2}(1+\mathsf{x})^2 \left(\frac{1 + 2\,\mathsf{Cos}[2^{-\mathsf{k}}\gamma]\mathsf{x} + \mathsf{x}^2}{2 + 2\,\mathsf{Cos}[2^{-\mathsf{k}}\gamma]} \right). \end{aligned}$$

Applying the subdivision mask $\mathsf{s}_{k-1}[x]$ requires evaluating $\mathsf{Cos}[2^{-k}\gamma]$ for successive values of k. As in the previous case, the trigonometric identity $\mathsf{Cos}[x] == 2\,\mathsf{Cos}[\frac{x}{2}]^2 - 1$ yields a pleasing recurrence for these values of the form

$$\mathsf{Cos}[2^{-k}\gamma] = \sqrt{\frac{1 + \mathsf{Cos}[2^{-(k-1)}\gamma]}{2}}.$$

The restriction $0 < \gamma \leq \pi$ ensures that $\mathsf{Cos}[2^{-k}\gamma] \geq 0$ for all $k > 0$ and causes the positive square root to always be used in this equation.

The three curves on the left-hand side of Figure 4.3 are examples of mixed trigonometric splines for an initial diamond p_0. The middle curve has tension $\gamma == 0$ and is a cubic B-spline curve. The next two curves to the left are mixed trigonometric splines with $\gamma = \frac{\pi}{2}$ and $\gamma = \pi$, respectively. By construction, all of these curves are C^2 continuous. Note that as γ increases the resulting splines are more rounded, the opposite of the behavior for splines in tension.

In this example, the trigonometric spline curve corresponding to $\gamma = \frac{\pi}{2}$ (second from the left in Figure 4.3) is exactly a circle. Setting $\gamma = \frac{\pi}{2}$ allows the spline to exactly reproduce the trigonometric functions $\{\mathsf{Cos}[\frac{\pi x}{2}], \mathsf{Sin}[\frac{\pi x}{2}]\}$ on the initial grid $\mathbb{Z}$. If x is treated as being periodic on the interval $[0, 4]$, this parameterization traces out a unit circle as x varies over this interval. Figure 4.4 shows three rounds of subdivision for this scheme starting from the initial diamond. Note that each successive approximation is a regular polygon.

Taking a tensor product version of this univariate subdivision scheme allows the construction of exact surfaces of revolution via subdivision. Figure 4.5 shows examples of an ellipsoid and an elliptical torus created using this scheme. In both examples, a diamond-shaped polygon has been revolved around the vertical axis to form a control polyhedron for a closed tensor product surface (the polyhedra on the left). In the first example, the limit surface is a double covering of the ellipsoid. Note that whereas the cross sections of the initial polyhedra with respect to the

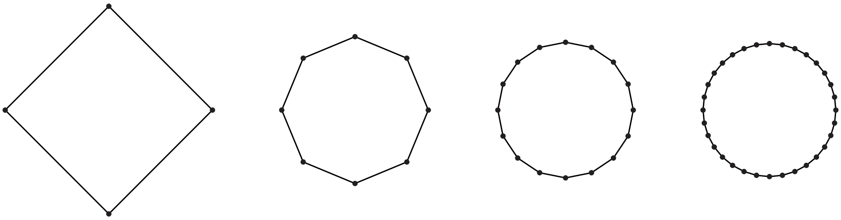


Figure 4.4 Three rounds of subdivision for a mixed trigonometric spline converging to a circle.

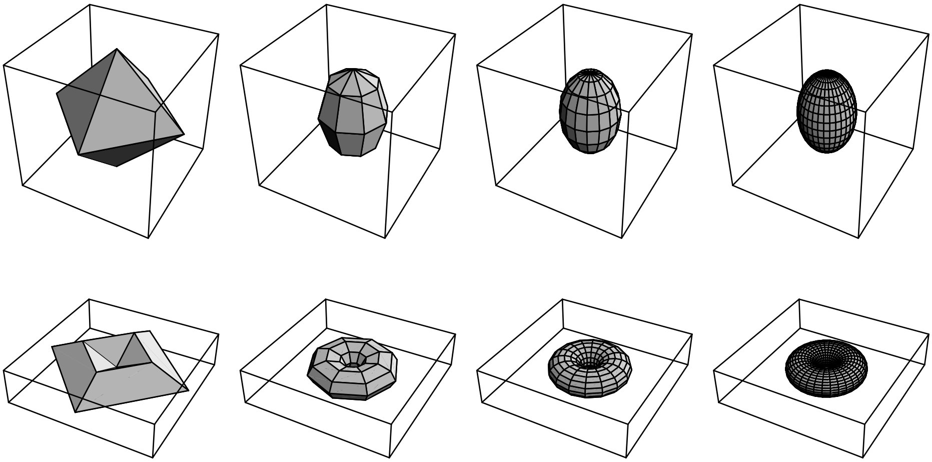


Figure 4.5 Three rounds of subdivision for two tensor product trigonometric splines.

axis of revolution are square diamonds, the cross sections of the limit surfaces taken along the axis of revolution are not circles, but ellipses. To generate a surface of revolution with a circular cross section (e.g., a sphere), the reader should consult section 7.2.3.

4.4.3 The Unified Subdivision Scheme

At this point, we have considered three examples of exponential splines of order four: cubic B-splines, splines in tension, and mixed trigonometric splines. These splines share enough structure that they can be viewed as instances of a single spline scheme of order four whose segments satisfy the differential equation $p^{(4)}[x] \pm \gamma^2 p^{(2)}[x] == 0$. Depending on the sign and the value of $\pm\gamma^2$, this differential equation captures the three schemes as special cases. Remarkably, the subdivision masks for these three schemes can be expressed as a single mask s_{k-1} involving a "tension" parameter σ_k. This mask depends on the tension σ_k in the following manner:

$$s_{k-1} = \frac{1}{4 + 4\sigma_k}(1, 2 + 2\sigma_k, 2 + 4\sigma_k, 2 + 2\sigma_k, 1). \tag{4.25}$$

For each of the three schemes, the choice of the tension σ_k depends on the sign and value of $\pm\gamma^2$. These cases are summarized as follows:

- For $\gamma == 0$ (i.e., the polynomial case), let $\sigma_k == 1$.

- For $+\gamma^2$ (i.e., the exponential case), let $\sigma_k = \text{Cosh}[2^{-k}\gamma]$. Note that in this case $\sigma_k > 1$ for all k.
- For $-\gamma^2$ (i.e., the trigonometric case), $\sigma_k = \text{Cos}[2^{-k}\gamma]$. The restriction $0 \le \gamma \le \pi$ causes σ_k to lie in the range $-1 \le \sigma_k < 1$.

In all three cases, the reader may verify by inspection that equation 4.25 reproduces the appropriate subdivision mask. At first glance, this separation of the values of σ_k into three cases seems to provide little help. However, the following theorem makes these three cases unnecessary once the initial tension σ_0 has been chosen.

THEOREM 4.4 If $\sigma_0 \ge -1$, for all three of the previously stated cases the tensions σ_{k-1} and σ_k satisfy the recurrence

$$\sigma_k = \sqrt{\frac{1+\sigma_{k-1}}{2}}. \tag{4.26}$$

Proof In the polynomial case, equation 4.26 follows from the fact that $1 == \sqrt{\frac{1+1}{2}}$. In the exponential case, equation 4.26 follows from the identity $\text{Cosh}[x]^2 == \frac{1+\text{Cosh}[2x]}{2}$. In the trigonometric case, equation 4.26 follows from the identity $\text{Cos}[x]^2 == \frac{1+\text{Cos}[2x]}{2}$ and the fact that $0 < \gamma \le \pi$.

Given the tension σ_{k-1}, the subdivision mask s_{k-1} is derived by first computing σ_k using equation 4.26 and by then substituting σ_k into equation 4.25. (Observe that the mask s_{k-1} uses the tension parameter σ_k, not σ_{k-1}.) This nonstationary subdivision scheme combines the three previous schemes in a very elegant manner. Instead of choosing γ, the user simply chooses an initial "tension" σ_0. In all three cases, the resulting limit curve is a $\mathcal{C}^2$ spline.

If the initial tension is $\sigma_0 = 1$, then $\sigma_k == 1$ for all k and the subdivision scheme is exactly the cubic B-spline subdivision algorithm of Lane and Riesenfeld. If $\sigma_0 > 1$, the scheme converges to a spline in tension. If $-1 \le \sigma_0 < 1$, the scheme converges to a "mixed" trigonometric spline. Figure 4.3 illustrates the effect of varying initial tensions σ_0 on the resulting splines. In particular, the figure shows a diamond-shaped polygon and the corresponding splines for $\sigma_0 = \{-1, 0, 1, 10, 100\}$. Varying σ_0 controls the distribution of the curvature of the spline. Using large values for σ_0 causes most of the curvature to be concentrated near the vertices of the control polygon (i.e., the spline "clings" to the control polygon). Using small values tends to distribute more of the curvature of the spline near edges of the control polygon. Finally, if the initial polygon p_0 is chosen to be a regular n-gon, choosing the initial tension σ_0 to be $\text{Cos}[\frac{2\pi}{n}]$ causes the subdivision scheme to converge to a circle.

4.4.4 Convergence Analysis for Nonstationary Schemes

The convergence analysis of Chapter 3 considered stationary subdivision schemes, that is, those schemes in which the mask used at each level of subdivision remained unchanged. However, the subdivision schemes for exponential B-splines in section 4.3.2 had the property that the subdivision masks $s_{k-1}[x]$ and their corresponding matrices S_{k-1} vary as a function of k. Given an initial control polygon p_0, this scheme constructs a sequence of new polygons p_k related by $p_k = S_{k-1} p_{k-1}$. By construction, the limit functions $p_k[x]$ associated with this scheme are $\mathcal{C}^2$. In this section, we consider the problem of determining the smoothness of the limit function $p_\infty[x]$ directly from the subdivision matrices S_{k-1}.

One approach would be to follow the general method of Chapter 3 and construct a subdivision scheme for various differences associated with S_{k-1}. The subdivision matrices T_{k-1} for these schemes should satisfy $D_k S_{k-1} = T_{k-1} D_{k-1}$, where D_k is an appropriate difference matrix. However, there are two difficulties with this approach. First, we still have to analyze the nonstationary subdivision matrices T_{k-1} for the difference schemes. Second, for many types of nonstationary schemes, these matrices T_{k-1} may not exist, because the subdivision scheme associated with S_{k-1} is not necessarily affinely invariant. For example, exponential B-splines are affinely invariant (i.e., have subdivision matrices whose rows sum to one) if and only if one of the roots α_i in equation 4.19 is zero.

In this section, we describe a different method, credited to Dyn and Levin in [54], for proving that nonstationary schemes converge to $\mathcal{C}^m$ limit functions. The key to this method is to observe that the subdivision matrices S_{k-1} for most nonstationary schemes converge to a limit matrix S_∞. If this convergence is sufficiently fast and the subdivision scheme for the matrix S_∞ produces $\mathcal{C}^m$ limit functions, the nonstationary scheme associated with the matrices S_{k-1} also produces $\mathcal{C}^m$ limit functions. The following, weaker version of Dyn and Levin's theorem bounds the rate at which the matrix S_{k-1} must converge to the limit matrix S_∞ to guarantee that the functions $p_k[x]$ defined by the nonstationary scheme converge to a limit function $p_\infty[x]$ that is continuous.

THEOREM 4.5 Consider a sequence of subdivision matrices S_{k-1} converging to a subdivision matrix S_∞ whose rows are non-negative and sum to one. If this convergence occurs at the rate

$$\|S_{k-1} - S_\infty\| < \beta\alpha^{k-1},$$

where $\beta > 0$ and $0 < \alpha < 1$, and the stationary scheme associated with S_∞ is uniformly convergent, the nonstationary scheme associated with the matrices S_{k-1} converges uniformly for all initial vectors p_0 with bounded norm.

Proof Because the scheme associated with S_∞ is convergent, let $n[x]$ be its associated scaling function. Given a sequence of vectors p_k defined by the nonstationary process $\mathsf{p}_k = \mathsf{S}_{k-1}\,\mathsf{p}_{k-1}$, our goal is to show that the successive functions $p_k[x]$ of the form

$$p_k[x] = \sum_i \mathsf{p}_k[\![i]\!]\, n[2^k x - i]$$

are converging uniformly. To prove this convergence, we examine the difference of successive functions $p_k[x] - p_{k-1}[x]$. These differences can be expressed as a linear combination of the coefficients of $(\mathsf{p}_k - \mathsf{S}_\infty \mathsf{p}_{k-1})$ multiplied by translates of the scaling function $n[2^k x]$. Because translates of the scaling function $n[2^k x]$ are non-negative and sum to one (due to the rows of S_∞ being non-negative and summing to one), the following relation holds:

$$\begin{aligned} \|p_k[x] - p_{k-1}[x]\| &\le \|\mathsf{p}_k - \mathsf{S}_\infty \mathsf{p}_{k-1}\| \\ &\le \|(\mathsf{S}_{k-1} - \mathsf{S}_\infty)\mathsf{p}_{k-1}\| \\ &\le \|\mathsf{S}_{k-1} - \mathsf{S}_\infty\| * \|\mathsf{p}_{k-1}\|. \end{aligned} \tag{4.27}$$

Given that the norm $\|\mathsf{S}_{k-1} - \mathsf{S}_\infty\|$ decays geometrically in k, we next bound the norm $\|\mathsf{p}_{k-1}\|$ in terms of the norm $\|\mathsf{p}_0\|$ to complete the proof. To this end, we observe that the vectors p_k and p_{k-1} defined by the nonstationary scheme satisfy

$$\begin{aligned} \|\mathsf{p}_k\| &\le \|\mathsf{S}_\infty + (\mathsf{S}_{k-1} - \mathsf{S}_\infty)\|\,\|\mathsf{p}_{k-1}\| \\ &\le (\|\mathsf{S}_\infty\| + \|\mathsf{S}_{k-1} - \mathsf{S}_\infty\|)\|\mathsf{p}_{k-1}\|. \end{aligned}$$

Because the rows of S_∞ are non-negative and sum to one, the ∞-norm $\|\mathsf{S}_\infty\|$ is exactly one. Moreover, by hypothesis, $\|\mathsf{S}_{k-1} - \mathsf{S}_\infty\| < \beta\alpha^{k-1}$ and, therefore,

$$\begin{aligned} \|\mathsf{p}_k\| &\le (1 + \beta\alpha^{k-1})\|\mathsf{p}_{k-1}\| \\ &\le \left(\prod_{i=1}^{k} (1 + \beta\alpha^{i-1})\right)\|\mathsf{p}_0\|. \end{aligned}$$

The product $\prod_{i=1}^{k}(1+\beta\alpha^{i-1})$ can be converted into an equivalent sum by taking the logarithm of the product and then summing the logarithms of the terms constituting the product. Because $\mathrm{Log}[1+\epsilon] < \epsilon$ for all $\epsilon > 0$, the product given previously is bounded by

$$\begin{aligned}\|p_k\| &\leq e^{\sum_{i=1}^{k}\beta\alpha^{i-1}}\|p_0\| \\ &\leq e^{\frac{\beta}{1-\alpha}}\|p_0\|.\end{aligned}$$

Substituting this bound into equation 4.27 and applying the hypothesis yields

$$\|p_k[x] - p_{k-1}[x]\| \leq \beta\,\alpha^{k-1} e^{\frac{\beta}{1-\alpha}}\|p_0\|.$$

Because $0 < \alpha < 1$, Theorem 3.1 guarantees that the continuous functions $p_k[x]$ converge uniformly to a continuous limit function $p_\infty[x]$ for all bounded $\|p_0\|$.

Requiring the rows of S_∞ to be non-negative and sum to one is actually unnecessary and was assumed in order to simplify the proof of Theorem 4.5. Dyn and Levin [54] give a more general proof that removes this restriction. In the case of higher-order continuity, Dyn and Levin show that if the convergence rate of the subdivision matrices S_{k-1} to a limit matrix S_∞ satisfies $0 < \alpha < 2^{-m}$ and the stationary scheme associated with S_∞ converges to $\mathcal{C}^m$ limit functions then the nonstationary scheme associated with the matrices S_{k-1} also converges to $\mathcal{C}^m$ limit functions.

To conclude this section, we use this theorem to analyze the smoothness of the nonstationary scheme of section 4.4.3. Given an initial tension $\sigma_0 \geq -1$, the scheme uses subdivision masks of the form

$$s_{k-1}[x] = \frac{(1+x)^2(1+2\sigma_k x + x^2)}{4x^2(1+\sigma_k)},$$

where σ is updated via $\sigma_k = \sqrt{\frac{1+\sigma_{k-1}}{2}}$. Given our previous analysis based on the governing differential equation, we know that this scheme converges to a $\mathcal{C}^2$ spline-in-tension if $\sigma_0 > 1$, a $\mathcal{C}^2$ trigonometric spline if $\sigma_0 < 1$, and a $\mathcal{C}^2$ cubic B-spline if $\sigma_0 == 1$. We now prove that the scheme is $\mathcal{C}^2$ without resorting to any underlying differential equation. Our first step is to construct a subdivision mask $t_{k-1}[x]$ for the second divided differences of the form

$$t_{k-1}[x] = \frac{(1 + 2\sigma_k x + x^2)}{x(1 + \sigma_k)}.$$

If the scheme associated with $t_{k-1}[x]$ is $\mathcal{C}^0$, the original scheme $s_{k-1}[x]$ is $\mathcal{C}^2$, as observed in Theorem 3.3.

To apply Theorem 4.5, our first task is to determine the behavior of $t_{k-1}[x]$ as $k \to \infty$. As observed previously, the tensions σ_k converge to one as $k \to \infty$. Therefore, the limit mask $t_\infty[x]$ is the subdivision mask for linear subdivision $\frac{(1+x)^2}{2x}$. Because linear subdivision converges to continuous limit functions, our final task is to bound the coefficients in $t_{k-1}[x] - t_\infty[x]$. After simplification, this expression has the form

$$t_{k-1}[x] - t_\infty[x] == -\frac{(1 - x)^2(1 - \sigma_k)}{2x(1 + \sigma_k)}.$$

Because $\sigma_k \geq -1$, we need only establish the rate at which σ_k converges to one. To bound the rate of convergence, we next show that the sequence of tensions σ_k obeys the bound $0 \leq \frac{1-\sigma_k}{1-\sigma_0} \leq \frac{1}{2^k}$ for all $k > 0$. To establish this bound, it suffices to show that $0 \leq \frac{1-\sigma_k}{1-\sigma_{k-1}} \leq \frac{1}{2}$. Due to the recurrence $\sigma_k = \sqrt{\frac{1+\sigma_{k-1}}{2}}$, this ratio satisfies

$$\frac{1 - \sigma_k}{1 - \sigma_{k-1}} == \frac{1}{2 + \sqrt{2}\sqrt{1 + \sigma_{k-1}}} \leq \frac{1}{2},$$

in that $\sigma_{k-1} \geq -1$. Because linear subdivision is convergent and $t_{k-1}[x]$ is converging to $t_\infty[x]$ at the rate of $O\left[\frac{1}{2^k}\right]$, the nonstationary scheme associated with $t_{k-1}[x]$ is also $\mathcal{C}^0$, and therefore the original scheme associated with the mask $s_{k-1}[x]$ is $\mathcal{C}^2$.

CHAPTER 5

Local Approximation of Global Differential Schemes

Chapter 4 described a method of constructing a subdivision mask $s_{k-1}[x]$ from the discretization $d_k[x]$ of a differential operator. In particular, these masks satisfy the finite difference relation $d_k[x]s_k[x] = 2d_{k-1}[x^2]$. For all three examples in Chapter 4 (polynomial B-splines, exponential B-splines, and box splines), the difference mask $d_k[x]$ divides the mask $d_{k-1}[x^2]$, yielding a finite subdivision mask $s_{k-1}[x]$. In this chapter, we consider two examples of differential operators for which this division fails and for which there is no local subdivision scheme that exactly models our physical problem.

The first section considers splines that are solutions to the polyharmonic equation $\mathcal{L}[x,y]^m p[x,y] == 0$, where $\mathcal{L}[x,y] = \mathcal{D}[x]^2 + \mathcal{D}[y]^2$ is the standard Laplace operator. These polyharmonic splines possess a globally supported, but highly localized, bell-shaped basis that can be expressed as a difference of integer translates of the traditional radial basis function. This bell-shaped basis function possesses a refinement relation that itself has a globally supported, but highly localized, subdivision mask. The next section in this chapter considers three techniques for computing finite approximations to this infinite mask: Laurent series expansion, Jacobi iteration, and linear programming. Each method yields a locally supported subdivision scheme that approximates the polyharmonic splines. The final section in the chapter considers the problem of generating a subdivision scheme for simple types of linear flows. As is the case of polyharmonic splines, the subdivision mask for these flows follows directly from the discretization of the differential operators governing these flows.

5.1 Subdivision for Polyharmonic Splines

One generalization of B-splines from the univariate setting to the multivariate setting is box splines. Recalling their differential definition, box splines satisfy a partial differential equation whose structure depends on a set of direction vectors Σ. One consequence of this dependence on Σ is that the box-spline scaling functions are not radially symmetric and exhibit certain directional preferences. This section considers an alternative generalization of B-splines based on the differential operator $\mathcal{L}[x, y] = \mathcal{D}[x]^2 + \mathcal{D}[y]^2$ associated with Laplace's equation $p^{(0,2)}[x, y] + p^{(2,0)}[x, y] == 0$. Due to the radial symmetry of the operator $\mathcal{L}[x, y]$ (also known as the *Laplace operator*), the space of solutions to the partial differential equation

$$\mathcal{L}[x, y]^{\mathsf{m}}\, p[x, y] == 0 \tag{5.1}$$

is invariant under rotation. Spline surfaces that satisfy this equation (with exceptions possible at a set of points) are the *polyharmonic splines* of order m.

Polyharmonic splines model many interesting physical phenomena. For example, harmonic splines (m == 1) model thin elastic membranes, heat conduction, and electromagnetic fields. Biharmonic splines (m == 2) model the bending behavior of thin elastic plates. (See Powers [121] for more examples and details.) The properties of polyharmonic splines have been studied in a number of papers, including [46, 47, 48, 72, 162].

Our approach in constructing a subdivision scheme for polyharmonic splines follows the differential approach for polynomial splines of Chapter 4. After initially constructing the Green's function for the polyharmonic equation, we next construct a bell-shaped basis function by taking linear combinations of translates of this Green's function. This basis function is the analog of the B-spline basis function for polynomial splines. Finally, we derive the partial differential equation that governs this bell-shaped basis function and discretize it. The subdivision mask for this new basis function follows from the resulting finite difference equations.

5.1.1 The Radial Basis for Polyharmonic Splines

As our first step, we construct a Green's function $c[x, y]$ associated with the differential operator $\mathcal{L}[x, y]^{\mathsf{m}}$ for the polyharmonic equation. Recall that a Green's function $c[x, y]$ is a solution to the partial differential equation

$$\mathcal{L}[x, y]^{\mathsf{m}}\, c[x, y] == \delta[x, y]. \tag{5.2}$$

As observed previously, this equation has many solutions. Given one Green's function $c[x, y]$, any other solution to equation 5.2 has the form $c[x, y] + p[x, y]$, where

$p[x, y]$ is a solution to equation 5.1. Observe that any Green's function $c[x, y]$ satisfying equation 5.2 also satisfies a scaling relation of the form

$$c[x, y] == 4^{1-\mathsf{m}} c[2x, 2y] + p[x, y], \tag{5.3}$$

where $p[x, y]$ is a solution to equation 5.1. The key to this equation is to note that both $c[x, y]$ and $4^{1-\mathsf{m}} c[2x, 2y]$ are solutions to equation 5.2, in that the Dirac delta function $\delta[x, y]$ satisfies a scaling relation of the form $\delta[x, y] = 4\delta[2x, 2y]$. (The factor of 4 arises from the fact that the integral of $\delta[2x, 2y]$ is $\frac{1}{4}$.)

Observe that the scaling relation of equation 5.3 is very similar to the scaling relation for cone splines given in section 2.3.2. This similarity is no coincidence. In fact, the Green's functions associated with a homogeneous differential operator always satisfy a scaling relation similar to that of equation 5.3. The existence of such a scaling relation is often an early indication that constructing a subdivision scheme using the integer translates of $c[x, y]$ is possible. Analytic solutions to equation 5.2 can be found in the literature [46, 56] and have the form

$$c[x, y] = \frac{(x^2 + y^2)^{\mathsf{m}-1} \operatorname{Log}[x^2 + y^2]}{2^{2\mathsf{m}}((\mathsf{m} - 1)!)^2 \pi}.$$

Note that we have carefully normalized the denominator in our definition of $c[x, y]$ such that the integral $\int \mathcal{L}[x, y]^{\mathsf{m}} c[x, y]\, \mathrm{d}x\, \mathrm{d}y$ is exactly one. This integral can be evaluated by restricting the range of this integral to the unit disc and then converting this integral over the unit disc to an equivalent integral over the unit circle using Green's theorem. (This observation accounts for the factor of π in the denominator.)

These functions $c[x, y]$ are often referred to as the *radial basis functions* because the resulting functions have radial symmetry. Figure 5.1 plots some examples of radial basis functions for small values of m. Note the behavior of these functions

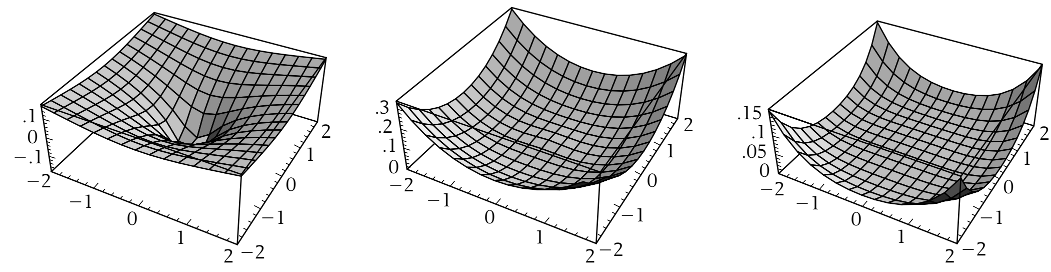


Figure 5.1 Radial basis function $c[x, y]$ for polyharmonic splines of order $\mathsf{m} = 1, 2, 3$.

at the origin. For $\mathsf{m} == 1$, $c[0,0]$ evaluates to $\frac{\mathsf{Log}[0]}{4\pi}$, which is $-\infty$. For $\mathsf{m} \geq 2$, $c[0,0]$ evaluates to a constant multiple of $0\,\mathsf{Log}[0]$, which appears to be indeterminate. However, taking the limit as $\{x, y\}$ approaches the origin yields $c[0,0] == 0$. In fact, closer analysis reveals that $c[x,y]$ has exactly $2\mathsf{m}-3$ continuous derivatives at the origin (i.e., $c[x,y]$ is a $\mathcal{C}^{2\mathsf{m}-3}$ function).

For the case of $\mathsf{m} == 2$, the radial basis function $c[x,y]$ is a smooth ($\mathcal{C}^1$) function. One important application of this particular radial basis function is in scattered data interpolation. Given a collection of n points $(x_{\mathsf{i}}, y_{\mathsf{i}})$, each with observed values v_{i}, this interpolation problem is that of constructing a function $p[x,y]$ that interpolates the data points with the observed values (i.e., $p[x_{\mathsf{i}}, y_{\mathsf{i}}] == v_{\mathsf{i}}$). Of course, many functions satisfy this interpolation problem. To uniquely determine a solution, an auxiliary constraint is to require this function $p[x,y]$ to "bend" as little as possible. One solution to this problem is to express $p[x,y]$ as a linear combination of translated radial basis functions $c[x-x_{\mathsf{i}}, y-y_{\mathsf{i}}]$ of the form

$$p[x,y] = \sum_{\mathsf{i}=1}^{\mathsf{n}} \alpha_{\mathsf{i}}\, c[x-x_{\mathsf{i}}, y-y_{\mathsf{i}}].$$

Given the n interpolation constraints $p[x_{\mathsf{i}}, y_{\mathsf{i}}] == v_{\mathsf{i}}$, the coefficients α_{i} are uniquely determined. Because the radial basis function $c[x,y]$ satisfies the biharmonic equation ($\mathsf{m} == 2$), which models a thin sheet of clamped metal, the resulting spline function $p[x,y]$ has the property that it "bends" as little as possible while interpolating the desired data points. This function $p[x,y]$ is often referred to as a *thin plate*

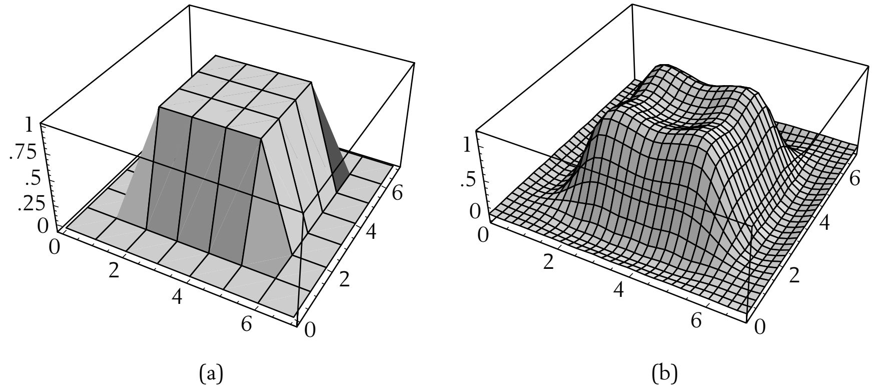


Figure 5.2 Given data points (a) interpolated by a thin plate spline (b).

spline. Figure 5.2 shows a thin plate spline that interpolates an 8 × 8 grid of data points with values of the form

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}.$$

There are many variants of this radial basis function approach to scattered data interpolation. For example, it is possible to add a polynomial term that forces the interpolation scheme to have polynomial precision. Other types of radial basis functions may also be used. Alfeld [3] and Hoschek and Lasser [76] provide a nice overview of the many approaches to this problem. However, all of these radial approaches have a major drawback: the basis functions $c[x, y]$ are highly nonlocal. The nonlocality of these bases often causes the associated systems of linear equations used in computing the α_i to be ill conditioned (see Franke [66] for details). Our approach in the next section is to develop a different basis for polyharmonic splines that is much more localized.

5.1.2 A Bell-shaped Basis for Polyharmonic Splines

Recall that in the polynomial case the B-spline basis functions could be expressed as linear combinations of integer translates of truncated powers. Likewise, the box-spline scaling functions could be expressed as a linear combination of integer translates of cone splines. Both the truncated powers and cone splines are themselves Green's functions. For polyharmonic splines, a similar construction is possible. Our approach is to construct a new basis function $n[x, y]$ as a linear combination of integer translates of the Green's function for the polyharmonic equation, the radial basis function $c[x, y]$. As before, the coefficients of this combination arise from the discretization for the differential operator $\mathcal{L}[x, y]^m$ on the integer grid $\mathbb{Z}^2$. The difference mask for $\mathcal{L}[x, y]$ on this grid is the *discrete Laplacian mask* $l[x, y] = \frac{(1-x)^2}{x} + \frac{(1-y)^2}{y}$. Note that the terms of the mask $l[x, y]$ have been centered at the origin by the division with x and y, respectively. The coefficients of this mask, plotted as a two-dimensional array, have the form

$$\begin{pmatrix}x^{-1} & 1 & x\end{pmatrix}\begin{pmatrix}0 & 1 & 0\\ 1 & -4 & 1\\ 0 & 1 & 0\end{pmatrix}\begin{pmatrix}y^{-1}\\ 1\\ y\end{pmatrix}.$$

More generally, the discretization of the differential operator $\mathcal{L}[x,y]^{m}$ on the grid $\mathbb{Z}^2$ is simply the corresponding power of the discrete Laplacian mask $l[x,y]^{m}$. For example, the coefficients of the difference mask $l[x,y]^2$ for biharmonic splines $(m == 2)$ have the form

$$\begin{pmatrix}x^{-2} & x^{-1} & 1 & x & x^{2}\end{pmatrix}\begin{pmatrix}0 & 0 & 1 & 0 & 0\\ 0 & 2 & -8 & 2 & 0\\ 1 & -8 & 20 & -8 & 1\\ 0 & 2 & -8 & 2 & 0\\ 0 & 0 & 1 & 0 & 0\end{pmatrix}\begin{pmatrix}y^{-2}\\ y^{-1}\\ 1\\ y\\ y^{2}\end{pmatrix}.$$

To discretize the operator $\mathcal{L}[x,y]^{m}$ on the grid $\frac{1}{2^k}\mathbb{Z}^2$, the difference mask $l[x,y]^{m}$ must be normalized by a power of 4^k to account for the effect of the grid spacing on the second derivative. As before, we denote this mask $(4^k l[x,y])^{m}$ by the generating function $d_k[x,y]$. Now, the coefficients of the difference mask $d_0[x,y]$ can be used to define the analog of the B-spline basis function for polyharmonic splines of order m. This analog is a function $n[x,y]$ of the form

$$n[x,y] == \sum_{i,j} d_0[\![i,j]\!]\, c[x-i, y-j], \tag{5.4}$$

where $d_0[\![i,j]\!]$ is the coefficient of $x^i y^j$ in $d_0[x,y]$.

Note that this basis function $n[x,y]$ is independent of the particular choice of Green's function. For example, adding a solution $p[x,y]$ of equation 5.1 to $c[x,y]$ yields another Green's function, $c[x,y]+p[x,y]$. However, because the coefficients of the difference mask $d_0[x]$ annihilate samples of this function on $\mathbb{Z}^2$, replacing $c[x-i,y-j]$ with $c[x-i,y-j]+p[x-i,y-j]$ in equation 5.4 yields the same basis function, $n[x,y]$.

Figure 5.3 shows plots of the basis function $n[x,y]$ for various values of m. These *bell-shaped basis functions* $n[x,y]$ for polyharmonic splines of order m have been studied in several papers, including Powell [120], Dyn et al. [47, 56], and Wahba

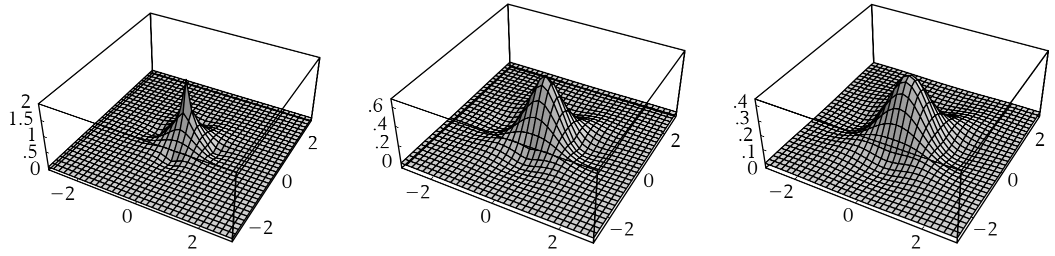


Figure 5.3 The bell-shaped basis functions for polyharmonic splines of order m = 1, 2, 3.

and Wendelberger [155]. Although not locally supported, these basis functions $n[x, y]$ exhibit a range of desirable properties. In particular:

- $n[x, y]$ is highly localized. For example, in the case of harmonic splines (m == 1), converting to polar coordinates via the transformation $\{x, y\} = \{\mathsf{r}\ \mathsf{Cos}[\alpha], \mathsf{r}\ \mathsf{Sin}[\alpha]\}$ and simplifying yields

  $$n[\mathsf{r}, \alpha] = \frac{1}{4\pi} \mathsf{Log}\left[1 + \frac{1}{\mathsf{r}^8} - \frac{2\,\mathsf{Cos}[4\alpha]}{\mathsf{r}^4}\right].$$

  Because $\mathsf{Log}[1 + \frac{1}{\mathsf{r}^4}]$ decays asymptotically at the rate $\mathsf{O}[\mathsf{r}^{-4}]$ as $\mathsf{r} \to \infty$, $n[x, y]$ also decays at the rate $\mathsf{O}[\mathsf{r}^{-4}]$ as $\mathsf{r} \to \infty$. Bell-shaped basis functions of higher orders exhibit a similar rate of decay.
- The basis function $n[x, y]$ of order m has smoothness $\mathcal{C}^{2\mathsf{m}-3}$. This fact follows directly from the smoothness of the corresponding radial basis function $c[x, y]$. Note that the harmonic (m == 1) basis function $n[x, y]$ is discontinuous (i.e., in $\mathcal{C}^{-1}$) due to the unbounded spikes at the grid points $\{\{0, 0\}, \{1, 0\}, \{0, 1\}, \{-1, 0\}, \{0, -1\}\}$. Only the central spike is positive; the remaining four spikes are negative. (These spikes in the leftmost plot of Figure 5.3 have been truncated to keep the scale of the plot in balance.)
- The sum of the integer translates of $n[x, y]$ is exactly one (i.e., the bell-shaped basis functions form a partition of unity). This fact can be proven with some effort using equation 5.3. Instead, we will give a much simpler proof in the next section, based on the structure of the subdivision mask associated with this scheme. A similar analysis leads the authors to hypothesize that this bell-shaped basis for polyharmonic splines of order m is also capable of reproducing all polynomial functions up to degree 2m − 1.

This last property is somewhat surprising because most scattered data schemes based on radial basis functions must explicitly include polynomial functions to

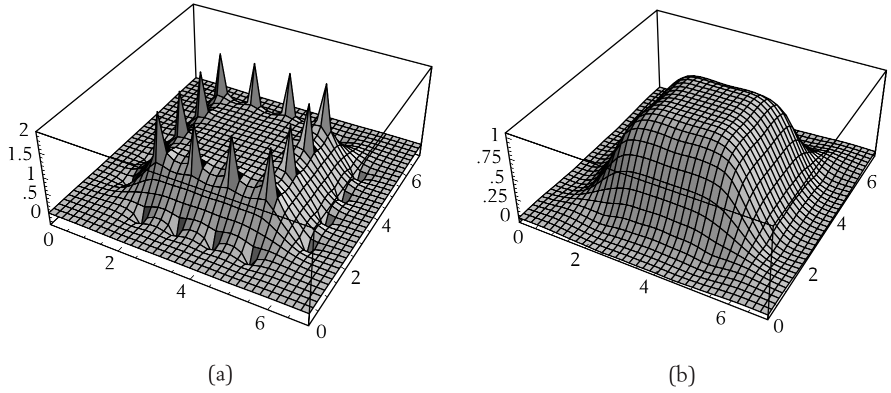


Figure 5.4 Examples of a harmonic spline (a) and a biharmonic spline (b) constructed using the bell-shaped basis.

achieve polynomial reproduction. For the bell-shaped basis, this explicit inclusion is no longer necessary; polynomials are reproduced naturally by the scheme.

Given this basis, a polyharmonic spline (in the bell-shaped basis) is a function $p[x,y]$ consisting of integer translates of the basis function $n[x,y]$; that is, $p[x,y] = \sum_{\mathrm{i},\mathrm{j}} \mathrm{p}_0\llbracket \mathrm{i}, \mathrm{j} \rrbracket n[x-\mathrm{i}, y-\mathrm{j}]$, where p_0 is a vector of control points. Figure 5.4 shows examples of the harmonic and biharmonic splines corresponding to the initial polyhedron of Figure 5.2. Observe that the upper portion of the large "bump" for both the harmonic and biharmonic splines is flat. This feature is due to the fact that the bell-shaped basis functions reproduce constants. The harmonic spline has infinite "spikes" at a subset of the integer grid $\mathbb{Z}^2$ due to the nature of the harmonic equation. (Note that these spikes cancel where the coefficients of $\mathrm{l}[\mathrm{x},\mathrm{y}]\mathrm{p}_0[\mathrm{x},\mathrm{y}]$ are zero.) Solutions to the harmonic equation model phenomena such as elastic membranes that naturally exhibit such unbounded behavior. The difference between the biharmonic spline of Figure 5.4 and the thin plate spline of Figure 5.2 is that the biharmonic spline approximates the initial control polyhedron whereas the thin plate spline interpolates the initial control polyhedron.

5.1.3 A Subdivision Scheme for Polyharmonic Splines in the Bell-shaped Basis

Given the definition of the bell-shaped basis $n[x,y]$ in terms of the radial basis function $c[x,y]$, the scaling relation of equation 5.3 clearly points to the existence of a refinement relation that relates translates of $n[x,y]$ to translates of $n[2x,2y]$. The key to deriving this refinement relation and its associated subdivision scheme is to derive the partial differential equation that governs a polyharmonic spline. Given the definition of a polyharmonic spline $p[x, y]$ in terms of the bell-shaped basis function $n[x, y]$ that in turn is defined in terms of the Green's function $c[x, y]$, the polyharmonic spline $p[x, y]$ satisfies the partial differential equation

$$\mathcal{L}[x, y]^{m} p[x, y] == \sum_{i,j} \mathrm{dp}_0[\![i, j]\!] \delta[x - i, y - j], \tag{5.5}$$

where $\mathrm{dp}_0[\![i, j]\!]$ is the coefficient of $x^i y^j$ in $d_0[x, y] p_0[x, y]$. To derive a subdivision scheme for these splines, we discretize equation 5.5 and construct a finite difference equation of the form

$$d_k[x, y] p_k[x, y] == 4^k d_0\left[x^{2^k}, y^{2^k}\right] p_0\left[x^{2^k}, y^{2^k}\right]. \tag{5.6}$$

As before, the coefficients of $p_k[x, y]$ form a discrete approximation to $p[x, y]$ on the grid $\frac{1}{2^k}\mathbb{Z}^2$. (Note that the factor of 4^k is the discretization of $\delta[x, y]$.) Just as in the univariate case, the beauty of this discretization is that successive approximations $p_{k-1}[x, y]$ and $p_k[x, y]$ are related by the subdivision relation $p_k[x, y] = s_{k-1}[x] p_{k-1}[x^2, y^2]$, where

$$s_{k-1}[x, y] = \frac{4 d_{k-1}[x^2, y^2]}{d_k[x, y]}.$$

Given this formula, we first analyze the structure of the subdivision mask $s_{k-1}[x, y]$ for harmonic splines (i.e., $m == 1$). This mask is independent of k (i.e., the various powers of four cancel out) and has the form $s[x, y] = \frac{l[x^2, y^2]}{l[x, y]}$, where $l[x, y]$ is the discrete Laplacian mask $\frac{(1-x)^2}{x} + \frac{(1-y)^2}{y}$. For polyharmonic splines of order m, the difference mask $d_k[x, y]$ has the form $(4^k l[x, y])^m$. Therefore, the subdivision mask $s_{k-1}[x, y]$ for these splines has the form

$$s_{k-1}[x, y] == 4 \frac{(4^{k-1} l[x^2, y^2])^m}{(4^k l[x, y])^m} == \frac{1}{4^{m-1}} \left(\frac{l[x^2, y^2]}{l[x, y]} \right)^m.$$

In particular, the subdivision mask for a polyharmonic spline of order m is the constant multiple of the mth power of the subdivision mask for harmonic splines. Thus, most of our focus in the subsequent analysis will be on the harmonic case (i.e., $m == 1$).

At this point, we arrive at a troubling realization. In the case of box splines, the difference mask $d[x, y]$ always divided the difference mask $d[x^2, y^2]$ exactly. As a result, the subdivision mask $s[x, y]$ for box splines was finite and defined a subdivision scheme with locally supported scaling functions. For polyharmonic splines, $l[x, y]$ does not divide $l[x^2, y^2]$. This difficulty is expected; the bell-shaped basis functions $n[x, y]$ were not locally supported. At this point, our main focus is to find a bi-infinite series expansion for $s[x, y]$ of the form $\sum_{i,j} s[\![i, j]\!] x^i y^j$. Given the correct expansion of $s[x, y]$, the resulting series coefficients $s[\![i, j]\!]$ define a refinement relation for the basis functions $n[x, y]$ and a corresponding subdivision scheme for polyharmonic splines $p[x, y]$.

5.2 Local Approximations to Polyharmonic Splines

In the previous section, we computed a globally supported bell-shaped basis for the polyharmonic splines and derived a subdivision scheme whose mask was the ratio of two difference masks (i.e., $s[x, y] = \frac{l[x^2,y^2]}{l[x,y]}$). Unfortunately, the difference mask $l[x, y]$ does not exactly divide $l[x^2, y^2]$. Given this fact, our focus in this section is on various series expansions for $\frac{l[x^2,y^2]}{l[x,y]}$. The first method computes the bi-infinite Laurent series expansion of $\frac{l[x^2,y^2]}{l[x,y]}$ whose coefficients form the exact refinement relation for the bell-shaped basis functions $n[x, y]$. Because manipulating this infinite series in practice is awkward, we next consider two other methods that produce locally supported approximations to this series. These locally supported series define subdivision schemes that provide a locally supported approximation to polyharmonic splines. The second method uses a traditional iterative technique, the Jacobi method, for computing a finite series approximation to $\frac{l[x^2,y^2]}{l[x,y]}$. The third method computes a locally supported expansion of $\frac{l[x^2,y^2]}{l[x,y]}$ using linear programming. For a series of fixed length, the coefficients produced by the latter scheme provide an optimal series expansion to $\frac{l[x^2,y^2]}{l[x,y]}$.

5.2.1 The Exact Scheme via Laurent Series

The most common type of series expansion in x^i is the *power series expansion* centered at the origin, $s[x] = \sum_{i=0}^{\infty} s[\![i]\!] x^i$. This expansion uses only non-negative powers of x and is convergent in the neighborhood of $x == 0$ for analytic functions. Unfortunately, such an expansion is not a suitable candidate for $s[x, y] = \frac{l[x^2,y^2]}{l[x,y]}$ because the resulting subdivision mask should be symmetric. However, there are series expansions of $s[x]$ that are convergent in other neighborhoods. One such expansion that allows for negative powers of x (and thus provides symmetric subdivision masks) is the *Laurent series expansion*. This expansion, again centered at the origin, has the form

$s[x] = \sum_{i=-\infty}^{\infty} s[\![i]\!]x^i$, where the coefficients $s[\![i]\!]$ are chosen such that the series expansion is convergent on an annulus containing the circle $|x| == 1$ in the complex plane.

Ahlfors [4] contains an introduction to complex analysis and Laurent series. Given a univariate function $s[x]$, the coefficients $s[\![i]\!]$ of the Laurent series expansion can be computed as complex (Cauchy) integrals of the form

$$s[\![i]\!] = \frac{1}{2\pi\mathbb{i}} \int_{|x|==1} \frac{s[x]}{x^{i+1}}\, \mathbb{d}x.$$

If $s[x]$ is a real analytic function, this integral is always real valued. For example, the rational function $\frac{x}{1-3x+x^2}$ has a Laurent series expansion $\sum_{i=-\infty}^{\infty} s[\![i]\!]x^i$ with coefficients $s[\![i]\!]$ of the form

$$\{\ldots, -0.009519, -0.02492, -0.06524, -0.17082,$$
$$-0.447214, -0.17082, -0.06524, -0.02492, -0.009519, \ldots\}.$$

Here, the values of $s[\![i]\!]$ are displayed as i ranges from -4 to 4. Analogously, the Laurent series expansion of a bivariate function $s[x, y]$ is a bi-infinite series expansion $\sum_{ij} s[\![i, j]\!]x^i y^j$ that is convergent on the Cartesian product of the unit circles $|x| == 1$ and $|y| == 1$. The coefficients $s[\![i, j]\!]$ of this expansion can be computed via the complex integral

$$s[\![i, j]\!] = \frac{-1}{4\pi^2} \int_{\substack{|x|==1 \\ |y|==1}} \frac{s[x, y]}{x^{i+1}y^{j+1}}\, \mathbb{d}x\, \mathbb{d}y. \tag{5.7}$$

For $s[x, y] = \frac{l[x^2,y^2]}{l[x,y]}$, it might appear that the integral of equation 5.7 is not convergent because the denominator $l[x, y]$ of the mask $s[x, y]$ is zero at the point $\{1, 1\}$, which lies on the range of integration. (This point is the only intersection of the algebraic curve $l[x, y] == 0$ with the circles $|x| == 1$ and $|y| == 1$ in complex space (♔).) Luckily, this pole in the denominator of $s[x, y]$ is canceled by a corresponding zero in the numerator of $s[x, y]$. In fact, the limit of the value of the mask $s[x, y]$ as $\{x, y\} \to \{1, 1\}$ is exactly 4, independent of the direction of approach. This fact follows from the observation that the value of the univariate mask $s[1 + az, 1 + bz]$ taken at $z == 0$ is 4, independent of the direction of approach $\{a, b\}$. Figure 5.5 shows a plot of the values of $s[x, y]$ in a small neighborhood of $\{1, 1\}$.

Observe that the numerator $l[x^2, y^2]$ of the mask $s[x, y]$ is zero at the three points $\{\{-1, 1\}, \{1, -1\}, \{-1, -1\}\}$. Therefore, as shown in section 3.3.3, the subdivision scheme corresponding to this mask reproduces constant functions, or, equivalently, the bell-shaped basis functions $n[x, y]$ form a partition of unity. More generally, the mask $s[x, y]$ has a zero of order $2m$ at these three points. This

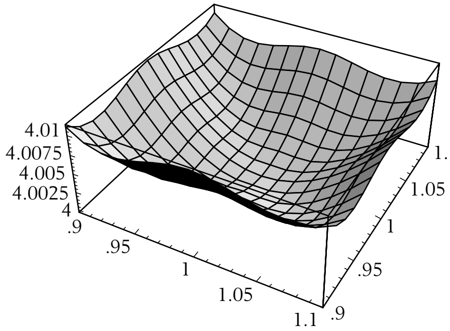


Figure 5.5 Plot of the harmonic mask s[x, y] on the range $[0.9, 1.1]^2$.

observation is consistent with the authors' hypothesis that the bell-shaped basis functions of order m can reproduce polynomial functions of up to degree 2m − 1.

Given that the integral of equation 5.7 is well defined, our next task is to develop a method for computing the coefficients s⟦i, j⟧. If we apply the transformation {x, y} = {Cos[α] + i Sin[α], Cos[β] + i Sin[β]} to the subdivision mask s[x, y], the resulting reparameterized mask s[α, β] is a real function of α and β and has the form

$$\text{s}[\alpha, \beta] == \frac{\text{Cos}[2\alpha] + \text{Cos}[2\beta] - 2}{\text{Cos}[\alpha] + \text{Cos}[\beta] - 2}.$$

Using this reparameterization, the complex integral of equation 5.7 can be reduced to a real integral of the form

$$\text{s}[\![\text{i}, \text{j}]\!] = \int_0^{2\pi}\int_0^{2\pi} \text{Cos}[\text{i}\alpha + \text{j}\beta]\left(\frac{\text{Cos}[2\alpha] + \text{Cos}[2\beta] - 2}{\text{Cos}[\alpha] + \text{Cos}[\beta] - 2}\right) d\alpha\, d\beta.$$

(Note that we have deleted the imaginary part of the integrand in this expression because its integral is always zero.) Unfortunately, this integral appears to be extremely difficult to evaluate symbolically when treating i and j as free parameters. However, the values s⟦i, j⟧ can be computed using numerical integration for various values of i and j. Figure 5.6 shows a table of the coefficients s⟦i, j⟧ for i and j ranging from −4 to 4, rounded to four digits.

Because working with this entire infinite expansion is impractical, we instead consider subdivision masks that are finite truncations of the infinite series s[x, y]. This approach seems reasonable because the coefficients s⟦i, j⟧ are converging to zero as i, j → ∞. If the size of the truncated mask is large enough, the difference between the truncated scheme and the infinite scheme is visually indiscernible. Figure 5.7 shows an example of the truncated subdivision mask of Figure 5.6

0.0016	0.0023	0.0019	−0.0022	−0.0106	−0.0022	0.0019	0.0023	0.0016
0.0023	0.0048	0.0073	0.0015	−0.0338	0.0015	0.0073	0.0048	0.0023
0.0019	0.0073	0.021	0.0347	−0.1277	0.0347	0.021	0.0073	0.0019
−0.0022	0.0015	0.0347	0.2441	0.4535	0.2441	0.0347	0.0015	−0.0022
−0.0106	−0.0338	−0.1277	0.4535	1.4535	0.4535	−0.1277	−0.0338	−0.0106
−0.0022	0.0015	0.0347	0.2441	0.4535	0.2441	0.0347	0.0015	−0.0022
0.0019	0.0073	0.021	0.0347	−0.1277	0.0347	0.021	0.0073	0.0019
0.0023	0.0048	0.0073	0.0015	−0.0338	0.0015	0.0073	0.0048	0.0023
0.0016	0.0023	0.0019	−0.0022	−0.0106	−0.0022	0.0019	0.0023	0.0016

Figure 5.6 Coefficients s⟦i, j⟧ of the exact subdivision mask for harmonic splines (to four digits).

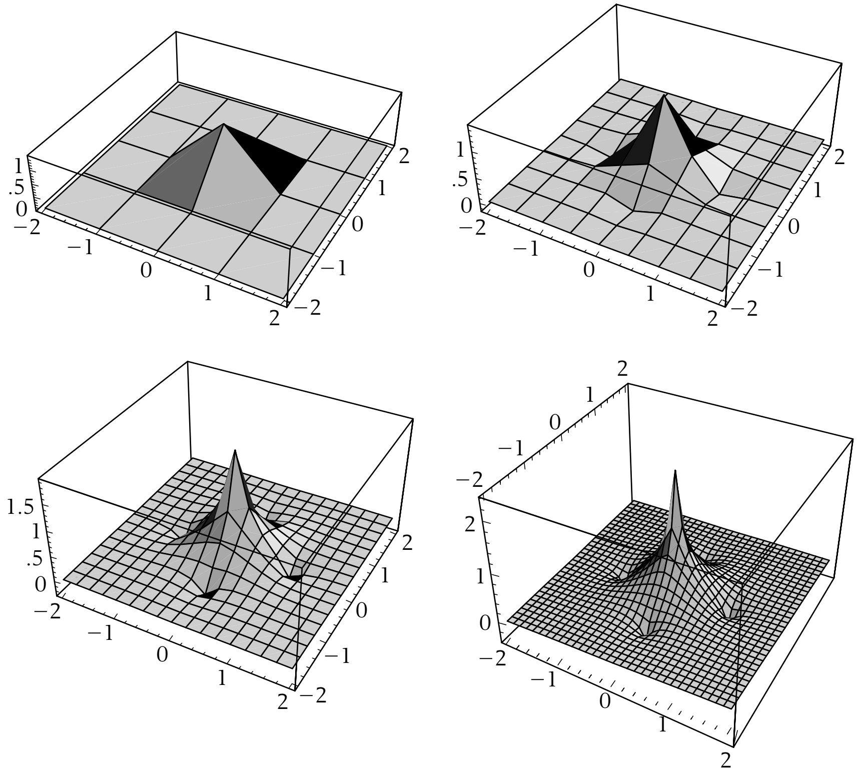


Figure 5.7 A basis function for harmonic splines constructed via subdivision based on the truncated Laurent series.

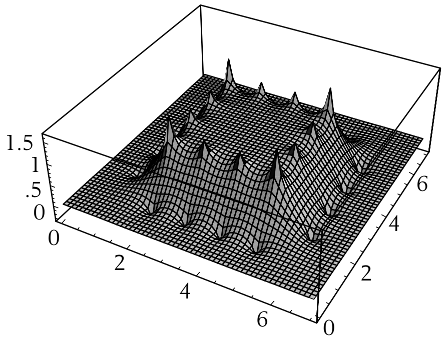


Figure 5.8 The harmonic spline of Figure 5.4 approximated by three rounds of subdivision using the truncated Laurent series.

applied to a standard unit vector. The result is an approximation of the harmonic basis function $n[x, y]$. Figure 5.8 shows three rounds of subdivision for the initial polyhedron of Figure 5.2. Note that the resulting polyhedron is a close approximation of the harmonic spline of Figure 5.4, with the exception that spikes of the subdivided version are slowly diverging toward the unbounded spikes of the limit solution.

Unfortunately, the truncated Laurent series has the drawback that the resulting scheme may not reproduce constant functions. Another defect is that the error induced by truncation is concentrated along the boundary of the mask where the truncation takes place. Ideally, this error should be distributed over the entire mask. In the rest of this section, we consider two alternative methods for constructing locally supported masks that approximate bi-infinite Laurent series.

5.2.2 Local Approximations via the Jacobi Iteration

We next consider a simple method for approximating the exact subdivision mask $\mathrm{s}[\mathrm{x}, \mathrm{y}] = \frac{\mathrm{l}[\mathrm{x}^2, \mathrm{y}^2]}{\mathrm{l}[\mathrm{x}, \mathrm{y}]}$ based on using an iterative method to solve the linear system

$$\mathrm{l}[\mathrm{x}, \mathrm{y}]\,\mathrm{s}[\mathrm{x}, \mathrm{y}] == \mathrm{l}[\mathrm{x}^2, \mathrm{y}^2]. \tag{5.8}$$

The main advantage of this approach is that iterative methods are simple to implement and have been extensively studied. Although many iterative methods (such as the Gauss-Seidel method, the Gauss-Seidel method with successive overrelaxation, the RF method, and the Peaceman-Rachford method) are available for this task, we focus on the *Jacobi iteration* due to its simplicity. For a comprehensive description and analysis of all of these methods, the authors suggest Varga [152] and Young [166]. Other excellent references on iterative solution methods for linear systems include Golub and Van Loan [67], Horn and Johnson [75], and Kelly [79].

Given an initial mask $s^0[x, y]$, the Jacobi iteration constructs a sequence of masks $s^i[x, y]$. For harmonic splines (i.e., masks satisfying equation 5.8), the Jacobi iteration has the form

$$s^i[x, y] = (\alpha * l[x, y] + 1)\, s^{i-1}[x, y] - \alpha * l[x^2, y^2]. \tag{5.9}$$

The mask $(\alpha * l[x, y]+1)$ is the *smoothing mask* associated with the iteration, whereas the mask $\alpha * l[x^2, y^2]$ is the *correction mask* associated with the iteration. For $0 \le \alpha \le \frac{1}{4}$, the smoothing mask takes convex combinations of coefficients of the mask $s^i[x, y]$. For example, if $\alpha = \frac{1}{6}$, the smoothing mask and correction mask have, respectively, the form

$$\begin{pmatrix} 0 & \frac{1}{6} & 0 \\ \frac{1}{6} & \frac{1}{3} & \frac{1}{6} \\ 0 & \frac{1}{6} & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & \frac{1}{6} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \frac{1}{6} & 0 & -\frac{2}{3} & 0 & \frac{1}{6} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{6} & 0 & 0 \end{pmatrix}.$$

Observe that the iteration of equation 5.9 has the property that any solution $s[x, y]$ to equation 5.8 is also a fixed point of the iteration.

The beauty of this iteration is that the masks $s^i[x, y]$ converge to the exact Laurent series $s[x, y]$ for a small range of α around $\alpha == \frac{1}{6}$. This constant α is the *relaxation factor* for the iteration and controls the rate of convergence of this iteration. (See Young [166] for more details on the convergence of iterative methods.) If we choose an initial mask $s^0[x, y]$ to be the mask for bilinear subdivision $\frac{(1+x)^2}{2x}\frac{(1+y)^2}{2y}$, Figure 5.9 shows the 9×9 mask resulting from three rounds of the Jacobi iteration.

Figure 5.10 shows an approximation to the basis function $n[x, y]$ based on three rounds of subdivision, with each round of subdivision using three iterations of equation 5.9. Note that although the subdivision mask of Figure 5.9 is not a particularly good approximation of the exact mask $s[x, y]$ the resulting approximation of $n[x, y]$ is visually realistic and difficult to distinguish from Figure 5.7. Figure 5.11 shows three rounds of subdivision applied to the initial polyhedron of Figure 5.2. Again, this figure and Figure 5.8 are nearly indistinguishable.

0.	0.	0.	0.0012	−0.0023	0.0012	0.	0.	0.
0.	0.	0.0035	0.0069	−0.0208	0.0069	0.0035	0.	0.
0.	0.0035	0.0185	0.0382	−0.1204	0.0382	0.0185	0.0035	0.
0.0012	0.0069	0.0382	0.2361	0.4352	0.2361	0.0382	0.0069	0.0012
−0.0023	−0.0208	−0.1204	0.4352	1.4167	0.4352	−0.1204	−0.0208	−0.0023
0.0012	0.0069	0.0382	0.2361	0.4352	0.2361	0.0382	0.0069	0.0012
0.	0.0035	0.0185	0.0382	−0.1204	0.0382	0.0185	0.0035	0.
0.	0.	0.0035	0.0069	−0.0208	0.0069	0.0035	0.	0.
0.	0.	0.	0.0012	−0.0023	0.0012	0.	0.	0.

Figure 5.9 Coefficients of the mask $s^3[x, y]$ formed by three rounds of Jacobi iteration.

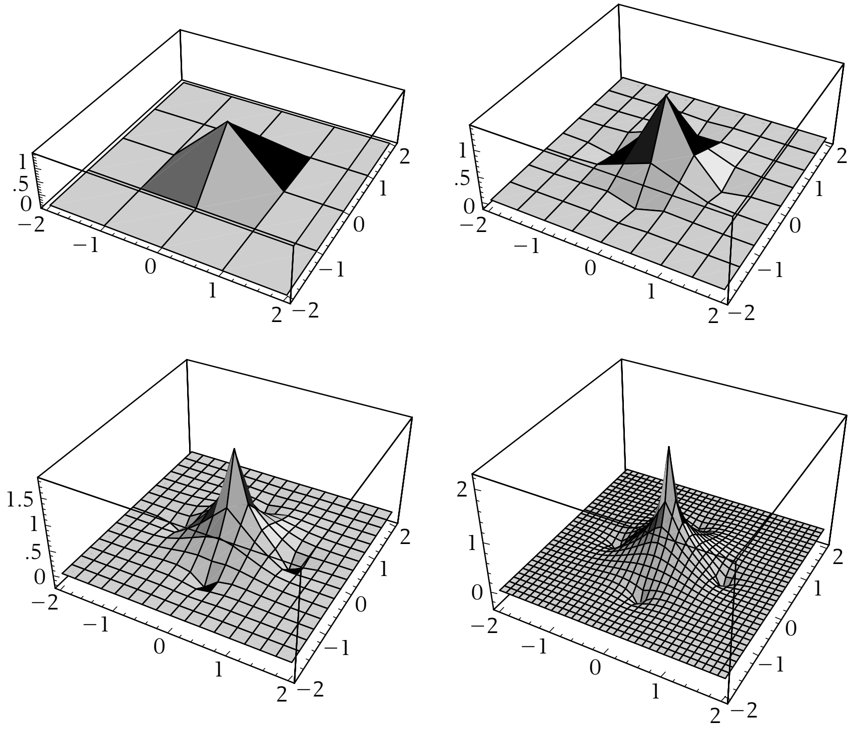


Figure 5.10 A basis function for harmonic splines constructed via subdivision based on the Jacobi iteration.

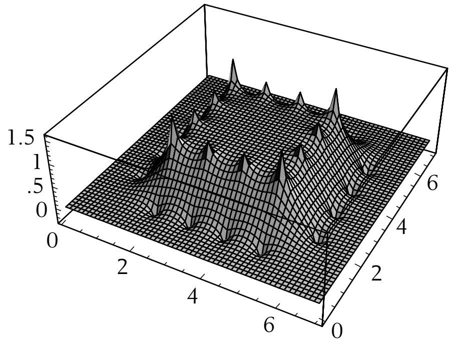


Figure 5.11 The harmonic spline of Figure 5.4 approximated by three rounds of subdivision using the Jacobi iteration.

For differential and variational problems, the idea of expressing subdivision as *prediction* using bilinear subdivision followed by *smoothing* with the Jacobi iteration has been examined by Kobbelt in a number of papers [80, 82, 83, 84, 86]. Note that the other types of iterative methods can be used in place of the Jacobi iteration. For example, Kobbelt and Schröder [87] use cyclic reduction in constructing subdivision schemes for variationally optimal curves. More generally, subdivision schemes of this type can be viewed as instances of multigrid schemes [13] in which the coarse grid correction phase has been omitted.

5.2.3 Optimal Local Approximations via Linear Programming

Our last method takes a more direct approach to constructing a local subdivision scheme that approximates harmonic splines. Because the exact subdivision mask $s[x, y]$ for harmonic splines is globally supported, any locally supported scheme has to be an approximation to the exact globally supported scheme. Our strategy is to compute a subdivision mask $\hat{s}[x, y]$ with a given support that approximately satisfies $l[x, y]\,\hat{s}[x, y] \simeq l[x^2, y^2]$. If the coefficients of $\hat{s}[x, y]$ are treated as unknowns, this approach leads to an optimization problem in these unknown coefficients.

For harmonic splines ($m == 1$), we compute a locally supported subdivision mask $\hat{s}[x, y]$ whose coefficients $\hat{s}[\![i, j]\!]$ are the unknowns. Because the bell-shaped basis function for harmonic splines is symmetric with respect to the axes $x == 0$, $y == 0$ and $x == y$, the number of unknowns in the symbolic representation of $\hat{s}[x, y]$ can be reduced accordingly by reusing variable names. In particular, for fixed i, j

the eight unknowns $\widehat{s}[\![\pm i, \pm j]\!]$, $\widehat{s}[\![\pm j, \pm i]\!]$ are constrained to be equal. The key to computing the remaining unknown coefficients lies in minimizing the residual mask $r[x, y]$ of the form

$$r[x, y] = l[x, y]\,\widehat{s}[x, y] - l[x^2, y^2].$$

Observe that any subdivision mask $\widehat{s}[x, y]$ that causes the residual mask $r[x, y]$ to be zero yields a subdivision scheme that produces solutions to equation 5.6. In keeping with the spirit of the analysis of Chapter 3, we choose to minimize the ∞-norm of the residual mask $r[x, y]$; that is, minimize

$$\text{Max}\left[\sum_{i,j} |r[\![2i, 2j]\!]|,\ \sum_{i,j} |r[\![2i+1, 2j]\!]|,\right. \tag{5.10}$$
$$\left.\sum_{i,j} |r[\![2i, 2j+1]\!]|,\ \sum_{i,j} |r[\![2i+1, 2j+1]\!]|\right].$$

This minimization problem can be expressed as a linear programming problem in the unknowns of the subdivision mask $\widehat{s}[x, y]$. To construct this linear program, we first convert the expressions $|r[\![i, j]\!]|$ appearing in expression 5.10 into a set of inequalities. To this end, we express the unknown $r[\![i, j]\!]$ as the difference of two related unknowns $r^+[\![i, j]\!] - r^-[\![i, j]\!]$, where $r^+[\![i, j]\!] \geq 0$ and $r^-[\![i, j]\!] \geq 0$. Given this decomposition, the absolute value of $[\![i, j]\!]$ satisfies the inequality $|r[\![i, j]\!]| \leq r^+[\![i, j]\!] + r^-[\![i, j]\!]$. Observe that during the course of minimizing the sum $r^+[\![i, j]\!] + r^-[\![i, j]\!]$ one of the variables $r^+[\![i, j]\!]$ and $r^-[\![i, j]\!]$ is forced to be zero, with remaining variables assuming the absolute value $|r[\![i, j]\!]|$.

Based on this transformation, minimizing expression 5.10 is equivalent to minimizing the variable obj subject to the four inequality constraints

$$\begin{aligned}
\textstyle\sum r^+[\![2i, 2j]\!] + r^-[\![2i, 2j]\!] &\leq obj,\\
\textstyle\sum r^+[\![2i, 2j+1]\!] + r^-[\![2i, 2j+1]\!] &\leq obj,\\
\textstyle\sum r^+[\![2i+1, 2j]\!] + r^-[\![2i+1, 2j]\!] &\leq obj,\\
\textstyle\sum r^+[\![2i+1, 2j+1]\!] + r^-[\![2i+1, 2j+1]\!] &\leq obj.
\end{aligned}$$

Because the coefficients $r[\![i, j]\!]$ are linear expressions in the unknown coefficients of $\widehat{s}[x, y]$, substituting these linear expressions for $r[\![i, j]\!]$ leads to a linear program whose minimizer is the approximate mask $\widehat{s}[x, y]$. For example, Figures 5.12 and 5.13 show the 5×5 and 9×9 masks $\widehat{s}[x, y]$, respectively, for harmonic splines. Observe that each of these masks is an increasingly accurate approximation of the exact mask shown in Figure 5.6.

0	0.0303	−0.1212	0.0303	0
0.0303	0.2424	0.4545	0.2424	0.0303
−0.1212	0.4545	1.4545	0.4545	−0.1212
0.0303	0.2424	0.4545	0.2424	0.0303
0	0.0303	−0.1212	0.0303	0

Figure 5.12 The optimal 5 × 5 approximation to the subdivision mask for harmonic splines.

0	0.0014	0.0014	−0.0014	−0.009	−0.0014	0.0014	0.0014	0
0.0014	0.0042	0.0072	0.0019	−0.0331	0.0019	0.0072	0.0042	0.0014
0.0014	0.0072	0.0211	0.0351	−0.1273	0.0351	0.0211	0.0072	0.0014
−0.0014	0.0019	0.0351	0.2445	0.4539	0.2445	0.0351	0.0019	−0.0014
−0.009	−0.0331	−0.1273	0.4539	1.4539	0.4539	−0.1273	−0.0331	−0.009
−0.0014	0.0019	0.0351	0.2445	0.4539	0.2445	0.0351	0.0019	−0.0014
0.0014	0.0072	0.0211	0.0351	−0.1273	0.0351	0.0211	0.0072	0.0014
0.0014	0.0042	0.0072	0.0019	−0.0331	0.0019	0.0072	0.0042	0.0014
0	0.0014	0.0014	−0.0014	−0.009	−0.0014	0.0014	0.0014	0

Figure 5.13 The optimal 9 × 9 approximation to the subdivision mask for harmonic splines.

Figure 5.14 shows an approximation of the basis function $n[x, y]$ for harmonic splines based on three rounds of subdivision with the 9 × 9 mask of Figure 5.13. Figure 5.15 shows the result of three rounds of subdivision with this mask applied to the initial polyhedron of Figure 5.2.

5.2.4 A Comparison of the Three Approaches

All three of the methods in this section (truncated Laurent series, Jacobi iteration, and linear programming) provide finite approximations to the exact subdivision mask $\mathtt{s[x, y]}$ for harmonic splines. To conclude this section, we compare the accuracy of these three methods in reproducing the exact subdivision scheme for harmonic splines. Figure 5.16 shows the ∞-norm of the residual expression $\mathtt{r[x, y]}$ for finite approximations of various sizes using these methods. The upper curve is the residual plot of the Jacobi iteration, the middle curve is the residual plot for the truncated Laurent series, and the lower curve is the residual plot for the linear programming method. The horizontal axis is the size of the approximate mask, and the vertical axis is the norm of the corresponding residual expression.

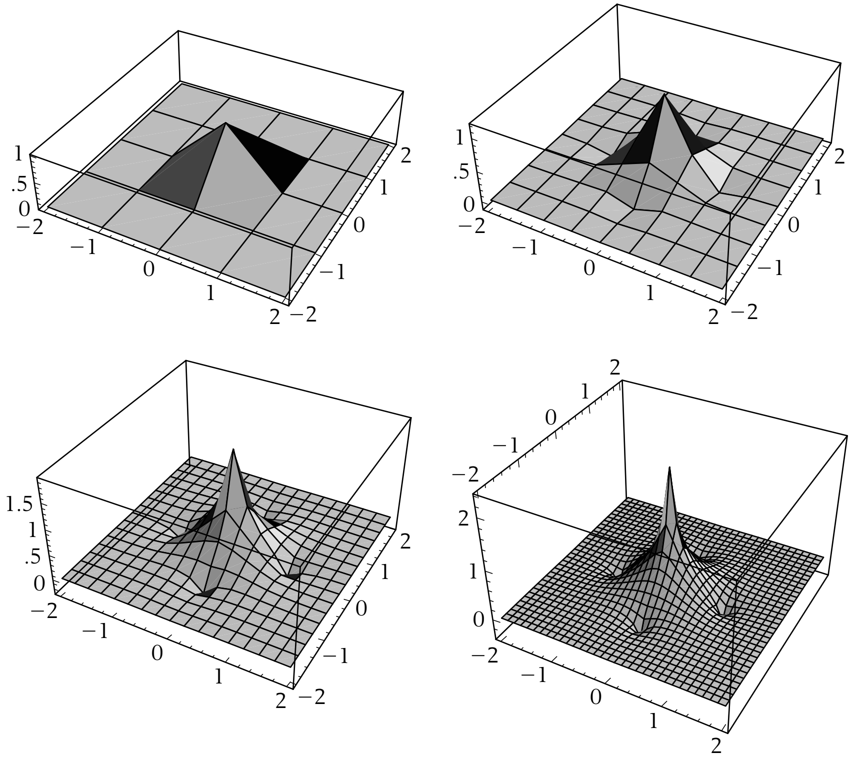


Figure 5.14 A basis function for harmonic splines constructed via subdivision based on the optimal 9 × 9 mask.

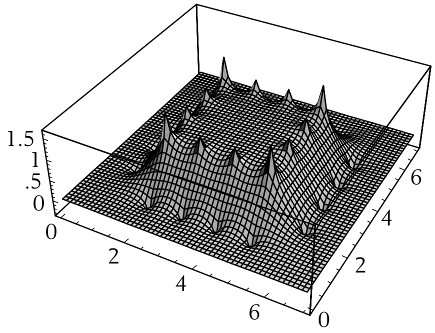


Figure 5.15 The harmonic spline of Figure 5.4 approximated by three rounds of subdivision using the optimal 9 × 9 mask.

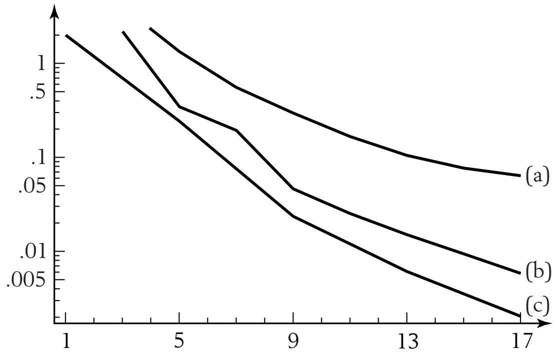


Figure 5.16 Residual plots for the Jacobi iteration (a), Laurent series expansion (b), and linear programming method (c).

As expected, the linear programming method yields masks with minimal residual. In general, the residual for the truncated Laurent mask is roughly twice that of the optimal mask, with all of the error being distributed at the boundary of the mask. Finite masks based on the Jacobi iteration typically have a residual that is at least an order of magnitude larger.

One interesting question involves the behavior of these residuals during several rounds of subdivision. The mask resulting from k rounds of subdivision has the form $\prod_{i=1}^{k} \widehat{s}_{i-1}[x^{2^{k-i}}]$. To measure the accuracy of this mask, we compute the norm of the multilevel residual:

$$l[x, y]\left(\prod_{i=1}^{k} \widehat{s}_{i-1}\left[x^{2^{k-i}}\right]\right) - l\left[x^{2^k}, y^{2^k}\right].$$

Figure 5.17 shows a plot of the multilevel residual for the masks of size 17 × 17 produced by each scheme. As before, the upper curve is the residual plot of the Jacobi iteration, the middle curve is the residual plot for the truncated Laurent series, and the lower curve is the residual plot for the linear programming method. The horizontal axis denotes the number of rounds of subdivision, whereas the vertical axis denotes the norm of the residual.

The truncation of the Laurent series expansion yields reasonable results. Because the Laurent series expansion corresponds to the exact subdivision scheme, the only error induced is that of the truncation. Expanding the support of the truncated series yields a sequence of subdivision schemes that provides increasingly accurate approximations of polyharmonic spline surfaces. On the other hand, the

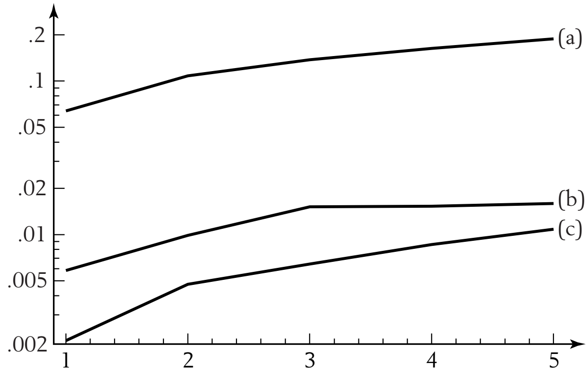


Figure 5.17 Multilevel residuals for Jacobi iteration (a), Laurent series expansion (b), and linear programming method (c).

Jacobi iteration yields masks that are substantially less accurate. In practice, a very large number of iterations may be necessary to obtain reasonably precise solutions, due to the slow convergence of the iteration.

The method based on linear programming exhibits the best performance in terms of minimizing the residual. For example, the 17×17 mask had a residual error of less than one percent, even after five rounds of subdivision. Although the size of this mask might seem prohibitive, applying such larger masks during subdivision can be greatly accelerated by using discrete convolution and the Fast Fourier Transformation. The associated implementation uses this method in computing the multilevel residual for large masks (♚). For example, the multilevel residual for five rounds of subdivision with a 17×17 mask was computed in less than 20 seconds on a laptop computer. One final advantage of the linear programming approach is that for higher-order polyharmonic splines (i.e., $m > 1$) it is easy to add extra constraints on the approximate mask $\widehat{s}[x, y]$ that correspond to polynomial precision and that guarantee the conditions for convergence and smoothness of the resulting subdivision schemes (as summarized in Chapter 3).

5.3 Subdivision for Linear Flows

Our previous examples considered the problem of constructing subdivision schemes for scalar-valued functions (with possibly vector coefficients in the parametric case). In this section, we consider an extension of the techniques of the

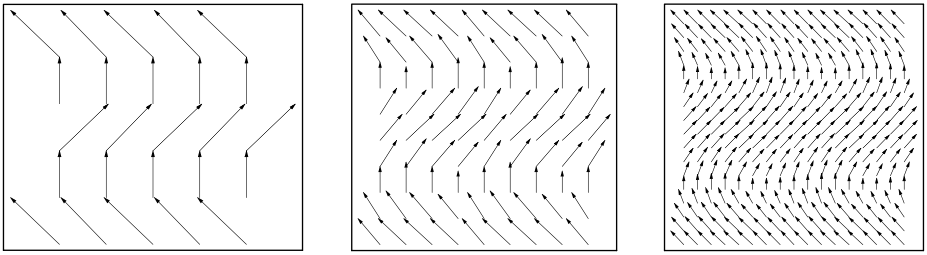


Figure 5.18 An example of subdivision for flow fields.

previous two sections to the problem of constructing subdivision schemes for vector fields. Figure 5.18 shows an example of this idea. Given a coarse vector field on the left, we wish to construct a set of subdivision rules that generate a sequence of increasingly dense vector fields (shown on the right). If these rules are chosen carefully, the limit of these vector fields is a continuous vector field that follows the initial vector field. As was the case for polyharmonic splines, our approach to constructing these rules is a differential one based on a set of partial differential equations that model simple types of flow. This method, developed by the authors, appears in [163].

Mathematically, flow in two dimensions is a vector-valued function $\{u[x, y], v[x, y]\}^T$ in two parameters, x and y. Flows can be visualized in a number of ways. Perhaps the simplest method is to evaluate $\{u[x, y], v[x, y]\}^T$ at a grid of parameter values $\{i, j\} \in \mathbb{Z}^2$. Each resulting vector $\{u[i, j], v[i, j]\}^T$ is then plotted, with its tail placed at $\{i, j\}$ to form a vector field. The behavior of flows is governed by a set of partial differential equations (PDEs) known as *Navier-Stokes equations*. Because the behavior of these equations is a field of study that can occupy an entire career, our goal is to give a brief introduction to several special cases of these equations. The source for most of the material in this introduction is *Fluid Mechanics* by Liggett [97].

5.3.1 Linear Flows

In their most general setting, the Navier-Stokes equations are nonlinear. Because subdivision is an intrinsically linear process, it is unreasonable to expect that the general solution to these equations can be modeled by a subdivision scheme. However, in several important settings, the Navier-Stokes equations reduce to a much simpler set of linear PDEs. We next consider two such cases: perfect flow and slow flow. Later, we derive subdivision schemes for perfect flow, noting that exactly the same methodology can be used to build schemes for slow flow.

Perfect flows are characterized by two properties: incompressibility and zero viscosity. A flow $\{u[x, y], v[x, y]\}^{T}$ is *incompressible* if it satisfies the partial differential equation

$$u^{(1,0)}[x, y] + v^{(0,1)}[x, y] == 0.$$

Flows $\{u[x, y], v[x, y]\}^{T}$ satisfying this equation are said to be *divergence free*. In most flows, viscosity induces rotation in the flow. However, if the fluid has zero viscosity, its flows are free of rotation. Such flows are often referred to as *irrotational*. Irrotational flows are characterized by the partial differential equation

$$u^{(0,1)}[x, y] == v^{(1,0)}[x, y].$$

Together, these two equations in two functions u and v uniquely characterize perfect flow:

$$\begin{aligned} u^{(1,0)}[x, y] + v^{(0,1)}[x, y] &== 0, \\ u^{(0,1)}[x, y] - v^{(1,0)}[x, y] &== 0. \end{aligned} \tag{5.11}$$

To facilitate manipulations involving such systems of PDEs, we use two important tools. As in the scalar case, we express various derivatives using the differential operators $\mathcal{D}[x]$ and $\mathcal{D}[y]$. Left multiplication by the differential operator $\mathcal{D}[x]$ is simply a shorthand method for taking a continuous derivative in the x direction; for example, $\mathcal{D}[x]u[x, y] = u^{(1,0)}[x, y]$ and $\mathcal{D}[y]v[x, y] = v^{(0,1)}[x, y]$. Our second tool is matrix notation. After replacing the derivatives in equation 5.11 with their equivalent differential operators, the two linear equations in two unknowns can be written in matrix form as

$$\begin{pmatrix} \mathcal{D}[x] & \mathcal{D}[y] \\ \mathcal{D}[y] & -\mathcal{D}[x] \end{pmatrix} \begin{pmatrix} u[x, y] \\ v[x, y] \end{pmatrix} == 0. \tag{5.12}$$

In conjunction, these techniques allow the generation of subdivision schemes from systems of PDEs using the same strategy as employed for polyharmonic splines.

Another important class of linear flow is *slow flow*. A slow flow is an incompressible flow in which the viscous behavior of the flow dominates any inertial component of the flow. For example, the flow of viscous fluids such as asphalt, sewage sludge, and molasses is governed almost entirely by its viscous nature. Other examples of slow flow include the movement of small particles through water or air and the swimming of microorganisms [97]. Slow flow may also be viewed as a minimum-energy elastic deformation applied to an incompressible material. This observation is based on the fact that the flow of an extremely viscous fluid is essentially equivalent to an elastic deformation of a solid. One by-product of this view is that any technique for creating slow flows can also be applied to create minimum-energy deformations.

Slow flows in two dimensions are also governed by two partial differential equations. The first partial differential equation corresponds to incompressibility. However, due to their viscosity, slow flows have a rotational component. This rotational component is governed by the partial differential equation

$$u^{(2,1)}[x,y] + u^{(0,3)}[x,y] - v^{(3,0)}[x,y] - v^{(1,2)}[x,y] == 0.$$

(For a complete derivation of this equation, see Liggett [97], page 161.) If $\mathcal{L}[x,y] = \mathcal{D}[x]^2 + \mathcal{D}[y]^2$ denotes the continuous Laplacian, these two partial differential equations can now be written together in matrix form as

$$\begin{pmatrix} \mathcal{D}[x] & \mathcal{D}[y] \\ \mathcal{L}[x,y]\mathcal{D}[y] & -\mathcal{L}[x,y]\mathcal{D}[x] \end{pmatrix} \begin{pmatrix} u[x,y] \\ v[x,y] \end{pmatrix} == 0. \tag{5.13}$$

In this form, the similarity between the governing equations for perfect flow and those for slow flow is striking: the second equation in 5.13 corresponds to the equation for irrotational flow multiplied by the Laplacian $\mathcal{L}[x,y]$. The effect of this extra factor of $\mathcal{L}[x,y]$ on the rotational component of slow flows is easy to explain: perfect flows exhibit infinite velocities at sources of rotational flow. The extra factor of $\mathcal{L}[x,y]$ smooths the velocity field at rotational sources and causes slow flows to be continuous everywhere. For the sake of simplicity, we focus most of our analysis on the case of subdivision for perfect flow. However, the same techniques can be used to derive a subdivision scheme for slow flow. Those readers interested in more details for the case of slow flow should consult [163]. The *Mathematica* implementation associated with this book includes code for computing approximate matrix masks for both schemes. This code was used to compute examples for both perfect and slow flows.

Traditionally, fluid flow has been modeled through either explicit or numerical solutions of the associated PDEs. Explicit solutions for perfect flow are known for a number of primitives, such as sources, sinks, and rotors. These can be combined into more complex fields using simple linear combinations (see [97] for more details).

Numerical solutions for the modeling and simulation of flow are a very active research area. Commonly, such solutions involve approximation using either finite difference or finite element schemes. Brezzi and Fortin [14] provide an introduction to some of the standard numerical techniques used for modeling flow. Recently, cellular automata have been applied with some success for the discrete modeling of physical problems, including gas and fluid flow [21, 133].

On the computer graphics side, Kass and Miller [78] simulate surface waves in water by approximately solving a two-dimensional shallow water problem. Chen and Lobo [19] solve the Navier-Stokes equations in two dimensions and use the resulting pressure field to define a fluid surface. Chiba et al. [20] simulate water currents using a particle-based behavioral model. Miller and Pearce [106] model viscous fluids through a connected particle system. Wejchert and Haumann [165] introduce notions from aerodynamics to the graphics community. Stam and Fiume [147] model turbulent wind fields for animation based on solving the underlying PDEs. Foster and Metaxas [64] suggest solving the Navier-Stokes equations on a coarse grid in three dimensions using a finite-difference approach and then interpolating the coarse solution locally as needed. They also extend their method to handle turbulent steam [65].

5.3.2 Primal Versus Dual Subdivision

Our goal is to use the differential method of Chapter 4 in deriving vector subdivision schemes for linear flows. In the previous scalar cases, the differential method produces a subdivision scheme that given an coarse polygon p_0 defines a sequence of polygons p_k that converge to a continuous curve $p[x]$ approximating the initial polygon p_0. In the current vector case, the differential method produces a subdivision scheme that given a coarse vector field $\{\mathsf{u}_0, \mathsf{v}_0\}^T$ on the coarse grid $\mathbb{Z}^2$ defines a sequence of increasingly dense vector fields $\{\mathsf{u}_k, \mathsf{v}_k\}^T$ on the finer grid $\frac{1}{2^k}\mathbb{Z}^2$ that converge to a continuous vector field $\{u[x,y], v[x,y]\}^T$ approximating the initial vector field $\{\mathsf{u}_0, \mathsf{v}_0\}^T$.

Before discussing the details of the subdivision scheme for flow, we first make an important observation concerning the differential method of Chapter 4. In the univariate case, the difference mask $\mathsf{d}_k[\mathsf{x}]$ for the order m divided differences involves only non-negative powers of x and leads to a subdivision mask $\mathsf{s}[\mathsf{x}]$ for B-splines of order m that also only involves non-negative powers of x. As a result, the corresponding B-spline basis function $n[x]$ is supported on the interval $[0, \mathsf{m}]$. For scalar subdivision schemes, we did not need to be careful about introducing extra powers of x into the difference mask $\mathsf{d_k[x]}$ and its associated subdivision mask $\mathsf{s[x]}$ because these extra powers simply translated the resulting scaling function $n[x]$.

For vector subdivision schemes, we must be much more careful about introducing extraneous translations into the components of the vector field. For example, translating only the first component of the flow $\{u[x,y], v[x,y]\}^T$ by one unit in the x direction yields a new flow $\{u[x-1,y], v[x,y]\}^T$. This new flow is not a translate of the original flow but a completely different flow. To translate the flow by one unit in the x direction, we must translate each component of the flow by one unit, that is, form the new flow $\{u[x-1,y], v[x-1,y]\}^T$. The main consequence of this observation is that when applying the differential method to vector schemes we must avoid introducing extraneous shifts into the difference masks used to model equations 5.12 and 5.13.

The solution to this problem is to center higher-order difference masks by taking powers of the centered first difference mask $\mathsf{d[x]} = \mathsf{x}^{-\frac{1}{2}} - \mathsf{x}^{\frac{1}{2}}$. Before constructing the finite difference equations for equation 5.11, we first consider some of the implications of using half-integer powers of x in our construction. For B-splines, using this centered mask $\mathsf{d[x]}$ in place of the uncentered mask $1 - \mathsf{x}$ leads to subdivision masks $\mathsf{s[x]}$ whose basis functions $n[x]$ are supported on the interval $[-\frac{\mathsf{m}}{2}, \frac{\mathsf{m}}{2}]$. If m is even, the powers $\mathsf{x}^{\frac{1}{2}}$ cancel out, yielding a mask $\mathsf{s[x]}$ involving only integral powers of x. Because m is even, the knots of the basis function $n[x]$ remain positioned at the integers $\mathbb{Z}$. Moreover, the coefficients attached to the integer translates of this basis function also remain positioned on the integer grid $\mathbb{Z}$. Such schemes are known as *primal* schemes. After k rounds of subdivision for a primal scheme, the entries of $\mathsf{p_k}$ are positioned on the grid $\frac{1}{2^k}\mathbb{Z}$.

If m is odd, the situation becomes more complicated. The powers of $\mathsf{x}^{\frac{1}{2}}$ do not cancel, and the resulting mask $\mathsf{s[x]}$ involves fractional powers of x. More important, because m is odd, the integer translates of the basis function $n[x]$ have knots that lie at the midpoints of segments in the integer grid $\mathbb{Z}$. We denote the grid consisting of such points $\mathsf{i} + \frac{1}{2}$, where $\mathsf{i} \in \mathbb{Z}$ by $\mathbb{D}[\mathbb{Z}]$. Because B-splines are typically viewed as having knots at the integers, this half-integer shift in $\mathbb{D}[\mathbb{Z}]$ can be cancelled by inducing a half-integer shift in the coefficients of $\mathsf{p_0[x]}$, that is, positioning the coefficients of $\mathsf{p_0}$ at the midpoints of segments in the integer grid $\frac{1}{2^k}\mathbb{Z}$. In this model, $\mathsf{p_0[x]}$ is now a generating function involving powers of x of the form $\mathsf{x^i}$, where $\mathsf{i} \in \mathbb{D}[\mathbb{Z}]$. Schemes with this structure are known as *dual* schemes. After k rounds of subdivision for a dual scheme, the entries of $\mathsf{p_k}$ are positioned on the grid $\mathbb{D}[\frac{1}{2^k}\mathbb{Z}]$, that is, at the midpoints of segments in the grid $\frac{1}{2^k}\mathbb{Z}$.

In the case of linear flow, the structure of the differential equations causes the resulting vector subdivision schemes to have an unusual primal/dual structure

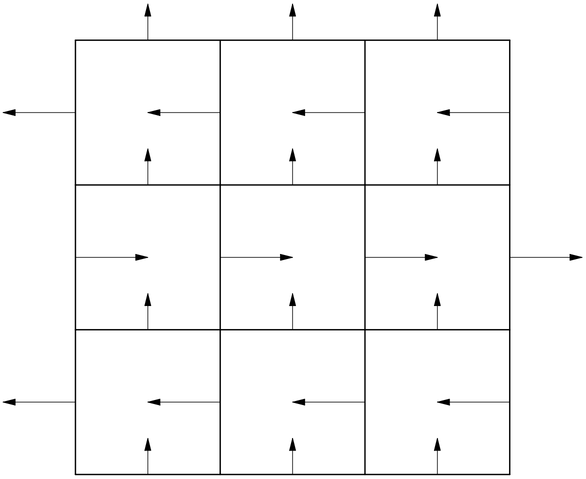


Figure 5.19 Plotting the components of the discrete flow $\{u_0, v_0\}^T$ on the integer grid.

that does not occur in the scalar case. In particular, the linear flows $\{u[x,y], v[x,y]\}^T$ defined by these schemes have the property that the $u[x,y]$ component is primal in x and dual in y, whereas the $v[x,y]$ component is dual in x and primal in y. In particular, the entries of the u_k component of the discrete flow $\{u_k, v_k\}^T$ lie on the grid $\frac{1}{2^k}\mathbb{Z}\times\mathbb{D}[\frac{1}{2^k}\mathbb{Z}]$, whereas the entries of the component v_k lie on the grid $\mathbb{D}[\frac{1}{2^k}\mathbb{Z}]\times\frac{1}{2^k}\mathbb{Z}$. One way to visualize this discrete flow is to define a square grid passing through the vertices of $\frac{1}{2^k}\mathbb{Z}^2$. The entries of u_k approximate flow in the x direction normal to the vertical walls of the square grid. The entries of v_k approximate flow in the y direction normal to the horizontal walls of the grid. Figure 5.19 illustrates this positioning, with the horizontal vectors corresponding to the u_k component and the vertical vectors corresponding to the v_k component. The length of the vectors corresponds to the value of the corresponding entry.

As in the scalar case, the vectors u_k and v_k can be represented as generating functions $u_k[x, y]$ and $v_k[x, y]$, with the dual component of these generating functions involving half-integer powers of x and y. (The associated implementation discusses a method for implementing dual subdivision schemes with generating functions based on integral powers of x and y (♚).)

5.3.3 A Finite Difference Scheme for Perfect Flows

To construct the subdivision scheme for perfect flow, we first discretize differential equation 5.12 on the grid $\frac{1}{2^k}\mathbb{Z}^2$. As suggested previously, we represent the differential operator $\mathcal{D}[x]$ in terms of a centered divided difference of the form $\mathrm{d_k[x]} = 2^{\mathrm{k}}(\mathrm{x}^{-\frac{1}{2}} - \mathrm{x}^{\frac{1}{2}})$. Under this discretization, the left-hand side of equation 5.12 reduces to

$$\begin{pmatrix} \mathrm{d_k[x]} & \mathrm{d_k[y]} \\ \mathrm{d_k[y]} & -\mathrm{d_k[x]} \end{pmatrix} \begin{pmatrix} \mathrm{u_k[x, y]} \\ \mathrm{v_k[x, y]} \end{pmatrix} == \begin{pmatrix} \mathrm{d_k[x]\, u_k[x, y] + d_k[y]\, v_k[x, y]} \\ \mathrm{d_k[y]\, u_k[x, y] - d_k[x]\, v_k[x, y]} \end{pmatrix}.$$

The coefficients of the upper generating function $\mathrm{d_k[x]\,u_k[x, y] + d_k[y]\,v_k[x, y]}$ estimate the compressibility of the discrete flow $\{\mathrm{u_k, v_k}\}^{\mathrm{T}}$ at the centers of squares in $\frac{1}{2^{\mathrm{k}}}\mathbb{Z}^2$. Each coefficient is associated with a square and consists of the difference of entries in $\mathrm{u_k}$ and $\mathrm{v_k}$ corresponding to flow entering and exiting the four sides of the square. The coefficients of the lower generating function $\mathrm{d_k[y]\,u_k[x, y] - d_k[x]\,v_k[x, y]}$ estimate the rotational component of the discrete flow at grid points of $\frac{1}{2^{\mathrm{k}}}\mathbb{Z}^2$. Each coefficient in this expression involves the difference of the four entries of $\mathrm{u_k}$ and $\mathrm{v_k}$ corresponding to the edges incident on a particular grid point. (The discrete flow of Figure 5.19 is incompressible but has a non-zero rotational component at each grid point.)

Given this discretization of the left-hand side of equation 5.12, our next task is to choose a right-hand side for equation 5.12 that relates a solution $\{u[x, y], v[x, y]\}^{\mathrm{T}}$ to an initial coarse flow $\{\mathrm{u_0, v_0}\}^{\mathrm{T}}$ in an intuitive manner. One particularly nice choice is

$$\begin{pmatrix} \mathcal{D}[x] & \mathcal{D}[y] \\ \mathcal{D}[y] & -\mathcal{D}[x] \end{pmatrix} \begin{pmatrix} u[x, y] \\ v[x, y] \end{pmatrix} == \begin{pmatrix} 0 \\ \sum_{\mathrm{i,j}} \mathrm{rot}[\![\mathrm{i, j}]\!]\, \delta[x - \mathrm{i}, y - \mathrm{j}] \end{pmatrix},$$

where $\mathrm{rot}[\![\mathrm{i, j}]\!]$ is an estimate of the rotational component of the flow $\{\mathrm{u_0, v_0}\}^{\mathrm{T}}$ at the integer grid point $\{\mathrm{i, j}\}$. In terms of generating functions, $\mathrm{rot[x, y]}$ is the expression $\mathrm{d_0[y]\,u_0[x, y] - d_0[x]\,v_0[x, y]}$. Due to this particular choice for the right-hand side, the resulting flows $\{u[x, y], v[x, y]\}^{\mathrm{T}}$ are divergence free everywhere and driven by rotational sources positioned at the integer grid points $\mathbb{Z}^2$.

Our last task before proceeding to the derivation of the subdivision scheme is to construct a set of finite difference equations analogous to the partial differential equations previously given. Given that the discrete analog of the Dirac delta $\delta[x, y]$ is 4^{k}, the appropriate set of finite difference equations on the grid $\frac{1}{2^{\mathrm{k}}}\mathbb{Z}^2$ has the form

$$\begin{aligned} \begin{pmatrix} \mathrm{d_k[x]} & \mathrm{d_k[y]} \\ \mathrm{d_k[y]} & -\mathrm{d_k[x]} \end{pmatrix} \begin{pmatrix} \mathrm{u_k[x, y]} \\ \mathrm{v_k[x, y]} \end{pmatrix} &== 4^{\mathrm{k}} \begin{pmatrix} 0 \\ \mathrm{rot}\left[\mathrm{x}^{2^{\mathrm{k}}}, \mathrm{y}^{2^{\mathrm{k}}}\right] \end{pmatrix} \qquad (5.14) \\ &== 4^{\mathrm{k}} \begin{pmatrix} 0 & 0 \\ \mathrm{d_0}\left[\mathrm{y}^{2^{\mathrm{k}}}\right] & -\mathrm{d_0}\left[\mathrm{x}^{2^{\mathrm{k}}}\right] \end{pmatrix} \begin{pmatrix} \mathrm{u_0}\left[\mathrm{x}^{2^{\mathrm{k}}}, \mathrm{y}^{2^{\mathrm{k}}}\right] \\ \mathrm{v_0}\left[\mathrm{x}^{2^{\mathrm{k}}}, \mathrm{y}^{2^{\mathrm{k}}}\right] \end{pmatrix}. \end{aligned}$$

5.3.4 **A Subdivision Scheme for Perfect Flows**

Given the previous finite difference equation, we can now derive the associated subdivision scheme for perfect flow. The subdivision mask for this scheme consists of a matrix of generating functions that relates successive solutions $\{u_{k-1}, v_{k-1}\}^T$ and $\{u_k, v_k\}^T$ and has the form

$$\begin{pmatrix} u_k[x, y] \\ v_k[x, y] \end{pmatrix} == \begin{pmatrix} s_{11}[x, y] & s_{12}[x, y] \\ s_{21}[x, y] & s_{22}[x, y] \end{pmatrix} \begin{pmatrix} u_{k-1}[x^2, y^2] \\ v_{k-1}[x^2, y^2] \end{pmatrix}.$$

Here, we let $s_{ij}[x]$ denote the ijth entry of this matrix of generating functions. (Note that we have dropped the subscript $k-1$ from s because the resulting scheme is stationary.) The key to computing this matrix of masks is to construct an associated finite difference equation that relates successive solutions $\{u_{k-1}, v_{k-1}\}^T$ and $\{u_k, v_k\}^T$. In the case of perfect flows, this two-scale relation has the form

$$\begin{pmatrix} d_k[x] & d_k[y] \\ d_k[y] & -d_k[x] \end{pmatrix} \begin{pmatrix} u_k[x, y] \\ v_k[x, y] \end{pmatrix} == 4 \begin{pmatrix} 0 & 0 \\ d_{k-1}[y^2] & -d_{k-1}[x^2] \end{pmatrix} \begin{pmatrix} u_{k-1}[x^2, y^2] \\ v_{k-1}[x^2, y^2] \end{pmatrix}.$$

Substituting for $\{u_k[x, y], v_k[x, y]\}^T$ and canceling common factors of $\{u_{k-1}[x^2, y^2], v_{k-1}[x^2, y^2]\}^T$ on both sides of this equation yields

$$\begin{pmatrix} d_k[x] & d_k[y] \\ d_k[y] & -d_k[x] \end{pmatrix} \begin{pmatrix} s_{11}[x, y] & s_{12}[x, y] \\ s_{21}[x, y] & s_{22}[x, y] \end{pmatrix} == 4 \begin{pmatrix} 0 & 0 \\ d_{k-1}[y^2] & -d_{k-1}[x^2] \end{pmatrix}. \tag{5.15}$$

The beauty of this relation is that the mask $s_{ij}[x, y]$ can now be computed by inverting the matrix of generating functions $\left(\begin{smallmatrix} d_k[x] & d_k[y] \\ d_k[y] & -d_k[x] \end{smallmatrix}\right)$. The resulting matrix of masks has the form

$$\begin{pmatrix} s_{11}[x, y] & s_{12}[x, y] \\ s_{21}[x, y] & s_{22}[x, y] \end{pmatrix} == \frac{2}{l[x, y]} \begin{pmatrix} d[y]\, d[y^2] & -d[y]\, d[x^2] \\ -d[x]\, d[y^2] & d[x]\, d[x^2] \end{pmatrix}, \tag{5.16}$$

where $d[x] = x^{-\frac{1}{2}} - x^{\frac{1}{2}}$ and $l[x, y] = d[x]^2 + d[y]^2$ is the discrete Laplacian mask. This matrix $(s_{ij}[x, y])$ encodes the subdivision scheme for perfect flow. Note that this subdivision scheme is a true vector scheme: u_k depends on both u_{k-1} and v_{k-1}. Such schemes, although rare, have been the object of some theoretical study. Dyn [49] investigates some of the properties of such vector schemes.

As in the polyharmonic case, the denominator $l[x, y]$ does not divide the various entries in the numerators of the masks $s_{ij}[x, y]$. In reality, this difficulty is to be expected. Flows, even linear ones, are intrinsically nonlocal. However, just as in the case of polyharmonic surfaces, the entries $s_{ij}[x, y]$ of the matrix of subdivision masks can be approximated by a matrix of finite masks $\widehat{s}_{ij}[x, y]$ using a variant of the linear programming technique discussed in the previous section.

To construct this linear program, we observe that due to symmetries in x and y in equation 5.16 the exact masks $s_{ij}[x, y]$ satisfy the relations $s_{11}[x, y] == s_{22}[y, x]$ and $s_{21}[x, y] == s_{12}[y, x]$. Moreover, the coefficients of $s_{11}[x, y]$ are symmetric in x and y, whereas the coefficients of $s_{21}[x, y]$ are antisymmetric in x and y. Thus, our task reduces to solving for approximate masks $\widehat{s}_{11}[x, y]$ and $\widehat{s}_{21}[x, y]$ that are symmetric and antisymmetric, respectively.

If these masks are fixed to have finite support, we can solve for the unknown coefficients of these approximate masks using equation 5.15. This matrix equation includes two coupled linear equations involving the exact masks $s_{11}[x, y]$ and $s_{21}[x, y]$. We wish to compute approximations $\widehat{s}_{11}[x, y]$ and $\widehat{s}_{21}[x, y]$ to these exact masks such that the approximate scheme is irrotational, that is,

$$d[y]\,\widehat{s}_{11}[x, y] - d[x]\,\widehat{s}_{21}[x, y] == 2d[y^2],$$

while minimizing the ∞-norm of the compression term

$$d[x]\widehat{s}_{11}[x, y] + d[y]\widehat{s}_{21}[x, y].$$

Additionally, we can force the approximate vector scheme to have constant precision by enforcing the four auxiliary constraints

$$\begin{aligned}\widehat{s}_{11}[1, 1] &= 4,\\ \widehat{s}_{11}[-1, 1] &= 0,\\ \widehat{s}_{11}[1, -1] &= 0\\ \widehat{s}_{11}[-1, -1] &= 0\end{aligned}$$

during the linear programming process. These four conditions in conjunction with the fact that $\widehat{s}_{21}[\pm 1, \pm 1] == 0$ (due to the antisymmetric structure of $\widehat{s}_{21}[x, y]$) automatically guarantee that the resulting vector scheme reproduces constant vector fields. The associated implementation contains a *Mathematica* version of this algorithm.

Given these approximate masks, we can apply the vector subdivision scheme to an initial, coarse vector field $\{u_0, v_0\}^T$ and generate increasingly dense vector fields $\{u_k, v_k\}^T$ via the matrix relation

$$\begin{pmatrix} u_k[x, y] \\ v_k[x, y] \end{pmatrix} = \begin{pmatrix} \widehat{s}_{11}[x, y] & \widehat{s}_{21}[y, x] \\ \widehat{s}_{21}[x, y] & \widehat{s}_{11}[y, x] \end{pmatrix} \begin{pmatrix} u_{k-1}[x^2, y^2] \\ v_{k-1}[x^2, y^2] \end{pmatrix}.$$

If the vector field $\{u_{k-1}, v_{k-1}\}^T$ is represented as a pair of arrays, the polynomial multiplications in this expression can be implemented very efficiently using discrete convolution. The resulting implementation allows us to model and manipulate flows in real time. Although matrix masks as small as 5×5 yield nicely behaved vector fields, we recommend the use of 9×9 or larger matrix masks if visually realistic approximations to perfect flows are desired.

Visualizing the discrete vector fields $\{u_k, v_k\}^T$ as done in Figure 5.19 is awkward due to the fact that entries of u_k and v_k are associated with the grids $\frac{1}{2^k}\mathbb{Z} \times \mathbb{D}[\frac{1}{2^k}\mathbb{Z}]$ and $\mathbb{D}[\frac{1}{2^k}\mathbb{Z}] \times \frac{1}{2^k}\mathbb{Z}$, respectively. One solution to this problem is to average pairs of entries in u_k that are adjacent in the y direction. The result is an averaged component whose entries lie on the grid $\frac{1}{2^k}\mathbb{Z}^2$. Similarly, applying a similar averaging to pairs of entries in v_k that are adjacent in the x direction yields a second component whose entries lie on the grid $\frac{1}{2^k}\mathbb{Z}^2$. The coefficients of these averaged components can be combined and be plotted as vectors placed at grid points of $\frac{1}{2^k}\mathbb{Z}^2$. Figures 5.20 and 5.21 show plots of several rounds of subdivision applied to a vector basis function defined in the x direction. Figure 5.20 uses the optimal 7×7 approximations $\widehat{s}_{ij}[x, y]$ to the mask for perfect flow. Figure 5.21 uses the optimal 7×7 approximations $\widehat{s}_{ij}[x, y]$ to the mask for slow flow. Figure 5.18, a slow flow, was also generated using this mask. Note that in both cases the averaging in the y direction has split the basis vector in the x direction into two separate vectors of length $\frac{1}{2}$. The effects of this averaging diminish as the vector field converges to the underlying continuous flows.

5.3.5 An Analytic Basis for Linear Flows

Due to the linearity of the process, the limiting vector field defined by the subdivision scheme can be written as a linear combination of translates of two vector basis functions $n_x[x, y]$ and $n_y[x, y]$ multiplied by the components u_0 and v_0 of the initial vector field, respectively. Specifically, the components of the limiting field $\{u[x, y], v[x, y]\}^T$ have the form

$$\begin{pmatrix} u[x,y] \\ v[x,y] \end{pmatrix} = \sum_{i \in \mathbb{Z},\, j \in \mathbb{D}[\mathbb{Z}]} u_0[\![i, j]\!]\, n_x[x - i, y - j] + \sum_{i \in \mathbb{D}[\mathbb{Z}],\, j \in \mathbb{Z}} v_0[\![i, j]\!]\, n_y[x - i, y - j].$$

(Note that both $n_x[x, y]$ and $n_y[x, y]$ are vector-valued functions.) Our goal in this final section is to find simple expressions for these vector basis functions in terms of the radial basis function $c[x, y]$ for harmonic splines. Recall that the continuous field $\{u[x, y], v[x, y]\}^T$ is the limit of the discrete fields $\{u_k, v_k\}^T$ as $k \to \infty$. By solving

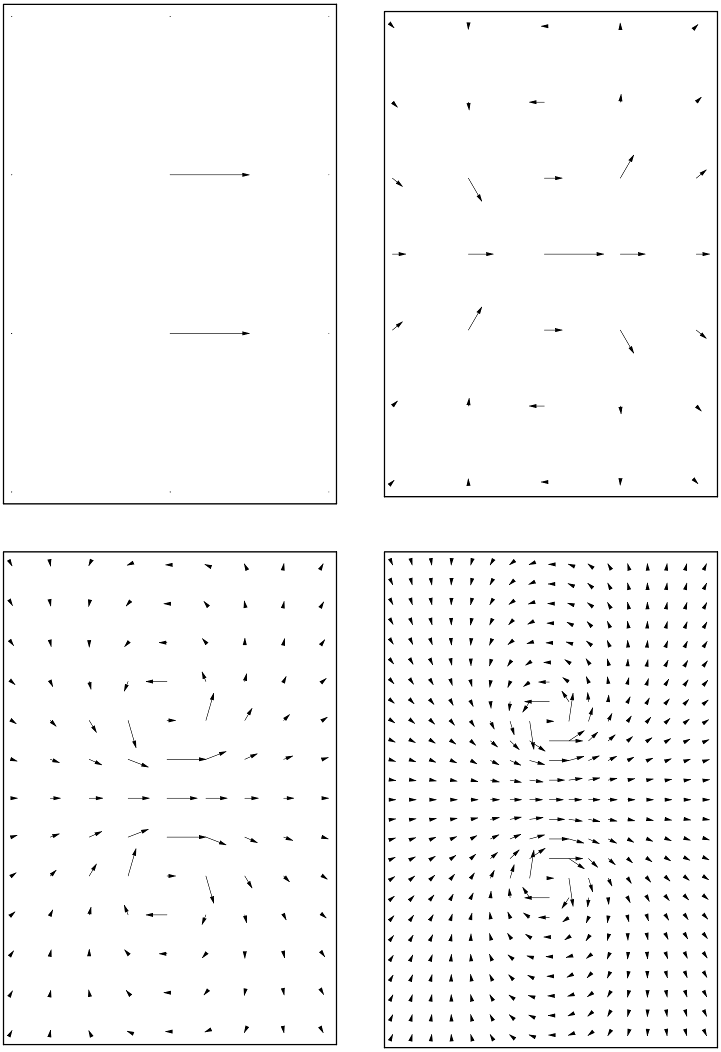


Figure 5.20 Three rounds of subdivision for a basis function in the x direction (perfect flow).

equation 5.14, we can express $\{u_k, v_k\}^T$ directly in terms of $\{u_0, v_0\}^T$:

$$\begin{pmatrix} u_k[x, y] \\ v_k[x, y] \end{pmatrix} = \frac{4^k}{l_k[x, y]} \begin{pmatrix} d_k[y] \\ -d_k[x] \end{pmatrix} \begin{pmatrix} d_0\left[y^{2^k}\right] & -d_0\left[x^{2^k}\right] \end{pmatrix} \begin{pmatrix} u_0\left[x^{2^k}, y^{2^k}\right] \\ v_0\left[x^{2^k}, y^{2^k}\right] \end{pmatrix}. \tag{5.17}$$

Our task is to find continuous analogs of the various parts of this matrix expression as $k \to \infty$. We first analyze the most difficult part of the expression, $\frac{4^k}{l_k[x,y]}$.

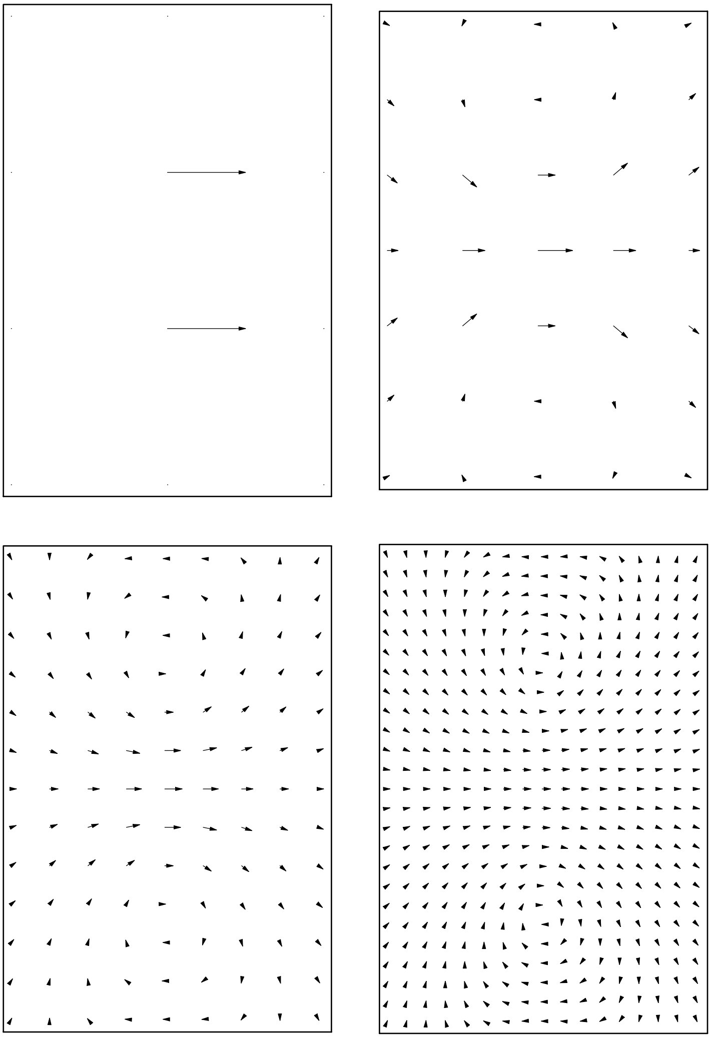


Figure 5.21 Three rounds of subdivision for a basis function in the x direction (slow flow).

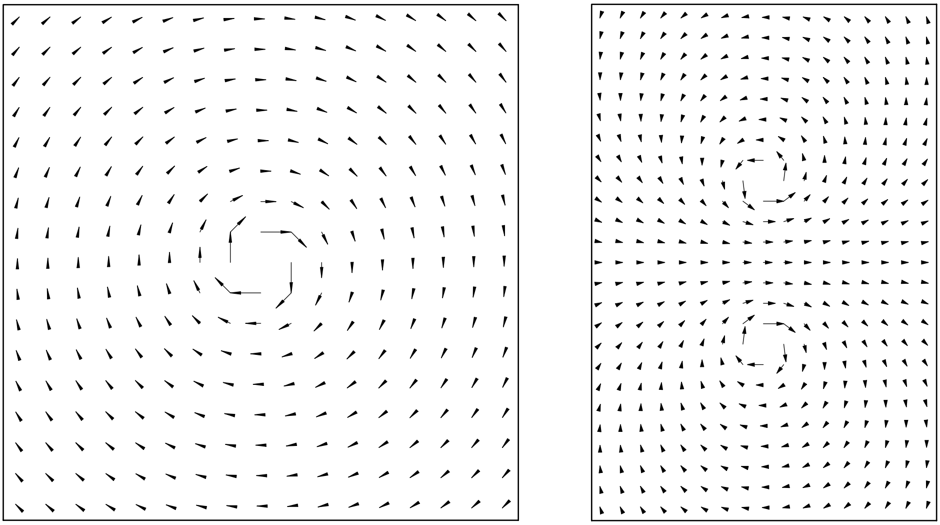
Consider a sequence of generating functions $c_k[x, y]$ such that $l_k[x, y]\,c_k[x, y] = 4^k$ for all k. The coefficients of the $c_k[x, y]$ are discrete approximations to the radial basis function $c[x, y]$ satisfying the differential equation $\mathcal{L}[x, y]\,c[x, y] = \delta[x, y]$.

The other components of this equation are easier to interpret. The differences $d_k[x]$ and $d_k[y]$ taken on $\frac{1}{2^k}\mathbb{Z}^2$ converge to the continuous derivatives $\mathcal{D}[x]$ and $\mathcal{D}[y]$, respectively. If $\mathsf{g}_\mathsf{k}[\mathsf{x},\mathsf{y}]$ corresponds to $\mathsf{c}_\mathsf{k}[\mathsf{x},\mathsf{y}]\left(\begin{smallmatrix}\mathsf{d}_\mathsf{k}[\mathsf{y}]\\ -\mathsf{d}_\mathsf{k}[\mathsf{x}]\end{smallmatrix}\right)$, the continuous analog is the vector function $g[x, y]$ of the form

$$g[x,y] = \begin{pmatrix} \mathcal{D}[y] \\ -\mathcal{D}[x] \end{pmatrix} c[x,y]. \tag{5.18}$$

The behavior of $g[x, y]$ gives us our first insight into the structure of perfect flow. In fact, $g[x, y]$ is a generator for a localized, rotational perfect flow. We can verify that $g[x, y]$ is truly a perfect flow by substituting the definition of $g[x, y]$ into equation 5.12. (Remember that the radial basis function $c[x, y]$ satisfies $(\mathcal{D}[x]^2 + \mathcal{D}[y]^2)c[x, y] == \delta[x, y]$.) Likewise, we can verify that this flow is highly localized (i.e., decays to zero very quickly, away from the origin), based on the analytic representation. The left-hand portion of Figure 5.22 shows a plot of $g[x, y]$ on the domain $[-1, 1]^2$.

To complete our construction, we observe that equation 5.17 can be written as $\mathsf{g}_\mathsf{k}[\mathsf{x},\mathsf{y}]\{\mathsf{d}_0[\mathsf{y}^{2^\mathsf{k}}], -\mathsf{d}_0[\mathsf{x}^{2^\mathsf{k}}]\}$. These differences $\mathsf{d}_0[\mathsf{x}^{2^\mathsf{k}}]$ and $\mathsf{d}_0[\mathsf{y}^{2^\mathsf{k}}]$ taken on the grid $\frac{1}{2^\mathsf{k}}\mathbb{Z}^2$ correspond to unit differences of $g[x, y]$ taken in the x and y directions, respectively, on the integer grid $\mathbb{Z}^2$. Thus, the coefficients of the generating functions $\mathsf{g}_\mathsf{k}[\mathsf{x},\mathsf{y}]\{\mathsf{d}_0[\mathsf{y}^{2^\mathsf{k}}], -\mathsf{d}_0[\mathsf{x}^{2^\mathsf{k}}]\}$ are converging to vector basis functions for our subdivision


(a)

(b)

Figure 5.22 A rotational generator $g[x, y]$ (a) and vector basis function $n_x[x, y]$ (b) for perfect flow.

scheme of the form

$$\begin{aligned} n_x[x,y] &= g\left[x, y+\tfrac{1}{2}\right] - g\left[x, y-\tfrac{1}{2}\right], \\ n_y[x,y] &= -g\left[x+\tfrac{1}{2}, y\right] + g\left[x-\tfrac{1}{2}, y\right]. \end{aligned} \tag{5.19}$$

The right-hand portion of Figure 5.22 contains a plot of the vector basis function $n_x[x, y]$. Note that this vector basis consists of a pair of rotational sources positioned so as to drive a flow along the x axis. Again, this flow is localized in the sense that it decays rapidly to zero away from the origin. More complex flows can be constructed by taking linear combinations of these vector basis functions. Due to the normalizing constant of $\frac{1}{4\pi}$ used in defining the original radial basis function $c[x, y]$, the scheme has constant precision. In particular, choosing the coarse vector field $\{\mathrm{u}_0, \mathrm{v}_0\}^{\mathrm{T}}$ to be translates of a constant vector defines a constant flow of the same magnitude in the same direction.

A similar approach can be used to construct the analytic bases for slow flow. The main change is that equation 5.18 now also involves a difference of the radial basis functions for $c[x, y] = \frac{(x^2+y^2)\,\mathrm{Log}[x^2+y^2]}{16\pi}$ for biharmonic splines as well as its derivatives. In particular, the rotational generator $g[x, y]$ for slow flow has the form

$$g[x,y] = \begin{pmatrix} \mathcal{D}[y] \\ -\mathcal{D}[x] \end{pmatrix} (-4c[x,y] + c[x-1,y] + c[x+1,y] + c[x,y-1] + c[x,y+1]). \tag{5.20}$$

The left-hand portion of Figure 5.23 shows a plot of $g[x, y]$ on the domain $[-1, 1]^2$.

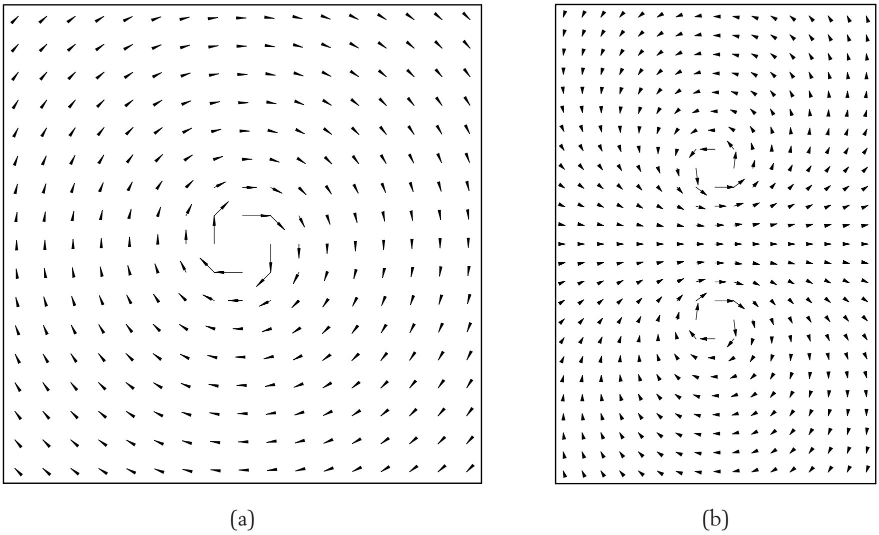


Figure 5.23 A rotational generator $g[x, y]$ (a) and vector basis function $n_x[x, y]$ (b) for slow flow.

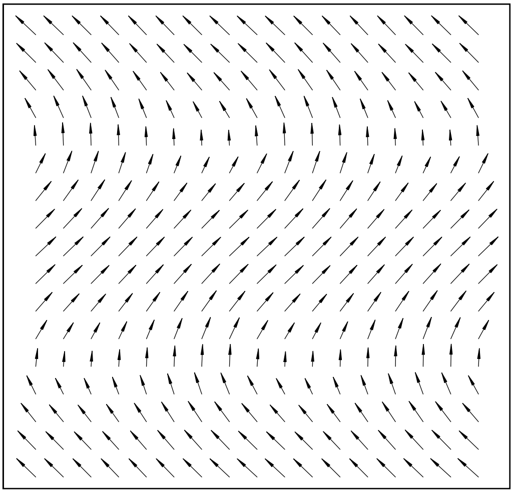


Figure 5.24 A plot of the flow from Figure 5.18 using the analytic representation for slow flow.

Now, the analytic basis functions $n_x[x, y]$ and $n_y[x, y]$ are again defined as in equation 5.19. The right-hand portion of Figure 5.23 shows a plot of the basis function $n_x[x, y]$. The final figure, Figure 5.24, shows a plot of the analytic representation of the flow of Figure 5.18.

CHAPTER 6

Variational Schemes for Bounded Domains

The first part of this book considered various subdivision schemes defined over unbounded, uniform grids. The rest of the book focuses on extending these schemes to domains that are nonuniform in nature. Note that nonuniform domains arise naturally due to the presence of boundaries, both internal and external, in most shapes. In the surface case, nonuniform schemes also necessarily arise in any attempt to model surfaces of genus other than one. In this chapter, we describe a generalization of the differential approach of Chapter 4 to the case of uniform grids over bounded domains.

The key to this approach is to develop a variational version of the original differential problem. Given this variational formulation, we construct an inner product matrix E_k that plays a role analogous to $d_k[x]$ in the differential case. Remarkably, such inner products can be computed exactly for limit functions defined via subdivision, even if these functions are not piecewise polynomial. Next, we derive a matrix relation of the form $\mathsf{E}_k \mathsf{S}_{k-1} == \mathsf{U}_{k-1} \mathsf{E}_{k-1}$, where U_{k-1} is an upsampling matrix. This new relation is a matrix version of the finite difference relation $d_k[x] s_{k-1}[x] == 2 d_{k-1}[x^2]$ that characterized subdivision in the differential case. Solutions S_{k-1} to this matrix equation yield subdivision schemes that converge to minimizers of the original variational problem. To illustrate these ideas, we consider the problem of constructing subdivision schemes for two types of splines with simple variational definitions: natural cubic splines and bounded harmonic splines.

6.1 Inner Products for Stationary Subdivision Schemes

One of the most promising applications of subdivision lies in its use to solve physical problems efficiently. As we saw in Chapters 4 and 5, there is a strong link between partial differential equations and subdivision. Via a finite difference approach, we were able to derive subdivision schemes that converge to the solution to a range of physical problems. Unfortunately, generalizing the difference rules used in the finite difference approach to nonuniform domains is difficult. For nonuniform domains, we suggest an alternate approach: pose the physical problem as an equivalent variational problem and construct a subdivision scheme that converges to the minimizers of this variational problem. Using a finite element approach, we can construct such subdivision schemes for nonuniform domains.

In the univariate case, the starting point for most variational methods is a continuous inner product $\langle p, q\rangle$ of the form $\int_{\Omega} p^{(\mathrm{i})}[x]\, q^{(\mathrm{j})}[x]\, dx$, where Ω is some bounded domain. Given this inner product, an associated variational functional $\mathcal{E}[p]$ is then defined to be $\langle p, p\rangle$. In this framework, a typical variational problem is to compute a function $p[x]$ that minimizes $\mathcal{E}[p]$ subject to a set of interpolation conditions on $p[x]$ (e.g., $p[\mathrm{i}] == \mathrm{b}[\![\mathrm{i}]\!]$ for integers $\mathrm{i} \in \Omega$). This section discusses two constructions useful in solving such variational problems: computing the exact values for various derivatives of a subdivision scheme and computing exact inner products for functions defined using subdivision.

Remarkably, both constructions rely only on the subdivision matrix S associated with a stationary scheme. In particular, they are based on the fact that this subdivision matrix S and its associated row vector of basis functions $\mathcal{N}[x]$ satisfy a matrix refinement relation of the form

$$\mathcal{N}[2x]\mathrm{S} == \mathcal{N}[x]. \tag{6.1}$$

Given this refinement relation, the constructions do not rely on any type of closed form representation (such as piecewise polynomials) for the basis function of $\mathcal{N}[x]$.

6.1.1 Exact Derivatives

Given the subdivision matrix S for a convergent, stationary scheme and its associated vector of basis functions $\mathcal{N}[x]$, we begin by considering the following problem: compute the values of the basis functions in $\mathcal{N}[x]$ taken on the integer grid $\mathbb{Z}$. The result of this computation is an *interpolation matrix* N whose ith row is the value of the vector $\mathcal{N}[x]$ taken at $x == \mathrm{i}$. Given a function $p[x]$ of the form $\mathcal{N}[x]\mathrm{p}$, multiplying the interpolation matrix N by the column vector p yields a column vector Np whose ith entry is the value of $p[x]$ at $x == \mathrm{i}$. For example, the interpolation matrix N for uniform cubic B-splines is a tridiagonal matrix of the form

$$\begin{pmatrix} . & . & . & . & . & . & . \\ . & \frac{2}{3} & \frac{1}{6} & 0 & 0 & 0 & . \\ . & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 & 0 & . \\ . & 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 & . \\ . & 0 & 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & . \\ . & 0 & 0 & 0 & \frac{1}{6} & \frac{2}{3} & . \\ . & . & . & . & . & . & . \end{pmatrix}.$$

The non-zero entries of these rows correspond to the values of a cubic B-spline basis function evaluated at its three interior knots. For polynomial splines, one approach to computing N would be to construct the polynomial pieces for each function in $\mathcal{N}[x]$ using the method of section 2.3.1 and evaluate these pieces directly on $\mathbb{Z}$. Unfortunately, there exist many subdivision schemes, such as the four-point scheme, that do not have an analytic definition as piecewise polynomials. Luckily, there exists an alternative approach that avoids constructing these pieces by using equation 6.1 to derive a recurrence that the interpolation matrix N must satisfy.

As a preliminary to constructing this recurrence, we recall that upsampling a vector p involves constructing a new vector by inserting zeros between the entries of p. If p[x] is the generating function associated with p, the coefficients of the generating function $\mathrm{p}[\mathrm{x}^2]$ are the upsampled version of p. Upsampling can also be modeled via matrix multiplication. If U is the *upsampling matrix* whose entries have the form $\mathrm{U}[\![\mathrm{i}, \mathrm{j}]\!] = 1$ if $\mathrm{i} == 2\mathrm{j}$ and 0 otherwise, the product Up returns the vector p with zero inserted between all of its entries. Note that multiplying the transpose of the upsampling matrix U by a column vector p has the effect of deleting every other entry of p. As a result, this matrix U^{T} is sometimes referred to as the *downsampling matrix*.

Given these two matrices, we are now ready to construct a recurrence involving the interpolation matrix N and the subdivision matrix S. The key to generating this recurrence is to evaluate equation 6.1 at $x == \mathrm{i}$ for all integers i. Each evaluation yields an equation of the form $\mathcal{N}[2\mathrm{i}]\mathrm{S} == \mathcal{N}[\mathrm{i}]$. Accumulating these row vectors $\mathcal{N}[\mathrm{i}]$ for all $\mathrm{i} \in \mathbb{Z}$ forms the interpolation matrix N on the right-hand side of this equation. On the left-hand side, accumulating the vectors $\mathcal{N}[2\mathrm{i}]$ yields a matrix that corresponds to the interpolation matrix N with all of its odd-indexed rows deleted.

This deletion can be modeled by multiplying on the expression NS on the left by the downsampling matrix U^{T}. Therefore,

$$\mathsf{U}^{\mathsf{T}}\mathsf{NS} == \mathsf{N}. \tag{6.2}$$

For example, the interpolation and subdivision matrices for cubic B-splines satisfy a recurrence of the form

$$\begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & 0 & 1 & 0 & 0 & 0 & \cdot \\ \cdot & 0 & 0 & 0 & 0 & 0 & \cdot \\ \cdot & 0 & 0 & 1 & 0 & 0 & \cdot \\ \cdot & 0 & 0 & 0 & 0 & 0 & \cdot \\ \cdot & 0 & 0 & 0 & 1 & 0 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}^{\mathsf{T}} \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \frac{2}{3} & \frac{1}{6} & 0 & 0 & 0 & \cdot \\ \cdot & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 & 0 & \cdot \\ \cdot & 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 & \cdot \\ \cdot & 0 & 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & \cdot \\ \cdot & 0 & 0 & 0 & \frac{1}{6} & \frac{2}{3} & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix} \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & 0 & 0 & \cdot \\ \cdot & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & \cdot \\ \cdot & 0 & \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & 0 & \cdot \\ \cdot & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & \cdot \\ \cdot & 0 & 0 & \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}$$

$$== \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \frac{2}{3} & \frac{1}{6} & 0 & 0 & 0 & \cdot \\ \cdot & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 & 0 & \cdot \\ \cdot & 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 & \cdot \\ \cdot & 0 & 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & \cdot \\ \cdot & 0 & 0 & 0 & \frac{1}{6} & \frac{2}{3} & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}. \tag{6.3}$$

For uniform schemes, the recurrence of equation 6.2 can be expressed very compactly in terms of generating functions. This conversion relies on the fact that the scaling vector $\mathcal{N}[x]$ consists of integer translates of a single scaling function $n[x]$. Thus, the rows of the interpolation matrix N consist of samples of this scaling function taken on the integer grid $\mathbb{Z}$, that is, $\{\ldots, n[2], n[1], n[0], n[-1], n[-2], \ldots\}$. These values can be used to define a generating function $\mathsf{n}[\mathsf{x}]$ of the form $\sum_{\mathsf{i}} n[-\mathsf{i}]\mathsf{x}^{\mathsf{i}}$. The generating function $\mathsf{n}[\mathsf{x}]$ is the *interpolation mask* for the scheme. Observe that multiplying the interpolation mask $\mathsf{n}[\mathsf{x}]$ by the generating function $\mathsf{p}_0[\mathsf{x}]$ yields a generating function $\mathsf{n}[\mathsf{x}]\mathsf{p}_0[\mathsf{x}]$ whose coefficients are the values of the function $\mathcal{N}[x]\mathsf{p}_0$, taken on the integer grid $\mathbb{Z}$.

If $\mathsf{s}[\mathsf{x}]$ is the subdivision mask for the subdivision matrix S, the matrix product NS in equation 6.2 can be modeled by the product $\mathsf{n}[\mathsf{x}]\,\mathsf{s}[\mathsf{x}]$. As observed previously, the action of left multiplication by U^{T} is to select every other coefficient of the generating function $\mathsf{n}[\mathsf{x}]\,\mathsf{s}[\mathsf{x}]$. In terms of generating functions, this action can be modeled by representing n[x] s[x] as the sum of two generating functions: one with only even powers and one with only odd powers. In particular, n[x] s[x] has the form

$$\mathsf{n}[x]\,\mathsf{s}[x] == \mathsf{n}[x^2] + x\,\mathsf{r}[x^2], \tag{6.4}$$

where the function $\mathsf{r}[x^2]$ is an arbitrary residual.

If the scaling function *n*[*x*] is supported over m intervals, the interpolation mask n[x] can be treated as a generating function with m − 1 unknown coefficients *n*[i] that are the value of *n*[*x*] at the m − 1 interior grid points of its support. Equation 6.4 yields a set of m − 1 homogeneous equations in these m − 1 unknowns. For convergent subdivision schemes, these equations are linearly dependent and have a single solution that is unique up to multiplication by a constant. (This observation is due to the fact that the subdivision matrix S for a convergent scheme has an eigenvalue 1 with multiplicity one. See section 8.2.4 for details.) To normalize this solution, we observe that for these convergent schemes the sum of the coefficients *n*[i] must be one (i.e., n[1] == 1). Given a stationary subdivision mask s[x], the associated *Mathematica* implementation computes the associated interpolation mask n[x] (♚). (Note that a similar construction can be used to compute the interpolation mask n[x, y] for a bivariate scheme with a stationary mask s[x, y].)

For example, the subdivision mask for cubic B-splines has the form $\mathsf{s}[x] = \frac{1}{8}x^{-2} + \frac{1}{2}x^{-1} + \frac{3}{4} + \frac{1}{2}x + \frac{1}{8}x^2$, whereas the associated interpolation mask n[x] has the form $\frac{1}{6}x^{-1} + \frac{2}{3} + \frac{1}{6}x$. For this example, equation 6.4 has the form

$$\begin{aligned}&\left(\frac{1}{6}x^{-1} + \frac{2}{3} + \frac{1}{6}x\right)\left(\frac{1}{8}x^{-2} + \frac{1}{2}x^{-1} + \frac{3}{4} + \frac{1}{2}x + \frac{1}{8}x^2\right)\\ &\quad == \left(\frac{1}{6}x^{-2} + \frac{2}{3} + \frac{1}{6}x^2\right) + x\left(\frac{1}{48}x^{-4} + \frac{23}{48}x^{-2} + \frac{23}{48} + \frac{1}{48}x^2\right).\end{aligned}$$

If the subdivision scheme has sufficient smoothness, it is also possible to compute *derivative matrices* for the scheme. These matrices return the exact value of derivatives of scaling functions in the vector $\mathcal{N}[x]$ taken on the grid $\mathbb{Z}$. One method for computing these matrices would be to take the derivative of both sides of equation 6.1. The resulting equation $\mathcal{N}'[2x]\mathsf{S} == \frac{1}{2}\mathcal{N}'[x]$ yields a recurrence involving the values of $\mathcal{N}'[x]$. (Note the extra factor of $\frac{1}{2}$ introduced by the chain rule.)

An alternative approach is to compute a subdivision scheme for the derivatives of the original scheme using the method of section 3.2.4 and compute the interpolation matrix for this derivative scheme. For uniform schemes, this construction is particularly simple. Given a subdivision scheme with mask s[x], the first derivative scheme has a subdivision mask of the form $\frac{2\mathsf{s}[\mathsf{x}]}{1+\mathsf{x}}$. Given the interpolation mask $\mathsf{n}[\mathsf{x}]$ for this derivative scheme, we observe that the derivative mask for the original scheme (on $\mathbb{Z}$) has the form $(1-\mathsf{x})\mathsf{n}[\mathsf{x}]$. The derivative mask on finer grids $\frac{1}{2^k}\mathbb{Z}$ has the form $2^k(1-\mathsf{x})\mathsf{n}[\mathsf{x}]$.

For example, the subdivision mask for the four-point scheme has the form $\mathsf{s}[\mathsf{x}] = -\frac{1}{16}\mathsf{x}^{-3} + \frac{9}{16}\mathsf{x}^{-1} + 1 + \frac{9}{16}\mathsf{x} - \frac{1}{16}\mathsf{x}^3$. Because this scheme is interpolatory, the interpolation mask for this scheme is simply 1. However, due to the fact that the four-point scheme is not piecewise polynomial, computing the exact value of first derivatives of this scheme at integer grid points is nontrivial. Dividing the subdivision mask $\mathsf{s}[\mathsf{x}]$ for the four-point scheme by $\frac{1}{2}(1+\mathsf{x})$ yields the subdivision mask for the first derivative of the four-point scheme, $-\frac{1}{8}\mathsf{x}^{-3} + \frac{1}{8}\mathsf{x}^{-2} + \mathsf{x}^{-1} + 1 + \frac{1}{8}\mathsf{x} - \frac{1}{8}\mathsf{x}^2$. After applying the recurrence of equation 6.4, the interpolation mask for this derivative scheme has the form $-\frac{1}{12}\mathsf{x}^{-2} + \frac{7}{12}\mathsf{x}^{-1} + \frac{7}{12} - \frac{1}{12}\mathsf{x}$. Finally, multiplying this mask by the difference mask $(1-\mathsf{x})$ yields the exact derivative mask for the four-point scheme of the form $-\frac{1}{12}\mathsf{x}^{-2} + \frac{2}{3}\mathsf{x}^{-1} - \frac{2}{3}\mathsf{x} + \frac{1}{12}\mathsf{x}^2$. (Remember that this mask should be multiplied by a factor of 2^k to compute the derivative on $\frac{1}{2^k}\mathbb{Z}$.)

6.1.2 Exact Inner Products

Having derived a method for computing the exact value and derivatives of the scaling functions $\mathcal{N}[x]$, we next derive a similar scheme that computes the exact inner product of two functions defined using subdivision. To begin, we consider a simple inner product $\langle p, q\rangle$ of the form $\int_\Omega p[x]\, q[x]\, dx$, where Ω is some bounded domain. Given a convergent, stationary scheme with a subdivision matrix S, our task is to compute the exact values of this inner product for functions of the form $p[x] = \mathcal{N}[x]\mathsf{p}$ and $q[x] = \mathcal{N}[x]\mathsf{q}$, where the vector $\mathcal{N}[x]$ satisfies the matrix refinement relation of equation 6.1. In particular, we are interested in a method that does not rely on any type of underlying piecewise polynomial definition for $\mathcal{N}[x]$. The key to this method is to compute an *inner product matrix* E of the form

$$\mathsf{E} = \int_\Omega \mathcal{N}[x]^{\mathsf{T}} \mathcal{N}[x]\, dx. \tag{6.5}$$

Note that $\mathcal{N}[x]^{\mathsf{T}}\mathcal{N}[x]$ is a matrix whose ijth entry is the product of the ith and jth functions in the vector $\mathcal{N}[x]$. Therefore, the ijth element of E is the inner product of the ith and jth functions in the vector $\mathcal{N}[x]$. The usefulness of this inner product matrix E lies in the fact that it can be used to compute the exact inner product of two functions $p[x]$ and $q[x]$ defined as $\mathcal{N}[x]\mathsf{p}$ and $\mathcal{N}[x]\mathsf{q}$, respectively. In particular, the inner product $\langle p, q\rangle$ is simply the scalar expression $\mathsf{p}^{T}\mathsf{E}\mathsf{q}$. As in the previous section, our basic approach in computing the inner product matrix E is to derive a recurrence that relates E and the subdivision matrix S. The following theorem characterizes this recurrence.

THEOREM 6.1 Given a convergent subdivision scheme whose subdivision matrix satisfies $\mathcal{N}[2x]\mathsf{S} == \mathcal{N}[x]$, the inner product matrix E of equation 6.5 satisfies

$$\mathsf{S}^{T}\,\mathsf{E}\,\mathsf{S} == 2\mathsf{E}. \tag{6.6}$$

Proof Substituting $\mathcal{N}[2x]\mathsf{S}$ for the subdivision matrix $\mathcal{N}[x]$ in equation 6.5 yields that the inner product matrix E has the form

$$\mathsf{E} == \int_{\Omega} (\mathcal{N}[2x]\mathsf{S})^{T}\,(\mathcal{N}[2x]\mathsf{S})\,dx.$$

Distributing the transpose inside the parentheses and pulling the matrices S^{T} and S outside the integral yields

$$\mathsf{E} == \mathsf{S}^{T}\left(\int_{\Omega} \mathcal{N}[2x]^{T}\mathcal{N}[2x]\,dx\right)\mathsf{S}.$$

To conclude, we apply the change of variable $x \to \frac{1}{2}x$ to the integral $\int_{\Omega}\mathcal{N}[2x]^{T}\mathcal{N}[2x]\,dx$ and observe that it reduces to $\frac{1}{2}\int_{\Omega}\mathcal{N}[x]^{T}\mathcal{N}[x]\,dx$. Because by definition $\int_{\Omega}\mathcal{N}[x]^{T}\mathcal{N}[x]\,dx$ is exactly the inner product matrix E, the theorem follows.

Observe that a similar recurrence of the form $\mathsf{S}^{T}\mathsf{E}\widetilde{\mathsf{S}} == 2\mathsf{E}$ holds when the inner product matrix E of equation 6.5 involves two different sets of scaling functions $\mathcal{N}[x]$ and $\widetilde{\mathcal{N}}[x]$ satisfying the recurrences $\mathcal{N}[2x]\widetilde{\mathsf{S}} == \mathcal{N}[x]$ and $\widetilde{\mathcal{N}}[2x]\widetilde{\mathsf{S}} == \widetilde{\mathcal{N}}[x]$.

Given the subdivision matrix S, our task is to compute the inner product matrix E using equation 6.6. If the matrix S is a finite matrix, this problem can be converted into an eigenvalue problem using direct products. In particular, E can be expressed as a left eigenvector of the direct product of S with itself. (See Chapter 4 of Horn and Johnson [75] or Chapter 8 of Lancaster [92] for an introduction to direct products.) Unfortunately, as currently posed, the matrix S is an infinite matrix. For now, we focus on a method credited to Dahmen and Micchelli [31] for computing this inner product matrix E in the uniform case. Later, in sections 6.2.2 and 6.4.1, we describe a more general method for computing the inner product matrix $\mathtt{E}$ based on linear algebra.

The key to Dahmen and Micchelli's method is to convert equation 6.6 into an equivalent expression in terms of generating functions. In the uniform case, the columns of the inner product matrix $\mathtt{E}$ are all one-shifts of a single column $\mathtt{e}$. The generating function $\mathtt{e}[x]$ associated with this column is the *inner product mask* $\mathtt{e}[x]$ associated with $\mathtt{E}$. If $s[x]$ is the subdivision mask for the subdivision matrix S, the columns of the matrix product $\mathtt{E}S$ are two-shifts of the coefficients of the generating function $\mathtt{e}[x]s[x]$. Multiplying the matrix S^T by $\mathtt{E}S$ yields a new matrix whose columns are two-shifts of the sequence formed by deleting every other coefficient of $s[\frac{1}{x}]\,\mathtt{e}[x]\,s[x]$. (This deletion is due to the two-shifting of the rows of S^T.) Therefore, equation 6.6 can be modeled by an expression of the form

$$s\left[\frac{1}{x}\right]\mathtt{e}[x]\,s[x] == 2\mathtt{e}[x^2] + x\,r[x^2],$$

where $r[x]$ is an arbitrary residual function. At this point, we note that this equation has a striking similarity to equation 6.4. Regrouping the terms on the left-hand side of this equation leads to a new equation of the form

$$\mathtt{e}[x]\left(\frac{1}{2}s\left[\frac{1}{x}\right]s[x]\right) == \mathtt{e}[x^2] + \frac{1}{2}x\,r[x^2].$$

In this view, the inner product mask $\mathtt{e}[x]$ is simply the interpolation mask for the uniform subdivision scheme with mask $\frac{1}{2}s[\frac{1}{x}]\,s[x]$. For example, if $s[x]$ is the subdivision mask for linear B-splines, $\frac{1}{2x}(1+x)^2$, the inner product mask $\mathtt{e}[x]$ is the interpolation mask associated with the subdivision mask for cubic B-splines, $\frac{1}{8x^2}(1+x)^4$. The reader may verify using the piecewise polynomial definition of linear B-splines that the inner product mask for linear B-splines has the form $\mathtt{e}[x] = \frac{1}{6}x^{-1} + \frac{2}{3} + \frac{1}{6}x$.

In many variational applications, one wishes to compute an inner product of the form $\int_\Omega p^{(i)}[x]\,q^{(j)}[x]\,dx$. The same technique used in computing derivative masks may be applied to computing masks for inner products involving derivatives. Given the subdivision mask $s[x]$, we first compute subdivision masks $\frac{2^i s[\frac{1}{x}]}{(1+\frac{1}{x})^i}$ and $\frac{2^j s[x]}{(1+x)^j}$ for the derivative schemes associated with $p^{(i)}[x]$ and $q^{(j)}[x]$. Next, we compute the inner product mask $\mathtt{e}[x]$ for these two derivative schemes. Finally, we multiply the inner product mask $\mathtt{e}[x]$ for the derivative schemes by the difference masks $(1-\frac{1}{x})^i$ and $(1-x)^j$ associated with the various derivatives on $\mathbb{Z}$. The resulting inner product mask corresponds to the original inner product $\int_\Omega p^{(i)}[x]\,q^{(j)}[x]\,dx$. (See the associated *Mathematica* implementation for details on computing such masks (♗).)

6.1.3 Example: Exact Enclosed Area for Parametric Curves

To illustrate the power of this approach, we consider a simple problem involving inner products: computing the area enclosed by a curve defined through uniform subdivision. In particular, our method will return the exact enclosed area even when the curve scheme is not piecewise polynomial. The key idea is to express the area enclosed by a parametric curve of the form $\{p[x], q[x]\}$ in terms of an inner product involving $p[x]$, $q[x]$, and their derivatives. The following theorem, a variant of Green's theorem, provides the necessary integral.

THEOREM 6.2 Given a closed parametric curve $\{p[x], q[x]\}$ that is periodic on the interval $\Omega = [0, 1]$, the signed area of the region enclosed by the curve is

$$\frac{1}{2}\int_0^1 \mathrm{Det}\left[\begin{pmatrix} p[x] & q[x] \\ p'[x] & q'[x] \end{pmatrix}\right] dx. \tag{6.7}$$

Proof The signed area can be approximated by the sum of the areas of m triangles with vertices $\{\{0, 0\}, \{p[\frac{i-1}{m}], q[\frac{i-1}{m}]\}, \{p[\frac{i}{m}], q[\frac{i}{m}]\}\}$, where $1 \le i \le m$. The area of each of these triangles can be expressed as a 3×3 determinant that is easily reduced to a 2×2 determinant:

$$\frac{1}{2}\mathrm{Det}\left[\begin{pmatrix} 1 & 1 & 1 \\ 0 & p\left[\frac{i-1}{m}\right] & q\left[\frac{i-1}{m}\right] \\ 0 & p\left[\frac{i}{m}\right] & q\left[\frac{i}{m}\right] \end{pmatrix}\right] == \frac{1}{2}\mathrm{Det}\left[\begin{pmatrix} p\left[\frac{i-1}{m}\right] & q\left[\frac{i-1}{m}\right] \\ p\left[\frac{i}{m}\right] - p\left[\frac{i-1}{m}\right] & q\left[\frac{i}{m}\right] - q\left[\frac{i-1}{m}\right] \end{pmatrix}\right].$$

Based on this observation, the sum of the areas of all m of these triangles can be expressed as

$$\frac{1}{2m}\sum_{i=1}^{m}\mathrm{Det}\left[\begin{pmatrix} p\left[\frac{i-1}{m}\right] & q\left[\frac{i-1}{m}\right] \\ m\left(p\left[\frac{i}{m}\right] - p\left[\frac{i-1}{m}\right]\right) & m\left(q\left[\frac{i}{m}\right] - q\left[\frac{i-1}{m}\right]\right) \end{pmatrix}\right].$$

Taking the limit of this sum as $m \to \infty$ yields the integral of equation 6.7.

To apply our exact inner product method from the previous section to the area integral of equation 6.7, we express this integral as the difference of two inner products $\frac{1}{2}(\langle p, q'\rangle - \langle p', q\rangle)$. If the functions $p[x]$ and $q[x]$ are constructed using the same subdivision mask $s[x]$, the subdivision mask for $q'[x]$ has the form $\frac{2s[x]}{1+x}$. Therefore, the inner product mask corresponding to $\langle p, q'\rangle$ is the interpolation mask for $s[\frac{1}{x}]\,\frac{s[x]}{(1+x)}$ multiplied by the difference mask $(1-x)$. Likewise, the inner product mask corresponding to $\langle p', q\rangle$ is the interpolation mask for $\frac{s[\frac{1}{x}]}{(1+\frac{1}{x})}\,s[x]$ multiplied by the difference mask $(1-\frac{1}{x})$. Because these two inner product masks are simply the negatives of each other, the corresponding inner products satisfy $\langle p, q'\rangle == -\langle p', q\rangle$. Thus, the area enclosed by the parametric curve $\{p[x], q[x]\}$ is simply the value of the inner product $\langle p, q'\rangle$.

As an example of this computation, we next compute the exact area enclosed by limit curves produced by the four-point scheme. If $s[x]$ is the subdivision mask for the four-point scheme $-\frac{1}{16}x^{-3}+\frac{9}{16}x^{-1}+1+\frac{9}{16}x-\frac{1}{16}x^{3}$, we recall that the subdivision mask for the first derivative of the four-point scheme has the form $-\frac{1}{8}x^{-3}+\frac{1}{8}x^{-2}+x^{-1}+1+\frac{1}{8}x-\frac{1}{8}x^{2}$. Therefore, the exact area mask $e[x]$ for the four-point scheme is the interpolation mask for the product of these two masks multiplied by the difference mask $(1-x)$. In particular, this mask has the form

$$e[x] = \frac{1}{665280x^5}+\frac{4}{10395x^4}-\frac{481}{73920x^3}+\frac{731}{6930x^2}-\frac{3659}{5280x} + \frac{3659x}{5280}-\frac{731x^2}{6930}+\frac{481x^3}{73920}-\frac{4x^4}{10395}-\frac{x^5}{665280}.$$

To compute the enclosed area for a given curve $\{p[x], q[x]\}$, we compute the value of the expression $p^T E q$, where $p[x] = \mathcal{N}[x]p$, $q[x] = \mathcal{N}[x]q$, and E is the inner product matrix corresponding to the mask $e[x]$ given previously. One quick check of the correctness of E is to choose the coefficients p and q such that the associated curve $\{p[x], q[x]\}$ forms an exact unit square. For the four-point scheme, this condition requires the use of five-fold points at each of the corners of the square. (The factor of five is due to the size of the mask associated with the four-point scheme.) The associated implementation verifies that in this case the expression $p^T E q$ evaluates to exactly 1 (♚). Figure 6.1 shows a plot of the area enclosed by the four-point curve associated with a regular n-gon inscribed in a unit circle. Observe that as n increases, the enclosed area rapidly converges to π.

Variants of equation 6.6 also hold for inner products defined on nonuniform domains. The traditional approach of Halstead et al. [71] is to isolate the nonuniformities and break the domain into an infinite union of increasingly small uniform domains. The inner product is then expressed as an infinite sum over these uniform domains. Instead, using equation 6.6, one can simply solve for the appropriate matrix E from the local subdivision rules of S. Section 6.2.2 illustrates an example of this nonuniform approach for natural cubic splines, and section 6.4.1 illustrates this approach for bounded harmonic splines. Other possible future applications of this method include finite elements on subdivision surfaces [22], enclosed volumes of subdivision surfaces [114], and construction of fair surfaces.

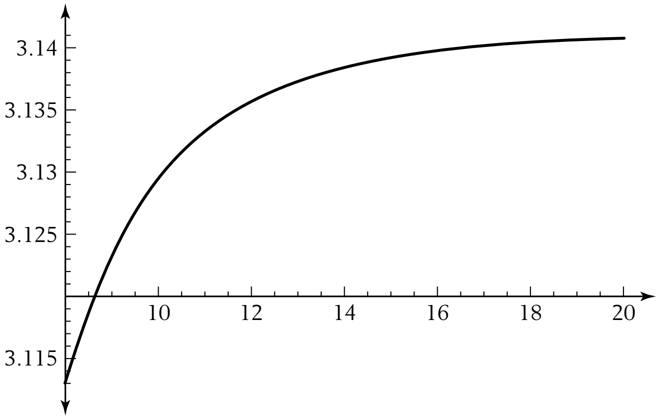


Figure 6.1 The exact area enclosed by four-point curves whose initial control polygon is a regular n-gon.

6.2 Subdivision for Natural Cubic Splines

Chapter 4 considered the problem of subdivision for cubic splines from a differential point of view. Given the differential equation $p^{(4)}[x] == 0$ that governed the segments of a cubic spline $p[x]$, we derived the subdivision mask $s[x]$ for cubic splines as a consequence of the finite difference relation $d_k[x]\,s[x] == 2d_{k-1}[x^2]$, where $d_k[x]$ was the discretization of the differential operator $\mathcal{D}[x]^4$ on the grid $\frac{1}{2^k}\mathbb{Z}$. One major drawback of the differential approach is that it yields little insight into how to generalize the subdivision rules for cubic splines to bounded domains. In this section, we recast cubic splines in a variational framework. This variational approach provides the insight necessary to generalize the subdivision rules for cubic splines to bounded domains.

6.2.1 A Variational Formulation of Cubic Splines

The evolution of geometric design as a discipline traces its roots all the way back to the beginning of the twentieth century. At that time, most drafting was done using a thin, flexible piece of metal known as a spline. The draftsman attached the spline to anchor points on his drafting table. The spline was then allowed to slide through the anchor points and to assume a smooth, minimum energy shape. In the 1940s and 1950s, mathematicians realized that the behavior of a spline could be modeled mathematically.

In the simplest model, the shape of a mechanical spline is described as the graph of a function $p[x]$. To capture the behavior of mechanical splines, we must enforce two conditions on $p[x]$. First, the graph of $p[x]$ must pass through pegs on the drafting table. If these pegs are spaced evenly in the x direction, this condition can be expressed mathematically as

$$p[\mathrm{i}] == \mathrm{b}[\![\mathrm{i}]\!], \tag{6.8}$$

where $\mathrm{b}[\![\mathrm{i}]\!]$ is the height of the ith peg. To capture the fact that the mechanical spline assumes a shape with minimal bending energy, we need to settle on a mathematical model for the bending energy of $p[x]$. Recall that the first derivative of the function, $p^{(1)}[x]$, represents the tangent of the curve p at parameter x. The second derivative of the function, $p^{(2)}[x]$, approximately measures how much the tangents of p change at x. In other words, $p^{(2)}[x]$ measures how much p bends at x. Thus, we can model the bending energy of the function $p[x]$ by computing its second derivative, taking its square, and integrating the result over the interval Ω. This functional $\mathcal{E}[p]$ has the form

$$\mathcal{E}[p] = \int_{\Omega} (p^{(2)}[x])^2 \, dx. \tag{6.9}$$

Now, a *natural cubic spline* is a function $p[x]$ that satisfies equation 6.8 while minimizing the functional $\mathcal{E}$ from equation 6.9. At this point, we note that a natural cubic spline is only an approximation to the shape of a mechanical spline, because the second derivative of $p[x]$ is only a weak approximation of the true curvature of a mechanical spline. Section 3.1 of Hoschek and Lasser [76] gives a more detailed analysis of the relationship between mechanical splines and natural cubic splines.

The Euler-Lagrange theorem, sometimes referred to as the Kuhn-Tucker relations, provides a link between the minimization of a variational functional, such as the functional $\mathcal{E}$ we just introduced for natural cubic splines, and a set of partial differential equations. Roughly speaking, the theorem states that the extremal functions for a variational problem satisfy a set of partial differential equations related to the variational functional. A more detailed explanation of the Euler-Lagrange theorem can be found in [14].

Applying Euler-Lagrange to the variational functional of equation 6.9 leads to the associated differential equation for natural cubic splines:

$$p^{(4)}[x] == 0. \tag{6.10}$$

Natural cubic splines satisfy this differential equation everywhere except at the integer knots in the interval Ω. In particular, a natural cubic spline $p[x]$ is a piecewise cubic function that has continuous second derivatives at the knots (i.e., $p[x] \in \mathcal{C}^2$) and whose second derivative is zero at the endpoints of the interval Ω (the natural boundary conditions).

In the differential approach to subdivision, we discretized equation 6.10 over a sequence of uniform grids $\frac{1}{2^k}\mathbb{Z}$ and constructed an associated sequence of finite difference equations. In the variational approach to subdivision, we construct an inner product matrix E_k corresponding to the functional of equation 6.9 taken on the restriction of the grid $\frac{1}{2^k}\mathbb{Z}$ to the interval Ω. As we shall see, the minimizers of this functional are captured by a matrix equation analogous to the finite difference equation $d_k[x]\,s[x] == 2d_{k-1}[x^2]$. The authors first published this method for constructing minimizers of variational problems in [160].

6.2.2 A Finite Element Scheme for Natural Cubic Splines

One standard approach to solving variational problems of this type is the *finite element* method. Given an interval Ω, the finite element method first constructs a set of finite element basis functions $\widetilde{\mathcal{N}}_k[x]$ defined on this interval whose members are indexed by points on the $\frac{1}{2^k}\mathbb{Z}$ grid. The basis functions comprising the vector $\widetilde{\mathcal{N}}_k[x]$ are chosen to be sufficiently smooth so that the variational functional $\mathcal{E}$ is well defined for any function $p_k[x]$ in the span of $\widetilde{\mathcal{N}}_k[x]$. Next, the finite element method computes the function $p_k[x]$ of the form $\widetilde{\mathcal{N}}_k[x]\mathrm{p}_k$ that minimizes the variational functional $\mathcal{E}$ while satisfying the interpolation constraints $p_k[i] == b[\![i]\!]$ for all integers $i \in \Omega$. For functionals such as equation 6.9 that are quadratic in the function $p[x]$ and its derivatives, this minimization problem reduces to solving a system of linear equations involving p_k. These linear equations can be viewed as a discrete approximation to the partial differential equation associated with the variational functional $\mathcal{E}$ by the Euler-Lagrange theorem.

The payoff from this construction is that under fairly weak conditions on the finite element basis $\widetilde{\mathcal{N}}_k[x]$ the limit of the minimizing functions $p_k[x]$ as $k \to \infty$ is the function $p_\infty[x]$ that minimizes $\mathcal{E}$ over the *space of all functions* for which $\mathcal{E}$ is well defined. Thus, the finite element method is simply a multilevel method for computing a sequence of vectors p_k that converge to the minimizer of the variational functional $\mathcal{E}$. For those readers interested in a more rigorous treatment of the finite element method, we recommend Oden and Reddy [113].

In the finite element method, the key step in constructing the linear equations that characterize the minimizing vector p_k is building the inner product matrix E_k associated with the variational functional $\mathcal{E}$. For natural cubic splines, the continuous inner product associated with the variational functional $\mathcal{E}$ of equation 6.9 has the form

$$\langle p, q\rangle = \int_{\Omega} p^{(2)}[x]\, q^{(2)}[x]\, dx. \tag{6.11}$$

If the functions $p[x]$ and $q[x]$ are expressed in terms of the finite element basis functions at level k (i.e., $p[x] = \widetilde{\mathcal{N}}_{\mathsf{k}}[x]\mathsf{p}_{\mathsf{k}}$ and $q[x] = \widetilde{\mathcal{N}}_{\mathsf{k}}[x]\mathsf{q}_{\mathsf{k}}$), the continuous inner product $\langle p, q\rangle$ given in equation 6.11 can be expressed as a discrete inner product of the form $\mathsf{p}_{\mathsf{k}}^{\mathsf{T}}\mathsf{E}_{\mathsf{k}}\mathsf{q}_{\mathsf{k}}$, where

$$\mathsf{E}_{\mathsf{k}} = \int_{\Omega} \widetilde{\mathcal{N}}_{\mathsf{k}}^{(2)}[x]^{\mathsf{T}}\, \widetilde{\mathcal{N}}_{\mathsf{k}}^{(2)}[x]\, dx.$$

In particular, the ijth element of E_{k} is the continuous inner product (defined in equation 6.11) of the ith and jth basis functions in $\widetilde{\mathcal{N}}_{\mathsf{k}}[x]$.

If the finite element basis functions in $\widetilde{\mathcal{N}}_{\mathsf{k}-1}[x]$ can be expressed as a linear combination of finite element basis functions in $\widetilde{\mathcal{N}}_{\mathsf{k}}[x]$, these basis functions satisfy a matrix refinement relation of the form $\widetilde{\mathcal{N}}_{\mathsf{k}}[x]\widetilde{\mathsf{S}}_{\mathsf{k}-1} == \widetilde{\mathcal{N}}_{\mathsf{k}-1}[x]$, where $\widetilde{\mathsf{S}}_{\mathsf{k}-1}$ is the subdivision matrix associated with this finite element scheme. For finite element bases of this form, the inner product matrices $\mathsf{E}_{\mathsf{k}-1}$ and E_{k} satisfy a recurrence similar to that of equation 6.6; that is,

$$\widetilde{\mathsf{S}}_{\mathsf{k}-1}^{\mathsf{T}}\mathsf{E}_{\mathsf{k}}\widetilde{\mathsf{S}}_{\mathsf{k}-1} == \mathsf{E}_{\mathsf{k}-1}. \tag{6.12}$$

To prove this recurrence relation, simply replace $\widetilde{\mathcal{N}}_{\mathsf{k}-1}[x]$ by $\widetilde{\mathcal{N}}_{\mathsf{k}}[x]\widetilde{\mathsf{S}}_{\mathsf{k}-1}$ in the definition of $\mathsf{E}_{\mathsf{k}-1}$ and simplify the resulting expression.

Given a finite interval Ω, our task for the remainder of this section is to choose an appropriate finite element basis $\widetilde{\mathcal{N}}_{\mathsf{k}}[x]$ and compute its corresponding inner product matrix E_{k} for the inner product of equation 6.11. To simplify our effort, we compute these inner product matrices E_{k} for the unbounded interval $\Omega = [0, \infty]$. The advantage of this particular choice of domain is that equation 6.12 reduces to the stationary case of the previous section and allows us to compute a single inner product matrix E. The rows of this inner product matrix E include the inner product rules for uniform grids as well as special inner product rules for the endpoint of an interval. Given these rules, the inner product matrix E_{k} for an arbitrary interval can be constructed without any further difficulty.

For the sake of simplicity, we choose the finite element basis functions in $\widetilde{\mathcal{N}}_{\mathsf{k}}[x]$ to be of as low a degree as possible. In this case, these basis functions are the restrictions of the uniform quadratic B-splines to the interval $[0, \infty]$. Because

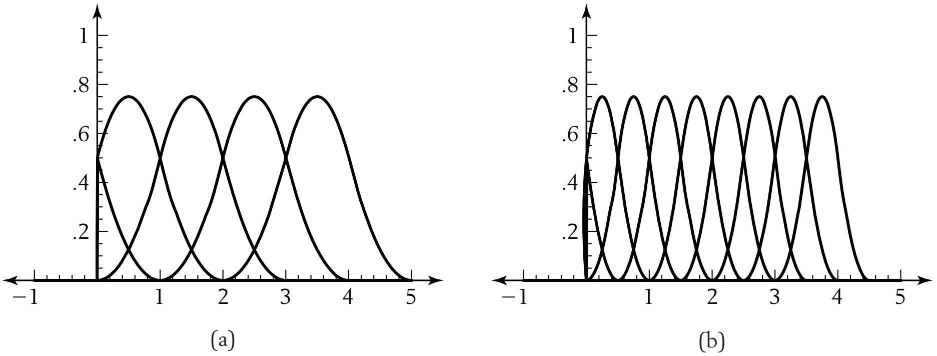


Figure 6.2 Plot of the finite element basis functions in (a) $\widetilde{\mathcal{N}}[x]$ and (b) its dilate $\widetilde{\mathcal{N}}[2x]$.

these basis functions are $\mathcal{C}^1$ piecewise quadratic functions, the inner product $\langle p, q\rangle$ of equation 6.11 is always guaranteed to be well defined for finite linear combinations of such functions.

If $\widetilde{n}[x]$ is the uniform quadratic basis function supported on the interval $[-2, 1]$, we define the vector $\widetilde{\mathcal{N}}[x]$ to consist of the restriction of integer translates of $\widetilde{n}[x]$ to the interval $[0, \infty]$. Because functions of the form $\widetilde{n}[x-\text{i}]$, where $\text{i} < 0$, are identically zero on $[0, \infty]$, we omit these functions from the basis vector $\widetilde{\mathcal{N}}[x]$. Therefore, the basis vector $\widetilde{\mathcal{N}}[x]$ has the form $\{\widetilde{n}[x], \widetilde{n}[x-1], \widetilde{n}[x-2], \widetilde{n}[x-3], \ldots\}$, with all functions being truncated at $x == 0$. Figure 6.2 shows a plot of the basis functions comprising the vector $\widetilde{\mathcal{N}}[x]$ and its dilate $\widetilde{\mathcal{N}}[2x]$.

Due to the choice of the domain $\Omega = [0, \infty]$ and the structure of the vector $\widetilde{\mathcal{N}}[x]$, the finite element basis functions at level k, $\widetilde{\mathcal{N}}_{\text{k}}[x]$, are simply dilates of $\widetilde{\mathcal{N}}[x]$ (i.e., $\widetilde{\mathcal{N}}_{\text{k}}[x] = \widetilde{\mathcal{N}}[2^{\text{k}}x]$). Given this definition, the basis functions $\widetilde{\mathcal{N}}[x]$ satisfy a matrix refinement relation of the form $\widetilde{\mathcal{N}}[2x]\widetilde{\text{S}} == \widetilde{\mathcal{N}}[x]$, where $\widetilde{\text{S}}$ is a subdivision matrix of the form

$$\widetilde{\text{S}} = \begin{pmatrix} \frac{3}{4} & \frac{1}{4} & 0 & 0 & . \\ \frac{1}{4} & \frac{3}{4} & 0 & 0 & . \\ 0 & \frac{3}{4} & \frac{1}{4} & 0 & . \\ 0 & \frac{1}{4} & \frac{3}{4} & 0 & . \\ 0 & 0 & \frac{3}{4} & \frac{1}{4} & . \\ 0 & 0 & \frac{1}{4} & \frac{3}{4} & . \\ . & . & . & . & . \end{pmatrix}.$$

Observe that this subdivision matrix is a submatrix of the bi-infinite subdivision matrix for uniform quadratic B-splines. Because the preceding matrix $\widetilde{\mathsf{S}}$ is independent of k, the subdivision scheme for this finite element basis is actually a stationary process.

Recall that by definition the ijth entry of E_k corresponds to the inner product of the ith and jth entries of $\widetilde{\mathcal{N}}[2^k x]$. Using a simple change of variables, it is easy to show that the inner product matrices E_k are all constant multiples of a single inner product matrix $\mathsf{E} = \mathsf{E}_0$ whose ijth entry is the inner product of the ith and jth entries of $\widetilde{\mathcal{N}}[x]$. In particular, the inner product matrices E_k satisfy the relation $\mathsf{E}_k = 8^k \mathsf{E}$. The constant 8 is the product of a factor of 4^2, corresponding to the effect of halving of the grid spacing on the second derivative squared, multiplied by a factor of $\frac{1}{2}$, arising from the effect of the change of variables on integration. Due to equation 6.12, the inner product matrices E_k also satisfy the recurrence $\widetilde{\mathsf{S}}^T \mathsf{E}_k \widetilde{\mathsf{S}} == \mathsf{E}_{k-1}$. Therefore, the inner product matrix E and the subdivision matrix $\widetilde{\mathsf{S}}$ for our finite element scheme satisfy the recurrence

$$\widetilde{\mathsf{S}}^T \mathsf{E} \widetilde{\mathsf{S}} == \frac{1}{8}\mathsf{E}. \tag{6.13}$$

The factor of $\frac{1}{8}$ (in place of 2) arises from the use of second derivatives in the inner product. Treating E as a matrix of unknowns while enforcing symmetry and sparseness conditions due to the structure of $\widetilde{\mathcal{N}}[x]$ yields a symbolic inner product matrix of the form

$$\mathsf{E} = \begin{pmatrix} e[-3] & e[-2] & e[2] & 0 & 0 & 0 & . \\ e[-2] & e[-1] & e[1] & e[2] & 0 & 0 & . \\ e[2] & e[1] & e[0] & e[1] & e[2] & 0 & . \\ 0 & e[2] & e[1] & e[0] & e[1] & e[2] & . \\ 0 & 0 & e[2] & e[1] & e[0] & e[1] & . \\ 0 & 0 & 0 & e[2] & e[1] & e[0] & . \\ . & . & . & . & . & . & . \end{pmatrix}.$$

The unknowns e[i] with negative index capture the special inner product rules at the origin, whereas the unknowns e[i] with non-negative index capture the inner product rule for uniform quadratic B-splines. In particular, the unknowns e[i], where $i \geq 0$, are the inner product of two uniform basis functions $\widetilde{n}[x]$ shifted by a relative difference of Abs[i] units on the integer grid. Based on the method of section 6.1.2, we observe that these unknowns have the values e[0] == 6, e[1] == −4, and e[2] == 1.

To solve for the remaining unknowns, we substitute the definitions of $\widetilde{\mathsf{S}}$ and E into the equation $\widetilde{\mathsf{S}}^T \mathsf{E} \widetilde{\mathsf{S}} == \frac{1}{8}\mathsf{E}$. This relation yields a set of homogeneous equations in the unknowns $e[i]$. Solving for the remaining $e[i]$ yields

$$E = \begin{pmatrix} 1 & -2 & 1 & 0 & 0 & 0 & . \\ -2 & 5 & -4 & 1 & 0 & 0 & . \\ 1 & -4 & 6 & -4 & 1 & 0 & . \\ 0 & 1 & -4 & 6 & -4 & 1 & . \\ 0 & 0 & 1 & -4 & 6 & -4 & . \\ 0 & 0 & 0 & 1 & -4 & 6 & . \\ . & . & . & . & . & . & . \end{pmatrix}. \qquad (6.14)$$

Given this inner product matrix E, we can construct inner product matrices E_k for finite grids, as described previously. Having these matrices, we could then follow the finite element approach and attempt to minimize the functional $\mathcal{E}$ subject to the interpolation conditions. Instead, we ignore the interpolation conditions and focus on constructing a variational analog of the finite difference equation $d_k[x]\, s_{k-1}[x] == 2d_{k-1}[x^2]$. This analog will lead directly to a matrix equation involving the inner product matrices E_k that characterizes a sequence of subdivision matrices S_{k-1} whose associated scheme converges to the minimizing function of the original variational problem.

6.2.3 A Multiscale Relation for Natural Cubic Splines

Given the inner product matrices E_k, we can now develop a set of linear equations that relate these inner product matrices and the subdivision matrices S_k for natural cubic splines. *Important note:* The matrices $\widetilde{S}_k$ (with a tilde) are the subdivision matrices for the finite element basis $\widetilde{\mathcal{N}}_k[x]$, whereas the matrices S_k (with no tilde) are the subdivision matrices for natural cubic splines. These matrices S_k define an associated basis vector $\mathcal{N}_k[x]$ for natural cubic splines on the grid $\frac{1}{2^k}\mathbb{Z}$. The distinction between these quantities is very important in the analysis that follows.

To generate these subdivision matrices S_k from the inner product matrices E_k, we return to the differential case for guidance. The subdivision scheme for cubic splines constructed generating functions $p_k[x]$ that were solutions to the multiscale finite difference equation

$$d_k[x]\, p_k[x] == 2^k d_0\left[x^{2^k}\right] p_0\left[x^{2^k}\right]. \qquad (6.15)$$

The beauty of this finite difference approach is that successive solutions of this scheme, $p_{k-1}[x]$ and $p_k[x]$, are related by a subdivision relation of the form $p_k[x] = s_{k-1}[x]\, p_{k-1}[x^2]$, where the subdivision mask $s_{k-1}[x]$ satisfies $d_k[x] s_{k-1}[x] == 2d_{k-1}[x^2]$. Our goal in this section is to derive an analogous matrix equation for the variational form of the problem. In the following section, we will show that solutions to this new equation minimize the original variational functional.

Assuming that the matrix E_k is the correct analog of the difference mask $d_k[x]$, we are left with deriving the correct matrix analog of $d_0[x^{2^k}]$. Note that the effect of replacing x by x^2 in a generating function (i.e., substituting $p[x] \to p[x^2]$) is to spread the coefficients of the generating function and to insert zeros between them (i.e., upsample the vector p). In matrix terms, upsampling the vector p from the coarse grid $\frac{1}{2^{k-1}}\mathbb{Z}$ onto the fine grid $\frac{1}{2^k}\mathbb{Z}$ can be modeled via multiplication by the upsampling matrix U_{k-1} whose ijth entry is 1 if $i == 2j$ and 0 otherwise. (Note that the ith row of U_{k-1} is indexed by the grid point $\frac{i}{2^k} \in \Omega$ and the jth column of U_{k-1} is indexed by the grid point $\frac{j}{2^{k-1}} \in \Omega$.) Based on this observation, the matrix analog of $d_0[x^2]$ is $U_0 E_0$. More generally, the matrix analog of $d_0[x^{2^k}]$ is the product of E_0 multiplied by a sequence of upsampling matrices $U_0, U_1, \ldots, U_{k-1}$. Consequently, the variational matrix analog of equation 6.15 is

$$E_k p_k == U_{k-1} \cdots U_0 E_0 p_0. \tag{6.16}$$

(Note that the extra power of 2^k in equation 6.15 is absorbed into the matrix E_k as a constant of integration for the grid $\frac{1}{2^k}\mathbb{Z}$.)

Equation 6.16 relates a vector p_k defined on the fine grid $\frac{1}{2^k}\mathbb{Z}$, with the vector p_0 defined on the coarse grid $\mathbb{Z}$. As in the differential case, our goal is to construct a sequence of subdivision matrices S_{k-1} that relates successive solutions to equation 6.16. These subdivision matrices S_{k-1} satisfy the following relation involving the inner product matrices E_{k-1} and E_k at successive scales.

THEOREM 6.3 For all $k > 0$, let S_k be a sequence of matrices satisfying the two-scale relation

$$E_k S_{k-1} == U_{k-1} E_{k-1}. \tag{6.17}$$

If the vectors p_k satisfy the subdivision relation $p_k = S_{k-1} p_{k-1}$ for all $k > 0$, then p_k and p_0 satisfy the multiscale relation of equation 6.16.

Proof The proof proceeds by induction. For $k = 1$, the theorem follows trivially. More generally, assume that $E_{k-1} p_{k-1} == U_{k-2} \cdots U_0 E_0 p_0$ holds by the inductive hypothesis. Multiplying both sides of this equation by U_{k-1} and replacing $U_{k-1} E_{k-1}$ by $E_k S_{k-1}$ on the left-hand side yields the desired result.

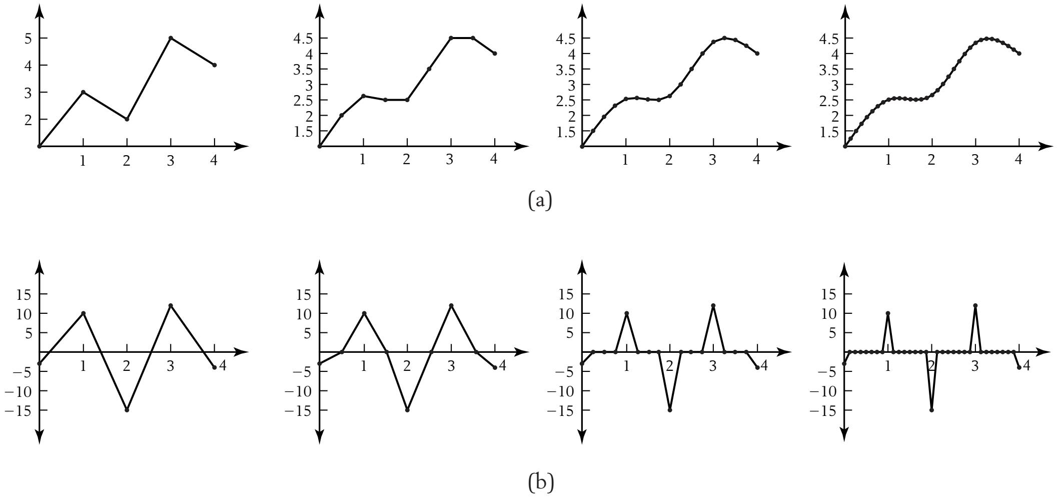
A subdivision scheme satisfying equation 6.17 requires a certain commutativity relationship to hold among the inner product matrices, subdivision matrices, and upsampling matrices. This relation has an interesting interpretation in terms of the Euler-Lagrange theorem. Recall that given a variational functional $\mathcal{E}$ this theorem constructs a partial differential equation whose solutions are extremal functions for $\mathcal{E}$. The discrete version of the variational functional $\mathcal{E}[p]$ on the grid $\frac{1}{2^k}\mathbb{Z}$ is the expression $p_k^T E_k p_k$, where E_k is the inner product matrix associated with $\mathcal{E}$. Now, observe that this quadratic function in the entries of the vector p_k is minimized by exactly those vectors p_k satisfying $E_k p_k == 0$. In other words, the linear equation $E_k p_k == 0$ can be viewed as the discretized version of the partial differential equation corresponding to $\mathcal{E}$.

If the products $E_{k-1} p_{k-1}$ and $E_k p_k$ are viewed as taking discrete differences of the vectors p_{k-1} and p_k corresponding to this partial difference equation, equation 6.17 states that differencing the vector p_{k-1} over the coarse grid $\frac{1}{2^{k-1}}\mathbb{Z}$ and then upsampling those differences to the next finer grid $\frac{1}{2^k}\mathbb{Z}$ should be the same as subdividing the vector p_{k-1} using the subdivision matrix S_{k-1} and then differencing those coefficients with respect to the finer grid $\frac{1}{2^k}\mathbb{Z}$ using the matrix E_k. Figure 6.3 illustrates this commutativity constraint in a diagram, stating that the two paths from the upper left to the lower right should yield the same result (i.e., the same differences over the fine grid $\frac{1}{2^k}\mathbb{Z}$).

The effect of several rounds of subdivision with these matrices S_k is to define a sequence of vectors p_k whose differences $E_k p_k$ replicate the differences $E_0 p_0$ on $\mathbb{Z}$ and are zero on the remaining grid points in $\frac{1}{2^k}\mathbb{Z}$. As a result, the limiting function

$$\begin{tikzcd}
p_{k-1} \arrow[r, "\text{Difference}"'] \arrow[d, "\text{Subdivide}"'] & E_{k-1}p_{k-1} \arrow[d, "\text{Upsample}"] \\
S_{k-1}p_{k-1} \arrow[r, "\text{Difference}"'] & E_k S_{k-1} p_{k-1} == U_k E_{k-1} p_{k-1}
\end{tikzcd}$$

Figure 6.3 The two-scale relation requires commutativity between inner products, upsampling, and subdivision.



(a)

(b)

Figure 6.4 A sequence of polygons p_k defined by natural cubic subdivision (a) and plots of the differences $E_k p_k$ (b).

$p_\infty[x]$ satisfies the partial differential equation associated with the variational functional everywhere except on the grid $\mathbb{Z}$. Figure 6.4 illustrates the behavior of these differences using the subdivision rules for natural cubic splines derived in the next section. The top row of plots shows the effects of three rounds of subdivision. The bottom row of plots shows the differences $E_k p_k$ for $k == 0 \ldots 3$.

6.2.4 Subdivision Rules for Natural Cubic Splines

Given equation 6.17, our approach is to use this relation in computing the subdivision matrix S_{k-1}. In the next section, we show that the limit functions produced by the resulting scheme are minimizers of the variational functional $\mathcal{E}[p]$. We proceed by noting that equation 6.17 does not uniquely determine the subdivision matrix S_{k-1}, because the inner product matrix E_k has a non-empty null space and thus is not invertible. For the inner product of equation 6.11, the differences $E_k p_k$ are zero when p_k is the constant vector or the linearly varying vector $p_k\llbracket i \rrbracket = i$. This observation is consistent with the fact that the continuous variational functional $\mathcal{E}$ yields zero energy for linear functions. Instead of attempting to invert E_k, we focus on computing a simple sequence of subdivision matrices S_{k-1} that satisfies equation 6.17 and defines a convergent subdivision scheme.

As done in computing the inner product matrices, our approach is first to compute the stationary subdivision matrix $\mathtt{S}$ for the domain $\Omega = [0, \infty]$. This matrix encodes the subdivision rules for the endpoint of a natural cubic spline and can be used to construct the subdivision matrices $\mathtt{S}_{k-1}$ for any other finite intervals. The main advantage of choosing this domain is that due to the inner product matrices $\mathtt{E}_k$ satisfying $\mathtt{E}_k == 8^k\mathtt{E}$ equation 6.17 reduces to a stationary equation of the form $8\mathtt{ES} == \mathtt{UE}$. The factor of 8 in this equation arises from integrating the product of second derivatives. If we assume the subdivision matrix $\mathtt{S}$ to be a uniform two-slanted matrix sufficiently far from the endpoint of $x == 0$, the equation $8\mathtt{ES} == \mathtt{UE}$ has the matrix form

$$8\begin{pmatrix} 1 & -2 & 1 & 0 & 0 & 0 & 0 & . \\ -2 & 5 & -4 & 1 & 0 & 0 & 0 & . \\ 1 & -4 & 6 & -4 & 1 & 0 & 0 & . \\ 0 & 1 & -4 & 6 & -4 & 1 & 0 & . \\ 0 & 0 & 1 & -4 & 6 & -4 & 1 & . \\ . & . & . & . & . & . & . & . \end{pmatrix}\begin{pmatrix} \mathtt{s}[-4] & \mathtt{s}[-3] & 0 & . \\ \mathtt{s}[-2] & \mathtt{s}[-1] & 0 & . \\ \mathtt{s}[2] & \mathtt{s}[0] & \mathtt{s}[2] & . \\ 0 & \mathtt{s}[1] & \mathtt{s}[1] & . \\ 0 & \mathtt{s}[2] & \mathtt{s}[0] & . \\ 0 & 0 & \mathtt{s}[1] & . \\ 0 & 0 & \mathtt{s}[2] & . \\ . & . & . & . \end{pmatrix}$$

$$== \begin{pmatrix} 1 & 0 & 0 & . \\ 0 & 0 & 0 & . \\ 0 & 1 & 0 & . \\ 0 & 0 & 0 & . \\ 0 & 0 & 1 & . \\ . & . & . & . \end{pmatrix}\begin{pmatrix} 1 & -2 & 1 & . \\ -2 & 5 & -4 & . \\ 1 & -4 & 6 & . \\ . & . & . & . \end{pmatrix}.$$

The unknowns $\mathtt{s[i]}$ with negative index encapsulate the boundary behavior of $\mathtt{S}$, whereas the unknowns $\mathtt{s[i]}$ with non-negative index encapsulate the uniform subdivision rules for cubic B-splines. Solving for these unknowns yields a subdivision matrix $\mathtt{S}$ for the domain $\Omega = [0, \infty]$ of the form

$$\mathtt{S} = \begin{pmatrix} 1 & 0 & 0 & . \\ \frac{1}{2} & \frac{1}{2} & 0 & . \\ \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & . \\ 0 & \frac{1}{2} & \frac{1}{2} & . \\ 0 & \frac{1}{8} & \frac{3}{4} & . \\ 0 & 0 & \frac{1}{2} & . \\ . & . & . & . \end{pmatrix}.$$

Note that the interior rows of S agree with the subdivision rules for uniform cubic B-splines with the subdivision rule at the endpoint forcing interpolation. In fact, the subdivision matrices S_k for an arbitrary interval Ω with integer endpoints have the same structure: the first and last row in the matrix are standard unit vectors, and all remaining rows are shifts of the fundamental sequences $\{\frac{1}{2}, \frac{1}{2}\}$ and $\{\frac{1}{8}, \frac{3}{4}, \frac{1}{8}\}$. The behavior of the scheme at the endpoints of this interval can be analyzed by constructing the explicit piecewise cubic representation for $\mathcal{N}[x]$. If $\Omega = [0, 4]$, the five basis functions comprising $\mathcal{N}[x]$ can be written in piecewise cubic Bézier form as follows:

$$\begin{array}{ccccccccccccc}
0 & \frac{1}{3} & \frac{2}{3} & 1 & \frac{4}{3} & \frac{5}{3} & 2 & \frac{7}{3} & \frac{8}{3} & 3 & \frac{10}{3} & \frac{11}{3} & 4 \\
\star & & & \star & & & \star & & & \star & & & \star \\
1 & \frac{2}{3} & \frac{1}{3} & \frac{1}{6} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{1}{3} & \frac{2}{3} & \frac{2}{3} & \frac{2}{3} & \frac{1}{3} & \frac{1}{6} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{1}{6} & \frac{1}{3} & \frac{2}{3} & \frac{2}{3} & \frac{2}{3} & \frac{1}{3} & \frac{1}{6} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{6} & \frac{1}{3} & \frac{2}{3} & \frac{2}{3} & \frac{2}{3} & \frac{1}{3} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{6} & \frac{1}{3} & \frac{2}{3} & 1
\end{array}.$$

The first row consists of the x coordinates for the control points of the four Bézier segments as x varies from 0 to 4. The last five rows are the corresponding Bézier coefficients of the five basis functions, each row consisting of four consecutive cubic Bézier functions. The reader can verify that these basis functions are consistent with the subdivision rules given previously. Note that the middle basis function is the uniform cubic basis function. A plot of these basis functions is shown in Figure 6.5.

This representation confirms that the basis functions are $\mathcal{C}^2$ piecewise cubics and satisfy the natural boundary conditions (i.e., their second derivatives at 0 and 4 are zero). Given the subdivision matrices S_k and an initial set of coefficients p_0, the scheme can be used to define increasingly dense sets of coefficients p_k via the relation $\mathsf{p}_k = \mathsf{S}_{k-1}\,\mathsf{p}_{k-1}$. If we interpolate the coefficients of p_k with the piecewise linear function $p_k[x]$ satisfying $p_k[\frac{i}{2^k}] = \mathsf{p}_k[\![i]\!]$, the functions $p_k[x]$ uniformly converge to a piecewise cubic function $p_\infty[x] \in \mathcal{C}^2$. (See [157] for a matrix version of the convergence analysis of Chapter 3.)

Given an initial set of coefficients p_0, applying the subdivision scheme yields increasingly dense approximations of the natural cubic spline. Figure 6.6 shows three rounds of this subdivision procedure applied to the initial coefficients p_0

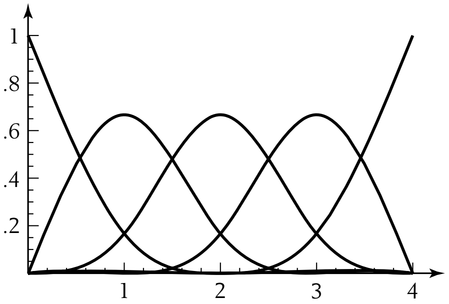


Figure 6.5 Plots of the five basis functions comprising the basis vector $\mathcal{N}_0[x]$ for the domain $\Omega = [0, 4]$.

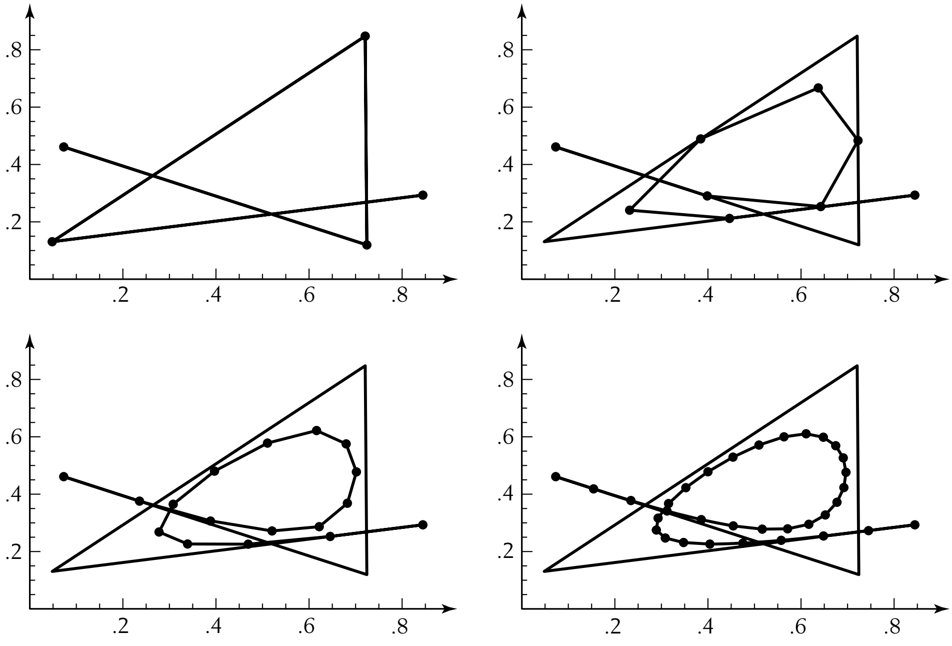


Figure 6.6 Progression of the natural cubic spline subdivision process starting from the initial control polygon (upper left).

(upper left). Note that here the scheme is applied parametrically; that is, the control points are points in two dimensions, and subdivision is applied to both coordinates to obtain the next denser shape.

6.3 Minimization of the Variational Scheme

The previous section constructed a relation of the form $\mathsf{E}_k \mathsf{S}_{k-1} == \mathsf{U}_{k-1} \mathsf{E}_{k-1}$ that models the finite difference relation $\mathsf{d}_k[\mathsf{x}]\mathsf{s}_{k-1}[\mathsf{x}] == 2\mathsf{d}_{k-1}[\mathsf{x}^2]$ arising from the differential method. However, we offered no formal proof that subdivision schemes constructed using the differential method converged to solutions to the differential equation. Instead, we simply analyzed the convergence and smoothness of the resulting schemes. In this section, we show that subdivision matrices satisfying equation 6.17 define a subdivision scheme whose limit functions $p_\infty[x]$ are minimizers of $\mathcal{E}[p]$ in the following sense: over the space of all functions (with well-defined energy) interpolating a fixed set of values on $\Omega \cap \mathbb{Z}$, the limit function $p_\infty[x]$ produced by this subdivision scheme has minimal energy.

Our approach in proving this minimization involves three steps. First, we consider the problem of enforcing interpolation for natural cubic splines using interpolation matrices. Next, we show that these interpolation matrices can also be used to compute the exact value of the variational functional $\mathcal{E}[\mathsf{p}_\infty]$. Finally, we set up a multiresolution framework based on the inner product associated with $\mathcal{E}$ and prove that $p_\infty[x]$ is a minimizer of $\mathcal{E}$ in this framework.

6.3.1 Interpolation with Natural Cubic Splines

Due to the linearity of the subdivision process, any limit function $p_\infty[x]$ can be written as the product of the row vector of basis functions $\mathcal{N}_0[x]$ associated with the subdivision scheme and a column vector p_0; that is, $p_\infty[x] = \mathcal{N}_0[x]\mathsf{p}_0$. Remember that $\mathcal{N}_0[x]$ represents the vector of basis functions associated with the variational subdivision scheme; $\widetilde{\mathcal{N}}_0[x]$ is the finite element basis used in constructing the inner product matrix E_0. Due to the convergence of the subdivision scheme for natural cubic splines, the continuous basis $\mathcal{N}_0[x]$ can be evaluated on the grid $\mathbb{Z}$ (restricted to Ω) to form the interpolation matrix N_0.

In the stationary case where $\Omega = [0, \infty]$, all of the interpolation matrices N_k agree with a single interpolation matrix N that satisfies the recurrence of equation 6.2 (i.e., $\mathsf{U}^T \mathsf{N}\,\mathsf{S} == \mathsf{N}$). This matrix N is tridiagonal because the natural cubic spline basis functions are supported over four consecutive intervals. If the non-zero entries of N are treated as unknowns, for natural cubic splines this recurrence has the matrix form

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & . \\ 0 & 0 & 1 & 0 & 0 & . \\ 0 & 0 & 0 & 0 & 1 & . \\ . & . & . & . & . & . \end{pmatrix} \begin{pmatrix} \mathsf{n}[-2] & \mathsf{n}[-1] & 0 & 0 & 0 & . \\ \mathsf{n}[1] & \mathsf{n}[0] & \mathsf{n}[1] & 0 & 0 & . \\ 0 & \mathsf{n}[1] & \mathsf{n}[0] & \mathsf{n}[1] & 0 & . \\ 0 & 0 & \mathsf{n}[1] & \mathsf{n}[0] & \mathsf{n}[1] & . \\ 0 & 0 & 0 & \mathsf{n}[1] & \mathsf{n}[0] & . \\ . & . & . & . & . & . \end{pmatrix} \begin{pmatrix} 1 & 0 & . \\ \frac{1}{2} & \frac{1}{2} & . \\ \frac{1}{8} & \frac{3}{4} & . \\ 0 & \frac{1}{2} & . \\ 0 & \frac{1}{8} & . \\ . & . & . \end{pmatrix}$$

$$== \begin{pmatrix} \mathsf{n}[-2] & \mathsf{n}[-1] & . \\ \mathsf{n}[1] & \mathsf{n}[0] & . \\ 0 & \mathsf{n}[1] & . \\ . & . & . \end{pmatrix}.$$

Again, the unknowns $\mathsf{n}[\mathsf{i}]$ with negative index encapsulate the boundary behavior of the natural cubic splines, whereas the unknowns $\mathsf{n}[\mathsf{i}]$ with non-negative index encapsulate the interpolation mask for cubic B-splines. After further constraining the rows of the interpolation matrix N to sum to one, there exists a unique solution for the interpolation matrix N of the form

$$\mathsf{N} = \begin{pmatrix} 1 & 0 & 0 & 0 & . \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 & . \\ 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & . \\ 0 & 0 & \frac{1}{6} & \frac{2}{3} & . \\ . & . & . & . & . \end{pmatrix}.$$

More generally, if Ω is an arbitrary finite interval with endpoints in $\mathbb{Z}$, then N_0 is a tridiagonal matrix whose first and last rows are of the form $\{1, 0, \ldots\}$ and $\{\ldots, 0, 1\}$, and whose interior rows have the form $\{\ldots, 0, \frac{1}{6}, \frac{2}{3}, \frac{1}{6}, 0, \ldots\}$.

The interpolation matrices play several vital roles in manipulating natural cubic splines. First, the interpolation matrix gives us a systematic method of handling the interpolation conditions associated with the original spline problem. Given a natural cubic spline of the form $p_\infty[x] = \mathcal{N}_0[x]\mathsf{p}_0$, the values of $p_\infty[x]$ on $\mathbb{Z}$ are exactly $\mathsf{N}_0\mathsf{p}_0$. Thus, $p_\infty[x]$ interpolates the values $\mathsf{N}_0\mathsf{p}_0$ on the grid $\mathbb{Z}$. To interpolate an arbitrary collection of values b, the initial coefficients p_0 must be chosen so as

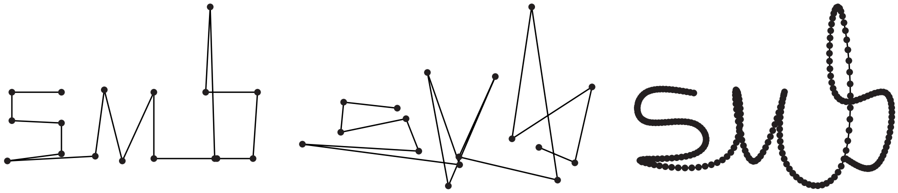


Figure 6.7 Using the inverse of the interpolation matrix to force interpolation of the vertices of a control polygon.

to satisfy $\mathrm{p}_0 = (\mathrm{N}_0)^{-1}\mathrm{b}$. For example, if the vertices of the curve in the left-hand image of Figure 6.7 are denoted by b, the center image of Figure 6.7 depicts the control polygon $\mathrm{p}_0 = (\mathrm{N}_0)^{-1}\mathrm{b}$. The right-hand image of the figure shows the result of applying four rounds of natural cubic spline subdivision to p_0, converging to a curve that interpolates the original points b.

6.3.2 Exact Inner Products for the Variational Scheme

Another important application of the interpolation matrices N_k lies in computing the inner product of limit functions produced by the subdivision scheme, that is, functions of the form $p_\infty[x] = \mathcal{N}_0[x]\mathrm{p}_0$ and $q_\infty[x] = \mathcal{N}_0[x]\mathrm{q}_0$ where $\mathcal{N}_0[x]$ is the vector of scaling functions associated with the variational scheme. In the previous section, we constructed the inner product matrices E_k with the property that $\langle p_\infty, q_\infty\rangle$ is the limit of $\mathrm{p}_k^T \mathrm{E}_k \mathrm{q}_k$ as $k \to \infty$. Instead of computing this inner product as a limit, the following theorem expresses the exact value of $\langle p_\infty, q_\infty\rangle$ in terms of a modified discrete inner product matrix of the form $\mathrm{E}_0\mathrm{N}_0$.

THEOREM 6.4 Consider a uniformly convergent subdivision scheme whose subdivision matrices S_{k-1} satisfy equation 6.17, with the matrices E_k being defined via a symmetric inner product taken on a bounded domain Ω. Given two limit curves $p_\infty[x]$ and $q_\infty[x]$ for this scheme of the form $\mathcal{N}_0[x]\mathrm{p}_0$ and $\mathcal{N}_0[x]\mathrm{q}_0$, the inner product $\langle p_\infty, q_\infty\rangle$ satisfies

$$\langle p_\infty, q_\infty\rangle == \mathrm{p}_0^T \mathrm{E}_0 \mathrm{N}_0 \mathrm{q}_0,$$

where N_0 is the interpolation matrix for the scheme.

Proof Given p_0 and q_0, let p_k and q_k be vectors defined via subdivision. Now, consider the continuous functions $p_k[x] = \widetilde{\mathcal{N}}_k[x]\mathrm{p}_k$ and $q_k[x] = \widetilde{\mathcal{N}}_k[x]\mathrm{q}_k$ defined in terms of *the finite element basis* $\widetilde{\mathcal{N}}_k[x]$ used in constructing the inner product matrix E_k. Due to the linearity of inner products, $\langle p_\infty, q_\infty \rangle$ satisfies

$$\langle p_\infty, q_\infty \rangle == \langle p_k, q_k \rangle + \langle p_\infty - p_k, q_k \rangle + \langle p_\infty, q_\infty - q_k \rangle.$$

Due to the uniform convergence of $p_k[x]$ to $p_\infty[x]$ on the bounded domain Ω, the inner product $\langle p_\infty - p_k, q_k \rangle$ converges to zero as $k \to \infty$. Because a similar observation holds for $\langle p_\infty, q_\infty - q_k \rangle$, the inner product $\langle p_\infty, q_\infty \rangle$ satisfies

$$\langle p_\infty, q_\infty \rangle = \lim_{k \to \infty} \langle p_k, q_k \rangle == \lim_{k \to \infty} \mathsf{p}_k^T \mathsf{E}_k \mathsf{q}_k.$$

Because the subdivision matrices S_{k-1} satisfy equation 6.17, the vectors p_k satisfy equation 6.16 for all $k > 0$. Replacing $\mathsf{p}_k^T \mathsf{E}_k == (\mathsf{E}_k \mathsf{p}_k)^T$ by its equivalent left-hand side, $(\mathsf{U}_{k-1} \cdots \mathsf{U}_0 \mathsf{E}_0 \mathsf{p}_0)^T$, yields

$$\lim_{k \to \infty} \mathsf{p}_k^T \mathsf{E}_k^T \mathsf{q}_k == \lim_{k \to \infty} \mathsf{p}_0^T \mathsf{E}_0^T \mathsf{U}_0^T \cdots \mathsf{U}_{k-1}^T \mathsf{q}_k.$$

Now, the effect of the product $\mathsf{U}_0^T \cdots \mathsf{U}_{k-1}^T$ on the vector q_k is to downsample the entries of q_k from the grid $\frac{1}{2^k}\mathbb{Z}$ to the grid $\mathbb{Z}$. Therefore, as $k \to \infty$, the vector $\mathsf{U}_0^T \cdots \mathsf{U}_{k-1}^T \mathsf{q}_k$ converges to the values of the limit function $q_\infty[x]$ taken on the integer grid $\mathbb{Z}$. However, these values can be computed directly using the interpolation matrix N_0 via the expression $\mathsf{N}_0 \mathsf{q}_0$. Thus, the theorem is proven; that is,

$$\lim_{k \to \infty} \mathsf{p}_0^T \mathsf{E}_0^T \mathsf{U}_0^T \cdots \mathsf{U}_{k-1}^T \mathsf{q}_k == \mathsf{p}_0^T \mathsf{E}_0 \mathsf{N}_0 \mathsf{q}_0.$$

For example, let p_0 be the vector consisting of the first coordinate of the vertices of the center polygon in Figure 6.7. If the subsequent vectors p_k are defined via the subdivision relation $\mathsf{p}_k = \mathsf{S}_{k-1} \mathsf{p}_{k-1}$ for natural cubic splines, the energy of the quadratic B-spline curves associated with these polygons is a scalar of the form $\mathsf{p}_k^T \mathsf{E}_k \mathsf{p}_k$. Evaluating this expression for $k = 0, 1, \ldots, 5$ yields the energies 71.3139, 42.2004, 34.922, 33.1025, 32.6476, and 32.5338. These energies are converging to the energy of the natural cubic spline $p_\infty[x]$ associated with the initial vector p_0. The energy of this limit curve can be computed directly by applying Theorem 6.4; that is,

$$\mathcal{E}[p_\infty] == \mathsf{p}_0^T \mathsf{E}_0 \mathsf{N}_0 \mathsf{p}_0 = 32.4959.$$

6.3.3 Multiresolution Spaces for Energy Minimization

We conclude this section by showing that the limit functions $p_\infty[x]$ produced by this subdivision scheme are minimizers of $\mathcal{E}$ in a well-defined sense. Typically, the space of functions over which this minimization is performed is restricted to those functions for which $\mathcal{E}$ is well defined. If $\mathcal{L}_2$ is the space of functions that are square integrable over Ω (in the Lebesgue sense), let $\mathcal{L}_2^{\mathsf{m}}$ denote those functions all of whose derivatives of up to order m are in $\mathcal{L}_2$. These spaces $\mathcal{L}_2^{\mathsf{m}}$ are the standard Sobolev spaces used in finite element analysis (see [113] or [134] for more details). By this definition, the variational functional for our problem, $\mathcal{E}[p]$, is well defined for those functions $p \in \mathcal{L}_2^2$.

Given a convergent subdivision scheme whose subdivision matrices S_{k-1} satisfy equation 6.17, let V_k be the span of the vector of scaling functions $\mathcal{N}_k[x]$ associated with the scheme. Due to the refinement relation for these vectors $\mathcal{N}_k[x]\mathsf{S}_{k-1} == \mathcal{N}_{k-1}[x]$, the spaces V_k are nested; that is,

$$\mathsf{V}_0 \subset \mathsf{V}_1 \subset \mathsf{V}_2 \subset \cdots \subset \mathsf{V}_k \subset \cdots \subset \mathcal{L}_2^2.$$

Our goal in this section is to prove that V_0 is exactly the space of functions $q[x]$ that minimize $\mathcal{E}[q]$ over all possible choices of interpolation conditions on $\Omega \cap \mathbb{Z}$. Our basic approach is to construct a multiresolution expansion of any function $q[x] \in \mathcal{L}_2^2$ in terms of the spaces V_k and a sequence of complementary spaces W_k. These complementary spaces W_k have the form

$$\mathsf{W}_k = \text{span}\left[\left\{r_k[x] \in \mathsf{V}_{k+1} \mid r_k[\mathsf{i}] == 0, \mathsf{i} \in \frac{1}{2^k}\mathbb{Z} \cap \Omega\right\}\right].$$

W_k consists of those functions in the space V_{k+1} defined on the fine grid $\frac{1}{2^{k+1}}\mathbb{Z}$ that vanish on the coarse grid $\frac{1}{2^k}\mathbb{Z}$. A function $q_{k+1}[x] \in \mathsf{V}_{k+1}$ can be written as a combination of a function $q_k[x] \in \mathsf{V}_k$ and a residual function $r_k[x] \in \mathsf{W}_k$ such that

$$\begin{aligned} q_k[\mathsf{i}] &== q_{k+1}[\mathsf{i}], \\ r_k[\mathsf{i}] &== 0, \end{aligned}$$

for all $\mathsf{i} \in \frac{1}{2^k}\mathbb{Z} \cap \Omega$. Therefore, the space V_{k+1} can be written as the sum of the spaces V_k and W_k (i.e., $\mathsf{V}_{k+1} = \mathsf{V}_k + \mathsf{W}_k$). The beauty of this decomposition is that the spaces V_k and W_k are orthogonal with respect to the continuous inner product used in defining the inner product matrices E_k.

THEOREM 6.5 Given a symmetric inner product $\langle\ \rangle$, let V_k and W_k be the spaces defined previously. Then the spaces V_k and W_k are orthogonal with respect to this inner product; that is, $q_k[x] \in \mathsf{V}_k$ and $r_k[x] \in \mathsf{W}_k$ implies that

$$\langle q_k, r_k \rangle == 0.$$

Proof Because $q_k[x] \in \mathsf{V}_k$, $q_k[x]$ can be expressed as $\mathcal{N}_k[x]\mathsf{q}_k$. Subdividing once, $q_k[x]$ can be expressed as $\mathcal{N}_{k+1}[x]\mathsf{S}_k\mathsf{q}_k$. Because $r_k[x] \in \mathsf{W}_{k+1}$, $r_k[x]$ can be expressed as $\mathcal{N}_{k+1}[x]\mathsf{r}_k$. Hence, the inner product in question can be expressed using Theorem 6.4 as

$$\langle q_k, r_k \rangle == \mathsf{q}_k^T \mathsf{S}_k^T \mathsf{E}_{k+1} \mathsf{N}_{k+1} \mathsf{r}_k.$$

Due to the symmetry of the inner product, the two-scale relation of equation 6.17 reduces to $\mathsf{S}_k^T \mathsf{E}_{k+1} == \mathsf{E}_k \mathsf{U}_k^T$. Substituting yields

$$\langle q_k, r_k \rangle == \mathsf{q}_k^T \mathsf{E}_k \mathsf{U}_k^T \mathsf{N}_{k+1} \mathsf{r}_k.$$

The expression $\mathsf{U}_k^T \mathsf{N}_{k+1} \mathsf{r}_k$ corresponds to the vector of values of the continuous function $r_k[x]$ sampled on $\frac{1}{2^k}\mathbb{Z} \cap \Omega$. By construction of $r_k[x]$, this vector is zero and the theorem is proven.

Given a function $q[x]$ for which the variational functional $\mathcal{E}[q]$ of equation 6.9 is well defined (i.e., $q[x] \in \mathcal{L}_2^2$), this function can be expressed as an infinite sum of the form

$$q[x] = q_0[x] + \sum_{i=0}^{\infty} r_i[x], \tag{6.18}$$

where $q_0[x] \in \mathsf{V}_0$ and $r_k[x] \in \mathsf{W}_k$. Due to this expansion, the variational functional $\mathcal{E}[q]$ can be expressed as the inner product $\langle q_0 + \sum_{i=0}^{\infty} r_i, q_0 + \sum_{i=0}^{\infty} r_i \rangle$. Due to the bilinearity of the inner product, this variational functional can be expanded to

$$\mathcal{E}[q] == \langle q_0, q_0 \rangle + 2\left\langle q_0, \sum_{i=0}^{\infty} r_i \right\rangle + \left\langle \sum_{i=0}^{\infty} r_i, \sum_{j=0}^{\infty} r_j \right\rangle.$$

Our next task is to move the infinite summations outside of the integrals comprising the inner products in this equation. Because $q[x] \in \mathcal{L}_2^2$, we observe that $q[x]$ must be continuous (see [113] for details) and therefore bounded on the closed, finite interval Ω. Consequently, the convergence of the partial sums $q_k[x]$ to $q[x]$ must be uniform (see Taylor [151], pages 498 and 594). This uniform convergence allows the previous equation to be rewritten as

$$\mathcal{E}[q] == \langle q_0, q_0\rangle + 2\sum_{i=0}^{\infty}\langle q_0, r_i\rangle + \sum_{i=0}^{\infty}\sum_{j=0}^{\infty}\langle r_i, r_j\rangle$$

(see Taylor [151], page 599). Finally, by Theorem 6.5, the inner product of q_0 and r_i is zero. Likewise, the inner product of r_i and r_j is zero for $i \neq j$. Therefore, this expansion reduces to

$$\mathcal{E}[q] == \langle q_0, q_0\rangle + \sum_{i=0}^{\infty}\langle r_i, r_i\rangle == \mathcal{E}[q_0] + \sum_{i=0}^{\infty}\mathcal{E}[r_i].$$

Based on this equation, it is clear that the minimum-energy function that interpolates the values of $q[x]$ on $\mathbb{Z} \cap \Omega$ is simply the function $q_0[x]$. More generally, the minimum-energy function that interpolates the values of $q[x]$ on $\frac{1}{2^k}\mathbb{Z} \cap \Omega$ is the function $q_k[x]$ whose energy satisfies

$$\mathcal{E}[q_k] == \mathcal{E}[q_0] + \sum_{i=0}^{k}\mathcal{E}[r_i].$$

To summarize, a convergent subdivision scheme whose subdivision matrices S_{k-1} satisfy equation 6.17 converges to limit functions that minimize the variational functional $\mathcal{E}$ used in defining the inner product matrices E_k.

We can now apply this technique to construct a multiresolution decomposition of a function $q_k[x]$ in V_k. If $q_k[x]$ is specified as $\mathcal{N}_k[x]\mathrm{q}_k$, our task is to compute the projection of $q_k[x]$ into the spaces V_{k-1} and W_{k-1}. By definition, the projection $\mathcal{N}_{k-1}[x]\mathrm{q}_{k-1}$ in V_{k-1} interpolates the values of $q_k[x]$ on the grid $\frac{1}{2^{k-1}}\mathbb{Z}$. Due to equation 6.2, the coefficient vectors q_{k-1} and q_k satisfy the relation $U_{k-1}^T N_k \mathrm{q}_k == N_{k-1}\mathrm{q}_{k-1}$, where the matrices N_{k-1} and N_k are interpolation matrices and U_{k-1}^T is a downsampling matrix. Multiplying both sides of this equation by N_{k-1}^{-1} yields the analysis equation

$$\mathrm{q}_{k-1} == N_{k-1}^{-1}U_{k-1}^T N_k \mathrm{q}_k.$$

The detail function $r_{k-1}[x]$ lies in V_k and can be expressed as $\mathcal{N}_k[x]\mathrm{r}_{k-1}$. Given that $r_{k-1}[x] = q_k[x] - q_{k-1}[x]$, the vector r_{k-1} can be generated via a second analysis

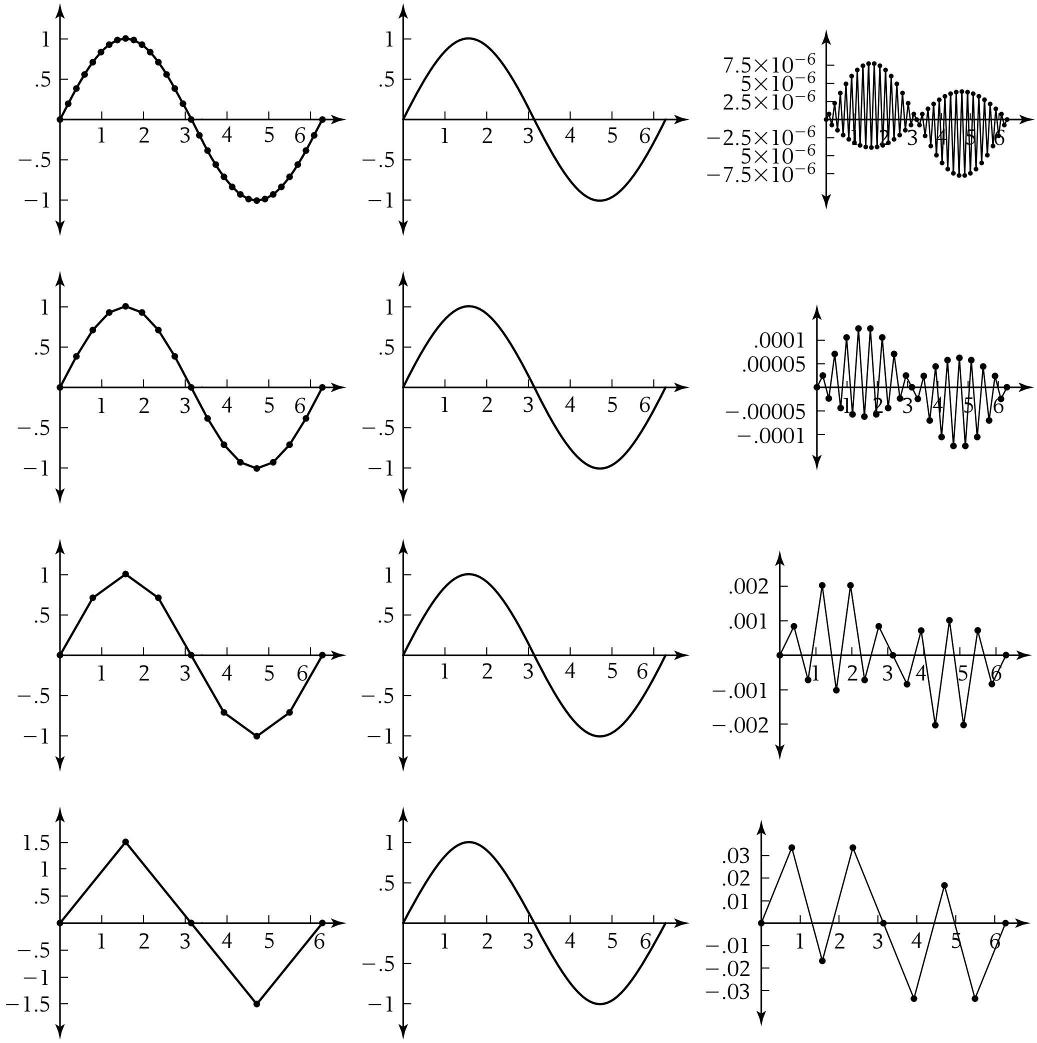


Figure 6.8 An example of a minimum-energy multiresolution decomposition for Sin[x].

equation of the form $\mathsf{r}_{k-1} = \mathsf{q}_k - \mathsf{S}_{k-1}\mathsf{q}_{k-1}$. Note that the detail coefficients r_{k-1} are overrepresented as a vector on the grid $\frac{1}{2^k}\mathbb{Z}$. (In particular, we are not using a wavelet basis for the space W_{k-1}.)

Figure 6.8 shows an example of this analysis technique. Starting from a vector q_3 whose limit function $q_3[x]$ interpolates the function Sin[x] on the grid $\frac{\pi}{16}\mathbb{Z}$, the analysis equation constructs the vectors q_3, q_2, q_1, and q_0 plotted in the left-hand column. Subdividing these vectors defines the natural cubic splines $q_3[x]$, $q_2[x]$, $q_1[x]$, and $q_0[x]$ in the center column. Each of these functions is the minimum-energy function that interpolates the values of $\mathsf{Sin}[x]$ on the grids $\frac{\pi}{16}\mathbb{Z}$, $\frac{\pi}{8}\mathbb{Z}$, $\frac{\pi}{4}\mathbb{Z}$, and $\frac{\pi}{2}\mathbb{Z}$, respectively. The right-hand column plots the residual vectors r_3, r_2, r_1, and r_0 produced by the analysis.

6.4 Subdivision for Bounded Harmonic Splines

The previous sections derived a variational subdivision scheme for natural cubic splines defined on a bounded interval Ω. The resulting subdivision rules produced splines that minimize a variational functional approximating bending energy. Here, we construct a subdivision scheme for a bounded variant of the harmonic splines $p[x, y]$ introduced in Chapter 5. These splines are the minimizers of the functional

$$\mathcal{E}[p] = \int_{\Omega} \left(p^{(1,0)}[x, y]^2 + p^{(0,1)}[x, y]^2 \right) dx\, dy. \tag{6.19}$$

Whereas Chapter 5 considered the case in which the domain of this integral is the entire plane, we consider here the case in which the domain Ω is a rectangle with corners on the integer grid $\mathbb{Z}^2$.

Our goal in this section is to construct a sequence of subdivision matrices S_{k-1} that given an initial vector p_0 define a sequence of vectors p_k satisfying $\mathsf{p}_k = \mathsf{S}_{k-1}\,\mathsf{p}_{k-1}$ and whose entries converge to a function $p[x, y]$ minimizing this functional $\mathcal{E}$. The entries of the vectors p_k are plotted on the restriction of the unbounded uniform grid $\frac{1}{2^k}\mathbb{Z}^2$ to the rectangle Ω. Our approach, introduced by the authors in [162] for biharmonic splines, is similar to that of the previous section. We construct discrete inner product matrices E_k that approximate $\mathcal{E}[p]$, and then solve for subdivision matrices S_{k-1} satisfying the relation $\mathsf{E}_k\mathsf{S}_{k-1} == \mathsf{U}_{k-1}\mathsf{E}_{k-1}$. As was true in the unbounded case of Chapter 5, there exists no locally supported solution S_{k-1} to this equation. Instead, we construct locally supported subdivision matrices $\widehat{\mathsf{S}}_{k-1}$ that approximately satisfy this equation (i.e., $\mathsf{E}_k\widehat{\mathsf{S}}_{k-1} \simeq \mathsf{U}_{k-1}\mathsf{E}_{k-1}$).

6.4.1 A Finite Element Scheme for Bounded Harmonic Splines

According to equation 6.17, the key to constructing a variational subdivision scheme is constructing a sequence of inner product matrices E_k for the variational functional of equation 6.19. In the case of harmonic splines, the continuous inner product corresponding to this energy function $\mathcal{E}$ has the form

$$\langle p, q\rangle = \int_{\Omega} \left(p^{(1,0)}[x,y]\,q^{(1,0)}[x,y] + p^{(0,1)}[x,y]\,q^{(0,1)}[x,y]\right) \mathrm{d}x\,\mathrm{d}y. \tag{6.20}$$

Given a sufficiently smooth finite element basis $\widetilde{\mathcal{N}}_{\mathsf{k}}[x,y]$, the entries of E_{k} are the inner product of pairs of basis functions in $\widetilde{\mathcal{N}}_{\mathsf{k}}[x,y]$. In the case of natural cubic splines, we chose a finite element basis consisting of translates of the quadratic B-spline basis function. In the case of bounded harmonic splines, we use a finite element basis consisting of translates of the piecewise linear hat function $\widetilde{n}[x,y]$. Again, this finite element basis leads to well-defined inner product matrices E_{k}.

Our approach in computing these matrices E_{k} is to restrict the piecewise linear hat functions to the quadrant $\Omega = [0,\infty]^2$, and then to set up a recurrence similar to that of equation 6.13 governing the corresponding inner product matrices over this domain. Due to this restriction, the initial finite element basis $\widetilde{\mathcal{N}}[x,y]$ consists of integer shifts of the form $\widetilde{n}[x-\mathsf{i}, y-\mathsf{j}]$, where $\mathsf{i}, \mathsf{j} \geq 0$. Subsequent finite element bases consist of dilates $\widetilde{\mathcal{N}}[2^{\mathsf{k}}x, 2^{\mathsf{k}}y]$ of this vector and are related by a stationary subdivision matrix $\widetilde{\mathsf{S}}$ satisfying $\widetilde{\mathcal{N}}[2x, 2y]\mathsf{S} == \widetilde{\mathcal{N}}[x,y]$, whose columns have the form

$$\begin{array}{ccc}
\begin{pmatrix} 1 & \frac{1}{2} & 0 & 0 & . \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & . \\ 0 & 0 & 0 & 0 & . \\ 0 & 0 & 0 & 0 & . \\ . & . & . & . & . \end{pmatrix} &
\begin{pmatrix} 0 & \frac{1}{2} & 1 & \frac{1}{2} & . \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} & . \\ 0 & 0 & 0 & 0 & . \\ 0 & 0 & 0 & 0 & . \\ . & . & . & . & . \end{pmatrix} & \ldots \\
\begin{pmatrix} 0 & 0 & 0 & 0 & . \\ \frac{1}{2} & 0 & 0 & 0 & . \\ 1 & \frac{1}{2} & 0 & 0 & . \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & . \\ . & . & . & . & . \end{pmatrix} &
\begin{pmatrix} 0 & 0 & 0 & 0 & . \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 & . \\ 0 & \frac{1}{2} & 1 & \frac{1}{2} & . \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} & . \\ . & . & . & . & . \end{pmatrix} & \ldots \\
\ldots & \ldots & \ldots
\end{array}$$

Each two-dimensional array here corresponds to a column of $\widetilde{\mathsf{S}}$ plotted using the natural two-dimensional grid induced by the grid $(\mathbb{Z}^+)^2$ (i.e., the restriction of $\mathbb{Z}^2$ to $[0,\infty]^2$). The rows of $\widetilde{\mathsf{S}}$ are also ordered in a similar manner. Technically, the rows and columns of $\widetilde{\mathsf{S}}$ (and E) are indexed by integer pairs $\{\mathsf{i}, \mathsf{j}\} \in (\mathbb{Z}^+)^2$. In *Mathematica*, these matrices are represented as four-dimensional tensors whose entries are indexed by two integer pairs, $\{i_1, j_1\}$ and $\{i_2, j_2\}$, the first pair corresponding to a row of the matrix and the second pair corresponding to a column of the matrix.

Having restricted our initial analysis to the quadrant $[0, \infty]^2$, we observe that the inner product matrices E_k associated with the basis $\widetilde{\mathcal{N}}[2^k x, 2^k y]$ satisfy the equation $E_k = E$ for all $k \geq 0$. (Observe that the factor 4^k induced by taking the square of the first derivative on $\frac{1}{2^k}(\mathbb{Z}^+)^2$ is canceled by the constant of integration $\frac{1}{4^k}$ arising from integrating on the grid $\frac{1}{2^k}(\mathbb{Z}^+)^2$.) Substituting this relation into equation 6.12 yields a recurrence relation of the form

$$\widetilde{S}^T E \widetilde{S} == E.$$

By treating the rows (or columns) of E as unknown masks, this equation can be used to set up a system of homogeneous linear equations whose single solution is a constant multiple of the inner product matrix E. This unique homogeneous solution can then be normalized to agree with the inner product mask for the uniform case. Based on the method of section 6.1, we observe that this inner product mask $e[x, y]$ for integer translates of the piecewise linear hat function $\widetilde{n}[x, y]$ has the form

$$e[x, y] = \begin{pmatrix} x^{-1} & 1 & x \end{pmatrix} \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} y^{-1} \\ 1 \\ y \end{pmatrix}.$$

The associated *Mathematica* implementation gives the details of setting up this unknown matrix in the bivariate case and solving for the entries (♔). The resulting inner product matrix E has the form shown in Figure 6.9.

$$\begin{array}{ccc}
\begin{pmatrix} 1 & -\frac{1}{2} & 0 & . \\ -\frac{1}{2} & 0 & 0 & . \\ 0 & 0 & 0 & . \\ . & . & . & . \end{pmatrix} & \begin{pmatrix} -\frac{1}{2} & 2 & -\frac{1}{2} & . \\ 0 & -1 & 0 & . \\ 0 & 0 & 0 & . \\ . & . & . & . \end{pmatrix} & \cdots \\
\begin{pmatrix} -\frac{1}{2} & 0 & 0 & . \\ 2 & -1 & 0 & . \\ -\frac{1}{2} & 0 & 0 & . \\ . & . & . & . \end{pmatrix} & \begin{pmatrix} 0 & -1 & 0 & . \\ -1 & 4 & -1 & . \\ 0 & -1 & 0 & . \\ . & . & . & . \end{pmatrix} & \cdots \\
\cdots & \cdots & \cdots
\end{array}$$

Figure 6.9 Rows of the inner product matrix E for translates of the piecewise linear hat function on $[0, \infty]^2$.

6.4.2 Subdivision for Harmonic Splines on a Quadrant

Having derived the inner product matrix E for the domain $\Omega = [0, \infty]^2$, our next task is to construct a subdivision matrix S for bounded harmonic splines that satisfies a two-dimensional stationary version of equation 6.17 (i.e., $ES == UE$, where U is the two-dimensional upsampling matrix). One approach to this problem would be to use the Jacobi iteration or the linear programming method of Chapter 5 to compute a finite approximation to S. Either of these methods produces a matrix $\widehat{S}$ whose rows are finite approximations to the infinitely supported row of the exact subdivision matrix S. In this section, we describe an alternative construction for the subdivision matrix S that exactly satisfies the relation $ES == UE$.

The key to this approach is to extend the subdivision scheme on a single quadrant to an equivalent uniform scheme defined over the entire plane. This extension from a quadrant to the plane relies on reflecting the entries of the initial vector p_0 with respect to each of the coordinate axes. Given an initial vector p_0 whose entries are defined on the grid $(\mathbb{Z}^+)^2$, this vector can be extended to the grid $\mathbb{Z}^2$ via the reflection rule $p_0[\![i, j]\!] = p_0[\![|i|, |j|]\!]$ for all $\{i, j\} \in \mathbb{Z}^2$. Plotted on the two-dimensional grid $\mathbb{Z}^2$, this reflected set of coefficients has the form

$$\begin{pmatrix} . & . & . & . & . & . & . \\ . & p_0[\![2,2]\!] & p_0[\![2,1]\!] & p_0[\![2,0]\!] & p_0[\![2,1]\!] & p_0[\![2,2]\!] & . \\ . & p_0[\![1,2]\!] & p_0[\![1,1]\!] & p_0[\![1,0]\!] & p_0[\![1,1]\!] & p_0[\![1,2]\!] & . \\ . & p_0[\![0,2]\!] & p_0[\![0,1]\!] & p_0[\![0,0]\!] & p_0[\![0,1]\!] & p_0[\![0,2]\!] & . \\ . & p_0[\![1,2]\!] & p_0[\![1,1]\!] & p_0[\![1,0]\!] & p_0[\![1,1]\!] & p_0[\![1,2]\!] & . \\ . & p_0[\![2,2]\!] & p_0[\![2,1]\!] & p_0[\![2,0]\!] & p_0[\![2,1]\!] & p_0[\![2,2]\!] & . \\ . & . & . & . & . & . & . \end{pmatrix}.$$

If the coefficients of this reflected vector are attached to a generating function $p_0[x, y]$, our approach in generating the subdivision matrix S for bounded harmonic splines is to subdivide $p_0[x, y]$ using the exact subdivision mask $s[x, y]$ for uniform harmonic splines (as defined in equation 5.7). The result is a sequence of generating functions $p_k[x, y]$ satisfying the multiscale relation

$$e[x, y]p_k[x, y] == e\left[x^{2^k}, y^{2^k}\right] p_0\left[x^{2^k}, y^{2^k}\right] \tag{6.21}$$

whose coefficients are also reflections of a sequence of vectors p_k defined over the grid $(\frac{1}{2^k}\mathbb{Z}^+)^2$. Now, we claim that these vectors p_k satisfy the related multiscale equation

$$E\, p_k == U^k E\, p_0 \tag{6.22}$$

for bounded harmonic splines, where E is the inner product matrix computed in the previous section. To verify this claim, we observe that for any vector p_k the coefficients of $e[x, y]p_k[x, y]$ are constant multiples of entries of the vector $E\, p_k$. For example, the coefficient of the x^0y^0 term in $e[x, y]p_k[x, y]$ is $4p_k[\![0, 0]\!] - 2p_k[\![1, 0]\!] - 2p_k[\![0, 1]\!]$, four times the entry of $E\, p_k$ corresponding to the origin. Likewise, coefficients of the terms $x^i y^0$ in $e[x, y]p_k[x, y]$, where $i > 0$, have the form $4p_k[\![i, 0]\!] - p_k[\![i-1, 0]\!] - p_k[\![i+1, 0]\!] - 2p_k[\![i, 1]\!]$. These coefficients are two times the corresponding entries of $E\, p_k$. Because these constants appear on both sides of equation 6.22 in E, restricting solutions of equation 6.21 to the first quadrant yields a solution to equation 6.22.

Given this observation, the subdivision relation $p_k[x, y] = s[x, y]p_{k-1}[x, y]$ for the unbounded case can be converted into an equivalent matrix form $p_k = S\, p_{k-1}$, where S is the subdivision matrix for harmonic splines on the quadrant $[0, \infty]^2$. In particular, the entry of S in the $\{i_1, j_1\}$th row and $\{i_2, j_2\}$th column is a sum of coefficients from the exact subdivision mask $s[x, y]$ for unbounded harmonic splines (equation 5.7) of the form

$$\sum s[\![i_1 \pm 2i_2,\ j_1 \pm 2j_2]\!]. \tag{6.23}$$

Note that when i_2 or j_2 is zero, the corresponding coefficient appears only once in the sum. For example, each entry of the column of S corresponding to the origin (i.e., $\{i_2, j_2\} == \{0, 0\}$) consists of a single coefficient of the exact mask $s[x, y]$. Plotted as a two-dimensional grid, this column has the form

$$\begin{pmatrix} 1.4535 & 0.4535 & -0.1277 & -0.0338 & -0.0106 & . \\ 0.4535 & 0.2441 & 0.0347 & 0.0015 & -0.0022 & . \\ -0.1277 & 0.0347 & 0.021 & 0.0073 & 0.0019 & . \\ -0.0338 & 0.0015 & 0.0073 & 0.0048 & 0.0023 & . \\ -0.0106 & -0.0022 & 0.0019 & 0.0023 & 0.0016 & . \\ . & . & . & . & . & . \end{pmatrix}.$$

Entries of columns of S corresponding to grid points that lie on the boundary of $\Omega = [0, \infty]^2$ are the sum of two coefficients from the exact mask $s[x, y]$. For example, the column of S corresponding to the grid point with index $\{i_2, j_2\} == \{1, 0\}$ has the form

$$\begin{pmatrix} -0.2554 & 0.0695 & 0.0421 & 0.0146 & 0.0038 & . \\ 0.4197 & 0.2456 & 0.0421 & 0.0063 & 0.0001 & . \\ 1.443 & 0.4513 & -0.1258 & -0.0315 & -0.0089 & . \\ 0.4496 & 0.2424 & 0.035 & 0.0025 & -0.0013 & . \\ -0.1295 & 0.0336 & 0.0208 & 0.0077 & 0.0024 & . \\ . & . & . & . & . & . \end{pmatrix}.$$

In general, entries of columns of S correspond to the interior grid points consisting of four coefficients. For example, the column of S corresponding to the grid point $\{i_2, j_2\} == \{1, 1\}$ has the form

$$\begin{pmatrix} 0.0841 & 0.0841 & -0.2516 & 0.0699 & 0.0417 & . \\ 0.0841 & 0.252 & 0.4198 & 0.2448 & 0.0413 & . \\ -0.2516 & 0.4198 & 1.4341 & 0.4483 & -0.1271 & . \\ 0.0699 & 0.2448 & 0.4483 & 0.2413 & 0.0343 & . \\ 0.0417 & 0.0413 & -0.1271 & 0.0343 & 0.021 & . \\ . & . & . & . & . & . \end{pmatrix}.$$

6.4.3 Subdivision for Harmonic Splines on Bounded Rectangular Domains

Given a bounded rectangular domain Ω whose corners are in $\mathbb{Z}^2$, we can construct a sequence of inner product matrices E_k using the boundary rules from Figure 6.9. Given these inner product matrices, our task is to compute the corresponding subdivision matrices S_{k-1} that satisfy the relation $E_k S_{k-1} == U_{k-1} E_{k-1}$. The simplest approach would be to invert the matrix E_k and explicitly solve for S_{k-1}. Unfortunately, the matrix E_{k-1} is not invertible because its null space is the constant vector. To uniquely specify the desired subdivision matrix S_{k-1}, we must add an extra constraint to each of the columns of the matrix S_{k-1}.

In the case where $\Omega = [0, \infty]^2$, each column of the subdivision matrix S defined by equation 6.23 satisfies a simple constraint arising from the observation that the sum of the coefficients $s[\![i, j]\!]$ for the unbounded harmonic mask $s[x, y]$ is exactly 4 (i.e., $s[1, 1] == 4$). If w is a row vector whose entries (plotted as a two-dimensional grid) have the form

$$w = \begin{pmatrix} 1 & 2 & 2 & 2 & . \\ 2 & 4 & 4 & 4 & . \\ 2 & 4 & 4 & 4 & . \\ 2 & 4 & 4 & 4 & . \\ . & . & . & . & . \end{pmatrix},$$

we claim that the subdivision matrix S defined in equation 6.23 satisfies the auxiliary relation $wS == 4w$. For each column of S, this relation follows from the fact that the harmonic mask $s[x, y]$ is symmetric and that its coefficients sum to 4. For example, the column of S corresponding to the origin had the form

$$\begin{pmatrix} s[\![0,0]\!] & s[\![0,1]\!] & s[\![0,2]\!] & . \\ s[\![1,0]\!] & s[\![1,1]\!] & s[\![1,2]\!] & . \\ s[\![2,0]\!] & s[\![2,1]\!] & s[\![2,2]\!] & . \\ . & . & . & . \end{pmatrix},$$

where $s[\![i, j]\!]$ was the ijth coefficient of the harmonic mask $s[x, y]$. Weighting the corner coefficient $s[\![0, 0]\!]$ by a factor of one, edge coefficients $s[\![i, 0]\!]$ and $s[\![0, i]\!]$ by a factor of two, and interior coefficients $s[\![i, j]\!]$ by a factor of four yields an infinite sum of the coefficients $s[\![i, j]\!]$ that is equivalent to requiring that $s[1, 1] == 4$ because the mask $s[x, y]$ is symmetric. This auxiliary constraint $wS == 4w$ can be generalized to the bounded case by defining a sequence of row vectors w_k whose entries correspond to grid points of $\frac{1}{2^k}\mathbb{Z}^2$ restricted to Ω and have the form

$$w_k = \begin{pmatrix} 1 & 2 & 2 & . & 2 & 1 \\ 2 & 4 & 4 & . & 4 & 2 \\ 2 & 4 & 4 & . & 4 & 2 \\ . & . & . & . & . & . \\ 2 & 4 & 4 & . & 4 & 2 \\ 1 & 2 & 2 & . & 2 & 1 \end{pmatrix}.$$

Now, the relation $E_k S_{k-1} == U_{k-1} E_{k-1}$ taken in conjunction with the auxiliary constraint $w_k S_{k-1} = 4w_{k-1}$ defines a sequence of unique subdivision matrices S_{k-1} for harmonic splines on the bounded domain Ω. Figure 6.10 shows an example of two rounds of exact harmonic subdivision applied to our test shape from Chapter 5 (defined on the domain $\Omega = [0, 7]^2$). The two subdivision matrices S_0 and S_1 for this example were computed exactly using linear algebra within *Mathematica*. Note

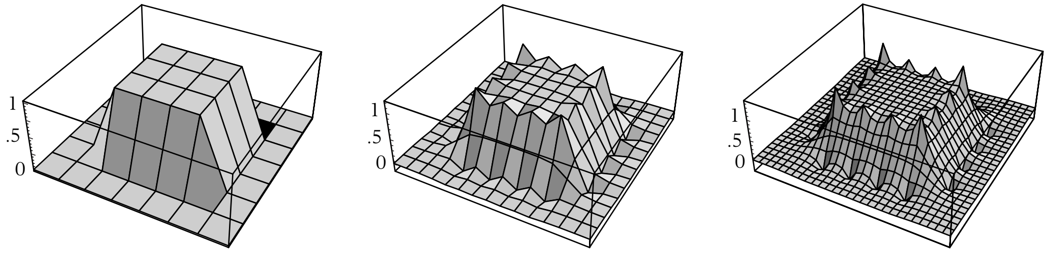


Figure 6.10 Two rounds of exact subdivision for bounded harmonic splines.

that we did not attempt to continue this process and compute the exact subdivision matrix S_2, in that this computation involves inverting a 3249×3249 matrix whose rows and columns are indexed by a pair of 57×57 grids.

As in the unbounded case, bounded harmonic splines $p_\infty[x, y]$ diverge at integer grid points $\mathbb{Z}^2$ for which the corresponding entry of the vector $E\,p_0$ is non-zero. Although this divergence causes the variational functional of equation 6.19 to also diverge, these functions $p_\infty[x, y]$ still satisfy the harmonic equation on the interior of the domain Ω (except at integer grid points in $\mathbb{Z}^2$) via equation 6.22. Moreover, due to the definition of the subdivision scheme in terms of reflection, the limit function $p_\infty[x, y]$ also satisfies the natural boundary condition of having its cross-boundary derivative be zero. (For the higher-order polyharmonic splines, the associated subdivision scheme is always convergent at the integer grid $\mathbb{Z}^2$, and as a result the minimization theory of section 6.3 follows without difficulty.)

As in the case when $\Omega = [0, \infty]^2$, entries of the columns of the subdivision matrices S_{k-1} for bounded domains are sums of coefficients $s[\![i, j]\!]$ of the uniform mask $s[x, y]$. (We leave the derivation of the exact formula to the interested reader.) Due to the fast decay of the coefficients in the mask $s[x, y]$, interior columns of the subdivision matrix S_0 for large domains provide a reasonable approximation to the uniform mask $s[x, y]$ itself. For example, Figure 6.11 shows the central 9×9 portion of column $\{4, 4\}$ of the subdivision matrix S_0 for the domain $[0, 8]^2$. Observe that the entries agree with the exact coefficients of $s[x, y]$ of Figure 5.6 to four decimal digits.

Another interesting consequence of the fast decay of the coefficients of $s[x, y]$ is that the subdivision rules (i.e., rows) associated with the initial matrix S_0 are finite approximations to the subdivision rules for subsequent matrices S_k. In particular, subdivision rules of S_0 for corners of Ω can be used to approximate the exact subdivision rules of S_k at corners of Ω. For example, if $\Omega = [0, 4]^2$, the two tables of Figure 6.12 are plots of the exact subdivision rules for S_0 and S_1 at a corner of Ω.

$$\begin{pmatrix}
. & . & . & . & . & . & . & . & . & . & . \\
. & 0.0016 & 0.0023 & 0.0018 & -0.0023 & -0.0107 & -0.0023 & 0.0018 & 0.0023 & 0.0016 & . \\
. & 0.0023 & 0.0048 & 0.0072 & 0.0014 & -0.0339 & 0.0014 & 0.0072 & 0.0048 & 0.0023 & . \\
. & 0.0018 & 0.0072 & 0.0209 & 0.0346 & -0.1278 & 0.0346 & 0.0209 & 0.0072 & 0.0018 & . \\
. & -0.0023 & 0.0014 & 0.0346 & 0.244 & 0.4534 & 0.244 & 0.0346 & 0.0014 & -0.0023 & . \\
. & -0.0107 & -0.0339 & -0.1278 & 0.4534 & 1.4534 & 0.4534 & -0.1278 & -0.0339 & -0.0107 & . \\
. & -0.0023 & 0.0014 & 0.0346 & 0.244 & 0.4534 & 0.244 & 0.0346 & 0.0014 & -0.0023 & . \\
. & 0.0018 & 0.0072 & 0.0209 & 0.0346 & -0.1278 & 0.0346 & 0.0209 & 0.0072 & 0.0018 & . \\
. & 0.0023 & 0.0048 & 0.0072 & 0.0014 & -0.0339 & 0.0014 & 0.0072 & 0.0048 & 0.0023 & . \\
. & 0.0016 & 0.0023 & 0.0018 & -0.0023 & -0.0107 & -0.0023 & 0.0018 & 0.0023 & 0.0016 & . \\
. & . & . & . & . & . & . & . & . & . & .
\end{pmatrix}$$

Figure 6.11 Approximating the subdivision mask for unbounded harmonic splines.

$$\begin{pmatrix}
1.4534 & -0.2556 & -0.0213 & -0.0039 & -0.001 \\
-0.2556 & 0.0838 & 0.0073 & -0.0012 & -0.0008 \\
-0.0213 & 0.0073 & 0.0065 & 0.0021 & 0.0005 \\
-0.0039 & -0.0012 & 0.0021 & 0.002 & 0.0009 \\
-0.001 & -0.0008 & 0.0005 & 0.0009 & 0.0004
\end{pmatrix}$$

$$\begin{pmatrix}
1.4535 & -0.2554 & -0.0211 & -0.0035 & -0.001 & -0.0004 & -0.0002 & -0.0001 & -0.0001 \\
-0.2554 & 0.0841 & 0.0076 & -0.0008 & -0.0009 & -0.0005 & -0.0003 & -0.0002 & -0.0001 \\
-0.0211 & 0.0076 & 0.0066 & 0.0019 & 0.0004 & 0 & -0.0001 & -0.0001 & -0.0001 \\
-0.0035 & -0.0008 & 0.0019 & 0.0014 & 0.0006 & 0.0002 & 0.0001 & 0 & 0 \\
-0.001 & -0.0009 & 0.0004 & 0.0006 & 0.0004 & 0.0003 & 0.0001 & 0.0001 & 0 \\
-0.0004 & -0.0005 & 0 & 0.0002 & 0.0003 & 0.0002 & 0.0001 & 0.0001 & 0 \\
-0.0002 & -0.0003 & -0.0001 & 0.0001 & 0.0001 & 0.0001 & 0.0001 & 0.0001 & 0.0001 \\
-0.0001 & -0.0002 & -0.0001 & 0 & 0.0001 & 0.0001 & 0.0001 & 0.0001 & 0.0001 \\
-0.0001 & -0.0001 & -0.0001 & 0 & 0 & 0 & 0.0001 & 0.0001 & 0
\end{pmatrix}$$

Figure 6.12 Approximating the exact subdivision rule at a corner of Ω.

Observe that padding out the subdivision rule for S_0 by zeros yields an approximation to the exact subdivision rule for S_1 that is accurate to within three decimal digits.

In a similar manner, subdivision rules of S_0 for the edges of Ω can be used to approximate the exact subdivision rules of S_k on the edges of Ω. Finally, subdivision rules of S_0 on the interior of Ω can be used to approximate the exact subdivision

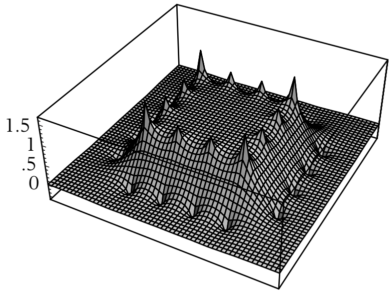


Figure 6.13 Three rounds of approximate subdivision for bounded harmonic splines.

rules of S_k on the interior of Ω. These observations lead to a simple algorithm for constructing locally supported approximations $\widehat{S}_k$ to the exact subdivision matrices S_k: compute the exact subdivision rules from S_0 and pad these rules to form the rows of $\widehat{S}_k$. Of course, this process could also be repeated with the exact subdivision matrix S_1, with the resulting subdivision matrices $\widehat{S}_k$ being a more accurate approximation to the exact matrices S_k at the cost of larger row supports. Figure 6.13 shows the result of three rounds of approximate harmonic subdivision using the padded subdivision rules from the matrix S_0 on the domain $[0, 7]^2$.

One final question that arises during the construction is the effect of the original finite element basis $\widetilde{\mathcal{N}}_k[x, y]$ on the resulting subdivision matrices S_k. In the differential case, the bi-infinite Laurent expansion of $\frac{l[x^2,y^2]}{l[x,y]}$ led to a unique choice for the subdivision mask for harmonic splines. However, in the variational case, choosing a different finite element basis, say translates of the biquadratic B-spline basis functions, leads to a different set of inner product matrices E_k that in turn leads to different subdivision matrices S_k satisfying equation 6.17. The theory of section 6.3 ensures that for any choice of finite element basis the resulting *exact* subdivision scheme converges to minimizers of the variational functional. Thus, different choices for the finite element basis lead to different subdivision schemes that all converge to the same space of minimizers. In the case of natural cubic splines, inner product matrices based on quadratic B-splines were carefully chosen so as to produce locally supported subdivision rules. For bounded harmonic splines, no choice of finite element basis leads to locally supported subdivision rules. Instead, we simply follow the heuristic of choosing the finite element basis with appropriate smoothness and support as small as possible. In practice, this rule appears to lead to a variational scheme whose subdivision rules are highly localized.

CHAPTER 7

Averaging Schemes for Polyhedral Meshes

Chapters 5 and 6 considered uniform surface schemes (both bounded and unbounded). Unfortunately, these schemes are not flexible enough for most practical modeling applications because even simple topological shapes such as the sphere cannot be represented by a single uniform surface grid. This chapter introduces a more flexible representation for surfaces—in terms of topological meshes capable of representing surfaces of arbitrary topology—that avoids this problem. The first part of the chapter investigates some of the basic properties of topological meshes and introduces two subdivision schemes for such meshes: linear subdivision for triangle meshes and bilinear subdivision for quadrilateral meshes. Next, the chapter introduces a smooth variant of bilinear subdivision and presents several extensions of this scheme. Finally, the chapter concludes with a brief survey of the wealth of current work on subdivision schemes for polyhedral meshes.

7.1 Linear Subdivision for Polyhedral Meshes

Chapter 6 considered uniform surface schemes whose coefficient vectors p_k are associated with the restriction of the uniform grid $\frac{1}{2^k}\mathbb{Z}^2$ to a bounded rectangular domain. In general, this restriction requires the resulting limit surfaces to be functional; that is, the graph of the bivariate function $p_\infty[x, y]$. Treating the vector p_k as a collection of control points allows us to construct parametric surfaces that although no longer functional are still topologically equivalent to a plane (or to a portion of a plane). This uniform approach can be generalized slightly if one treats the underlying parametric domain as periodic. Taking this view, the resulting parametric surface is now a closed surface that is topologically equivalent to a torus (or a portion of a torus). Unfortunately, other topological shapes such as spheres cannot be modeled by this method without introducing degeneracies such as poles. For example, the latitude/longitude system used to assign two-dimensional coordinates to the surface of the earth is degenerate at the North and South Poles.

To understand the reason for this topological restriction, consider the closed square grid induced by restricting the grid points $\mathbb{Z}^2$ to a periodic domain. (See the doughnut-shaped grid of Figure 7.7 for an example.) This grid consists of a collection of $\mathtt{v}$ vertices, $\mathtt{e}$ edges, and $\mathtt{f}$ square faces. Now, consider the value of the expression $\mathtt{v} - \mathtt{e} + \mathtt{f}$ for any such surface grid. For each vertex in this grid, there exist two corresponding edges (say one pointing "north" and another pointing "east") and one corresponding face (say the "northeast" face). Therefore, the expression $\mathtt{v} - \mathtt{e} + \mathtt{f}$ is always zero for any size grid. This expression $\mathtt{v} - \mathtt{e} + \mathtt{f}$ is the *Euler characteristic* for the surface grid and is related to the number of "handles" $\mathtt{g}$ associated with the surface grid (and its underlying closed surface) via the relation $\mathtt{v} - \mathtt{e} + \mathtt{f} = 2 - 2\mathtt{g}$. (See Firby and Gardiner [62] and Lakatos [91] for more details.) Because the Euler characteristic is always zero for such closed uniform grids, this surface scheme is capable of representing only surfaces, such as a torus, that have one handle. To model other types of topological surfaces, we must use a more general type of surface representation.

7.1.1 Polyhedral Meshes

A two-dimensional *topological mesh* consists of a list of polygonal faces in which each polygon is represented by a list of vertices, with each vertex denoted by a distinct index. For example, an octahedron can be thought of as eight triangles, each consisting of three vertices. If we number the vertices 1 through 6, one possible topological mesh for the octahedron, has the form

$$\begin{gathered}\{\{1, 2, 5\}, \{2, 3, 5\}, \{3, 4, 5\},\\ \{4, 1, 5\}, \{2, 1, 6\}, \{3, 2, 6\}, \{4, 3, 6\}, \{1, 4, 6\}\}.\end{gathered}$$

As a second example, consider a hexahedron (i.e., a topological cube) consisting of six quadrilateral faces. If each face consists of four vertices numbered 1 through 8, one possible topological mesh for the hexahedron has the form

$$\begin{gathered}\{\{1, 4, 3, 2\}, \{1, 2, 6, 5\},\\ \{2, 3, 7, 6\}, \{3, 4, 8, 7\}, \{4, 1, 5, 8\}, \{5, 6, 7, 8\}\}.\end{gathered}$$

Note that this representation does not encode the geometric shape of the mesh. It encodes only the connectivity of the edges and faces of the mesh. To specify actual geometric locations for each of these vertices, we need only add a second vector consisting of geometric positions for each of the vertices (the vector p in previous chapters). In the hexahedral example, letting the vertices 1 through 8 have the geometric positions

$$\{\{0,0,0\},\{1,0,0\},\{1,1,0\},\\ \{0,1,0\},\{0,0,1\},\{1,0,1\},\{1,1,1\},\{0,1,1\}\}$$

defines a unit cube. As before, we can access the particular coordinates for a vertex v via the expression p⟦v⟧.

Thus, a *polyhedral mesh* is a pair {M, p}, where M is the topological mesh and p is a vector of vertex positions. One might ask, Why not substitute the vertex positions of p directly into the corresponding elements of M? Separating topology from geometry has several advantages. First, two faces sharing a common edge or vertex can immediately be identified by comparing vertex indices. If a face were simply a list of vertex positions, determining whether two faces share a common edge or vertex would involve geometric comparison of numerical vertex positions that themselves may be subject to error. Second, separating topology from geometry also saves space. A vertex position that consists of three floating-point numbers is stored only once in p. Subsequent references in M require only an integer index. Finally, and most important, the subdivision rules described in this chapter depend solely on the local topology of the mesh. Explicitly encoding this topology makes implementing the schemes much easier. Figure 7.1 depicts the polyhedral meshes for the octahedron and the hexahedron referred to previously.

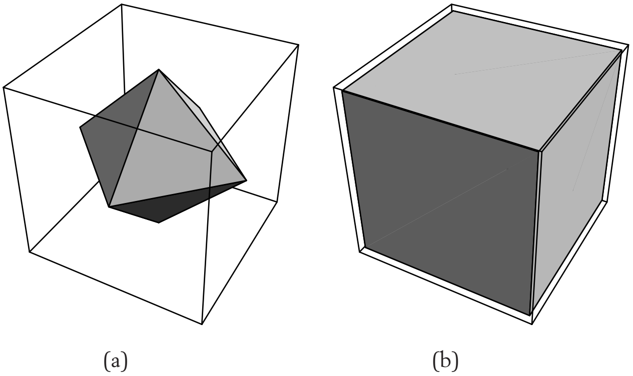


Figure 7.1 Polyhedral meshes representing an octahedron (a) and a hexahedron (b).

We next introduce two important quantities associated with the vertices of a topological mesh M. The set of faces in M that contains the vertex v is denoted by ring[v], whereas val[v], the *valence* of v, is the number of faces in ring[v]. Given these definitions, a topological mesh M defines a *surface mesh* (i.e., a manifold) if ring[v] is topologically equivalent to a disc (an interior vertex) or a half-disc (a boundary vertex) for every vertex v of M. For surface meshes, an edge is an *interior* edge if it is shared by exactly two polygons, and is a *boundary* edge if it lies on a single polygon. A maximally connected chain of boundary edges forms a boundary curve of the surface mesh. *Closed* surface meshes have no boundary curves, whereas *open* surface meshes have one or more boundary curves.

Surface meshes consisting entirely of triangular faces are *triangle meshes*. For example, three-direction quartic box splines generate a sequence of triangle meshes by splitting each triangle into four subtriangles. The resulting meshes have vertices of only valence six. Likewise, surface meshes consisting entirely of quadrilateral faces are *quad meshes*. The subdivision scheme for bicubic B-splines generates a sequence of quad meshes by splitting each quad into four subquads.

7.1.2 Topological Subdivision of Polyhedral Meshes

We proceed by considering two simple subdivision schemes for polyhedral meshes: linear subdivision for triangle meshes and bilinear subdivision for quad meshes. These methods avoid the topological limitations of uniform schemes and provide a foundation for subsequent smooth schemes. Each method produces a sequence of polyhedral meshes $\{M_k, p_k\}$ via a two-step process. The first step, topological subdivision, splits each face in M_{k-1} into a set of subfaces in M_k. The second step, geometric positioning, computes positions for the new vertices p_k in terms of the positions of the old vertices p_{k-1} via a subdivision relation $p_k = S_{k-1} p_{k-1}$. The beauty of these two subdivision schemes is that the subdivision matrix S_{k-1} depends only on the topology M_{k-1} of the mesh. We first discuss the details of topological subdivision for these schemes.

Given a topological mesh M_{k-1}, topological subdivision splits each face in M_{k-1} into a collection of subfaces and collects the results as a new topological mesh M_k. There are two traditional variants of topological subdivision for surface meshes. In triangle subdivision, each triangle is split into four subtriangles, as shown in the upper left of Figure 7.2. In one sense, this topological split "chops off" the three corners of the triangle, leaving a triangle in the middle. This split can be generalized to other types of polygons, as shown in the upper portion of Figure 7.2. Given an

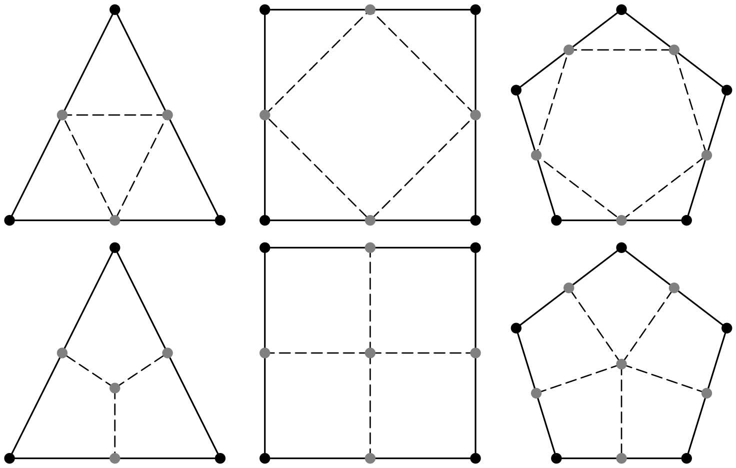


Figure 7.2 Two types of topological subdivision for various polygons.

n-gon, this split returns n triangles and a new n-gon connecting the midpoints of the edges of the original n-gon. For most applications, triangle subdivision is used only when the initial mesh consists only of triangles.

An alternative approach is quad subdivision. In this method, each n-gon in M_{k-1} is split into n quads in M_k by placing a single vertex in the center of the n-gon and adding edges from this vertex to the midpoints of each edge. The lower portion of Figure 7.2 illustrates this quadrilateral decomposition. This subdivision has the nice property that after one round of topological subdivision the refined mesh consists solely of quads.

Having computed the new topology M_k, we complete the subdivision process by computing the vertex positions p_k from coarse vertex positions p_{k-1} via the subdivision relation $p_k = S_{k-1} p_{k-1}$. Of course, in practice we never actually construct the matrix S_{k-1}, in that this matrix consists almost entirely of zeros. Instead, during topological subdivision we compute the geometric position of new vertices in M_k directly from the position of their parent vertices in M_{k-1}. For triangle subdivision, new vertices are positioned at the midpoint of edges in the coarse mesh M_{k-1}. For quad subdivision, new vertices are positioned at the midpoint of edges in M_{k-1} and at the centroid of faces of M_{k-1} using bilinear subdivision.

The advantage of these two types of subdivision is that iterating them leads to increasingly dense topological meshes that are nearly uniform. For example, triangle subdivision of a triangle mesh produces a new triangle mesh. Moreover, all new interior vertices introduced into this new mesh have valence six. For triangle meshes, interior vertices of valence six are *ordinary vertices*. Interior vertices of valence other than six are *extraordinary vertices*. (On the boundaries of a triangle mesh, ordinary vertices have valence four.) Likewise, quad subdivision always produces a quad mesh and introduces only ordinary interior vertices of valence four. During quad subdivision, interior vertices with valence other than four are extraordinary and can only be inherited from the initial mesh.

Under these definitions, only ordinary vertices are introduced by topological subdivision. Furthermore, those extraordinary vertices present in the initial mesh are increasingly isolated as subsequent rounds of topological subdivision are performed. It is this *isolation of extraordinary* vertices that constitutes the foundation of most subdivision schemes for surface meshes. These schemes rely on the fact that almost all of the mesh produced by topological subdivision is uniform. For these portions of the mesh, the rules from uniform subdivision schemes developed in previous chapters can be applied. Figure 7.3 depicts the octahedron and the cube after three rounds of subdivision. Note that the original vertices of these shapes are the only extraordinary vertices in the new, refined mesh.

These topological splits in which edges of the mesh M_{k-1} are split into two edges, both appearing in M_k, are *edge-splitting* subdivisions. Other types of edge-splitting subdivisions are certainly possible. For example, Figure 7.4 shows a two-step refinement process for a triangulation of a square grid. This split, referred to as 4-8 subdivision by Velho et al. [153], involves repeatedly splitting triangles along

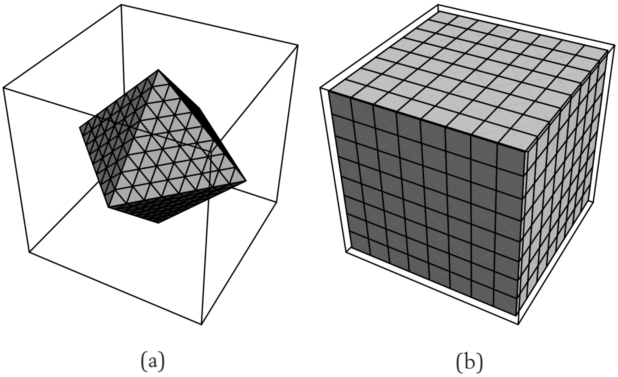


Figure 7.3 An octahedron (a) and cube (b) after three rounds of subdivision. The extraordinary vertices are separated by a uniform mesh.

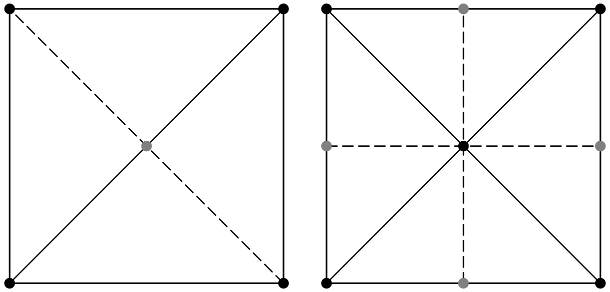


Figure 7.4 The two steps of 4-8 subdivision of triangle meshes.

their longest edge. The authors show that several well-known quad schemes can be reproduced using 4-8 subdivision and also argue that 4-8 subdivision is well suited for adaptive applications.

We conclude this section by considering some practical requirements for implementing edge-splitting subdivision. Given the topology M_{k-1} for a coarse mesh, we can generate the topology M_k for the refined mesh by splitting each triangle or quad in M_{k-1} into its children. The decomposition of a single triangle (or quad) in M_{k-1} into its children is relatively easy. The only difficult detail lies in correctly generating new indices for the vertices of M_k that lie at the midpoints of edges in M_{k-1}. For these "edge" vertices in M_k, several faces in M_{k-1} may contain this edge and spawn new faces in M_k that contain this vertex. Thus, we need to ensure that all of these faces in M_k use the same vertex index.

One simple solution is to store the indices for "edge" vertices in M_k in a hash table. To look up the index for a given "edge" vertex in M_k, we query the hash table with the indices of the endpoints of the edge in M_{k-1} containing the "edge" vertex. If the table entry is uninitialized, the vertex is assigned a new index and that index is stored in the hash table. Otherwise, the hash table returns the previously stored index for the edge vertex. A global counter can be used to assign new indices and to keep track of the total number of vertices.

7.2 Smooth Subdivision for Quad Meshes

For both triangle and quad meshes, topological subdivision produces meshes that are tantalizingly close to uniform. The resulting meshes M_k have the property that they are uniform everywhere except at extraordinary vertices, which themselves are restricted to be a subset of the initial vertices of M_0. For uniform portions of M_k, the uniform subdivision rules developed in the previous chapters can be used to produce smooth limit surfaces. Only near the extraordinary vertices of M_k are new, nonuniform rules needed. Our approach in this section is to develop a geometric characterization of the bicubic subdivision rules for uniform quad meshes, and to then generalize these rules to extraordinary vertices of nonuniform meshes.

7.2.1 Bilinear Subdivision Plus Quad Averaging

To develop a geometric characterization for bicubic subdivision, we first return to the subdivision rules for cubic B-splines. The Lane and Riesenfeld algorithm of Chapter 2 subdivided a polygon p_{k-1} and produced a new polygon p_k via the subdivision relation

$$p_k = \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & 0 & 0 & \cdot \\ \cdot & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & \cdot \\ \cdot & 0 & \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & 0 & \cdot \\ \cdot & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & \cdot \\ \cdot & 0 & 0 & \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix} p_{k-1} == \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & 0 & 0 & \cdot \\ \cdot & 0 & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & 0 & \cdot \\ \cdot & 0 & 0 & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix} \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & 1 & 0 & 0 & \cdot \\ \cdot & \frac{1}{2} & \frac{1}{2} & 0 & \cdot \\ \cdot & 0 & 1 & 0 & \cdot \\ \cdot & 0 & \frac{1}{2} & \frac{1}{2} & \cdot \\ \cdot & 0 & 0 & 1 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix} p_{k-1}.$$

Note that on the right-hand side of this relation the subdivision matrix for cubic B-splines has been factored into the subdivision matrix for linear B-splines, followed by averaging via the mask $\{\frac{1}{4}, \frac{1}{2}, \frac{1}{4}\}$. The subdivision rules for bicubic B-splines exhibit a similar structure. These subdivision rules can be factored into two separate transformations: bilinear subdivision, followed by averaging with the tensor product of the univariate mask with itself; that is, the two-dimensional mask,

$$\begin{pmatrix} \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \end{pmatrix}.$$

Figure 7.5 illustrates this factorization for a single step of bicubic subdivision. A coarse mesh is first split using bilinear subdivision. The resulting mesh is then smoothed using the averaging mask given previously.

There are several advantages of expressing bicubic subdivision as bilinear subdivision followed by averaging. First, bilinear subdivision neatly encapsulates the

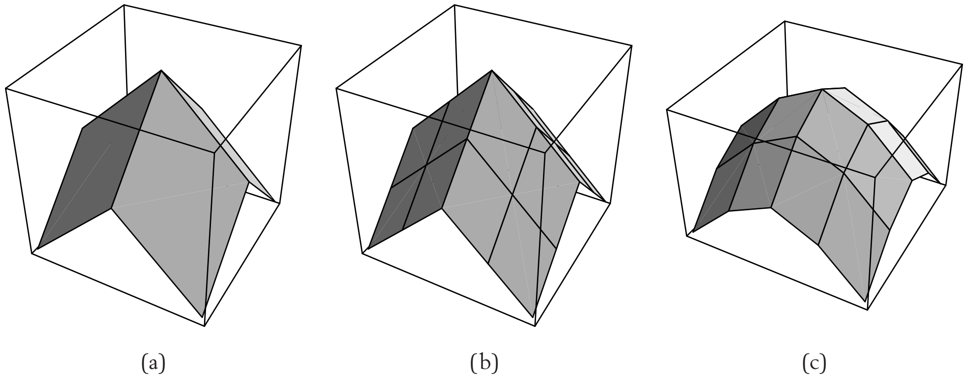


Figure 7.5 Initial shape (a), result of bilinear subdivision (b), result of quad averaging (c).

topological modifications of the mesh, with the latter averaging step only perturbing the geometric positions of the resulting vertices. Another advantage of this decomposition is that there is only a single averaging mask for uniform tensor product meshes (as opposed to three types of subdivision rules for bicubic B-splines). This simplification allows us to focus on developing a generalization of this averaging rule that reproduces the tensor product rule on tensor product meshes, and that yields smooth limit surfaces for arbitrary quad meshes. As we shall see, this generalization (developed simultaneously by Morin et al. [107], Zorin and Schröder [170], and Stam [146]) is remarkably simple.

The key to finding a suitable averaging operator for non-tensor product meshes is to understand the structure of the averaging operator in the tensor product case. In the univariate case, the subdivision rule for cubic B-splines can be expressed as linear subdivision followed by averaging with the mask $\{\frac{1}{4}, \frac{1}{2}, \frac{1}{4}\}$. Note that this mask can be rewritten as $\frac{1}{2}\{\frac{1}{2}, \frac{1}{2}, 0\} + \frac{1}{2}\{0, \frac{1}{2}, \frac{1}{2}\}$. This new expression has a simple geometric interpretation: reposition a vertex at the midpoint of the midpoints of the two segments containing that vertex. In the bivariate case, the bivariate averaging mask can be decomposed into the sum of four submasks:

$$\begin{pmatrix} \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \end{pmatrix} == \frac{1}{4}\left(\begin{pmatrix} 0 & 0 & 0 \\ \frac{1}{4} & \frac{1}{4} & 0 \\ \frac{1}{4} & \frac{1}{4} & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{4} & \frac{1}{4} \\ 0 & \frac{1}{4} & \frac{1}{4} \end{pmatrix} + \begin{pmatrix} 0 & \frac{1}{4} & \frac{1}{4} \\ 0 & \frac{1}{4} & \frac{1}{4} \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} \frac{1}{4} & \frac{1}{4} & 0 \\ \frac{1}{4} & \frac{1}{4} & 0 \\ 0 & 0 & 0 \end{pmatrix}\right). \tag{7.1}$$

This decomposition has a geometric interpretation analogous to the univariate case: reposition a vertex at the centroid of the centroids of the four quadrilaterals that contain the vertex. This geometric interpretation of the tensor product bivariate rule leads to our general rule for smoothing a mesh of quadrilaterals:

Quad averaging: Given a vertex v, compute the centroids of those quads that contain v. Reposition v at the centroid of these centroids.

Due to its simplicity, quad averaging can be implemented in a very straightforward manner, with a minimal amount of topological computation. Given a mesh $\{\mathsf{M}_k, \tilde{\mathsf{p}}_k\}$ produced by bilinear subdivision of $\{\mathsf{M}_{k-1}, \mathsf{p}_{k-1}\}$, first compute val[v], the number of quads in M_k that contain the vertex v. This quantity can easily be computed during a single pass through M_k (or maintained during topological subdivision). Next, initialize a table of new vertex positions p_k to be zero. Finally, make a second pass through M_k. For each quad in M_k, compute the centroid cent of its vertices in $\tilde{\mathsf{p}}_k$ and update the position of vertex v of the quad via

$$\mathsf{p}_k[\![\mathsf{v}]\!] += \frac{\mathsf{cent}}{\mathsf{val}[\mathsf{v}]}, \tag{7.2}$$

where $\mathsf{p}_k[\![\mathsf{v}]\!]$ is the entry of p_k corresponding to vertex v. Because there are exactly val[v] quads containing v, $\mathsf{p}_k[\![\mathsf{v}]\!]$ accumulates the centroid of the val[v] centroids.

The combination of bilinear subdivision plus quad averaging yields a smooth subdivision scheme for quad meshes. Figure 7.6 shows the effect of applying this subdivision scheme to an initial surface mesh consisting of six squares forming a cube. The leftmost figure is the initial cube. The next figure to the right is the result of bilinear subdivision. The middle figure is the result of next applying quad averaging and corresponds to one round of subdivision applied to the initial cube. The remaining two figures are the results of applying bilinear subdivision and then averaging to the middle figure. Note that the resulting mesh has a tensor product structure everywhere except at the eight extraordinary vertices. Because bilinear subdivision followed by quad averaging reproduces the subdivision rule for bicubic B-splines on tensor product meshes, the limit surfaces are $\mathcal{C}^2$ everywhere except at

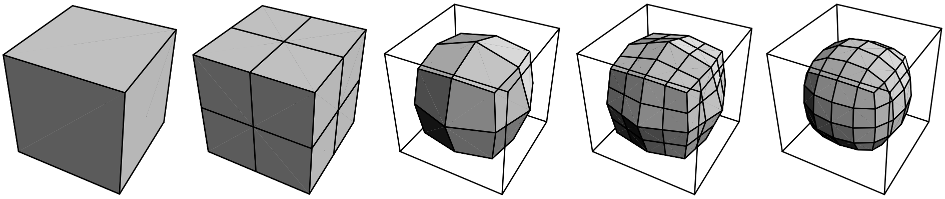


Figure 7.6 Two rounds of subdivision of a cube expressed as alternating steps of bilinear subdivision and quad averaging.

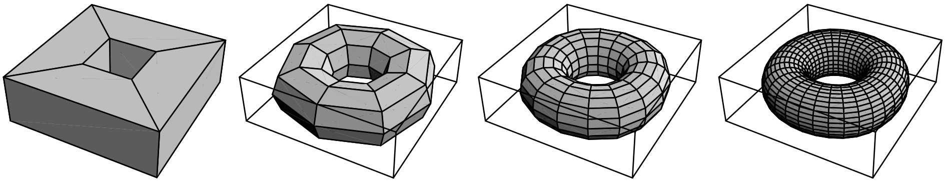


Figure 7.7 Three rounds of subdivision of a doughnut-shaped mesh using bilinear subdivision combined with quad averaging.

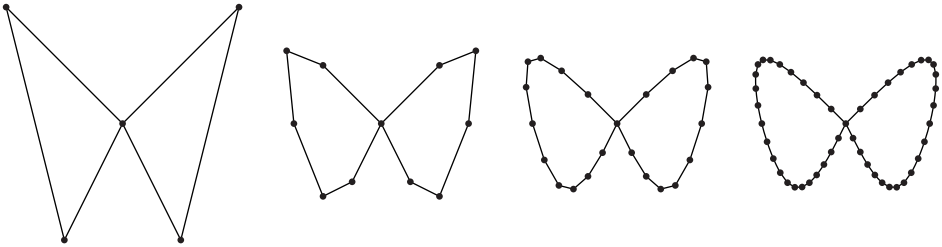


Figure 7.8 Subdivision of a non-manifold curve network using linear subdivision plus averaging.

extraordinary vertices. At these extraordinary vertices, the scheme produces limit surfaces that are provably $\mathcal{C}^1$ (see Chapter 8 for a formal analysis).

Figure 7.7 shows another example of bilinear subdivision with quad averaging applied to a doughnut-shaped quad mesh. Note that the initial surface mesh has only valence-four vertices. As a result, the limit surface is not only smooth but has continuous curvature (i.e., the surface is $\mathcal{C}^2$). Observe that the limit surface is not precisely a torus, but instead has a slightly flattened shape along the sides of the bounding box. The problem of constructing exact surfaces of revolution is considered later in this chapter.

One particular advantage of this subdivision scheme is that it makes no restrictions on the local topology of the mesh. In particular, the scheme can be applied without any change to meshes with non-manifold topology. In the univariate case, non-manifold topology gives rise to curve networks with $n > 2$ segments meeting at a vertex. Applying this averaging rule in conjunction with linear subdivision yields a subdivision rule at a non-manifold vertex that takes $\frac{3}{4}$ of the original vertex position plus $\frac{1}{4n}$ multiplied by the position of each of its n neighbors. Figure 7.8 shows an example of the scheme applied to a non-manifold, butterfly-shaped curve network. The method yields similarly nice results for non-manifold surface meshes.

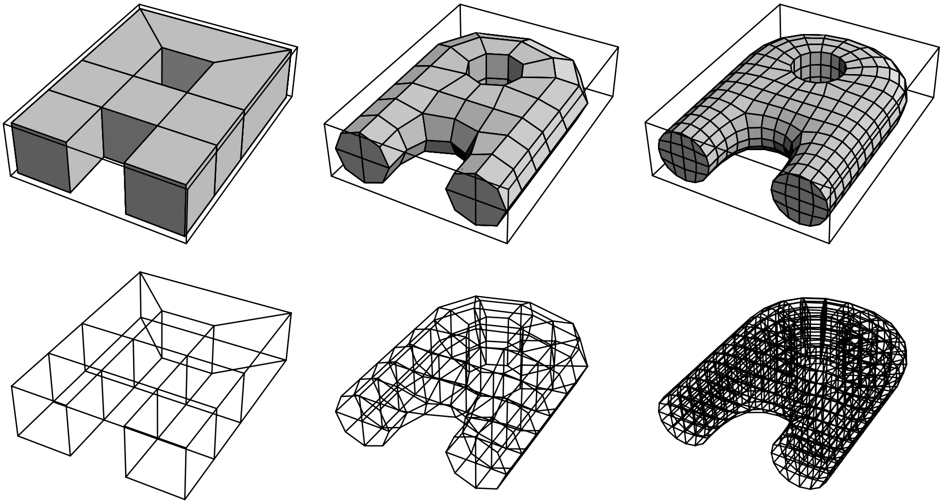


Figure 7.9 Two rounds of subdivision for an A-shaped volume mesh.

Another advantage of this scheme is that it easily generalizes to higher dimensions. For example, MacCracken and Joy [101] first proposed a subdivision scheme for volume meshes that could be used to represent free-form deformations. Bajaj et al. [5] propose an improved variant of this volume scheme that consists of trilinear subdivision followed by hexahedral averaging. Figure 7.9 shows two rounds of this scheme applied to an A-shaped volume mesh consisting initially of eight hexahedra. The lower portion of the figure shows wireframe plots of the internal hexahedra.

7.2.2 Comparison to Other Quad Schemes

Traditionally, the subdivision rules for surface schemes have been expressed in terms of weights that combine the action of bilinear subdivision and averaging simultaneously. For example, Catmull and Clark [16] were the first to derive a smooth scheme for quad meshes that generalizes bicubic subdivision for B-splines. In [81], Kobbelt describes a smooth interpolating scheme for quad meshes that generalizes the tensor product version of the four-point rule. In each paper, topological and geometric subdivision are performed together in a single step. The positions p_k of "new" vertices in M_k are specified as combinations of positions p_{k-1} of "old" vertices in M_{k-1} via the subdivision relation $p_k = S_{k-1} p_{k-1}$.

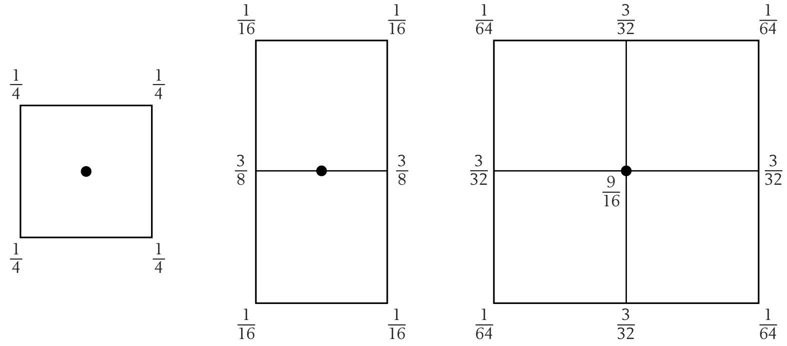
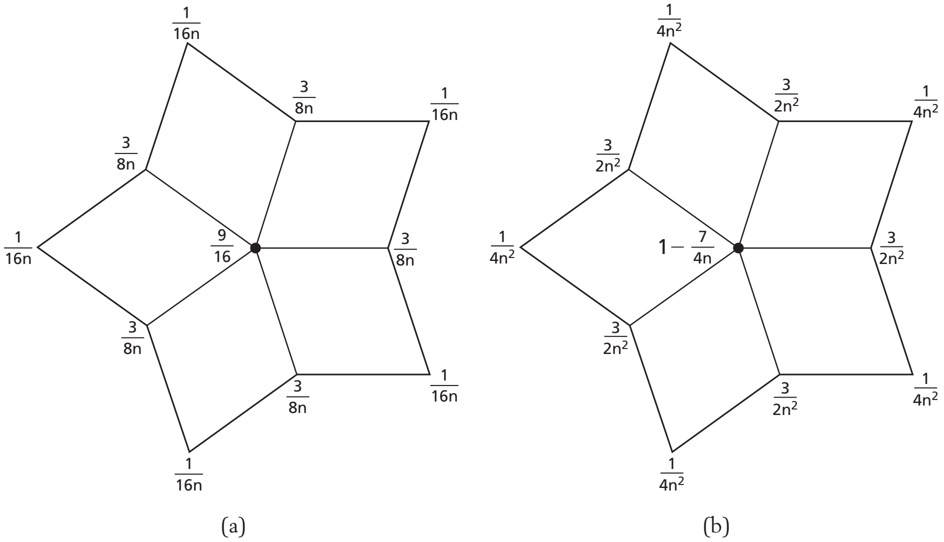


Figure 7.10 Face, edge, and vertex subdivision rules for bicubic B-splines.

Each row of the subdivision matrix S_{k-1} specifies the position of a vertex in M_k relative to the positions of vertices in M_{k-1}. These rows (i.e., subdivision rules associated with S_{k-1}) typically come in three variants: a "face" subdivision rule that specifies the relative positions of vertices in M_k that lie on the faces of M_{k-1}, an "edge" subdivision rule that specifies the relative positions of vertices in M_k that lie on edges of M_{k-1}, and a "vertex" subdivision rule that specifies the positions of vertices in M_k that lie near vertices of M_{k-1}. Given a vertex in M_k, one way to visualize the structure of its corresponding subdivision rule is to plot a portion of the nearby mesh in M_{k-1} and to attach to each vertex in this plot the corresponding weight used in computing the position of the new vertex. Figure 7.10 shows a plot of the three subdivision rules for bicubic B-splines. The grid lines correspond to the mesh M_{k-1}, whereas the dark dot is the vertex of M_k in question.

To apply these subdivision rules to an arbitrary quad mesh, we need only generalize the "vertex" rule from the valence-four case to vertices of arbitrary valence. Specifically, this modified "vertex" rule specifies the new position of v as a weighted combination of vertex positions in ring[v] on M_{k-1}. If the vertex v happens to be of valence four, the weights reproduce the uniform rule of Figure 7.10. If the vertex v is extraordinary, the weights should lead to a subdivision scheme that is smooth at the extraordinary vertex.

Our next task is to derive the subdivision rule for an extraordinary vertex of valence n for bilinear subdivision plus averaging. Observe that after bilinear subdivision the n vertices that are edge-adjacent to v in M_k lie on the midpoints of the n edges incident on v in M_{k-1}. The n vertices that are face-adjacent to v in M_k lie at the centroid of the quadrilateral faces containing v in M_{k-1}. After quad averaging, the new position of v is $\frac{1}{16n}$ times the position of each of its n face-adjacent neighbors plus $\frac{3}{8n}$ times the position of its n edge-adjacent neighbors plus $\frac{9}{16}$ times



(a)

(b)

Figure 7.11 Vertex rules for bilinear subdivision plus averaging (a) and standard Catmull-Clark subdivision (b).

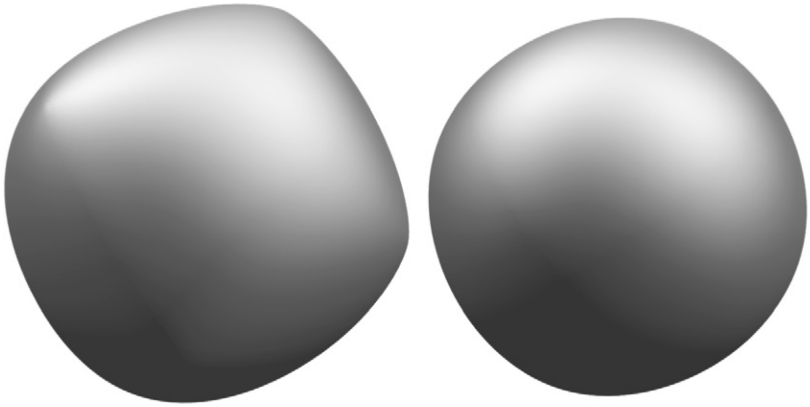


Figure 7.12 Bilinear subdivision plus quad averaging (left) and standard Catmull-Clark (right) applied to an initial cube.

the original position of v. The left-hand image of Figure 7.11 schematically depicts this rule for a vertex of valence n (not just valence five). The right-hand image of Figure 7.11 schematically depicts the standard Catmull-Clark rule. (Catmull and Clark also mentioned the left-hand rule in their original paper [16] but were unaware of its expression in terms of bilinear subdivision plus averaging.)

Figure 7.12 shows a high-resolution rendering of the limit surface produced by each of these schemes when applied to an initial cube. Note that the standard

Catmull-Clark rule tends to produce "rounder" surfaces than the rule based on bilinear subdivision and averaging. In fact, this behavior is not an accident: inducing this "roundness" in the resulting limit surface was one of the main criteria that Catmull and Clark used in the design of their subdivision scheme. Luckily, the standard Catmull-Clark rule can also be expressed in terms of bilinear subdivision followed by a modified form of averaging. (See the associated implementation for details (♚).)

7.2.3 Weighted Averaging for Surfaces of Revolution

The previous schemes yielded smooth limit surfaces, even in the presence of extraordinary vertices. For quad meshes, the final limit surface is a bicubic B-spline away from extraordinary vertices. A fundamental limitation of this scheme is that many surfaces of great practical importance (such as spheres, cylinders, and tori) cannot be exactly represented as bicubic B-spline surfaces. This limitation is due to the fact that a circle does not possess a polynomial parameterization (see Abhyankar and Bajaj [1] for more details). However, because circles do possess rational parameterizations, one possible solution would be to move from the polynomial domain to the rational domain. The resulting rational subdivision scheme would then manipulate the control points in homogeneous space. Sederberg et al. [139] construct one such subdivision scheme based on a generalization of nonuniform rational B-spline surfaces (NURBS).

Such a rational approach has a major drawback. Rational parameterizations for circles are nonuniform. Using a rational quadratic parameterization, only a portion of the circle (such as one or two quadrants) can be modeled using a single parameterization. As a result, rational parameterizations of spheres, tori, and other surfaces of revolution typically only cover a portion of the surface. Several parameterizations are required to cover the entire surface. A better solution is to use the arc-length parameterization of the circle based on the functions Sin[x] and Cos[x]. This parameterization is uniform and covers the entire circle. Luckily, a univariate subdivision scheme that reproduces Sin[x] and Cos[x] was developed in section 4.4. This curve scheme had a subdivision matrix S_{k-1} whose columns have the form

$$\frac{1}{4+4\sigma_k}(1, 2+2\sigma_k, 2+4\sigma_k, 2+2\sigma_k, 1), \tag{7.3}$$

where the tension parameter σ_k is updated via the rule $\sigma_k = \sqrt{\frac{1+\sigma_{k-1}}{2}}$. Given an initial tension $\sigma_0 == 1$, the scheme produces cubic B-splines. For initial tensions

$\sigma_0 > 1$, the scheme produces splines in tension. For initial tensions $-1 \leq \sigma_0 < 1$, the scheme produces splines whose segments are linear combinations of linear functions and the trigonometric functions $\mathsf{Sin}[x]$ and $\mathsf{Cos}[x]$. Our task here is to develop a geometric interpretation for this mask, and then to generalize this interpretation to arbitrary quad meshes. The resulting scheme, introduced in Morin et al. [107], is a simple nonstationary subdivision scheme that can represent general surfaces of revolution.

To begin, we note that the action of the scheme of equation 7.3 can be decomposed into three separate operations. Equation 7.4 shows this decomposition in terms of the subdivision matrix S_{k-1}. (Note that only finite portions of the infinite matrices are shown.) The columns of the matrix S_{k-1} on the left-hand side of equation 7.4 are exactly two-shifts of the mask of equation 7.3. The subdivision mask of equation 7.3 can be expressed as linear subdivision followed by two rounds of averaging (the right-hand side in equation 7.4):

$$\begin{pmatrix} \frac{1}{4+4\sigma_k} & \frac{1+2\sigma_k}{2+2\sigma_k} & \frac{1}{4+4\sigma_k} & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{4+4\sigma_k} & \frac{1+2\sigma_k}{2+2\sigma_k} & \frac{1}{4+4\sigma_k} & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{4+4\sigma_k} & \frac{1+2\sigma_k}{2+2\sigma_k} & \frac{1}{4+4\sigma_k} \end{pmatrix} == \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix}$$
$$\begin{pmatrix} \frac{1}{1+\sigma_k} & \frac{\sigma_k}{1+\sigma_k} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{\sigma_k}{1+\sigma_k} & \frac{1}{1+\sigma_k} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{1+\sigma_k} & \frac{\sigma_k}{1+\sigma_k} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\sigma_k}{1+\sigma_k} & \frac{1}{1+\sigma_k} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{1+\sigma_k} & \frac{\sigma_k}{1+\sigma_k} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{\sigma_k}{1+\sigma_k} & \frac{1}{1+\sigma_k} \end{pmatrix} \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix}. \tag{7.4}$$

Given a coarse polygon p_{k-1}, let $\widetilde{\mathsf{p}}_k$ be the new polygon produced by linear subdivision. The first round of averaging applies weighted combinations of the tension σ_k to the vertices of $\widetilde{\mathsf{p}}_k$. For each edge $\{\mathsf{v}, \mathsf{u}\}$ in M_k, this weighted averaging computes a new point of the form

$$\frac{\sigma_k \widetilde{\mathsf{p}}_k[\![\mathsf{v}]\!] + \widetilde{\mathsf{p}}_k[\![\mathsf{u}]\!]}{\sigma_k + 1},$$

where v is a vertex of M_{k-1} and u is a vertex of M_k that lies on an edge of M_{k-1}. Note that this weighted rule reverses the averaging mask on consecutive segments of M_k.

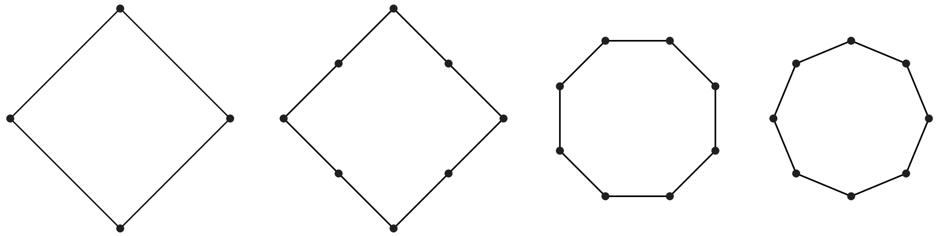


Figure 7.13 The three separate transformations comprising a single round of subdivision.

Finally, the second round of averaging places the vertices of p_k at the midpoints of these weighted averages.

For example, Figure 7.13 shows the three separate transformations of equation 7.4 used to perform a single round of subdivision applied to a diamond. The leftmost polygon p_0 is a diamond whose initial tension is $\sigma_0 = \mathrm{Cos}[\frac{2\pi}{4}] == 0$. The polygon immediately to its right, $\widetilde{p}_1$, is the result of applying linear subdivision. The next polygon (a regular octagon) is the result of weighted averaging using the tension $\sigma_1 = \sqrt{\frac{1}{2}}$. Note that applying midpoint averaging here (i.e., $\sigma_1 == 1$) would not produce a regular octagon. The rightmost polygon, p_1, is another regular octagon that results from a final round of averaging using midpoints.

The beauty of this reformulated curve scheme is that it has a simple generalization to quad meshes. For each quad in the initial mesh M_0, we assign two tension parameters, σ_0 and ρ_0, one per pair of parallel edges in the quad. Each round of subdivision takes a coarse polyhedron $\{M_{k-1}, p_{k-1}\}$; applies bilinear subdivision to produce a finer, intermediate mesh $\{M_k, \widetilde{p}_k\}$; and then applies weighted quad averaging to produce the final polyhedral mesh $\{M_k, p_k\}$. During bilinear subdivision, the tensions σ_{k-1} and ρ_{k-1} for a given quad are updated via $\sigma_k = \sqrt{\frac{1+\sigma_{k-1}}{2}}$ and $\rho_k = \sqrt{\frac{1+\rho_{k-1}}{2}}$. These new tensions are then assigned to the appropriate pairs of edges for each of the four quad subfaces. These new tensions are then used to compute *weighted centroids* of each quad during the averaging phase. Given a quad $\{v, u, t, s\}$ in M_k, this weighted centroid is the tensor product of the univariate weighting rule and has the form

$$\frac{\rho_k(\sigma_k \widetilde{p}_k[\![v]\!] + \widetilde{p}_k[\![u]\!]) + (\sigma_k \widetilde{p}_k[\![t]\!] + \widetilde{p}_k[\![s]\!])}{(\sigma_k + 1)(\rho_k + 1)}, \tag{7.5}$$

where v is a vertex of M_{k-1}, s lies on a face of M_{k-1}, and u and t lie on edges of M_{k-1} tagged by ρ_k and σ_k, respectively. (The necessary orientation information for this rule can be maintained for each quad by storing its topological indices in a

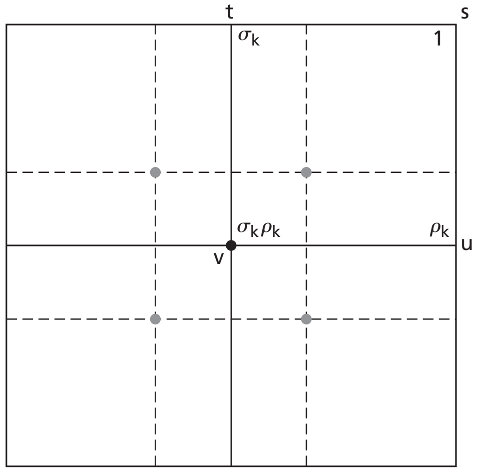


Figure 7.14 Computing the weighted average of the vertices on the solid grid. Gray vertices are intermediate points used in computing the position of the black vertex.

systematic order.) The second round of averaging is done by computing the centroid of these weighted centroids. The update rule for these two rounds of averaging is exactly that of equation 7.2, in which cent is the weighted centroid of equation 7.5.

Figure 7.14 illustrates weighted quad averaging in action. The solid lines denote the mesh $\{M_k, \tilde{p}_k\}$ produced by bilinear subdivision. To compute the new vertex position of the central black vertex, weighted quad averaging computes the weighted gray centroids for each quad in M_k. Note that the weight $\sigma_k\rho_k$ is always attached to the vertex of the quad that is also a vertex of M_{k-1}. Finally, the central black vertex is repositioned at the centroid of the gray vertices.

This subdivision scheme has several remarkable properties. Because the initial tensions σ_0 and ρ_0 are greater than or equal to -1, the scheme involves only convex combinations of control points. If all initial tensions are chosen to be 1, the scheme reduces to bilinear subdivision plus unweighted averaging. For tensor product meshes with a single global tension for each coordinate, the resulting limit surfaces are the tensor product of two $\mathcal{C}^2$ curve schemes. Consequently, the surfaces must be $\mathcal{C}^2$. For tensor product meshes whose associated tensions are not the tensor of two sets of curve tensions, the scheme is no longer necessarily $\mathcal{C}^2$. However, numerical tests show that the limit surfaces still appear to be $\mathcal{C}^2$. At extraordinary vertices, the limit surfaces remain $\mathcal{C}^1$. The smoothness of this scheme is analyzed in greater detail in Morin et al. [107].

To complete this section, we describe a simple construction for generating surface meshes whose limit surfaces are surfaces of revolution. Given a control polygon p_0 (and a list of tensions for its segments) in the *xz* plane, we construct a tensor product surface mesh s_0 (and associated set of tensions) in *xyz* space whose limit s_∞ is the surface formed by revolving p_∞ around the *z* axis. We construct s_0 by making m copies of p_0 and revolving the i th copy around the *z* axis by $\frac{2\pi i}{m}$ radians. If we decompose p_0 into its components in *x* and *z* (i.e., $p_0 = \{p_0^x, p_0^z\}$), the i th copy of p_0 has the form

$$s_0 = \left\{\mathrm{Cos}\left[\frac{2\pi i}{m}\right] p_0^x, \mathrm{Sin}\left[\frac{2\pi i}{m}\right] p_0^x, p_0^z\right\}.$$

Joining these m revolved curves (and their tensions) yields a tensor product mesh. If the tensions in the direction of revolution for this mesh are initialized to $\mathrm{Cos}[\frac{2\pi}{m}]$, cross sections of the associated limit surface s_∞ taken orthogonal to the *z* axis are circles (i.e., s_∞ is a surface revolution).

However, as observed in section 4.4, this limit surface s_∞ is not the surface of revolution formed by revolving p_∞ around the *z* axis. Due to the averaging that takes place between adjacent revolved copies of p_0, s_∞ is "squeezed" toward the *z* axis. To counteract this effect, we must scale the *x* and *y* coordinates of the initial polyhedron s_0 by a factor of $\frac{2\pi}{m}\mathrm{Csc}[\frac{2\pi}{m}]$ (see [107] for a derivation of this constant). For $m \geq 4$, this scaled version of s_0 has the property that s_∞ is the surface of revolution formed by revolving the limit curve p_∞ around the *z* axis.

Figures 7.15 and 7.16 show an example of this process used to construct a pawn. The leftmost polygon in Figure 7.15 is the control polygon p_0. The polygons to its right are refined polygons p_1 and p_2. The leftmost surface in Figure 7.16 is the scaled tensor product surface mesh s_0 formed by revolving p_0 around the vertical *z* axis. Note that a vertical cross section of s_0 is "fatter" than p_0 due to the scaling in the *x* direction by $\frac{2\pi}{m}\mathrm{Csc}[\frac{2\pi}{m}]$. Subsequent meshes s_1 and s_2 appearing to the right are the result of subdivision using weighted averaging. Note that vertical cross sections of these meshes s_k are converging to p_k. (This example also revolves crease vertices on p_0 to introduce crease edges on the limit surface s_∞. See the next section for details on creating crease vertices and crease edges.)

Figures 7.17 and 7.18 show two other examples of simple surfaces of revolution. Figure 7.17 shows an octahedron being subdivided into a sphere. Figure 7.18 shows a torus being created through subdivision. In both cases, the original profile curves p_∞ were circles. In the case of the torus, the profile curve does not intersect the *z* axis, and thus the surface of revolution is a smooth, single-sheeted surface.

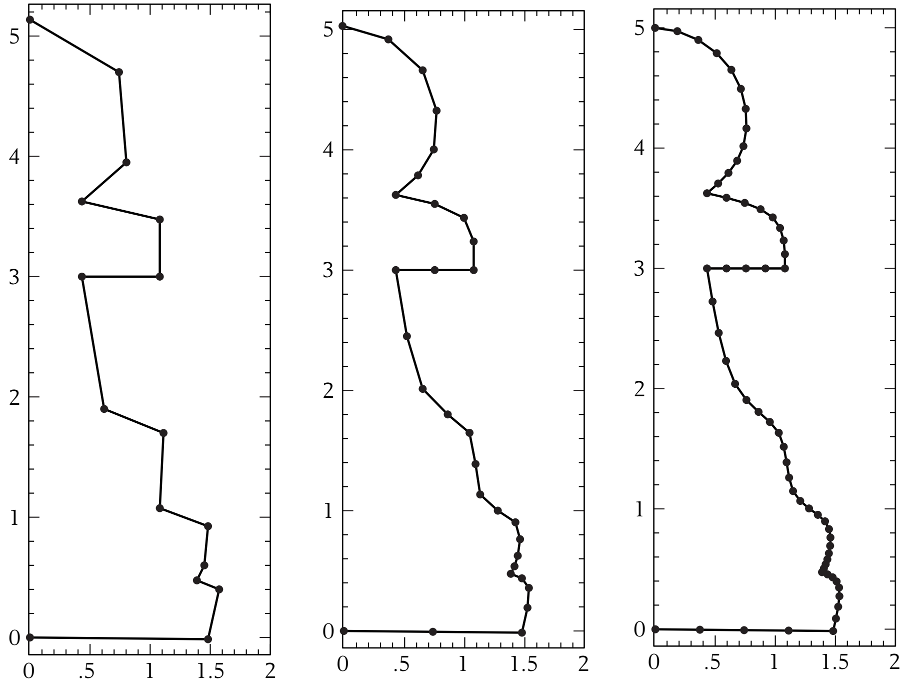


Figure 7.15 Subdivision of the profile curve for a pawn.

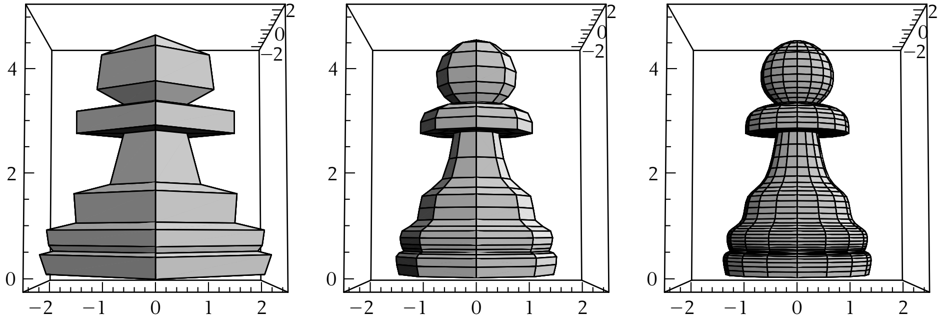


Figure 7.16 A pawn created by subdividing the revolved profile curve.

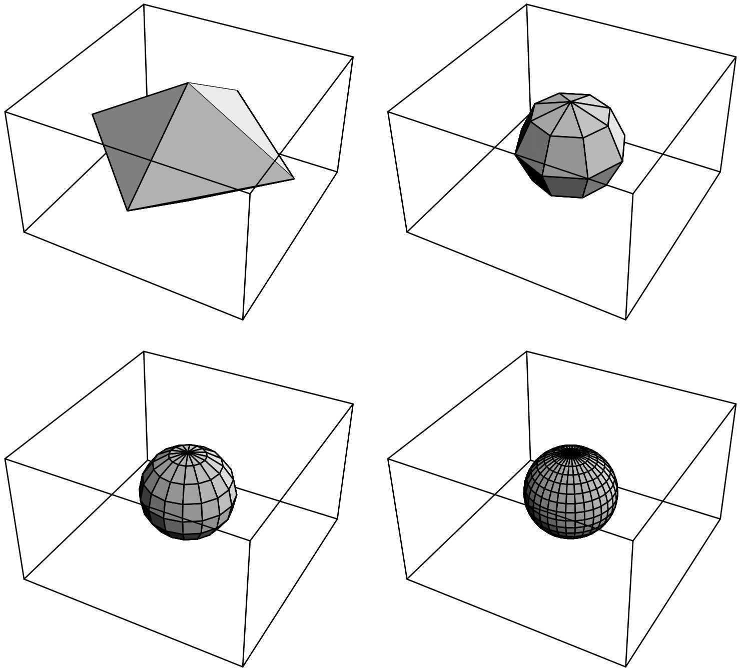


Figure 7.17 Three rounds of subdivision converging to a sphere.

In the case of the sphere, the profile curve is a circle symmetrically positioned on the z axis. Due to this positioning, the resulting surface of revolution is a smooth, double-sheeted surface (i.e., a surface that consists of two copies of the same surface). More generally, revolving any profile curve that is symmetric with respect to the z axis through 2π radians results in a double-sheeted surface.

In practice, we wish to avoid this double-sheetedness when the profile curve p_∞ is symmetric with respect to the z axis. The solution to this problem is to revolve the initial symmetric polygon p_0 through only π radians. Because the polygon p_0 is symmetric with respect to the z axis, the new polyhedron s_0 is closed (and single-sheeted). If the intersection of the z axis and polygon p_0 occurs at vertices of p_0, this quad mesh s_0 has poles at these vertices. These poles consist of a ring of degenerate quads (triangles) surrounding the pole (as shown by the dark lines in Figure 7.19).

Applying bilinear subdivision plus weighted averaging to this single-sheet polyhedron s_0 yields a single-sheeted surface of revolution s_∞ that agrees with the double-sheeted surface of revolution produced by revolving p_0 through 2π radians. The key to this observation is to note that applying this subdivision scheme to

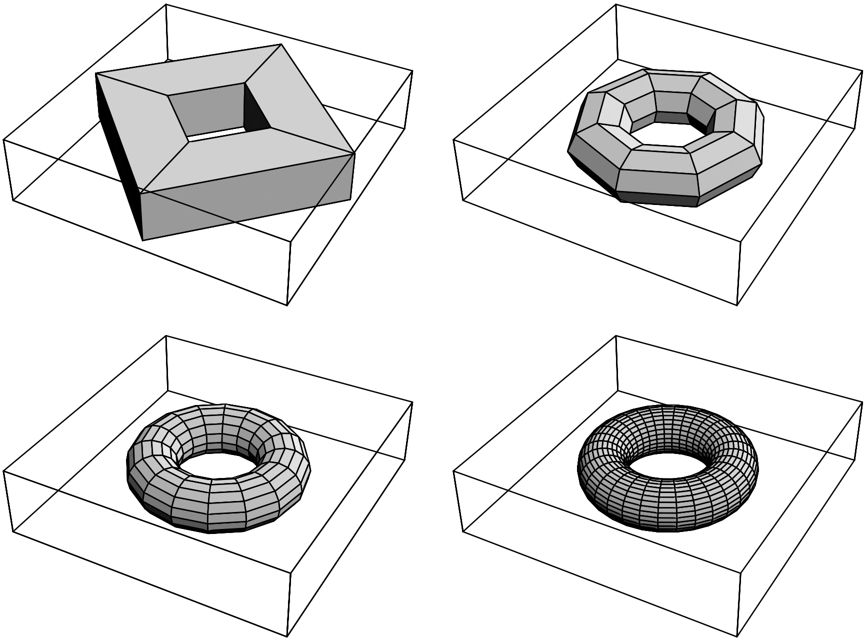


Figure 7.18 Three rounds of subdivision converging to a torus.

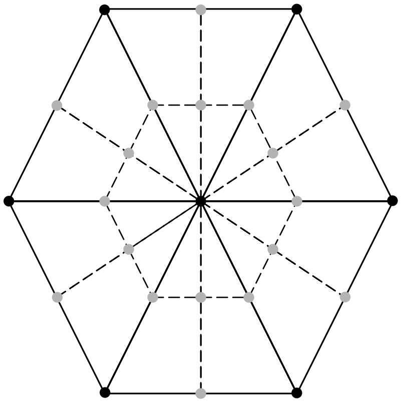


Figure 7.19 Subdividing a degenerate quad mesh around a pole.

both the single-sheeted and double-sheeted variants of s_0 yields the same surface (single-sheeted and double-sheeted, respectively) as long as the degenerate quads at the poles of the single-sheet version are treated as such and are subdivided as shown in Figure 7.19 (the dotted lines).

7.2.4 Averaging for Quad Meshes with Embedded Creases

Up to now, we have focused on constructing subdivision schemes that yield smooth limit surfaces everywhere. However, most real-world objects have boundaries and internal creases where the surface of the object is not smooth. Our goal in this section is to modify the averaging rules developed in the previous two sections to allow for the controlled introduction of "crease" curves and vertices onto the limit surface. The key to this modification is to restrict the application of the averaging rules to a subset of the mesh M_k corresponding to the crease curves and vertices.

To model such features, the topological base mesh M_0 associated with the scheme is expanded to allow the inclusion of vertices and edges, as well as faces. For example, the base mesh M_0 for a quadrilateral surface may contain line segments and vertices, as well as quad faces. These lower-dimensional cells define crease curves and crease vertices in the resulting surface mesh. For example, the initial topology M_0 for the bounded quad surface patch of Figure 7.20 consists of four faces, eight boundary edges, and four corner vertices. If the vertices in this 3×3 grid are numbered $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$, the initial topology M_0 is a list of the form

$$\begin{gathered}\{\{1,2,5,4\},\{4,5,8,7\},\{2,3,6,5\},\{5,6,9,8\},\\ \{1,4\},\{4,7\},\{1,2\},\{2,3\},\{3,6\},\{6,9\},\{7,8\},\{8,9\},\\ \{1\},\{3\},\{7\},\{9\}\}.\end{gathered} \tag{7.6}$$

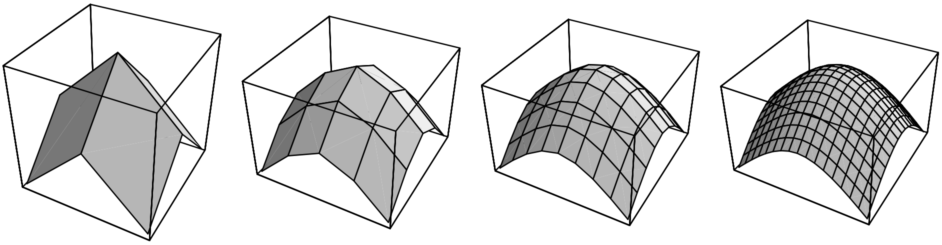


Figure 7.20 Bilinear subdivision plus averaging for a surface patch with boundary creases.

Given this new representation, bilinear subdivision is performed as usual, with each quad split into four quads, each edge split into two edges, and each vertex copied.

Given the mesh M_k produced by bilinear subdivision, we define the dimension dim[v] of a vertex v in M_k to be the dimension of the lowest-dimensional cell in M_k that contains v. In the previous example, dim[v] has the values

$$\begin{array}{rccccccccccc} \mathrm{v} & : & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \mathrm{dim[v]} & : & 0 & 1 & 0 & 1 & 2 & 1 & 0 & 1 & 0 \end{array}.$$

Having computed dim[v] for M_k, the rule for quad averaging at vertex v is modified to use only those cells whose dimension is dim[v]. In particular, the update process of equation 7.2 is adjusted as follows:

Quad averaging with creases: Given a vertex v, compute the centroids of those cells in M_k containing v whose dimension equals dim[v]. Reposition v at the centroid of these centroids.

This modification ensures that the subdivision rules for vertices on a crease are influenced only by vertices on the crease itself. For example, if M_0 contains a vertex (a cell of dimension 0), this modified averaging rule leaves the position of the vertex unchanged and forces the resulting limit surface to interpolate the vertex. If M_0 contains a sequence of edges forming a curve, the resulting limit surface interpolates the cubic B-spline associated with the control points along the edges.

To implement this modified version of averaging, we compute dim[v] and val[v] (now the number of cells of dimension dim[v] containing v) simultaneously during a first pass through M_k. Given these two quantities, the averaging rules described in the previous sections can then be applied. Quad averaging is particularly amenable to this change because the update rule for the vertices of a cell is independent of its dimension. Figure 7.20 shows a surface patch produced by bilinear subdivision plus averaging applied to the example mesh of equation 7.6. The four edges in M_0 force the boundary of the surface patch to interpolate the B-spline curves defined by the boundary of the initial mesh. Similarly, the four vertices in M_0 force the limit surface to interpolate the four corners of the initial mesh.

Figures 7.21 and 7.22 show high-resolution renderings of two more complex objects, a ring and an axe, during subdivision. The ring includes a crease curve where the stone meets the ring and four crease vertices at the corners of the stone. The axe incorporates crease edges at the top and bottom of its handle, crease edges along its blade, and crease vertices at its corners. Note that in both examples a compact polyhedral mesh leads to a nontrivial, realistic model.

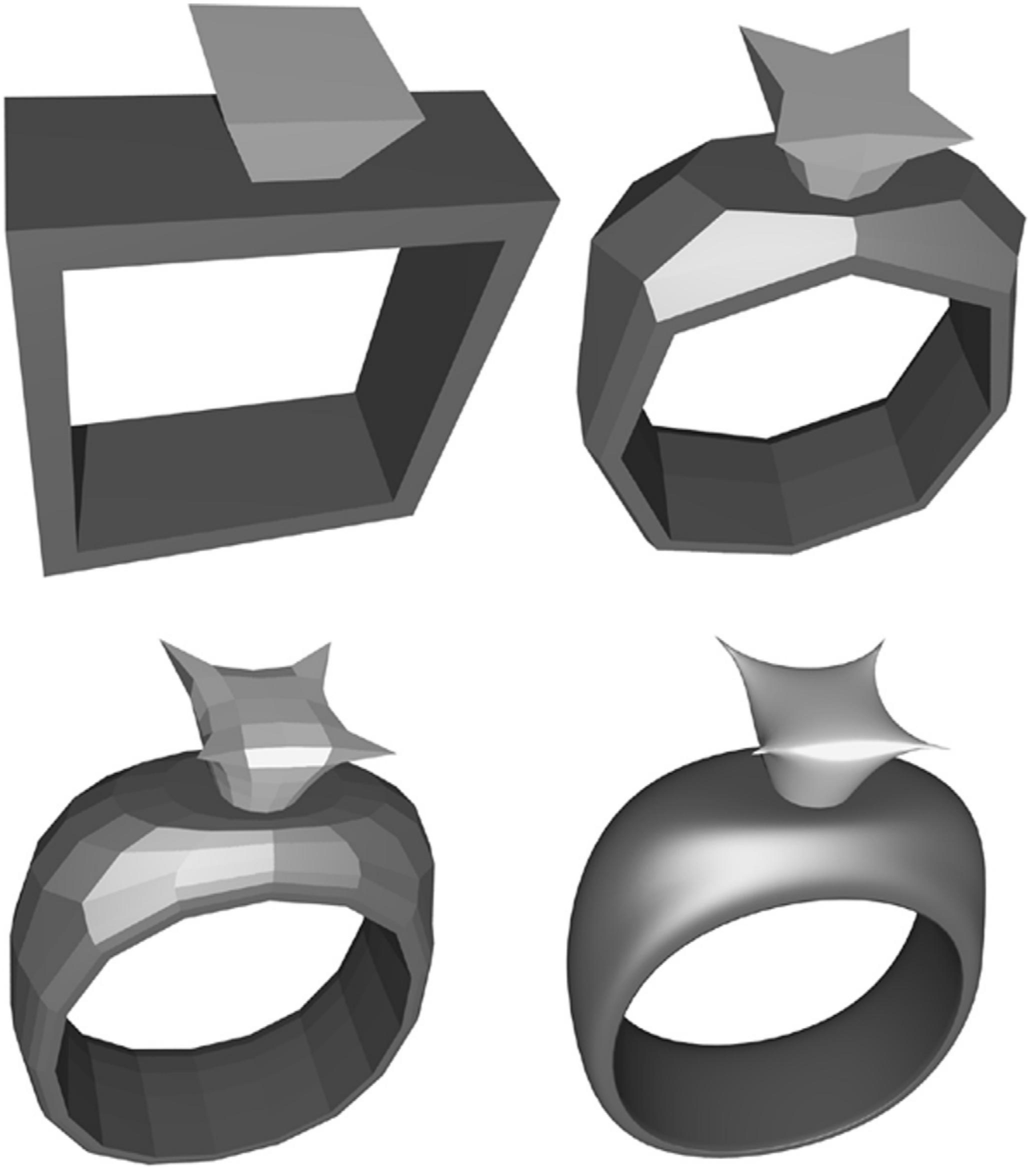


Figure 7.21 A ring modeled using a subdivision surface with creases. (Thanks to Scott Schaefer for his help with this figure.)

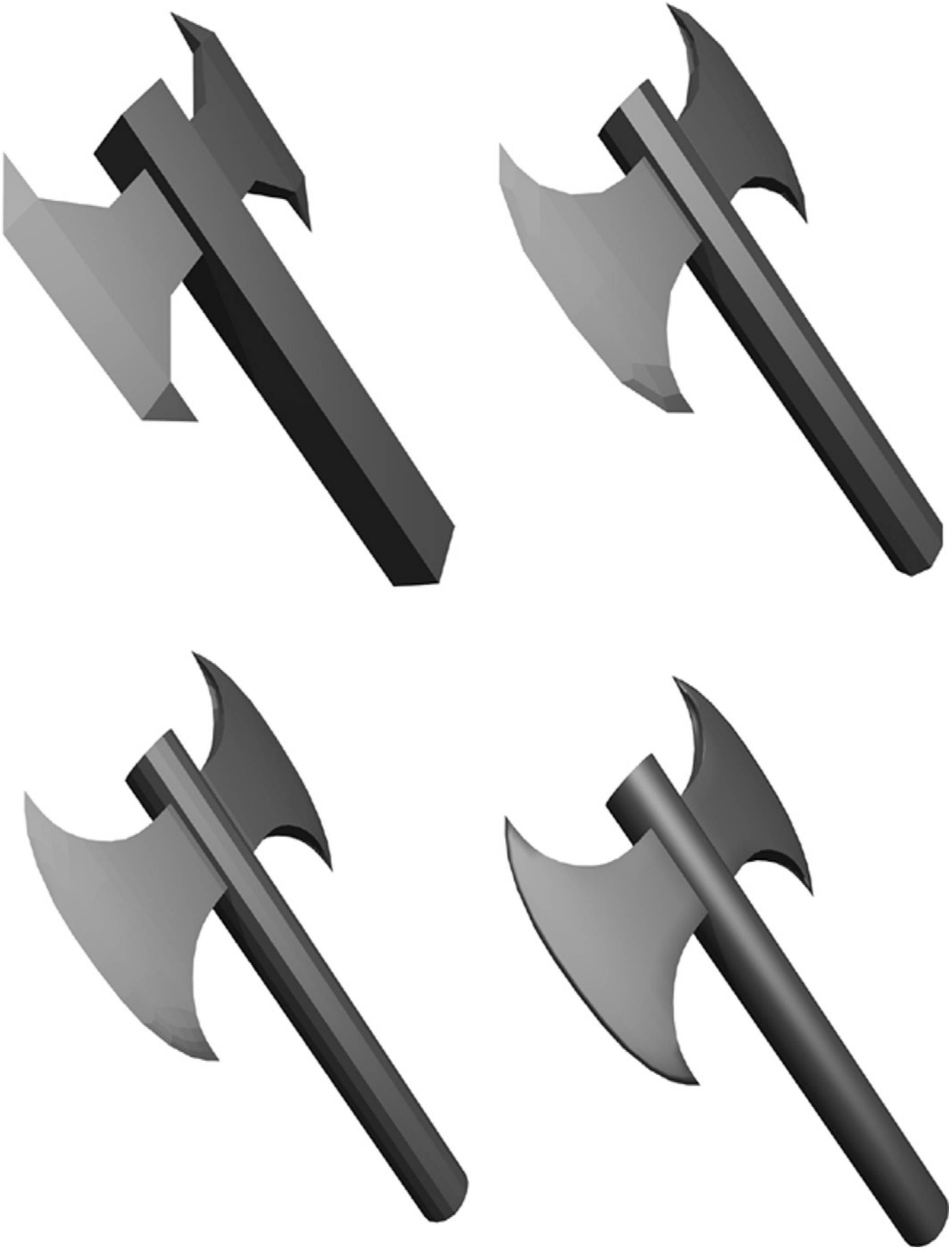


Figure 7.22 An axe modeled using a subdivision surface with creases. (Thanks to Scott Schaefer for his help with this figure.)

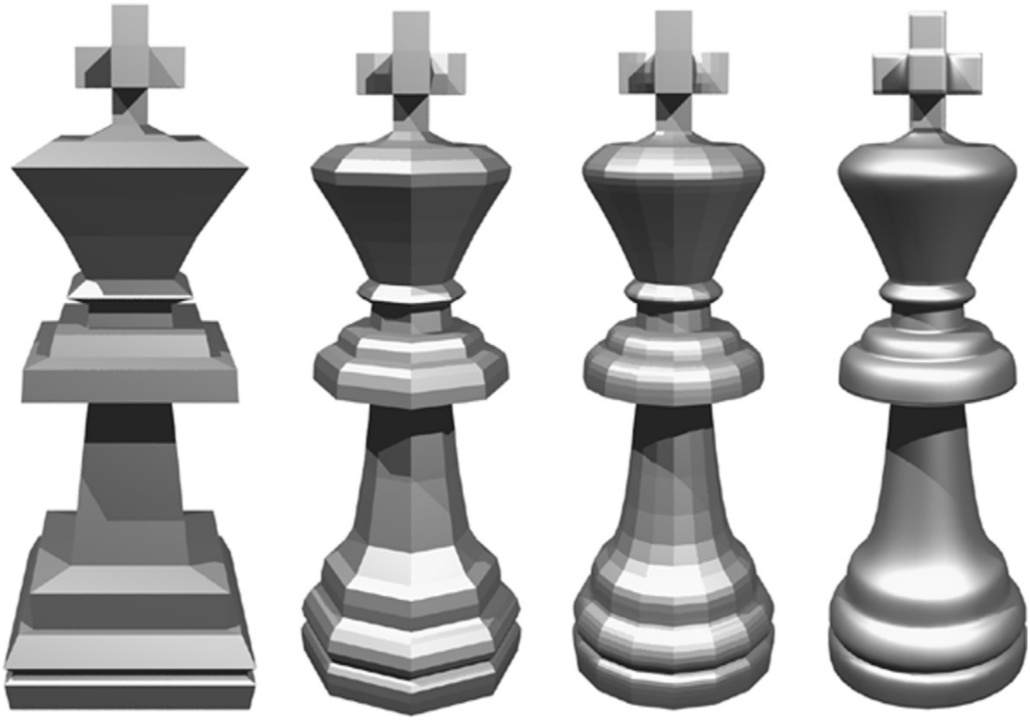


Figure 7.23 A king modeled using weighted subdivision in conjunction with creases. (Thanks to Scott Schaefer for his help with this figure.)

Using crease vertices and crease edges in conjunction with weighted subdivision allows us to model complex shapes using coarse initial meshes. Figure 7.23 shows a high-resolution rendering of another chess piece, a king, during subdivision. This model was created as a surface of revolution and then hand edited to add the three-dimensional cross to the king's head. Figure 7.24 shows a high-resolution rendering of an umbilic torus defined via subdivision. This object was modeled as a twisted surface of revolution in which the profile curve, a triangle with concave edges, was rotated $\frac{2\pi}{3}$ radians around its center during a single sweep around the axis of revolution. Due to this rotation, the sharp edge on the umbilic torus is modeled as a single crease curve that wraps around the axis of revolution three times.

Developing subdivision rules for boundaries and creases of subdivision surfaces is an active area of research. For example, Hoppe et al. [74] first proposed specialized rules that allow the introduction of B-spline crease curves onto triangle

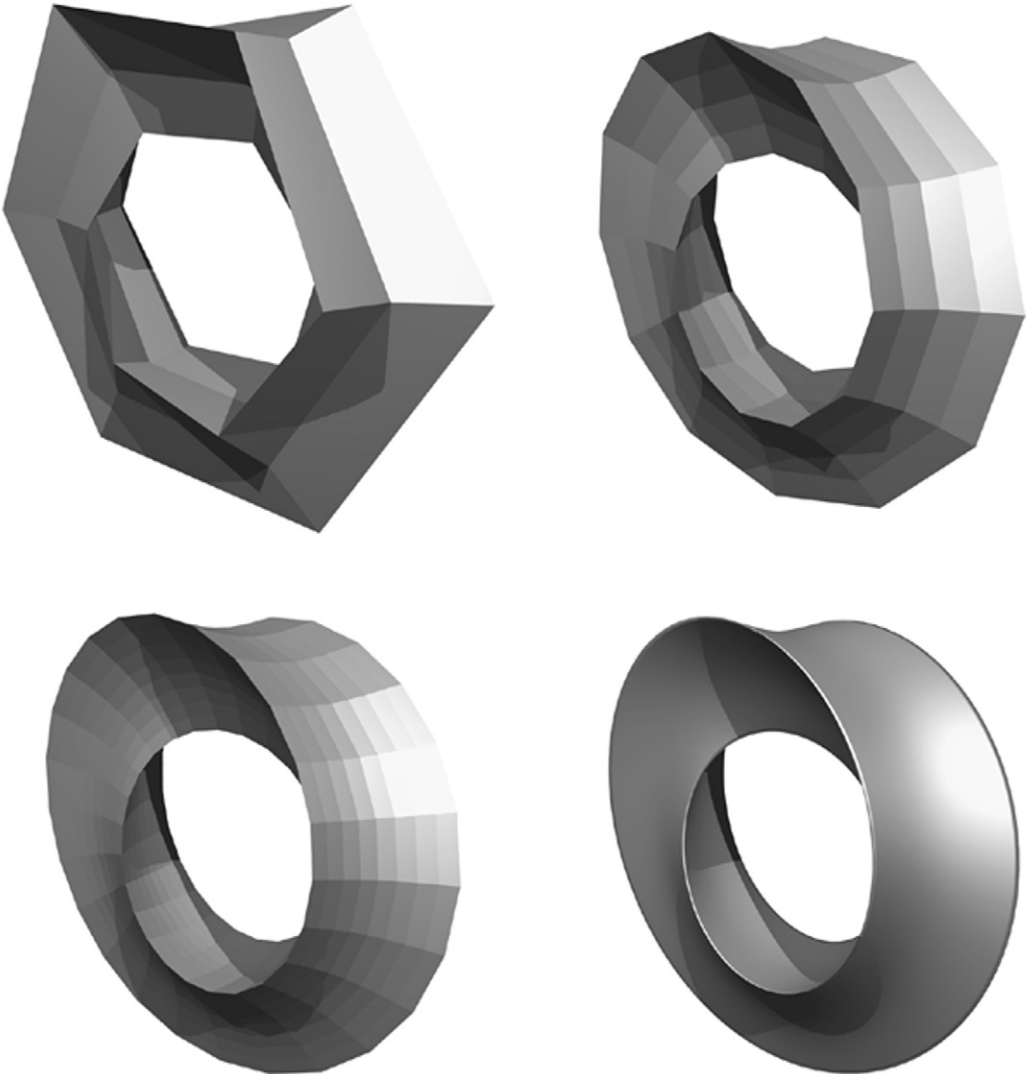


Figure 7.24 An umbilic torus modeled as a twisted surface of revolution with a single crease edge. (Thanks to Scott Schaefer for his help with this figure.)

subdivision surfaces. Biermann et al. [10] propose an improved set of boundary rules tailored to deliver better behavior along concave portions of the boundary. DeRose, Kass, and Truong [42] describe a simple modification to an existing crease rule that allows the user to control the "sharpness" of a crease. In [94] and [95], Levin describes a subdivision method called "combined subdivision" that allows for interpolation of arbitrary curves using subdivision. This scheme is the basis of a trimming algorithm used to compute the intersection of two subdivision surfaces [98]. In [109], [110], and [112], Nasri describes several schemes used to interpolate points and normals along curves.

7.3 Smooth Subdivision for Triangle Meshes

The previous section described a scheme for smoothly subdividing a quadrilateral mesh in which a single round of subdivision is decomposed into two simpler steps: bilinear subdivision and quad averaging. One of the major advantages of this scheme is that almost all of the topological computation takes place during bilinear subdivision. The only topological information computed during quad averaging is the valence val[v] for each vertex v in M_k. Maintained as an array, val[v] can easily be computed during a single pass through M_k or simply updated during bilinear subdivision, in that all old vertices in M_{k-1} inherit the same valence and all new vertices introduced into M_k have valence four. This decomposition of the subdivision scheme into two steps is superior to the single-step approach because it avoids the need to compute or maintain an explicit set of neighbors for each vertex in M_{k-1}.

Our goal in this section is to develop a similar two-step scheme for smoothly subdividing triangle meshes. Again, the main benefit of this scheme is that it avoids the need to compute or maintain explicit neighbor information for the meshes M_k. Clearly, the first step of our scheme should be linear subdivision. As described in section 7.1, linear subdivision can easily be implemented as a single pass through the triangles of M_{k-1}. All that remains is to construct an averaging method for triangle meshes that is analogous to quad averaging. Once we have derived this averaging operator for uniform triangle meshes, an extension to meshes with extraordinary vertices follows naturally.

7.3.1 Linear Subdivision Plus Triangle Averaging

The key to constructing an averaging rule for triangle meshes is to recall the three-direction quartic box spline of Chapter 2. This spline is a $\mathcal{C}^2$ piecewise quartic function defined over a three-direction triangle mesh. In particular, the set of direction vectors Σ for this scheme has the form

$$\Sigma = \{\{1, 0\}, \{1, 0\}, \{0, 1\}, \{0, 1\}, \{1, 1\}, \{1, 1\}\}.$$

Observe that Σ consists of the direction vectors for piecewise linear splines $\{\{1, 0\}, \{0, 1\}, \{1, 1\}\}$ repeated twice. Thus, the subdivision scheme for the three-direction quartic box spline can be expressed as linear subdivision followed by averaging with the mask corresponding to these three direction vectors. This two-dimensional averaging mask is of the form

$$\begin{pmatrix} \frac{1}{8} & \frac{1}{8} & 0 \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ 0 & \frac{1}{8} & \frac{1}{8} \end{pmatrix}.$$

For quad meshes, we decomposed the averaging mask of equation 7.1 into the average of four submasks, each supported over one of the four quads meeting at the vertex. For triangle meshes, we can decompose the averaging mask into six submasks, each supported over a triangle incident on the vertex:

$$\begin{aligned}\begin{pmatrix} \frac{1}{8} & \frac{1}{8} & 0 \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ 0 & \frac{1}{8} & \frac{1}{8} \end{pmatrix} == \frac{1}{6}\Biggl(&\begin{pmatrix} \frac{3}{8} & 0 & 0 \\ \frac{3}{8} & \frac{1}{4} & 0 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} \frac{3}{8} & \frac{3}{8} & 0 \\ 0 & \frac{1}{4} & 0 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & \frac{3}{8} & 0 \\ 0 & \frac{1}{4} & \frac{3}{8} \\ 0 & 0 & 0 \end{pmatrix} \\ &+ \begin{pmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{4} & \frac{3}{8} \\ 0 & 0 & \frac{3}{8} \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{4} & 0 \\ 0 & \frac{3}{8} & \frac{3}{8} \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ \frac{3}{8} & \frac{1}{4} & 0 \\ 0 & \frac{3}{8} & 0 \end{pmatrix}\Biggr).\end{aligned} \tag{7.7}$$

This decomposition again suggests a geometric interpretation for this averaging mask applied at a vertex v: compute the weighted centroid of each triangle that contains v using the weights $\frac{1}{4}$, $\frac{3}{8}$, and $\frac{3}{8}$ at v and its two neighbors; then reposition v at the centroid of these weighted centroids. This generalized averaging operation, *triangle averaging,* can now be applied to triangle meshes with vertices of any valence. Figure 7.25 shows an example of two rounds of linear subdivision plus triangle averaging applied to an octahedron. The upper left-hand mesh is the initial octahedron after linear subdivision. The mesh to its right is the result of applying triangle averaging to this mesh. The lower left-hand mesh shows the results of applying linear subdivision to the upper right-hand mesh. The lower right-hand mesh shows the results of applying triangle averaging to the lower left-hand mesh.

Triangle averaging differs from quad averaging in that the weights used during the averaging process vary, depending on which vertex of the triangle is being updated. Specifically, consider a triangle $\{s, t, v\}$ in the triangle mesh $\{M_k, \widetilde{p}_k\}$ produced by linear subdivision. Triangle averaging computes a weighted centroid of

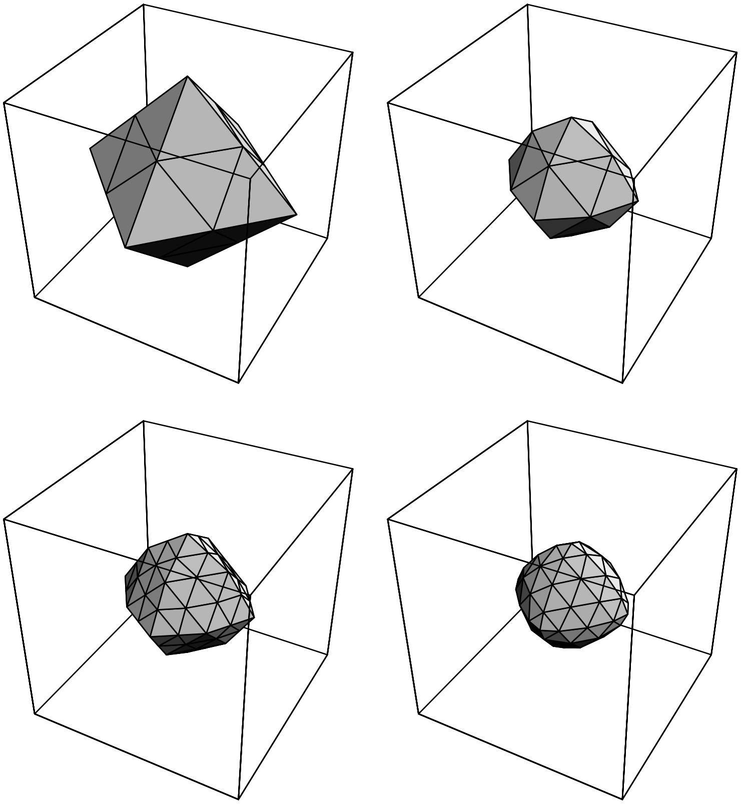


Figure 7.25 Two rounds of subdivision for an octahedron expressed as alternating steps of linear subdivision and triangle averaging.

the form $\frac{1}{4}\widetilde{p}_k[\![v]\!] + \frac{3}{8}\widetilde{p}_k[\![s]\!] + \frac{3}{8}\widetilde{p}_k[\![t]\!]$ when updating the position of vertex v. Similarly, triangle averaging uses weighted centroids of the form $\frac{1}{4}\widetilde{p}_k[\![s]\!] + \frac{3}{8}\widetilde{p}_k[\![t]\!] + \frac{3}{8}\widetilde{p}_k[\![v]\!]$ and $\frac{1}{4}\widetilde{p}_k[\![t]\!] + \frac{3}{8}\widetilde{p}_k[\![s]\!] + \frac{3}{8}\widetilde{p}_k[\![v]\!]$ when updating the position of s and t, respectively. (Note that using the centroid of the triangle $\{s, t, v\}$ in the averaging phase leads to a subdivision scheme whose limit functions are only $\mathcal{C}^1$ in the uniform case.)

Luckily, triangle averaging can still be implemented as a single pass through the topology M_k. As for quad averaging, a table of new vertex position p_k is initialized to 0, with the positions of the averaged mesh being accumulated on a triangle-by-triangle basis. The main distinction here from quad averaging is that instead of computing a single centroid for a triangle $\{s, t, v\}$ the three weighted centroids described previously must be computed and used to update the appropriate vertex of $\{s, t, v\}$. The update rule for this triangle can be expressed in matrix form as the product of an update matrix multiplied by the column vector of old positions $\widetilde{p}_k\llbracket\{s, t, v\}\rrbracket$:

$$p_k\llbracket\{s, t, v\}\rrbracket \mathrel{+}= \frac{1}{8}\begin{pmatrix} \frac{2}{\mathrm{val}[s]} & \frac{3}{\mathrm{val}[s]} & \frac{3}{\mathrm{val}[s]} \\ \frac{3}{\mathrm{val}[t]} & \frac{2}{\mathrm{val}[t]} & \frac{3}{\mathrm{val}[t]} \\ \frac{3}{\mathrm{val}[v]} & \frac{3}{\mathrm{val}[v]} & \frac{2}{\mathrm{val}[v]} \end{pmatrix} \widetilde{p}_k\llbracket\{s, t, v\}\rrbracket. \tag{7.8}$$

Here, $p_k\llbracket\{s, t, v\}\rrbracket$ is a column vector consisting of the three entries $p_k\llbracket s\rrbracket$, $p_k\llbracket t\rrbracket$, and $p_k\llbracket v\rrbracket$. Note that the row of the update matrix corresponding to v sums to $\frac{1}{\mathrm{val}[v]}$. This property is necessary if we desire the corresponding averaging scheme to be affinely invariant (i.e., the sum of the weights at a vertex is one).

Because triangle averaging is designed to reproduce the averaging rule for three-direction quartic box splines on valence-six triangle meshes, the resulting limit surface is $\mathcal{C}^2$ everywhere except at extraordinary vertices. The next chapter analyzes the smoothness of the limit surfaces at extraordinary vertices and shows that the scheme produces limit surfaces that are $\mathcal{C}^1$ for all valences except three. In this case, the surface is only $\mathcal{C}^0$. The middle portion of Figure 7.26 shows a stellated octahedron (an octahedron with a regular tetrahedron attached to each face) after three rounds of subdivision using triangle averaging. Note that the limit surface is not smooth in the neighborhood of the original valence-three vertices. This problem is addressed later in this chapter.

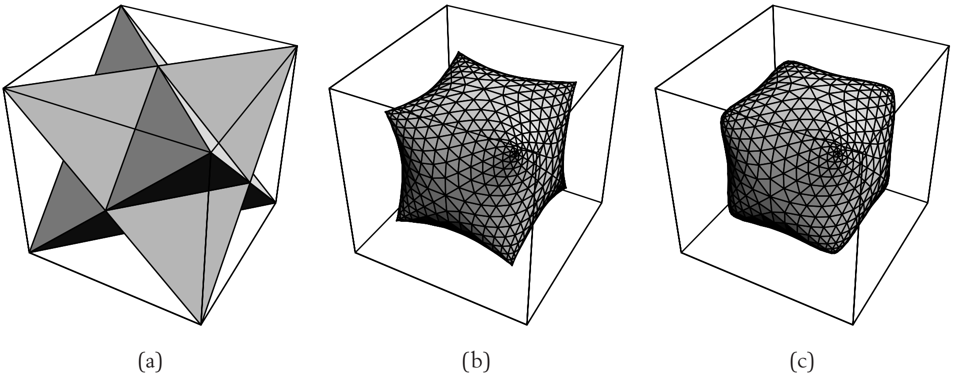


Figure 7.26 A stellated octahedron (a), after three rounds of subdivision with triangle averaging (b) and after three rounds of subdivision using Loop's scheme (c).

7.3.2 Comparison to Other Triangle Schemes

The combined action of linear subdivision followed by triangle averaging can be expressed in a single step as multiplication by a subdivision matrix S_{k-1}. For triangle schemes, there are essentially two types of rows in S_{k-1}, each corresponding to two types of subdivision rules. "Edge" rules position vertices of M_k that are introduced during edge splits of M_{k-1}. This rule, the edge rule for the four-direction quartic box spline, is shown on the left in Figure 7.27.

The remaining type of subdivision rules are "vertex" rules that position those vertices of M_k that are also vertices of M_{k-1}. For such a vertex v, the n vertices adjacent to v in M_k lie on the midpoints of n edges incident on v in the original mesh M_{k-1} after linear subdivision. Each of these midpoints lies on triangles containing v. After triangle averaging, the new position of vertex v is $\frac{3}{8n}$ times the position of each of its n neighboring vertices in M_{k-1} plus $\frac{5}{8}$ times the original position of v in M_{k-1}. The middle portion of Figure 7.27 shows this rule plotted diagrammatically.

The first smooth subdivision scheme for arbitrary triangle meshes was developed by Loop in his Master's thesis [99]. This scheme, also based on a generalization of the subdivision rules for the three-direction quartic box spline, converges to surfaces that smooth at extraordinary vertices of all valences. For example, the rightmost portion of Figure 7.26 was produced using three rounds of Loop subdivision. Note that the limit surface is smooth even at extraordinary vertices of valence three. The rightmost portion of Figure 7.27 shows Loop's rule for positioning an extraordinary vertex of valence n. The constant $w[n]$ used in this rule has

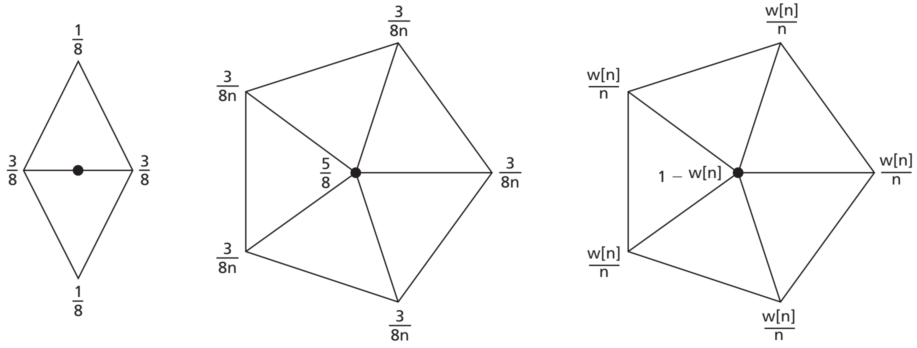


Figure 7.27 Subdivision rules for smooth triangle schemes.

the form

$$w[n] = \frac{5}{8} - \left(\frac{3}{8} + \frac{1}{4}\,\mathrm{Cos}\left[\frac{2\pi}{n}\right]\right)^2. \tag{7.9}$$

This weight function of equation 7.9, although mysterious looking, was chosen by Loop so as to make the resulting surfaces close to having continuous curvature (i.e., be $\mathcal{C}^2$) for low-valence vertices. Although the resulting limit surfaces do not have continuous curvature (except, of course, at valence-six vertices), the limit surfaces do have bounded curvature in the neighborhood of extraordinary vertices of valence three to six (see Peters and Umlauf [117, 118] for details). Both Sabin [135] and Holt [73] give alternative subdivision rules that also attempt to bound the curvature of a subdivision surface in the neighborhood of an extraordinary vertex.

Figure 7.28 shows a side-by-side comparison of linear subdivision with averaging versus Loop's scheme when the base mesh is an octahedron. The left-hand surface was produced by linear subdivision plus triangle averaging. The right-hand surface was produced by Loop's scheme. Note that the right-hand surface is "rounder" due to the careful curvature "tuning" of Loop's subdivision rule. (Section 8.3 provides a more detailed analysis of the differences between these schemes.)

The subdivision rules of Figure 7.27 position a vertex of M_k directly in terms of the positions p_{k-1} of vertices of M_{k-1}. Ideally, we would like to recast Loop's rule in terms of linear subdivision plus some type of modified averaging with a simplicity similar to that of triangle averaging. The beauty of the matrix update used in equation 7.8 is that it can also be modified to reproduce Loop's extraordinary

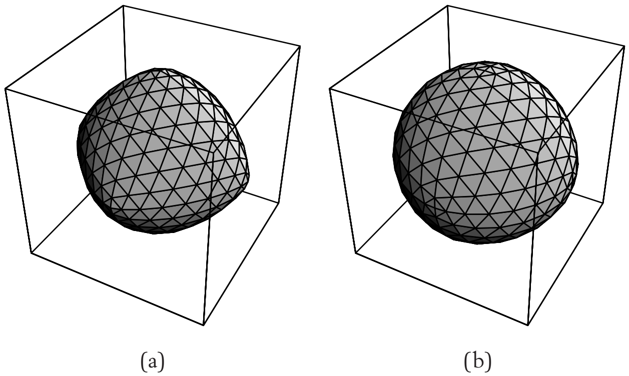


Figure 7.28 A comparison of linear subdivision with triangle averaging (a) versus Loop's scheme (b).

vertex rule for triangle meshes. In particular, Loop subdivision can be expressed as linear subdivision followed by averaging via the matrix update

$$\mathsf{p}_k[\![\{s,t,v\}]\!] \mathrel{+}= \begin{pmatrix} \frac{1-2w[val[s]]}{val[s]} & \frac{w[val[s]]}{val[s]} & \frac{w[val[s]]}{val[s]} \\ \frac{w[val[t]]}{val[t]} & \frac{1-2w[val[t]]}{val[t]} & \frac{w[val[t]]}{val[t]} \\ \frac{w[val[v]]}{val[v]} & \frac{w[val[v]]}{val[v]} & \frac{1-2w[val[v]]}{val[v]} \end{pmatrix} \widetilde{\mathsf{p}}_k[\![\{s,t,v\}]\!],$$

where $w[n]$ satisfies equation 7.9. The advantage of this formulation of Loop's method is that the positions of vertices in M_k can now be accumulated on a triangle-by-triangle basis as done using triangle averaging.

Although both of the schemes described previously are approximating, interpolatory schemes for triangle meshes are also possible. Dyn, Levin, and Gregory [55] developed the Butterfly scheme, an interpolatory scheme that is $\mathcal{C}^1$ on three-direction triangle meshes (see Chapter 3 for details). Although applying the Butterfly scheme to arbitrary triangle meshes is possible, the resulting surfaces are not always $\mathcal{C}^1$ for all possible vertex valences. Zorin et al. [171] describe a modified version of the Butterfly scheme that is provably $\mathcal{C}^1$ at extraordinary vertices of all valences.

7.4 Other Types of Polyhedral Schemes

The previous three sections covered the core subdivision methods for polyhedral meshes. However, there are a number of other interesting types of subdivision schemes available for polyhedral meshes. Because subdivision is an active area of research in which new types of schemes are constantly being invented, it is difficult to summarize all possible schemes in the subdivision "zoo." This section provides a quick overview of some of these schemes, focusing in particular on subdivision schemes based on other types of topological splits.

7.4.1 Face-splitting Schemes

The two types of topological splits considered previously were based on edge splitting. However, these are not the only types of topological splits that can be applied to a polyhedral mesh. Another interesting class of topological splits involves face-splitting subdivision. During face-splitting subdivision of a mesh M_{k-1}, new vertices (gray) are added to M_k on the faces of M_{k-1}. Existing edges in M_{k-1} are then removed

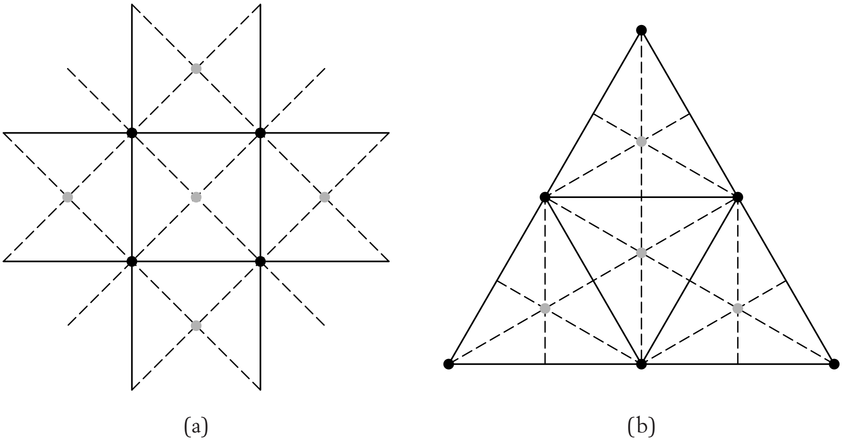


Figure 7.29 Face-splitting subdivisions for quad (a) and triangle meshes (b).

with new edges (dashed) added into M_k, as shown in Figure 7.29. The left-hand split is for quad meshes, and the right-hand split is for triangle meshes. As in the case of edge-splitting subdivision, these topological splits introduce only ordinary vertices and isolate those extraordinary vertices present in the original mesh. An important feature of these two splits is that performing two rounds of subdivision is equivalent to one round of binary subdivision for quad meshes and one round of ternary subdivision for triangle meshes.

Kobbelt [85] has built an interesting subdivision scheme based on face-splitting subdivision for triangle meshes. This scheme, which he calls $\sqrt{3}$-subdivision, positions "face" vertices in M_k (the gray vertices in Figure 7.29) at the centroid of the triangular face in M_{k-1} that contains the vertex. The remaining vertices (inherited from M_{k-1}) are repositioned by applying the weight $1 - w[n]$ to the vertex and weights $\frac{w[n]}{n}$ to each of its n neighbors where

$$w[n] = \frac{4 - 2\,\mathrm{Cos}\left[\frac{2\pi}{n}\right]}{9}.$$

In the ordinary case (i.e., $n == 6$), this rule leads to a weighting of $\frac{2}{3}$ at the vertex and weights of $\frac{1}{18}$ applied to its six neighbors. Kobbelt shows that this rule is C^2 in the uniform case and delivers C^1 limit surfaces at extraordinary vertices. Kobbelt argues that $\sqrt{3}$-subdivision is particularly well suited for adaptive subdivision.

Another area in which face-splitting subdivision appears to have promise is in creating simpler interpolatory surface schemes. The advantage of face-splitting

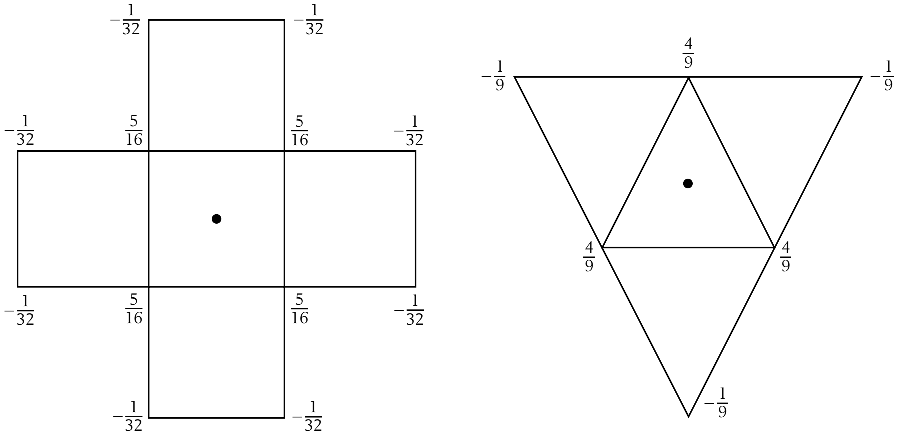


Figure 7.30 Subdivision rules for interpolatory face-splitting subdivision.

subdivision over edge-splitting subdivision is that in the interpolatory case only one type of subdivision rule is necessary: a positioning rule for new vertices on faces of M_{k-1}. In the uniform case, this rule can easily be derived so as to reproduce polynomials. Figure 7.30 shows the face rules for a quad and a triangle scheme, respectively. These rules are chosen so as to reproduce cubic functions for the quad rule and quadratic functions for the triangle rule. Note that the support of the rule for triangle meshes is smaller than the Butterfly rule.

A preliminary analysis of the quad scheme (included in the associated *Mathematica* implementation (♔)) shows that the scheme converges to C^1 limit surfaces for uniform meshes and produces smooth limit surfaces at extraordinary vertices for a small range of valences. For the triangle rules, we have no estimates of the smoothness of the resulting surfaces and leave the analysis of this scheme to the interested reader. In both cases, we suspect that it will be necessary to develop variants of these rules to produce schemes that are C^1 for all valences.

7.4.2 Dual Subdivision Schemes

In section 5.3.2, we observed that the uniform B-splines of odd order had the property that the control points in p_k could be viewed as being positioned at $\mathbb{D}[\frac{1}{2^k}\mathbb{Z}]$, the midpoints of the intervals defined by the grid $\frac{1}{2^k}\mathbb{Z}$. Such spline schemes

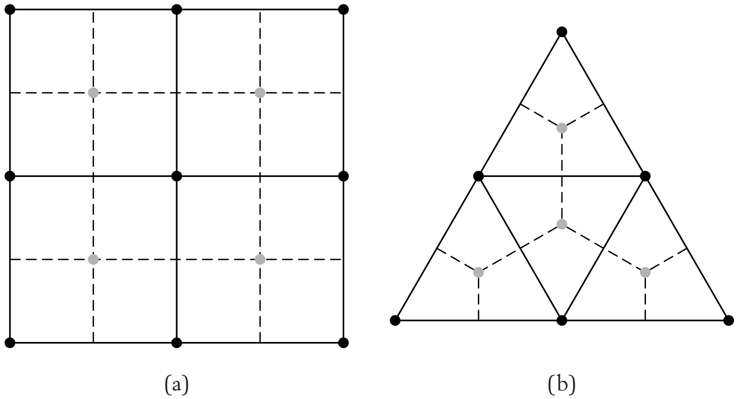


Figure 7.31 Dual meshes (dotted) for quad (a) and triangle (b) meshes.

were referred to as dual schemes. Now, a surface scheme is a *dual* scheme if the control points in the vector p are attached to the faces of the surface mesh M. Note that placing these control points on the faces of M defines a new topological surface mesh $\mathbb{D}[M]$ whose vertices correspond to the faces of M, whose faces correspond to vertices of M, and whose edges connect two vertices in $\mathbb{D}[M]$ if and only if the associated faces in M share a common edge. This surface mesh $\mathbb{D}[M]$ is the *topological dual* of a surface mesh M.

Figure 7.31 shows a plot of the topological duals of a portion of a uniform quad mesh and a portion of a uniform triangle mesh. These duals (dotted lines) are, respectively, another quad mesh and a hexagonal mesh. Note that constructing the dual of M "flips" the edges of M in constructing $\mathbb{D}[M]$. One important feature of the topological dual is that for closed surface meshes the dual of the dual of a mesh yields the original mesh (i.e., $\mathbb{D}[\mathbb{D}[M]] == M$). For example, the dual of the topological mesh for a cube is the mesh of an octahedron. Likewise, the dual of the topological mesh for an octahedron is a topological cube.

Now, given a primal subdivision scheme that maps a coarse mesh M_{k-1} to a refined mesh M_k, the corresponding dual scheme maps the coarse mesh $\mathbb{D}[M_{k-1}]$ to the refined mesh $\mathbb{D}[M_k]$. Figure 7.32 shows an example of dual topological subdivision applied to a quad mesh. The solid lines are the meshes M_{k-1} and M_k, respectively. The dotted lines are the dual meshes $\mathbb{D}[M_{k-1}]$ and $\mathbb{D}[M_k]$, respectively. Note that the vertices of $\mathbb{D}[M_{k-1}]$ are not a subset of the vertices of $\mathbb{D}[M_k]$, as they were for the previous primal schemes. Luckily, dual subdivision retains the key

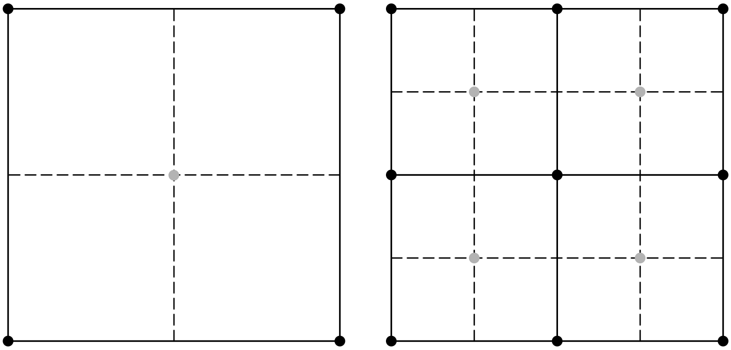


Figure 7.32 Topological subdivision of a dual quad mesh.

property that made primal subdivision work: isolation of extraordinary features. In the primal case, topological subdivision introduces only ordinary vertices and isolates existing extraordinary vertices. In the dual case, topological subdivision introduces only ordinary faces and isolates those extraordinary faces that are dual to extraordinary vertices in M_k.

The simplest example of a dual subdivision scheme is constant subdivision. This scheme takes an existing mesh $\{M_{k-1}, p_{k-1}\}$ and splits each face f of M_{k-1} into four subfaces in M_k. Given the control point in p_{k-1} corresponding to this face f, we next make four copies of this control point for the four subfaces of f. Unfortunately, this scheme is not particularly interesting for parametric meshes because the scheme converges to a set of points corresponding to those in the original vector p_0. (In the functional case, this scheme converges to piecewise constant functions; hence the name.)

The main application of constant subdivision lies in serving as a preprocessor for higher-order dual schemes. The method for generating these schemes is the same as in the primal case: apply an appropriate averaging operator to the results of some type of simple subdivision operation. For primal meshes, applying quad averaging to the result of linear subdivision leads to smooth primal schemes. For dual meshes, applying quad averaging to the result of constant subdivision leads to smooth dual schemes.

For example, one round of quad averaging leads to a dual scheme that generalizes biquadratic B-splines to polyhedral meshes. The left-hand portion of Figure 7.33 shows the subdivision rule for this scheme at an extraordinary vertex v of valence n in M_{k-1}. The dark lines bound the face in $\mathbb{D}[M_{k-1}]$ dual to v; the

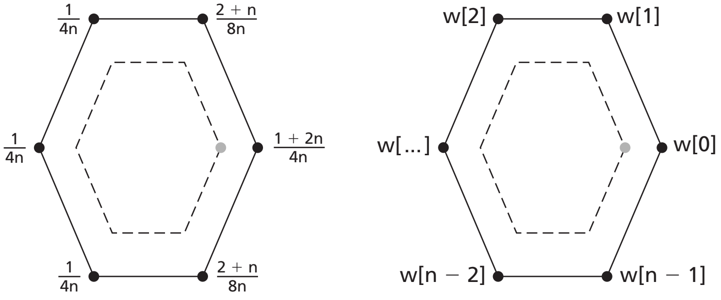


Figure 7.33 Two subdivision rules for a dual quad scheme at an extraordinary vertex.

dashed lines bound the face in $\mathbb{D}[M_k]$ dual to v. The position of a vertex (gray dot) on $\mathbb{D}[M_k]$ is expressed as a weighted combination of the vertices of $\mathbb{D}[M_{k-1}]$ (black dots), as shown. This subdivision rule defines a dual scheme that converges to smooth limit surfaces even in the presence of extraordinary vertices.

This subdivision rule was first proposed by Doo and Sabin in their original paper [45]. Doo and Sabin also propose an alternate subdivision rule shown on the right in Figure 7.33. The weights w[i] have the form $w[0] = \frac{n+5}{4n}$ and $w[i] = \frac{3+2\,\mathrm{Cos}[\frac{2i\pi}{n}]}{4n}$ for i == 1...n − 1. This alternate rule also produces limit surfaces that are smooth everywhere.

Higher-order dual schemes can be constructed by applying several rounds of quad averaging. For example, Zorin and Schröder [170] generate a dual quartic subdivision scheme via constant subdivision, followed by two rounds of quad averaging.

Dual schemes based on other types of topological splits are also possible. For example, Peters and Reif [115] devised a scheme that they refer to as the "mid-edge" scheme. Topologically, this scheme is the dual of the face-splitting scheme for quads described previously; that is, vertices in M_k are placed at the midpoints of edges in M_{k-1}. On uniform grids, two rounds of the mid-edge scheme correspond to one round of subdivision for the four-direction quadratic box spline. Peters and Reif show that their scheme is $\mathcal{C}^1$ even in the presence of extraordinary vertices. Habib and Warren [70] describe another variant of the quadratic box spline rule that converges to a $\mathcal{C}^1$ surface at extraordinary vertices and allows more precise control over boundary behavior.

Dual schemes based on triangle subdivision of M_{k-1} are also possible (although rarely used in practice). These schemes lead to primal subdivision schemes on the hexagonal mesh $\mathbb{D}[M_{k-1}]$. For example, Goodman [68] describes a construction for what he refers to as a "cubic half-boxspline." This spline is a dual spline defined on a three-direction triangle mesh. The subdivision scheme for this spline is, therefore, a primal subdivision scheme for hexagonal meshes. The authors leave the details of deriving the actual subdivision rules for this scheme to the interested reader.

CHAPTER 8

Spectral Analysis at an Extraordinary Vertex

Given a polyhedral mesh $\{M_{k-1}, p_{k-1}\}$, the subdivision schemes considered in Chapter 7 consisted of two major steps: topological subdivision and geometric positioning. For most schemes, topological subdivision consisted of splitting each face in M_{k-1} into four new subfaces to produce a new topological mesh M_k. Geometric updating involved positioning new vertices p_k via the subdivision relation $p_k = S_{k-1}p_{k-1}$. For these schemes, the choice of the subdivision rules used in S_{k-1} depended only on the topology of M_{k-1}, not on the level k. Such schemes in which the same subdivision rules are used at all levels of subdivision are *stationary subdivision schemes*. Note that the scheme for surfaces of revolution was not stationary, because the tensions σ_k, ρ_k used in constructing S_{k-1} varied as a function of k.

The goal of this chapter is to develop a set of tools for analyzing the behavior of stationary subdivision schemes at extraordinary vertices. The first part of the chapter introduces the main tool used in understanding the behavior of a subdivision process at an extraordinary vertex: spectral analysis. The second portion of the chapter uses spectral analysis to derive necessary and sufficient conditions for stationary subdivision schemes to produce smooth limit surfaces. The third portion of the chapter applies these analysis techniques to an example problem, Loop's subdivision scheme, and shows that this scheme converges to a smooth limit surface at extraordinary vertices. Finally, we conclude the book with a brief overview of some current areas of research in subdivision.

8.1 Convergence Analysis at an Extraordinary Vertex

The averaging schemes of Chapter 7 generalized uniform subdivision schemes to arbitrary polyhedral meshes. The key idea behind these schemes was to apply uniform subdivision within each triangle or quad of the original mesh M_0. Because

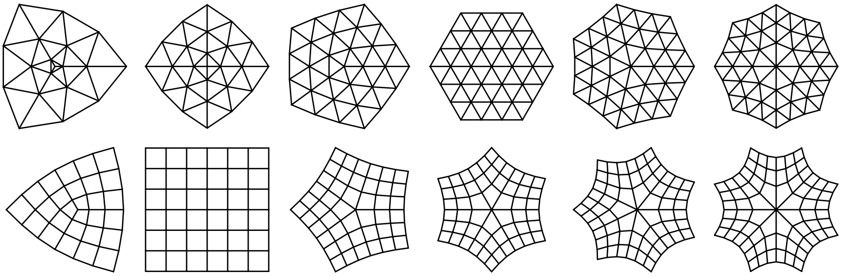


Figure 8.1 Triangular and quadrilateral meshes M consisting of a single extraordinary vertex.

the only nonuniformities present in the resulting meshes M_k are those that are inherited from the original mesh M_0, these nonuniformities (the extraordinary vertices) are isolated inside an otherwise uniform mesh. Our approach in the next two sections is to analyze the behavior of averaging schemes on a mesh M consisting of a single extraordinary vertex of valence n surrounded by an otherwise uniform mesh.

In the two-dimensional case, such meshes M can be formed by combining several copies of a quadrant of the uniform mesh $\mathbb{Z}^2$. These *sectors* correspond to the mesh $(\mathbb{Z}^+)^2$ where $\mathbb{Z}^+$ denotes the grid of non-negative integers. If M consists of a single extraordinary vertex v of valence n, M can be formed by joining n sectors around a common, extraordinary vertex v of valence n. Figure 8.1 shows several examples of such triangular and quadrilateral meshes for an extraordinary vertex of valence three to eight. (Only a finite portion of the infinite mesh M centered around the extraordinary vertex v is shown.)

These infinite meshes M have the nice property that they are invariant under repeated topological subdivision; that is, if $M_0 == M$, then $M_k == M$ for all $k \geq 0$. In other words, topologically subdividing such a mesh M always returns M back. Because the subdivision matrices S_k are functions of the topology M_k, the subdivision process for such an initial mesh M is independent of the level of subdivision. In particular, if S is the bi-infinite subdivision matrix associated with M, the geometric update rule for the mesh M has the form

$$p_k = Sp_{k-1} == S^k p_0, \tag{8.1}$$

where p_0 is the vector of initial vertex positions.

8.1.1 The Limit Surface at an Extraordinary Vertex

Chapter 3 discussed developing tools for analyzing the behavior of subdivision schemes defined over uniform meshes. The key observation there was to define a function $p_k[x, y]$ associated with a vector of coefficients $\mathsf{p_k}$ on the grid $\frac{1}{2^k}\mathbb{Z}^2$ and then to consider the convergence of this sequence of functions. In our current case, we wish to analyze the behavior of this vector $\mathsf{p_k}$ when associated with the mesh M. Because M consists of n sectors, we can number the vertices of $\mathsf{p_k}$ using a triple index $\{\mathsf{h}, \mathsf{i}, \mathsf{j}\}$, where h denotes a sector of M and $\{\mathsf{i}, \mathsf{j}\}$ denotes a particular vertex in that sector where $\mathsf{i}, \mathsf{j} \geq 0$. To model the connection between sectors, this indexing is also subject to the following rules:

- The indices $\{\mathsf{h}, 0, 0\}$ coincide with the extraordinary vertex v for $0 \leq \mathsf{h} < \mathsf{n}$.
- The boundary vertices of each sector overlap subject to the indexing rules $\{\mathsf{h}, 0, \mathsf{i}\} == \{\mathrm{mod}[\mathsf{h}+1, \mathsf{n}], \mathsf{i}, 0\}$ for $0 \leq \mathsf{h} < \mathsf{n}$.

Figure 8.2 shows an example of this indexing for an extraordinary vertex of valence four in a triangular mesh.

Given a vector of coefficients $\mathsf{p_k}$, we let $\mathsf{p_k}[\![\mathsf{h}, \mathsf{i}, \mathsf{j}]\!]$ denote the coefficient in $\mathsf{p_k}$ with index $\{\mathsf{h}, \mathsf{i}, \mathsf{j}\}$. Based on this indexing, we now define n associated piecewise polynomial functions $p_k[\mathsf{h}, x, y]$ for $0 \leq \mathsf{h} < \mathsf{n}$. For triangular meshes, these functions $p_k[\mathsf{h}, x, y]$ are typically piecewise linear over each sector of M, whereas in the quad case the functions are piecewise bilinear over each sector of M. In both cases, these

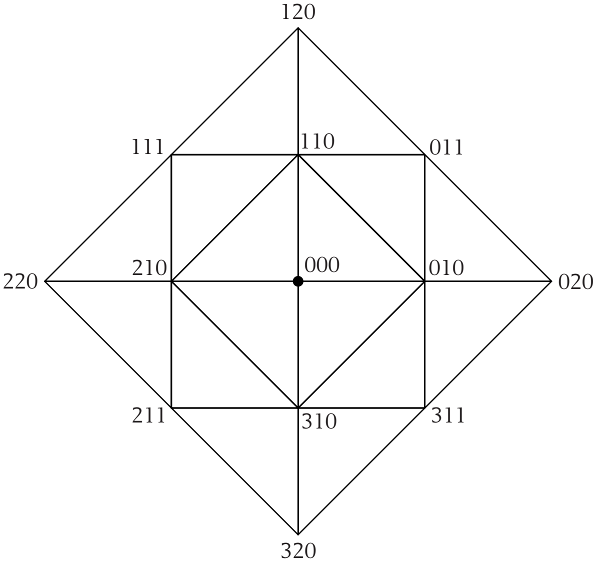


Figure 8.2 An example of vertex indexing for the two-ring of an extraordinary vertex of valence four.

functions interpolate the values of p_k in a manner similar to that of the uniform case; that is,

$$p_k\left[\mathsf{h}, \frac{\mathsf{i}}{2^k}, \frac{\mathsf{j}}{2^k}\right] = \mathsf{p}_k\llbracket \mathsf{h}, \mathsf{i}, \mathsf{j}\rrbracket. \tag{8.2}$$

Due to the compatibility conditions associated with this indexing, these functions coincide along the common boundary shared by two adjacent sectors; that is, they satisfy $p_k[\mathsf{h}, 0, x] == p_k[\mathsf{mod}[\mathsf{h}+1, \mathsf{n}], x, 0]$ for $0 \le \mathsf{h} < \mathsf{n}$ and $x \ge 0$. Given a subdivision process of the form $\mathsf{p}_k = \mathsf{S}^k \mathsf{p}_0$, we define the limit surface associated with this process to be the parametric surface of the form

$$p_\infty[\mathsf{h}, x, y] = \lim_{k \to \infty} p_k[\mathsf{h}, x, y], \tag{8.3}$$

where $0 \le \mathsf{h} < \mathsf{n}$ and $x, y \ge 0$. As we shall see, the existence of this limit surface $p_\infty[\mathsf{h}, x, y]$ depends on the structure of the subdivision matrix S. Away from the extraordinary vertex v, the subdivision rules in S are typically chosen to reproduce a uniform subdivision scheme. Because the mesh M is uniform except at v, the limit surface $p_\infty[\mathsf{h}, x, y]$ shares all of the convergence and smoothness properties of the uniform scheme everywhere except at the extraordinary vertex v (with coordinates $\{x, y\} == \{0, 0\}$). All that remains is to analyze the behavior of this surface at v.

To illustrate the concepts discussed in this chapter, we will consider a single running example throughout the entire chapter. The subdivision scheme for this example is the univariate stationary scheme associated with the following bi-infinite subdivision matrix S:

$$\mathsf{S} = \begin{pmatrix}
. & . & . & . & . & . & . & . & . & . & . \\
. & 0 & \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & 0 & 0 & 0 & 0 & 0 & . \\
. & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & . \\
. & 0 & 0 & \frac{1}{8} & \frac{25}{32} & \frac{3}{32} & 0 & 0 & 0 & 0 & . \\
. & 0 & 0 & 0 & \frac{5}{8} & \frac{3}{8} & 0 & 0 & 0 & 0 & . \\
. & 0 & 0 & 0 & \frac{5}{24} & \frac{29}{40} & \frac{1}{15} & 0 & 0 & 0 & . \\
. & 0 & 0 & 0 & 0 & \frac{3}{5} & \frac{2}{5} & 0 & 0 & 0 & . \\
. & 0 & 0 & 0 & 0 & \frac{3}{20} & \frac{29}{40} & \frac{1}{8} & 0 & 0 & . \\
. & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & . \\
. & 0 & 0 & 0 & 0 & 0 & \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & 0 & . \\
. & . & . & . & . & . & . & . & . & . & .
\end{pmatrix}.$$

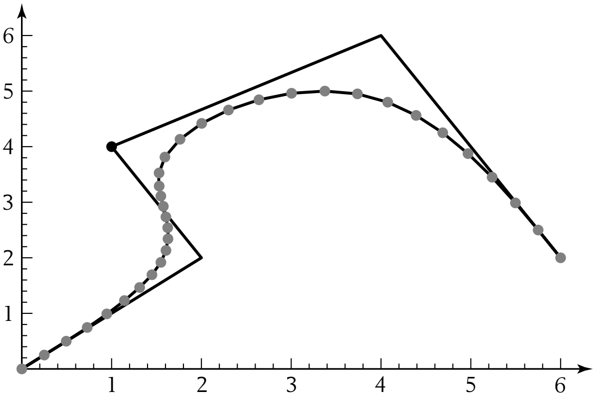


Figure 8.3 Three rounds of subdivision with a nonuniform curve scheme.

This matrix has three nonuniform columns, with the remaining columns corresponding to the subdivision mask $\{\frac{1}{8}, \frac{1}{2}, \frac{3}{4}, \frac{1}{2}, \frac{1}{8}\}$ for a uniform cubic B-spline. Figure 8.3 shows several rounds of subdivision applied to an initial control polygon. The large, dark vertex on p_0 is the extraordinary vertex v.

Our approach for this univariate example is to treat the central vertex v as an extraordinary vertex joining two sectors (each a uniform mesh $\mathbb{Z}^+$). The three special columns of S arise from the need to form a smooth transition between these two meshes. Following our indexing scheme for surfaces, entries of vectors p_k associated with this scheme can be indexed by a pair of the form {h, i}, where h == 0, 1 and i is a non-negative integer. Index h == 0 corresponds to the upper-left-hand portion of the vector p_k, whereas indices with h == 1 correspond to lower-right-hand portions of the vector p_k. Enumerated in order, these indices have the form

$$\{\ldots, \{0, 2\}, \{0, 1\}, \{0, 0\} == \{1, 0\}, \{1, 1\}, \{1, 2\}, \ldots\}.$$

Our task in the rest of this section and the next two is to analyze the convergence and smoothness of the composite parametric limit curve $p_\infty[\mathrm{h}, x]$ defined via a univariate version of equation 8.3.

8.1.2 Local Spectral Analysis

The key to analyzing the behavior of the limit surface $p_\infty[\mathrm{h}, x, y]$ near an extraordinary vertex v is to construct a finite submatrix $\overline{S}$ of the infinite subdivision matrix S that encapsulates the behavior of the scheme near the extraordinary vertex v. In practice, the size of this finite submatrix depends on the locality of the subdivision rules used in constructing S. Given the initial mesh M, let $\overline{M}$ be the submesh of M that determines the behavior of the limit function $p_\infty[h, x, y]$ on the one-ring of the extraordinary vertex v, ring[v]. This submesh $\overline{M}$ is the *neighborhood* of the extraordinary vertex v. For example, in the case of uniform cubic B-splines, any vertex v has a neighborhood that consists of v and its two neighbors on either side. In the surface case, the neighborhood of an extraordinary vertex v is the one-ring of v in the case of linear and bilinear subdivision. For Loop and Catmull-Clark subdivision, the neighborhood of v consists of the two-ring of v (i.e., ring[ring[v]]).

Given this neighborhood $\overline{M}$, we can restrict the subdivision matrix S to a square, finite submatrix $\overline{S}$ whose rows and columns correspond to the vertices in $\overline{M}$. As we shall show, the spectral properties of this matrix $\overline{S}$ determine the convergence and smoothness of the scheme at the extraordinary vertex v. For our running example, the neighborhood of the central vertex v consists of v and its two neighbors on either side. Therefore, the matrix $\overline{S}$ is the 5 × 5 submatrix corresponding to the five vertices lying in the two-ring of the central vertex v:

$$\overline{S} = \begin{pmatrix} \frac{1}{8} & \frac{25}{32} & \frac{3}{32} & 0 & 0 \\ 0 & \frac{5}{8} & \frac{3}{8} & 0 & 0 \\ 0 & \frac{5}{24} & \frac{29}{40} & \frac{1}{15} & 0 \\ 0 & 0 & \frac{3}{5} & \frac{2}{5} & 0 \\ 0 & 0 & \frac{3}{20} & \frac{29}{40} & \frac{1}{8} \end{pmatrix}.$$

If the subdivision matrix S relates infinite vectors via $p_k = Sp_{k-1}$, these infinite vectors p_k can also be restricted to the neighborhood of v. This restriction $\overline{p}_k$ is a finite subvector of p_k corresponding to the vertices in the neighborhood $\overline{M}$. Given a sequence of vectors satisfying equation 8.1, the corresponding localizations to the neighborhood $\overline{M}$ satisfy

$$\overline{p}_k = \overline{S}\,\overline{p}_{k-1} == \overline{S}^k \overline{p}_0. \tag{8.4}$$

To understand the behavior of this iteration as $k \to \infty$, we consider the eigenvalues and eigenvectors of $\overline{S}$. A vector $\overline{z}$ is a right *eigenvector* $\overline{z}$ of $\overline{S}$ with associated *eigenvalue* λ if $\overline{S}\,\overline{z} == \lambda\overline{z}$. Note that multiplying $\overline{S}^k$ by an eigenvector $\overline{z}$ is equivalent to multiplying λ^k by $\overline{z}$ (i.e., $\overline{S}^k\overline{z} == \lambda^k\overline{z}$). As we shall see, the magnitude of the eigenvalues of $\overline{S}$ governs the behavior of equation 8.4 as $k \to \infty$. The process of computing the properties of the powers $\overline{S}^k$ is known as spectral analysis. (See

Strang [149] for an introduction to eigenvalues, eigenvectors, and spectral analysis.) Our goal in this chapter is to use spectral analysis to understand the behavior of subdivision schemes at extraordinary vertices.

The eigenvalues and eigenvectors of $\overline{S}$ can be expressed collectively in matrix form as $\overline{S}\,\overline{Z} == \overline{Z}\Lambda$. For our running example, this spectral decomposition of the subdivision matrix $\overline{S}$ has the form

$$\begin{pmatrix} \frac{1}{8} & \frac{25}{32} & \frac{3}{32} & 0 & 0 \\ 0 & \frac{5}{8} & \frac{3}{8} & 0 & 0 \\ 0 & \frac{5}{24} & \frac{29}{40} & \frac{1}{15} & 0 \\ 0 & 0 & \frac{3}{5} & \frac{2}{5} & 0 \\ 0 & 0 & \frac{3}{20} & \frac{29}{40} & \frac{1}{8} \end{pmatrix} \begin{pmatrix} 1 & -2 & \frac{11}{3} & 6 & 0 \\ 1 & -1 & \frac{2}{3} & 0 & 0 \\ 1 & \frac{1}{3} & \frac{-2}{3} & 0 & 0 \\ 1 & 2 & \frac{8}{3} & 0 & 0 \\ 1 & 4 & \frac{44}{3} & 0 & 48 \end{pmatrix} \tag{8.5}$$

$$== \begin{pmatrix} 1 & -2 & \frac{11}{3} & 6 & 0 \\ 1 & -1 & \frac{2}{3} & 0 & 0 \\ 1 & \frac{1}{3} & \frac{-2}{3} & 0 & 0 \\ 1 & 2 & \frac{8}{3} & 0 & 0 \\ 1 & 4 & \frac{44}{3} & 0 & 48 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{4} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{8} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{8} \end{pmatrix}.$$

The eigenvalues of $\overline{S}$, 1, $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$, and $\frac{1}{8}$, are entries of the rightmost diagonal matrix Λ. These values are also the non-zero eigenvalues of the matrix S. The eigenvectors of $\overline{S}$ are the corresponding columns of the matrix $\overline{Z}$. For example, the eigenvector associated with $\frac{1}{2}$ is $\{-2, 1, \frac{1}{3}, 2, 4\}$.

Given this decomposition, we can now answer the question of whether the limit of $p_k[h, 0, 0]$ exists as $k \to \infty$. Let λ_i and $\overline{z}_i$ be the eigenvalues and eigenvectors of $\overline{S}$, respectively. For convenience, we index the eigenvalues in order of decreasing magnitude. If the subdivision matrix $\overline{S}$ is *non-defective* (i.e., $\overline{S}$ has a full complement of linearly independent eigenvectors), $\overline{p}_0$ can be expressed as a combination of these eigenvectors $\overline{z}_i$ of the form $\overline{p}_0 = \sum_i a_i \overline{z}_i$. All of the schemes discussed in this book have local subdivision matrices $\overline{S}$ that are non-defective. For schemes with defective subdivision matrices $\overline{S}$, a similar analysis can be performed using the Jordan decomposition of $\overline{S}$. (See page 191 of Strang [149] for more details on defective matrices and the Jordan decomposition.)

Given this spectral decomposition, we can now determine the limit of $\overline{p}_k$ as $k \to \infty$. In our running example, the initial vector $\overline{p}_0 = \{0, 0, 1, 0, 0\}$ can be expressed as a linear combination of the eigenvectors of $\overline{\mathrm{S}}$ via

$$\begin{pmatrix} 1 & -2 & \frac{11}{3} & 6 & 0 \\ 1 & -1 & \frac{2}{3} & 0 & 0 \\ 1 & \frac{1}{3} & \frac{-2}{3} & 0 & 0 \\ 1 & 2 & \frac{8}{3} & 0 & 0 \\ 1 & 4 & \frac{44}{3} & 0 & 48 \end{pmatrix} \begin{pmatrix} \frac{3}{5} \\ \frac{3}{10} \\ \frac{-9}{20} \\ \frac{11}{40} \\ \frac{1}{10} \end{pmatrix} == \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}. \tag{8.6}$$

Given this decomposition of $\overline{\mathrm{p}}_0$ into a linear combination of eigenvectors $\overline{\mathrm{z}}_i$, subsequent vectors $\overline{\mathrm{p}}_k$ satisfying equation 8.4 can be expressed as

$$\overline{\mathrm{p}}_k = \overline{\mathrm{S}}^k \overline{\mathrm{p}}_0 = \sum_i \lambda_i^k \mathrm{a}_i \overline{\mathrm{z}}_i. \tag{8.7}$$

For our running example, the eigenvalues satisfy $\lambda_0 == 1 > |\lambda_i|$ for $i > 0$. Hence, for $i > 0$ the λ_i^k converge to zero as $k \to \infty$. Therefore, the vectors $\overline{\mathrm{p}}_k$ converge to the vector $\mathrm{a}_0\overline{\mathrm{z}}_0 == \frac{3}{5}\{1, 1, 1, 1, 1\}$ as $k \to \infty$ or, equivalently, $p_\infty[\mathrm{h}, 0] == \frac{3}{5}$. Note that this method computes the exact limit value of the scheme at the origin in a manner similar to that of section 6.1.1.

In the surface case, this decomposition leads to a simple condition for a scheme to converge at an extraordinary vertex v. If the spectrum of $\overline{\mathrm{S}}$ has the form $\lambda_0 == 1 > |\lambda_i|$ for all $i > 0$, the vectors $\overline{\mathrm{p}}_k$ converge to the vector $\mathrm{a}_0\mathrm{z}_0$ as $k \to \infty$. For the surface schemes of the previous chapter, this condition on the eigenvalues of $\overline{\mathrm{S}}$ can be checked using the techniques discussed in section 8.3. Because these schemes are affinely invariant (i.e., the rows of their subdivision matrices S sum to one), the eigenvector z_0 associated with the eigenvalue 1 is the vector $\{\ldots, 1, 1, 1, \ldots\}$, and via equation 8.7 the limit of the $p_k[\mathrm{h}, 0, 0]$ as $k \to \infty$ is simply a_0. Note that this type of convergence analysis first appeared in Doo and Sabin [45]. Ball and Storry [6] later refined this analysis to identify the "natural configuration" associated with the scheme, a precursor of the "characteristic map" introduced in section 8.2.

8.1.3 Exact Evaluation Near an Extraordinary Vertex

The method of the previous section used spectral analysis to compute the exact value of $p_\infty[\mathrm{h}, 0, 0]$. Remarkably, Stam [145] shows that this analysis can be extended to compute the exact value of the limit function $p_\infty[\mathrm{h}, x, y]$ anywhere within ring[v] for any subdivision scheme whose uniform rules are those of box splines. To understand Stam's method, we first extend eigenvectors $\overline{\mathrm{z}}$ of $\overline{\mathrm{S}}$ to form eigenvectors

z of S. Specifically, the non-zero eigenvalues (and associated eigenvectors) of S and $\overline{S}$ are related as follows.

THEOREM 8.1 Let $\bar{z}$ be an eigenvector of $\overline{S}$ with associated right eigenvalue $\lambda > 0$ (i.e., $\overline{S}\,\bar{z} == \lambda\bar{z}$). Then there exists an extension of $\bar{z}$ to an infinite right eigenvector z of S with associated eigenvalue λ such that $Sz == \lambda z$.

The proof is straightforward and is left to the reader; the basic idea is to apply the subdivision scheme to $\bar{z}$ and multiply by $\frac{1}{\lambda}$. Due to the size of the neighborhood used in constructing $\overline{S}$, the result of this application is an extension of the eigenvector $\bar{z}$ to a larger portion of the mesh surrounding the extraordinary vertex v. For our running example, the eigenvector $\{-2, -1, \frac{1}{3}, 2, 4\}$ was defined over the two-ring of v. Applying the subdivision rules for the scheme and multiplying by 2 yields an extension of this vector to the three-ring of v; that is,

$$\begin{pmatrix} -3 \\ -2 \\ -1 \\ \frac{1}{3} \\ 2 \\ 4 \\ 6 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{8} & \frac{25}{32} & \frac{3}{32} & 0 & 0 \\ 0 & \frac{5}{8} & \frac{3}{8} & 0 & 0 \\ 0 & \frac{5}{24} & \frac{29}{40} & \frac{1}{15} & 0 \\ 0 & 0 & \frac{3}{5} & \frac{2}{5} & 0 \\ 0 & 0 & \frac{3}{20} & \frac{29}{40} & \frac{1}{8} \\ 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix} \cdot \begin{pmatrix} -2 \\ -1 \\ \frac{1}{3} \\ 2 \\ 4 \end{pmatrix}.$$

Applying the subdivision rules for S to this new vector $\{-3, -2, -1, \frac{1}{3}, 2, 4, 6\}$ yields an even longer extension of $\bar{z}$ of the form $\{-5, -4, -3, -2, -1, \frac{1}{3}, 2, 4, 6, 8, 10\}$. Further repetitions yield increasingly long extensions of the eigenvector $\{-2, -1, \frac{1}{3}, 2, 4\}$. Due to the structure of these eigenvectors, the limit functions associated with these eigenvectors have an intrinsically self-similar structure. The following theorem establishes a fundamental recurrence governing these limit functions.

THEOREM 8.2 Let λ be the eigenvalue associated with the right eigenvector z. If $p_0 = z$, consider the sequence of vectors defined by the recurrence $p_k = Sp_{k-1}$. The limit surface $p_\infty[h, x, y]$ associated with this process satisfies the relation

$$p_\infty[h, x, y] == \lambda p_\infty[h, 2x, 2y].$$

Proof Recall the definition of the associated functions $p_k[h, x, y]$ from equation 8.2:

$$p_k\left[h, \frac{i}{2^k}, \frac{j}{2^k}\right] = p_k[\![h, i, j]\!].$$

Because z is an eigenvector of S, the associated vectors p_{k-1} and p_k satisfy $p_k = \lambda p_{k-1}$. Therefore, the functions p_k and p_{k-1} are related via

$$p_k\left[h, \frac{i}{2^k}, \frac{j}{2^k}\right] == p_k[\![h, i, j]\!] == \lambda p_{k-1}[\![h, i, j]\!] == \lambda p_{k-1}\left[h, \frac{2i}{2^k}, \frac{2j}{2^k}\right]$$

for all $\{i, j\} \in (\mathbb{Z}^+)^2$. Because the remaining values of the functions p_k and p_{k-1} are defined by piecewise linear (or bilinear) interpolation, the two functions must agree over their entire domain; that is, $p_k[h, x, y] == \lambda p_{k-1}[h, 2x, 2y]$ for all $x, y \geq 0$. Given that $p_\infty[h, x, y]$ is the limit of the $p_k[h, x, y]$ as $k \to \infty$, the theorem follows immediately.

The recurrence of Theorem 8.2 is the key ingredient of Stam's exact evaluation algorithm for the limit function $p_\infty[h, x, y]$. For the sake of simplicity, we sketch this evaluation algorithm in the case of Catmull-Clark surfaces. Outside the one-ring of the extraordinary vertex v (i.e., where $\text{Max}[x, y] \geq 1$), Catmull-Clark surfaces are simply bicubic B-spline surfaces. There, the value of this tensor product surface can be computed at an arbitrary parameter location using Böhm's knot insertion algorithm for B-splines [11]. (For schemes whose uniform rules are derived from other types of box splines, the evaluation algorithm of section 2.3 based on cone splines can be used.) However, within the one-ring of an extraordinary vertex v the mesh structure is no longer tensor product, and this algorithm is no longer applicable.

To evaluate a Catmull-Clark surface within the one-ring of v (i.e., where $\text{Max}[x, y] < 1$), we instead express $\overline{p}_0$ in terms of the eigenvectors $\overline{z}_i$ for the scheme and evaluate these eigenvectors using Böhm's algorithm at $\{h, 2^k x, 2^k y\}$, where $2^k\,\text{Max}[x, y] \geq 1$. Theorem 8.2 can then be used to derive the values of the z_i and, consequently, p_0 at $\{h, x, y\}$. Note that this algorithm is not particularly efficient, and Stam suggests several nice improvements to this basic algorithm that greatly improve its speed.

To illustrate this method, we consider the problem of computing $p_\infty[1, \frac{1}{3}]$ for our running curve example. Due to the structure of uniform rules used in defining the subdivision matrix S for our example, the limit curve $p_\infty[\mathrm{h}, x]$ is a uniform cubic B-spline for $x \geq 2$. Therefore, we can use Böhm's algorithm in conjunction with Theorem 8.2 to compute the value of $\mathrm{p}_\infty[1, x]$ for any $x \in [0, 2]$. For example, if $\overline{\mathrm{p}}_0 = \{0, 0, 1, 0, 0\}$, the value of $p_\infty[1, \frac{1}{3}]$ can be computed as follows (♔):

- Express $\overline{\mathrm{p}}_0$ in terms of the eigenvectors of $\overline{\mathrm{S}}$, as shown in equation 8.6. Observe that the coefficients a_i are $\{\frac{3}{5}, \frac{3}{10}, -\frac{9}{20}, \frac{11}{40}, \frac{1}{10}\}$.
- For each eigenvector $\overline{\mathrm{z}}_\mathrm{i}$, compute its value at $\{1, \frac{8}{3}\}$ using Böhm's algorithm. (Note that we use a larger neighborhood than for Catmull-Clark surfaces due to the larger extent of the nonuniform rules in our curve example.) The values of the limit functions associated with these eigenvectors are, in order, $\{1, \frac{16}{3}, \frac{256}{9}, 0, \frac{4096}{27}\}$.
- Apply Theorem 8.2 three times to compute values of the limit functions associated with these eigenvectors at $\{1, \frac{1}{3}\}$. In particular, the values at $\{1, \frac{8}{3}\}$ are scaled by the cubes of the eigenvalues of $\overline{\mathrm{S}}$ (i.e., $\{1, \frac{1}{8}, \frac{1}{64}, \frac{1}{512}, \frac{1}{512}\}$). The resulting values at $\{1, \frac{1}{3}\}$ are, in order, $\{1, \frac{2}{3}, \frac{4}{9}, 0, \frac{8}{27}\}$.
- Multiply these values by their corresponding coefficients a_i, given previously. Thus, the value of $p_\infty[1, \frac{1}{3}]$ is exactly $\frac{17}{27}$.

8.2 Smoothness Analysis at an Extraordinary Vertex

Although the previous section established conditions for a subdivision scheme to converge at an extraordinary vertex, it did not answer a second, more difficult question: Do the surface schemes of the previous chapter produce limit surfaces that are smooth at extraordinary vertices? To answer this question, we must first settle on a suitable method for measuring the smoothness of a surface. The standard definition from mathematics requires that the surface in question be a smooth manifold. In particular, a limit surface $p_\infty[\mathrm{h}, x, y]$ is a $\mathcal{C}^\mathrm{k}$ *manifold* in the neighborhood of an extraordinary vertex v if $p_\infty[\mathrm{h}, x, y]$ is locally the graph of a $\mathcal{C}^\mathrm{k}$ function (see Fleming [63] for more details). Our goal is to develop a local reparameterization for $p_\infty[\mathrm{h}, x, y]$ such that the resulting surface is the graph of a function and then to develop conditions on the spectral structure of S for a scheme to produce smooth limit functions. We warn the reader that the mathematics of this section is rather involved. Those readers interested in simply testing whether a given scheme is smooth at an extraordinary vertex are advised to skip to section 8.3.

8.2.1 The Characteristic Map

For convergent schemes with a spectrum of the form $\lambda_0 == 1 > |\lambda_1| \geq |\lambda_2| > \ldots$, the key to this reparameterization is the eigenvectors z_1 and z_2 corresponding to the subdominant eigenvalues λ_1 and λ_2. These eigenvectors determine the local structure of the limit surface $p_\infty[h, x, y]$ in the neighborhood of v. Given an initial mesh of the form $\{M, p_0\}$ with $p_0 = \{z_1, z_2\}$, we define the *characteristic map* $\psi = \{\psi_s, \psi_t\}$ associated with the subdominant eigenvectors z_1 and z_2 to be the limit of the subdivision process $p_k = Sp_{k-1}$; that is,

$$\psi[h, x, y] = p_\infty[h, x, y].$$

Figure 8.1 shows plots of the mesh $\{M, \{z_1, z_2\}\}$ for both Loop and Catmull-Clark subdivision for valences three to eight. Reif introduced the characteristic map in his ground-breaking analysis of the smoothness of subdivision schemes at extraordinary vertices [128]. The characteristic map ψ is *regular* if it is $1 - 1$ and onto everywhere. Subdivision schemes with regular characteristic maps define limit surfaces that are locally manifolds at the extraordinary vertex v. In particular, if ψ is regular, the inverse map ψ^{-1} exists everywhere and provides the reparameterization needed to convert the parametric surfaces $p_\infty[h, x, y]$ into functional form. Once in this functional form, analyzing the smoothness of the limit surface reduces to determining whether certain derivatives of the functional form exist and whether they are continuous at v.

Up to now, the possibility of the eigenvalues λ_i and their corresponding eigenvectors z_i being complex has not affected our analysis. However, if the subdominant eigenvalues are complex, the characteristic map is complex valued and the analysis that follows is much more difficult. Luckily, for most schemes considered in this book, the subdominant eigenvalues λ_1 and λ_2 are real and positive. For a few of the schemes, such as the face-splitting quad scheme of section 7.4.1, one round of the subdivision induces some type of rotation in the mesh M. As a result, the subdominant eigenvalues for the associated subdivision matrix are typically complex conjugates. In these cases, constructing a subdivision matrix that represents two or more rounds of subdivision usually cancels out the rotational component and yields a subdivision matrix with real eigenvalues. Consequently, we restrict our analysis to those schemes for which λ_1 and λ_2 are real and positive. For the reader interested in a more general analysis of the fully complex case, we recommend consulting Zorin [169].

To illustrate the nature of the characteristic map, we explicitly construct this map for our running curve example. In the univariate case, the characteristic map

$\psi[\mathrm{h}, x]$ is the scalar limit function associated with the subdominant eigenvector $\mathrm{z}_1 = \{\ldots, -3, -2, -1, \frac{1}{3}, 2, 4, 6, \ldots\}$. Now, observe that the entries of this vector satisfy $\mathrm{z}_1[\![0, \mathrm{i}]\!] = -\mathrm{i}$ and $\mathrm{z}_1[\![1, \mathrm{i}]\!] = 2\mathrm{i}$ for $\mathrm{i} > 0$. Because the subdivision rules for uniform cubic B-splines have linear precision, the associated characteristic map ψ satisfies $\psi[0, x] = -x$ and $\psi[1, x] = 2x$. Before attempting to invert ψ, we must first verify that ψ is regular (i.e., a $1 - 1$ and onto covering of the parameter line). To this end, we observe that $\psi[0, x]$ covers the negative portion of the parameter line in a $1 - 1$ manner, whereas $\psi[1, x]$ covers the positive portion of the line in a $1 - 1$ manner. Therefore, ψ is regular and possesses an inverse map ψ^{-1} of the form

$$\psi^{-1}[s] = \begin{cases} \{0, -s\} & \text{if } s \leq 0, \\ \left\{1, \frac{1}{2}s\right\} & \text{if } s \geq 0. \end{cases}$$

In the curve case, verifying that the characteristic map ψ is regular is easy. In the surface case, determining whether the characteristic map is regular is actually quite difficult. One important trick in simplifying this task lies in applying Theorem 8.2. The characteristic map ψ satisfies the recurrence

$$\psi[\mathrm{h}, x, y] == \psi[\mathrm{h}, 2x, 2y] \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}. \tag{8.8}$$

If λ_1 and λ_2 are real and positive, applying the affine transformation $\left(\begin{smallmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{smallmatrix}\right)$ to the map ψ does not affect its regularity. Thus, if we can prove that ψ is regular on an annulus surrounding v of sufficient width, the recurrence of equation 8.8 implies that ψ is regular everywhere except at v. As long as the characteristic map winds around v exactly once, it is also regular at the origin. The details of verifying the regularity of the characteristic map of Loop's scheme on such an annulus are considered in section 8.3.

The main use of the characteristic map ψ lies in the fact that it provides an extremely convenient parameterization for the limit surface $p_\infty[\mathrm{h}, x, y]$ at the extraordinary vertex v. Given an initial vector q of scalar values, consider the limit surface $p_\infty[\mathrm{h}, x, y]$ associated with the initial vector $\mathrm{p}_0 = \{\mathrm{z}_1, \mathrm{z}_2, \mathrm{q}\}$. If the characteristic map ψ is regular, this three-dimensional limit surface can be viewed as the graph of a single-valued function with respect to the first two coordinates. More precisely, we define a bivariate function $\phi[\mathrm{q}]$ associated with a vector q of scalar values as follows:

$$\phi[\mathrm{q}][s, t] = p_\infty[\psi^{(-1)}[s, t]], \tag{8.9}$$

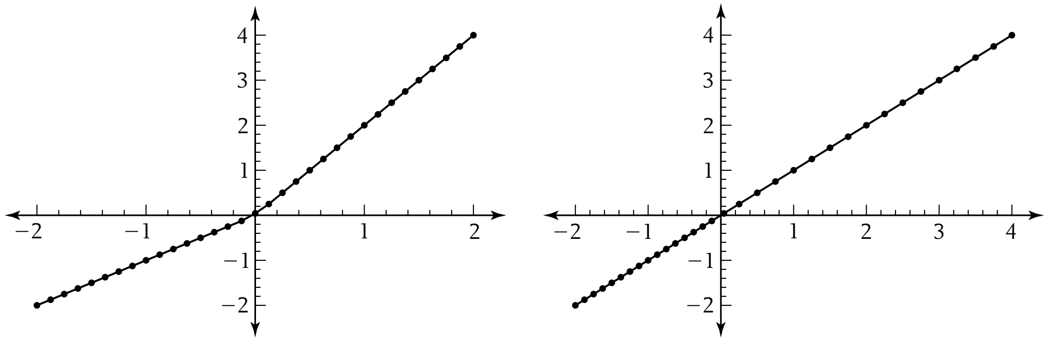


Figure 8.4 Reparameterization using the inverse of the characteristic map.

where $p_0 = q$ is the initial vector associated with p_∞. Observe that this definition is equivalent to requiring that $\phi[q][\psi[h, x, y]] == p_\infty[h, x, y]$. Note that the function $\phi[q]$ is linear in q due to the linearity of the subdivision process.

Figure 8.4 illustrates the effect of reparameterization by the inverse of the characteristic map for our running curve example. This figure shows two plots of the eigenvector $\bar{z}_1 = \{-2, -1, \frac{1}{3}, 2, 4\}$ after three rounds of subdivision. The left-hand plot shows the result plotted on the grid $\frac{1}{8}\mathbb{Z}$. Note that the resulting curve has a discontinuity in its derivative at the origin. In the right-hand plot, the grid $\frac{1}{8}\mathbb{Z}$ has been reparameterized using ψ^{-1}. The effect on the resulting function $\phi[z_1]$ is to stretch the positive portion of the parameter line by a factor of two and to smooth out the discontinuity at the origin. Observe that if z_1 is the eigenvalue used in defining ψ, then $\phi[z_1]$ is always exactly the linear function s by construction.

8.2.2 Eigenfunctions

For subdivision schemes with regular characteristic maps, we can now introduce our main theoretical tool for analyzing the behavior of these schemes. Recall that if the local subdivision matrix $\bar{S}$ is non-defective any local coefficient vector $\bar{q}$ can be written in the form $\sum_i a_i \bar{z}_i$. Therefore, its corresponding function $\phi[q]$ has an expansion of the form

$$\phi[q][s, t] == \sum_i a_i \phi[z_i][s, t]$$

for all $\{s, t\}$ in the one-ring of v. Note that this sum need only include those eigenvectors z_i that are the extensions of the eigenvectors $\bar{z}_i$ of $\bar{S}$. Due to this decomposition, we can now focus our attention on analyzing the smoothness of the *eigenfunctions* $\phi[z]$. These eigenfunctions $\phi[z]$ satisfy a fundamental scaling recurrence analogous to the recurrence of Theorem 8.2. This recurrence is the main tool for understanding the behavior of stationary schemes at extraordinary vertices. The following theorem, first appearing in Warren [158], establishes this recurrence.

THEOREM 8.3 If λ is an eigenvalue of S with associated eigenvector z, the eigenfunction $\phi[z]$ satisfies the recurrence

$$\phi[z][s, t] == \lambda\, \phi[z]\left[\frac{s}{\lambda_1}, \frac{t}{\lambda_2}\right], \tag{8.10}$$

where λ_1 and λ_2 are the subdominant eigenvalues of S.

Proof Consider the limit function $p_\infty[h, x, y]$ associated with the initial vector $p_0 = z$. Applying Theorem 8.2 to this limit surface yields $p_\infty[h, x, y] == \lambda p_\infty[h, 2x, 2y]$. Applying the definition of ϕ (i.e., equation 8.9) to both sides of this equation yields a functional relation of the form

$$\phi[z][\psi[h, x, y]] == \lambda\, \phi[z][\psi[h, 2x, 2y]]. \tag{8.11}$$

If the characteristic map $\psi[h, x, y]$ is assigned the coordinates $\{s, t\}$ (i.e., $\{s, t\} = \psi[h, x, y]$), the dilated map $\psi[h, 2x, 2y]$ produces the coordinates $\{\frac{s}{\lambda_1}, \frac{t}{\lambda_2}\}$ due to the recurrence relation of equation 8.8. Replacing $\psi[h, x, y]$ and $\psi[h, 2x, 2y]$ by their equivalent definitions in terms of s and t in equation 8.11 yields the theorem.

Returning to our running example, the eigenfunction $\phi[z_0][s]$ had an associated eigenvalue of 1 and therefore satisfied the scaling relation $\phi[z_0][s] = \phi[z_0][2s]$. This observation is compatible with the fact that $\phi[z_0][s] == 1$. Likewise, the eigenfunction $\phi[z_1][s]$ had an associated eigenvalue of $\frac{1}{2}$ and satisfied the scaling relation $\phi[z_0][s] = \frac{1}{2}\phi[z_0][2s]$. Again, this relation is compatible with $\phi[z_1][s] == s$. In the next two sections, we deduce the structure of the remaining eigenfunctions.

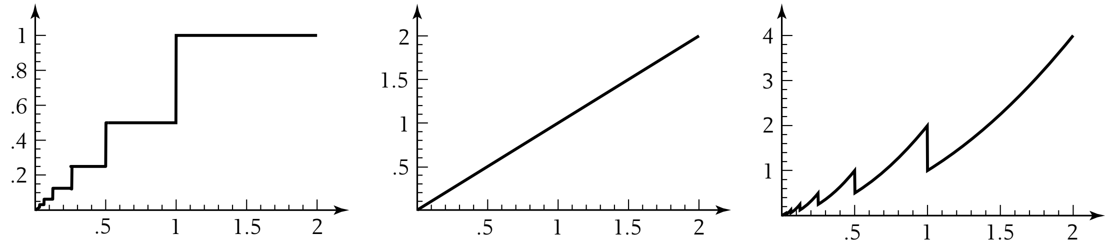


Figure 8.5 Three functions that satisfy the scaling relation $f[s] = \frac{1}{2}f[2x]$.

8.2.3 Sufficient Conditions for $\mathcal{C}^m$ Continuity

The smoothness of the eigenfunction $\phi[z]$ away from the origin depends entirely on the structure of the uniform rules associated with the subdivision scheme. All that remains is to analyze the smoothness of these eigenfunctions at the origin. The key to this analysis is the scaling recurrence of equation 8.10. In particular, the magnitude of the eigenvalue λ (taken in relation to λ_1 and λ_2) determines the smoothness of the eigenfunction $\phi[z]$ at the origin.

To illustrate the effect of λ, consider a univariate function $f[s]$ satisfying a scaling relation of the form $f[s] = \lambda f[2s]$. Once $f[s]$ has been specified on the interval $[1, 2)$, the scaling relation determines the remaining values of $f[s]$ for all $s \geq 0$. If $0 < |\lambda| < 1$, the value of $f[s]$ on the interval $[1, 2)$ has no effect on the behavior of $f[s]$ at the origin. Substituting $s = 0$ into the scaling relation yields $f[0] == \lambda f[0]$, and therefore that $f[0] == 0$. More important, the scaling relation forces the function $f[s]$ to converge to zero in the neighborhood of the origin. For example, Figure 8.5 shows plots of three possible functions $f[s]$ satisfying the scaling relation $f[s] = \frac{1}{2}f[2x]$. Observe that all three functions converge to zero at the origin.

By taking the derivative of both sides of the scaling relation of equation 8.10, a similar method can be used to analyze various derivatives of $f[s]$ at the origin. The following theorem, credited to the authors, ties these observations together and relates the magnitude of the eigenvalue λ to the smoothness of its associated eigenfunction $\phi[z]$ at the origin.

THEOREM 8.4 Let S be a bivariate subdivision matrix with spectrum $\lambda_0 == 1 > \lambda_1 \geq \lambda_2 > |\lambda_3| > \ldots$. Consider an eigenfunction $\phi[z]$ of S whose associated eigenvalue λ satisfies $|\lambda| < \lambda_2^m$. If the function $\phi[z]$ is $\mathcal{C}^m$ continuous everywhere except at the origin, then $\phi[z]$ is $\mathcal{C}^m$ continuous at the origin.

Proof Consider the case when $m == 0$. By hypothesis, the eigenvalue λ satisfies $|\lambda| < 1$. Evaluating the recurrence of Theorem 8.3,

$$\phi[z][s, t] == \lambda\, \phi[z]\left[\frac{s}{\lambda_1}, \frac{t}{\lambda_2}\right],$$

at $\{s, t\} == \{0, 0\}$ yields $\phi[z][0, 0] == 0$. We next show that the limit of $\phi[z][s, t]$ as $\{s, t\}$ approaches the origin is zero. Consider the behavior of $\phi[z]$ on the annulus A_k of the form

$$A_k = \{\psi[h, x, y] \mid 0 \le h < n, 2^{-k} \le x + y \le 2^{1-k}\}.$$

Because $\phi[z]$ is continuous on the bounded annulus A_0, its absolute value must be bounded on this annulus by some value ν. Due to the recurrence of Theorem 8.3, the absolute value of $\phi[z]$ on the larger annulus A_k is bounded by $\lambda^k \nu$. Because $|\lambda| < 1$, the function $\phi[z][s, t]$ is converging to zero as $\{s, t\}$ converges to the origin. Therefore, $\phi[z]$ is continuous at the origin.

For $m > 0$, we show that $\phi[z]^{(i,j)}$ is continuous (with value zero) at the origin for $0 \le i + j \le m$. The key step in this proof is to construct a recurrence for $\phi[z]^{(i,j)}$ of the form

$$\phi[z]^{(i,j)}[s, t] == \frac{\lambda}{\lambda_1^i \lambda_2^j} \phi[z]^{(i,j)}\left[\frac{s}{\lambda_1}, \frac{t}{\lambda_2}\right]$$

by taking the appropriate derivatives of both sides of the recurrence of Theorem 8.3. Due to the hypothesis that $|\lambda| < \lambda_2^m$, the eigenvalue λ satisfies $|\lambda| < \lambda_1^i \lambda_2^j$ because $0 < \lambda_2 \le \lambda_1$. The remainder of the proof follows the structure of the case $m == 0$.

At this point, we have assembled all of the tools necessary to prove that the surface schemes of Chapter 7 (i.e., Catmull-Clark and Loop) produce limit surfaces that are $\mathcal{C}^1$ at an extraordinary vertex v. In particular, these schemes have subdivision matrices S centered at v that satisfy three properties:

- The spectrum of S has real eigenvalues of the form $\lambda_0 == 1 > \lambda_1 \ge \lambda_2 > |\lambda_3| \ge \ldots$.
- The dominant eigenvector z_0 is the vector of ones.
- The characteristic map defined by the subdominant eigenvectors z_1 and z_2 is regular.

Now, given an arbitrary vector of coefficients q, the smoothness of the function $\phi[\mathrm{q}]$ is determined by the smoothness of the eigenfunctions $\phi[\mathrm{z}_i]$ associated with the subdivision scheme. Because the dominant eigenvector z_0 is the vector of ones, the corresponding eigenfunction $\phi[\mathrm{z}_0]$ is simply the constant function. The eigenfunctions $\phi[\mathrm{z}_1]$ and $\phi[\mathrm{z}_2]$ correspond to the functions *s* and *t*, respectively, due to their use in defining the characteristic map ψ. All three of these eigenfunctions are polynomial, and therefore $\mathcal{C}^\infty$ everywhere.

Because the uniform rules associated with these schemes produce $\mathcal{C}^2$ limit surfaces, we can use Theorem 8.4 to analyze the smoothness of the remaining eigenfunctions at the origin. By assumption, the eigenvalues associated with the remaining eigenfunctions satisfy $|\lambda_i| < \lambda_2$ for $i > 2$. Therefore, these eigenfunctions $\phi[\mathrm{z}_i]$ are at least $\mathcal{C}^1$ at the origin, and consequently any linear combination of these eigenfunctions $\phi[\mathrm{q}]$ must be $\mathcal{C}^1$ at the origin. Note the fact that the associated uniform scheme converges to $\mathcal{C}^2$ limit functions does not automatically guarantee that all of these eigenfunctions $\phi[\mathrm{z}_i]$ are $\mathcal{C}^2$ at the origin. Theorem 8.4 guarantees only that those eigenfunctions with eigenvalues λ_i satisfying $|\lambda_i| < \lambda_2^2$ are guaranteed to be $\mathcal{C}^2$ continuous at the origin. Those eigenfunctions whose eigenvalues λ_i satisfy $\lambda_2 > |\lambda_i| > \lambda_2^2$ may only be $\mathcal{C}^1$ continuous at the origin.

This observation gives insight into constructing surface schemes that are $\mathcal{C}^m$ continuous at extraordinary vertices. The trick is to construct a set of subdivision rules whose subdivision matrix S has a spectrum with $|\lambda_i| < \lambda_2^m$ for all $i > 2$. If the uniform rules in S are $\mathcal{C}^m$, the corresponding eigenfunctions $\phi[\mathrm{z}_i]$ are also $\mathcal{C}^m$ continuous. Note, however, that all derivatives up to order m of these eigenfunctions are necessarily zero at the origin. For example, the subdivision rules at the endpoint of a natural cubic spline have a subdivision matrix $\overline{\mathrm{S}}$ with a spectral decomposition $\overline{\mathrm{S}}\,\overline{\mathrm{Z}} == \overline{\mathrm{Z}}\Lambda$ of the form

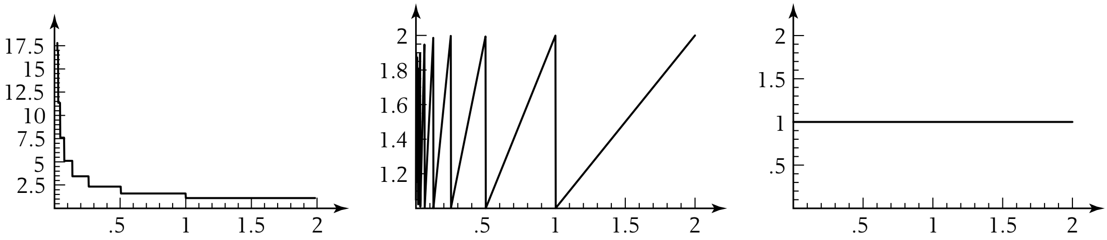
$$\begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{8} & \frac{3}{4} & \frac{1}{8} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 2 & \frac{1}{3} \end{pmatrix} == \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 2 & \frac{1}{3} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{8} \end{pmatrix}.$$

Due to Theorem 8.4, this scheme is at least $\mathcal{C}^2$ at the endpoint, because the eigenfunction $\phi[\mathrm{z}_2]$ has $\lambda_2 < \lambda_1^2$. Note that this eigenfunction $\phi[\mathrm{z}_2]$ is *flat* (i.e., has a second derivative of zero) at the endpoint. Prautzsch and Umlauf [125] have designed surface schemes that are $\mathcal{C}^2$ at extraordinary vertices based on this idea. They construct subdivision rules at an extraordinary vertex v such that the resulting subdivision matrix S has a spectrum of the form $1 > \lambda_1 == \lambda_2 > |\lambda_3| > \ldots$, where $|\lambda_3| < \lambda_2^2$. Just as in the previous curve case, the resulting limit surfaces are flat (i.e., have zero curvatures) at the extraordinary vertex v.

8.2.4 Necessary Conditions for $\mathcal{C}^m$ Continuity

Just as the scaling relation of equation 8.10 was the key to deriving sufficient conditions for $\mathcal{C}^m$ continuity at an extraordinary vertex, this relation is also the main tool in deriving necessary conditions for a surface scheme to converge to a $\mathcal{C}^m$ limit function at an extraordinary vertex. The key technique is to examine those functions that satisfy a scaling relation of the type $f[s] = \lambda f[2s]$, where $|\lambda| \geq 1$. The left-hand plot of Figure 8.6 shows an example of a function satisfying this scaling relation $f[s] = \frac{3}{2}f[2s]$; this function diverges at the origin independent of the value of $f[s]$ on the defining interval $[1, 2)$. The middle and right-hand plots show examples of functions satisfying $f[s] = f[2s]$. In the middle plot, the function $f[s]$ varies (i.e., is not a constant) on the interval $[1, 2)$. As a consequence of the scaling relation, the function $f[s]$ cannot be continuous at the origin. The right-hand plot shows a function that is constant on the interval $[1, 2)$. Observe that only constant functions can both satisfy the scaling relation and be continuous at the origin.

By taking various derivatives of both sides of the scaling relation of equation 8.10, this observation can be generalized to functions with higher-order smoothness. In particular, if a subdivision scheme converges to smooth limit functions, its eigenfunctions corresponding to dominant eigenvalues must be polynomials. The following theorem deals with the case when the eigenvalues λ_1 and λ_2 are equal. Variants of this theorem appear in [158] and [123].



(a) (b) (c)

Figure 8.6 A function satisfying $f[s] = \frac{3}{2}f[2s]$ (a) and two functions satisfying $f[s] = f[2s]$ (b and c).

THEOREM 8.5 Let S be a subdivision matrix with spectrum $\lambda_0 == 1 > \lambda_1 == \lambda_2 > |\lambda_3| > \ldots$. If $\phi[z]$ is a non-zero eigenfunction of S that is C^m continuous everywhere and whose associated eigenvalue λ satisfies $1 \geq \lambda \geq \lambda_1^m$, there exists an integer $0 \leq i \leq m$ such that $\lambda = \lambda_1^i$, with $\phi[z][s, t]$ being a homogeneous polynomial of degree i in s and t.

Proof Consider the functions $\phi[z]^{(i,j)}$ for all $i + j == m$. Taking the appropriate derivatives of both sides of equation 8.10 yields a new recurrence of the form

$$\phi[z]^{(i,j)}[s, t] == \frac{\lambda}{\lambda_1^i \lambda_2^j} \phi[z]^{(i,j)}\left[\frac{s}{\lambda_1}, \frac{t}{\lambda_2}\right].$$

Based on this recurrence, we claim that the functions $\phi[z]^{(i,j)}$ must be constant functions. By hypothesis, $\lambda \geq \lambda_1^m$, and therefore $\lambda \geq \lambda_1^i \lambda_2^j$, because $\lambda_1 == \lambda_2$. If $\lambda > \lambda_1^i \lambda_2^j$, either the function $\phi[z]^{(i,j)}$ is identically zero or it diverges as $\{s, t\} \to \{0, 0\}$. If $\lambda == \lambda_1^i \lambda_2^j$, then $\phi[z]^{(i,j)}$ either is the constant function or has a discontinuity at $\{s, t\} == \{0, 0\}$. Because the function $\phi[z]^{(i,j)}$ is continuous by hypothesis, $\phi[z]^{(i,j)}$ must be a constant function in either case.

Given that $\phi[z]^{(i,j)}$ is a constant function for all $i + j == m$, the original function $\phi[z]$ is a polynomial function of, at most, degree m. To conclude, we observe that equation 8.10 is satisfied only by homogeneous polynomial functions of degree i where $\lambda == \lambda_1^i$.

Note that a similar theorem holds in the case when $\lambda_1 > \lambda_2$. In this case, there exist integers $0 \leq i + j \leq m$ such that $\lambda == \lambda_1^i \lambda_2^j$, with $\phi[z]$ being a multiple of the monomial $s^i t^j$.

To illustrate this theorem, we complete our analysis of the running curve example by constructing explicit representations for the eigenfunctions associated with the scheme. At this point, we reveal that the matrix S associated with this example is the subdivision matrix that maps the control points for a nonuniform cubic B-spline with knots at $\{\ldots, -3, -2, -1, 0, 2, 4, 6, \ldots\}$ to the new set of control points associated with a B-spline whose knots lie at $\frac{1}{2}\{\ldots, -3, -2, -1, 0, 2, 4, 6, \ldots\}$. The particular entries of S were computed using the blossoming approach to B-splines described in Ramshaw [126] and Seidel [140]. One consequence of this observation is that the eigenfunctions of S must be cubic B-splines and therefore C^2 piecewise polynomials. Given this fact, we may now apply Theorem 8.3.

Given the spectral decomposition $\overline{\mathsf{S}}\,\overline{\mathsf{Z}} == \overline{\mathsf{Z}}\Lambda$, as shown in equation 8.5, the eigenfunctions $\phi[\mathsf{z}_0]$, $\phi[\mathsf{z}_1]$, and $\phi[\mathsf{z}_2]$ must reproduce constant multiples of the functions 1, s, and s^2, respectively, by Theorem 8.5. Via blossoming, the reader may verify that the remaining eigenfunctions $\phi[\mathsf{z}_3]$ and $\phi[\mathsf{z}_4]$ are piecewise cubic functions of the form

$$\phi[\mathsf{z}_3][s] = \begin{cases} -s^3 & \text{if} \quad s \leq 0, \\ 0 & \text{if} \quad s \geq 0, \end{cases}$$

$$\phi[\mathsf{z}_4][s] = \begin{cases} 0 & \text{if} \quad s \leq 0, \\ s^3 & \text{if} \quad s \geq 0. \end{cases}$$

As observed in the previous section, constructing a surface scheme that satisfies these necessary conditions for $\mathsf{m} == 1$ is easy. The characteristic map ψ automatically yields eigenfunctions $\phi[\mathsf{z}_1]$ and $\phi[\mathsf{z}_2]$ that reproduce the functions s and t. Unfortunately, Theorem 8.5 is much more restrictive in the case of $\mathsf{m} > 1$. To construct a non-flat scheme that is C^2 continuous at extraordinary vertices, the scheme must possess eigenvectors z_3, z_4, and z_5 whose associated eigenfunctions $\phi[\mathsf{z}_3]$, $\phi[\mathsf{z}_4]$, and $\phi[\mathsf{z}_5]$ reproduce linearly independent quadratic functions with respect to the characteristic map ψ. Peters and Umlauf [117] give an interesting characterization of this condition in terms of a system of differential equations.

Building a stationary subdivision matrix S that satisfies these conditions for arbitrary valence extraordinary vertices is extremely difficult. For example, one consequence of these conditions is that the uniform rules used in constructing S must have high degree. Prautzsch and Reif [124] show that the uniform rules of S must reproduce polynomials of at least degree $2\mathsf{m} + 2$ if S defines a C^{m} non-flat scheme. Both Prautzsch [122] and Reif [129] have developed C^{m} schemes that achieve this bound. Unfortunately, both schemes are rather complex and beyond the scope of this book.

8.3 Verifying the Smoothness Conditions for a Given Scheme

The previous two sections developed conditions on the dominant eigenvalues and eigenvectors of $\overline{\mathsf{S}}$ for a subdivision scheme to be convergent/smooth at an extraordinary vertex v. In the running curve example, computing these eigenvalues and eigenvectors was easy using *Mathematica*. For surface meshes with extraordinary vertices, this computation becomes more difficult because the valence of the vertex affects the spectral structure of $\overline{S}$. In this section, we discuss techniques for computing these eigenvalues/eigenvectors based on the block cyclic structure of $\overline{S}$. Using these techniques, we show that the triangle schemes discussed in section 7.3 converge to smooth limit surfaces at extraordinary vertices. A similar analysis can be used to show that the quad schemes of section 7.2 also converge to a smooth limit surface at extraordinary vertices.

8.3.1 Computing Eigenvalues of Circulant Matrices

For curve schemes, the construction of $\overline{S}$ and the computation of its spectral decomposition are easy given the subdivision rules for the scheme. For surface schemes on uniform meshes, this spectral analysis is equally easy because each vertex of a uniform mesh has the same valence. As a result, the local subdivision matrix $\overline{S}$ centered at a vertex is the same for every vertex in a uniform mesh. However, for a surface scheme over meshes with extraordinary vertices, the problem of computing the spectral decomposition at these extraordinary vertices becomes more complicated because the local topology at an extraordinary vertex is parameterized by its valence n. The structure of the local subdivision matrix $\overline{S}$ centered at this extraordinary vertex depends on this valence n.

Fortunately, given the subdivision rules for a surface scheme, the eigenvalues and eigenvectors of $\overline{S}$ can be computed in closed form as a function of n. This section outlines the basic idea underlying this computation. The key observation is that almost all of $\overline{S}$ can be expressed as a block matrix whose blocks are each circulant matrices. An $(n \times n)$ matrix C is *circulant* if the rows of C are successive rotational shifts of a single fundamental row c; that is,

$$C[\![i, j]\!] == c[\![\mathrm{mod}[j - i, n]]\!] \tag{8.12}$$

for $0 \leq i, j < n$. (Note that for the rest of this section we index the rows and columns of C starting at zero.) As done in previous chapters, the row c can be used to define an associated generating function $c[x]$ of the form $\sum_{i=0}^{n-1} c[\![i]\!]x^i$.

The usefulness of this generating function $c[x]$ lies in the fact that eigenvalues of C are the values of $c[x]$ taken at the n solutions to the equation $x^n == 1$. These solutions, *the* n *roots of unity,* are all powers of a single fundamental solution ω_n of the form

$$\omega_n = \mathbb{e}^{\frac{2\pi \mathbb{i}}{n}} == \mathrm{Cos}\left[\frac{2\pi}{n}\right] + \mathrm{Sin}\left[\frac{2\pi}{n}\right]\mathbb{i}.$$

The following theorem, from Davis [36], makes this relationship explicit.

THEOREM 8.6 An $n \times n$ circulant matrix C has n eigenvalues λ_i of the form $c[\omega_n^i]$ for $0 \le i < n$. Moreover, their associated eigenvectors z_i have the form $\{1, x, x^2, \ldots, x^{n-1}\}^T$, where $x = \omega_n^i$ for $0 \le i < n$.

Proof Consider the product of the circulant matrix C and the column vector $\{1, x, x^2, \ldots, x^{n-1}\}^T$. By definition, the zeroth entry of the resulting column vector is $c[x]$. If we restrict x to satisfy $x^n == 1$, the jth entry of this product can be rewritten as $c[x]x^n$. More generally, the product of the matrix C and the vector $\{1, x, x^2, \ldots, x^{n-1}\}^T$ is exactly the vector $c[x]\{1, x, x^2, \ldots, x^{n-1}\}^T$, where x is one of the n solutions to the equation $x^n == 1$.

To illustrate this theorem, consider a circulant matrix C of the form

$$C = \begin{pmatrix} 2 & 1 & 0 & 0 & 0 & 1 \\ 1 & 2 & 1 & 0 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 & 0 \\ 0 & 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 0 & 1 & 2 & 1 \\ 1 & 0 & 0 & 0 & 1 & 2 \end{pmatrix}.$$

Using *Mathematica* (♔), the reader can verify that this matrix has eigenvalues $\{0, 1, 1, 3, 3, 4\}$. The associated generating function $c[x]$ has the form $2 + x + x^5$. Evaluating $c[x]$ at various powers of $\omega_6 == \frac{1}{2} + \frac{i\sqrt{3}}{2}$ also yields the desired eigenvalues. One common trick used in manipulating $c[x]$ is to reduce the degree of high powers of x in $c[x]$ via the relation $x^n == 1$. This reduction does not affect the value of $c[x]$ at powers of ω_n^h because $(\omega_n^h)^n == (\omega_n^n)^h == 1$. For our current example, the mask $c[x]$ can be rewritten as $2 + x + x^{-1}$. This trick is particularly useful in expressing $c[x]$ in such a way that it does not explicitly depend on n.

One nice feature of this analysis is that it can be generalized to block circulant matrices. For example, consider an $(m \times m)$ block matrix (C_{ij}) whose components C_{ij}

are themselves $(n \times n)$ circulant matrices. If c_{ij} is the zeroth row of the circulant matrix C_{ij}, we can construct an $(m \times m)$ matrix of generating functions $(c_{ij}[x])$ corresponding to the block matrix (C_{ij}). Now, the $m\,n$ eigenvalues of the original block circulant matrix (C_{ij}) correspond to the m eigenvalues of the matrix $(c_{ij}[x])$. Each of these m eigenvalues of $(c_{ij}[x])$ is a function of x. Evaluating each of these m functions at the n roots of unity yields the mn eigenvalues of (C_{ij}). If the vector $\{z_0[x], z_1[x], \ldots, z_{m-1}[x]\}^T$ is one of the m right eigenvectors of $(c_{ij}[x])$, then (C_{ij}) has n associated right eigenvectors of the form

$$\begin{array}{cccc} \{z_0[x], & z_0[x]x, & \ldots, & z_0[x]x^{n-1}, 1 \\ z_1[x], & z_1[x]x, & \ldots, & z_1[x]x^{n-1}, \\ \ldots, & \ldots, & \ldots, & \ldots, \\ z_{m-1}[x], & z_{m-1}[x]x, & \ldots, & z_{m-1}[x]x^{n-1}\}^T, \end{array} \tag{8.13}$$

where $x = \omega_n^h$ for $0 \leq h < n$. As an example, consider a block circulant matrix of the form

$$\begin{pmatrix} 2 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 4 \end{pmatrix}.$$

Using *Mathematica*, the reader may verify that this block circulant matrix has the twelve eigenvalues $\{0, 1, 1, 2, 3, 3, 3, 3, 4, 5, 5, 6\}$ (♔). The associated (2×2) matrix of generating functions $(c_{ij}[x])$ has the form

$$\begin{pmatrix} 2 + x + x^{-1} & 1 \\ 0 & 4 + x + x^{-1} \end{pmatrix}.$$

Because this matrix is upper triangular, its eigenvalues (taken as functions of x) are the entries of its diagonal (i.e., $2 + x + x^{-1}$ and $4 + x + x^{-1}$). Evaluating these two eigenvalues at $x = \omega_6^h$ for $0 \leq h < 6$ yields the twelve eigenvalues of the original block circulant matrix. The eigenvectors of $(c_{ij}[x])$ are independent of x and have the form $\{1, 0\}$ and $\{1, 2\}$. Therefore, the twelve eigenvectors of (C_{ij}) have the form

$$\{1, x, x^2, x^3, x^4, x^5, 0, 0, 0, 0, 0, 0\},$$
$$\{1, x, x^2, x^3, x^4, x^5, 2, 2x, 2x^2, 2x^3, 2x^4, 2x^5\},$$

where $x = \omega_6^h$ for $0 \leq h < 6$.

8.3.2 Computing Eigenvalues of Local Subdivision Matrices

Given the block circulant method described in the previous section, we can now compute the eigenvalues and eigenvectors of the subdivision matrices for Loop's scheme (and its triangular variants). For Loop's scheme, the neighborhood of an n-valent extraordinary vertex v consists of the two-ring of v. The vertices in this neighborhood can be partitioned into v plus three disjoint sets of n vertices with indices ij of the form 10, 11, and 20 (see Figure 8.2). If we order the rows and columns of the local subdivision matrix $\overline{S}$ according to this partition, the resulting matrix has the form (for $n == 4$)

$$\overline{S} = \begin{pmatrix}
1 - w[n] & \frac{w[n]}{n} & \frac{w[n]}{n} & \frac{w[n]}{n} & \frac{w[n]}{n} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{3}{8} & \frac{3}{8} & \frac{1}{8} & 0 & \frac{1}{8} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{3}{8} & \frac{1}{8} & \frac{3}{8} & \frac{1}{8} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{3}{8} & 0 & \frac{1}{8} & \frac{3}{8} & \frac{1}{8} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{3}{8} & \frac{1}{8} & 0 & \frac{1}{8} & \frac{3}{8} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{8} & \frac{3}{8} & \frac{3}{8} & 0 & 0 & \frac{1}{8} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{8} & 0 & \frac{3}{8} & \frac{3}{8} & 0 & 0 & \frac{1}{8} & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{8} & 0 & 0 & \frac{3}{8} & \frac{3}{8} & 0 & 0 & \frac{1}{8} & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{8} & \frac{3}{8} & 0 & 0 & \frac{3}{8} & 0 & 0 & 0 & \frac{1}{8} & 0 & 0 & 0 & 0 \\
\frac{1}{16} & \frac{5}{8} & \frac{1}{16} & 0 & \frac{1}{16} & \frac{1}{16} & 0 & 0 & \frac{1}{16} & \frac{1}{16} & 0 & 0 & 0 \\
\frac{1}{16} & \frac{1}{16} & \frac{5}{8} & \frac{1}{16} & 0 & \frac{1}{16} & \frac{1}{16} & 0 & 0 & 0 & \frac{1}{16} & 0 & 0 \\
\frac{1}{16} & 0 & \frac{1}{16} & \frac{5}{8} & \frac{1}{16} & 0 & \frac{1}{16} & \frac{1}{16} & 0 & 0 & 0 & \frac{1}{16} & 0 \\
\frac{1}{16} & \frac{1}{16} & 0 & \frac{1}{16} & \frac{5}{8} & 0 & 0 & \frac{1}{16} & \frac{1}{16} & 0 & 0 & 0 & \frac{1}{16}
\end{pmatrix}.$$

Note that we have not assumed an explicit subdivision rule at the extraordinary vertex v. Instead, we have written this rule so as to weight the vertex v by weight $1 - w[n]$ and the neighbors of v by weights $\frac{w[n]}{n}$.

Deleting the uppermost row and leftmost column of $\overline{S}$ yields a (3×3) block circulant submatrix (C_{ij}), where $0 \le i < 3$ and $0 \le j < 3$. To compute the eigenvalues and eigenvectors of $\overline{S}$, we first compute the eigenvalues and eigenvectors of (C_{ij}) and then extend these vectors to form eigenvectors of $\overline{S}$. The eigenvalues and eigenvectors of (C_{ij}) can be computed by examining the spectral structure of the matrix $(c_{ij}[x])$:

$$\begin{pmatrix} \frac{3}{8} + \frac{1}{8}x + \frac{1}{8}x^{-1} & 0 & 0 \\ \frac{3}{8} + \frac{3}{8}x & \frac{1}{8} & 0 \\ \frac{5}{8} + \frac{1}{16}x + \frac{1}{16}x^{-1} & \frac{1}{16} + \frac{1}{16}x^{-1} & \frac{1}{16} \end{pmatrix}. \tag{8.14}$$

This lower triangular matrix has three eigenvalues that correspond to the entries on its diagonal: $\frac{3}{8} + \frac{1}{8}x + \frac{1}{8}x^{-1}$, $\frac{1}{8}$, and $\frac{1}{16}$. Evaluating these three eigenvalues at $x = \omega_n^h$ for $0 \le h < n$ yields the corresponding $3n$ eigenvalues of (C_{ij}). For example, if $n == 4$, the four eigenvalues of (C_{ij}) associated with $\frac{3}{8} + \frac{1}{8}x + \frac{1}{8}x^{-1}$ are $\frac{5}{8}$, $\frac{3}{8}$, $\frac{1}{8}$, and $\frac{3}{8}$. The remaining eight eigenvalues of (C_{ij}) consist of four copies of $\frac{1}{8}$ and four copies of $\frac{1}{16}$.

Note that the n-fold symmetry of the mesh M used in defining $\overline{S}$ gave rise to the block circulant structure of (C_{ij}). This symmetry also manifests itself during the construction of the eigenvectors of (C_{ij}). In particular, the eigenvectors of the matrix $(c_{ij}[x])$ correspond to the restriction of the eigenvectors of (C_{ij}) to a single sector of the underlying mesh M. The multiplication of the components of these eigenvectors of $(c_{ij}[x])$ by powers of x where $x^n == 1$ corresponds to rotating the components of this eigenvector in the complex plane to form the associated eigenvector of (C_{ij}) defined over all n sectors of M.

The matrix $(c_{ij}[x])$ of equation 8.14 has three eigenvectors of the form $\{1, \frac{3x}{1+x}, \frac{1+13x+x^2}{2+5x+2x^2}\}$, $\{0, x, 1 + x\}$, and $\{0, 0, 1\}$. The corresponding $3n$ eigenvectors of (C_{ij}) are as shown in equation 8.13, with x being evaluated at ω_n^h for $0 \le h < n$. For example, if $n == 4$, the four eigenvectors of (C_{ij}) corresponding to $\{0, x, 1 + x\}$ have the form

$$\{0, 0, 0, 0, x, x^2, x^3, x^4, (1 + x), (1 + x)x, (1 + x)x^2, (1 + x)x^3\},$$

where $x = i^h$ for $0 \leq h < 4$. The remaining eight eigenvectors of (C_{ij}) corresponding to $\left\{1, \frac{3x}{1+x}, \frac{1+13x+x^2}{2+5x+2x^2}\right\}$ and $\{0, 0, 1\}$ are constructed in a similar manner.

Given these eigenvectors for (C_{ij}), we next form potential eigenvectors for the subdivision matrix $\overline{S}$ by prepending an initial zero to these vectors. For the $3n - 3$ eigenvectors (C_{ij}) corresponding to x taken at ω_n^h where $0 < h < n$, these extended vectors have the property that they also annihilate the uppermost row of $\overline{S}$. Therefore, these extended vectors are eigenvectors of $\overline{S}$ whose associated eigenvalues are simply the corresponding eigenvalues of (C_{ij}). Most importantly, the eigenvalue $\frac{3}{8} + \frac{1}{8}x + \frac{1}{8}x^{-1}$ of $(c_{ij}[x])$ gives rise to $n - 1$ eigenvalues of $\overline{S}$ of the form

$$\frac{3}{8} + \frac{1}{8}\omega_n^h + \frac{1}{8}\omega_n^{-h} == \frac{3}{8} + \frac{1}{4}\,\mathrm{Cos}\left[\frac{2\pi h}{n}\right]$$

for $0 < h < n$. Note that these eigenvalues lie in the range $(\frac{1}{8}, \frac{5}{8})$ and reach a maximum at $h = 1, n - 1$. In particular, we let $\lambda[n]$ denote this maximal double eigenvalue of $\overline{S}$; that is,

$$\lambda[n] = \frac{3}{8} + \frac{1}{4}\,\mathrm{Cos}\left[\frac{2\pi}{n}\right].$$

For appropriate choices of $w[n]$, this double eigenvalue $\lambda[n]$ is the subdominant eigenvalue of the subdivision matrix $\overline{S}$. Observe that the associated eigenvectors used in defining the characteristic map are independent of the weight function $w[n]$. (In the next section, we will show that the characteristic map defined by these eigenvectors is regular.)

The remaining four of the $3n + 1$ eigenvectors of $\overline{S}$ have entries that are constant over each block of $\overline{S}$. The four eigenvalues associated with these eigenvectors are eigenvalues of the 4×4 matrix formed by summing any row of each block in the matrix $\overline{S}$; that is,

$$\begin{pmatrix} 1 - w[n] & w[n] & 0 & 0 \\ \frac{3}{8} & \frac{5}{8} & 0 & 0 \\ \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & 0 \\ \frac{1}{16} & \frac{3}{4} & \frac{1}{8} & \frac{1}{16} \end{pmatrix}.$$

(Note that the 3×3 submatrix on the lower right is simply $(c_{ij}[1])$.)

Because the eigenvalues of this matrix are $1, \frac{5}{8} - w[n], \frac{1}{8}, \frac{1}{16}$, the full spectrum of the local subdivision matrix $\overline{S}$ (and its infinite counterpart S) has the form

$1, \lambda[n], \lambda[n], \frac{5}{8} - w[n]$ with all of the remaining eigenvalues having an absolute value less than $\lambda[n]$ (♚). Our goal is to choose the weight function $w[n]$ such that $\lambda[n] > |\frac{5}{8} - w[n]|$. To this end, we substitute the definition of $\lambda[n]$ into this expression and observe that this inequality simplifies to

$$\frac{1 - \mathrm{Cos}\left[\frac{2\pi}{n}\right]}{4} < w[n] < \frac{4 + \mathrm{Cos}\left[\frac{2\pi}{n}\right]}{4}. \tag{8.15}$$

Any choice for $w[n]$ that satisfies this condition yields a subdivision matrix $\overline{S}$ whose spectrum has the form $1 > \lambda[n] == \lambda[n] > |\frac{5}{8} - w[n]|$. Therefore, by Theorem 8.4, the limit functions associated with such a scheme are smooth at extraordinary vertices. For example, linear subdivision plus averaging (see section 7.3.1) causes the weight function $w[n]$ to be the constant $\frac{3}{8}$. This choice for $w[n]$ satisfies equation 8.15 for all $n > 3$, and consequently linear subdivision plus averaging converges to smooth limit functions at extraordinary vertices with valences greater than three. However, for $n == 3$, the subdivision matrix $\overline{S}$ has the spectrum $1, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \ldots$, and, as observed previously, the resulting scheme is continuous, but not smooth, at the extraordinary vertices of this valence. (An alternative choice for $w[n]$ that satisfies equation 8.15 for all valences and reproduces the uniform weight $w[6] == \frac{3}{8}$ is $w[n] = \frac{3}{n+2}$.)

Loop's subdivision rule uses a subtler choice for $w[n]$. In particular, Loop sets the weight function $w[n]$ to have the value $\frac{5}{8} - \lambda[n]^2$. This choice satisfies equation 8.15 for all $n \geq 3$ and yields a subdivision matrix $\overline{S}$ whose spectrum has the form $1, \lambda[n], \lambda[n], \ldots, \lambda[n]^2, \ldots$. This particular choice for $w[n]$ is motivated by an attempt to define a subdivision rule that is $\mathcal{C}^2$ continuous at extraordinary vertices. Unfortunately, the eigenfunction associated with the eigenvalue $\lambda[n]^2$ does not reproduce a quadratic function as required by Theorem 8.5, and therefore cannot be $\mathcal{C}^2$ at the extraordinary vertex v.

However, this associated eigenfunction does have the property that its curvature is bounded (but discontinuous) at v (see Peters and Umlauf [118] for details). For valences $3 \leq n \leq 6$, the spectrum of $\overline{S}$ has the form $1, \lambda[n], \lambda[n], \lambda[n]^2, \ldots$ (i.e., there are no other eigenvalues between $\lambda[n]$ and $\lambda[n]^2$). Therefore, due to Theorem 8.4, Loop's scheme produces limit surfaces with bounded curvature for these valences. This bound on the curvature is particularly important in some types of milling applications. In a similar piece of work, Reif and Schröder [130] show that the curvature of a Loop surface is square integrable, and therefore Loop surfaces can be used in a finite element method for modeling the solution to thin shell equations [22].

An almost identical analysis can be used to derive the spectral structure for Catmull-Clark subdivision (and its quadrilateral variants). The crux of the analysis is deriving an analog to the 3×3 matrix of equation 8.14. For quad meshes, there are $6n$ vertices in the two-ring of the vertex v (excluding v). If we order the indices ij of the six vertices in a single quadrant as 10, 11, 20, 12, 21, and 22, the appropriate 6×6 matrix is

$$\begin{pmatrix} \frac{3}{8} + \frac{1}{16}x + \frac{1}{16}x^{-1} & \frac{1}{16} + \frac{1}{16}x^{-1} & 0 & 0 & 0 & 0 \\ \frac{1}{4} + \frac{1}{4}x & \frac{1}{4} & 0 & 0 & 0 & 0 \\ \frac{9}{16} + \frac{1}{64}x + \frac{1}{64}x^{-1} & \frac{3}{32} + \frac{3}{32}x^{-1} & \frac{3}{32} & \frac{1}{64}x^{-1} & \frac{1}{64} & 0 \\ \frac{1}{16} + \frac{3}{8}x & \frac{3}{8} & \frac{1}{16}x & \frac{1}{16} & 0 & 0 \\ \frac{3}{8} + \frac{1}{16}x & \frac{3}{8} & \frac{1}{16} & 0 & \frac{1}{16} & 0 \\ \frac{3}{32} + \frac{3}{32}x & \frac{9}{16} & \frac{1}{64} + \frac{1}{64}x & \frac{3}{32} & \frac{3}{32} & \frac{1}{64} \end{pmatrix}.$$

Computing the eigenvalues and eigenvectors of this matrix and evaluating them at $x = \omega_n^h$ for $0 \leq h < n$ leads to the desired spectral decomposition of $\overline{S}$. Reif and Peters [116] perform such an analysis in full detail and show that Catmull-Clark subdivision produces C^1 limit functions at extraordinary vertices of all valences.

8.3.3 Proving Regularity of the Characteristic Map

To complete our proof that Loop's scheme produces smooth limit surfaces, we must verify that the characteristic map for this scheme is regular at an extraordinary vertex of arbitrary valence. In this section, we outline a proof of regularity for the characteristic map arising from Loop's scheme. The basic approach is to compute the pair of eigenvectors corresponding to the double eigenvalue $\lambda[n]$ and show that the *Jacobian* of the resulting characteristic map $\psi[h, x, y]$ has the same sign everywhere; that is,

$$\text{Det}\left[\begin{pmatrix} \psi_s^{(1,0)}[h, x, y] & \psi_s^{(0,1)}[h, x, y] \\ \psi_t^{(1,0)}[h, x, y] & \psi_t^{(0,1)}[h, x, y] \end{pmatrix}\right] \neq 0 \tag{8.16}$$

for all $x, y \geq 0$, where $\psi = \{\psi_s, \psi_t\}$. This condition on the Jacobian of ψ prevents ψ from folding back on itself locally and is sufficient to guarantee that the map is regular (see Fleming [63] for more details). In practice, we orient the meshes used in computing this Jacobian such that the resulting determinant is strictly positive for regular characteristic maps.

As observed in the previous section, the range on which this condition must be verified can be reduced significantly via equation 8.8. Because $\lambda[n]$ is real and positive, applying the affine transformation $\left(\begin{smallmatrix}\lambda[n] & 0\\ 0 & \lambda[n]\end{smallmatrix}\right)$ to the map ψ does not affect its regularity. Thus, if we can prove that ψ satisfies equation 8.16 on an annulus surrounding v of sufficient width, the recurrence of equation 8.8 implies that ψ satisfies equation 8.16 everywhere except at v. Because the subdominant eigenvectors used in defining the characteristic map arise from the principal nth root of unity ω_n, this annulus winds around v exactly once, and the corresponding characteristic map ψ is also regular at the origin.

For surface schemes supported on the two-ring, this annulus typically consists of those faces that lie in the two-ring but not in the one-ring of v. The advantage of this choice is that the behavior of the scheme on this annulus is determined entirely by the uniform subdivision rules used in S. For triangular schemes, such as Loop's scheme, this annulus consists of the region $1 \le x+y \le 2$ composed of 3n triangular faces with 3 triangles per wedge (see Figure 8.7). In this section, we sketch a computational proof that the characteristic map ψ for Loop's scheme (and its triangular variants) is regular. This method is very similar to an independently derived method given in Zorin et al. [171]. Variants of either the current method or Zorin's method can be used to prove that the characteristic map for quadrilateral schemes is also regular.

Our first task in this proof is to compute the two eigenvectors used in defining the characteristic map, that is, the eigenvectors associated with the double eigenvalue $\lambda[n]$. In particular, we need the entries of these eigenvectors over the three-ring of the extraordinary vertex v because this ring determines the behavior of the characteristic map on the two-ring of v (i.e., the annulus $1 \le x+y \le 2$). Recall that this double eigenvalue arose from evaluating the eigenvalue $\frac{1+3x+x^2}{8x}$ of the matrix $(c_{ij}[x])$ of equation 8.14 at $x = \omega_n$ and $x = \omega_n^{-1}$. Extending the matrix $(c_{ij}[x])$ to the

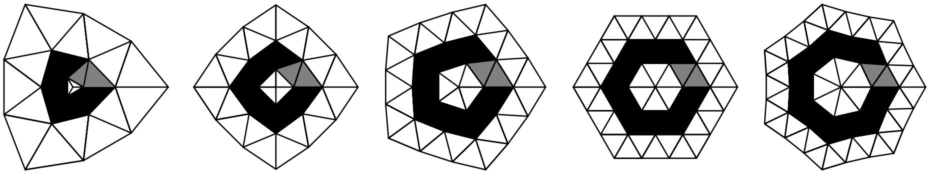


Figure 8.7 The annulus used in verifying regularity of the characteristic map for Loop's scheme.

three-ring of v yields a new matrix of the form

$$\begin{pmatrix} \frac{3}{8}+\frac{1}{8}x+\frac{1}{8}x^{-1} & 0 & 0 & 0 & 0 & 0 \\ \frac{3}{8}+\frac{3}{8}x & \frac{1}{8} & 0 & 0 & 0 & 0 \\ \frac{5}{8}+\frac{1}{16}x+\frac{1}{16}x^{-1} & \frac{1}{16}+\frac{1}{16}x^{-1} & \frac{1}{16} & 0 & 0 & 0 \\ \frac{1}{8}+\frac{3}{8}x & \frac{3}{8} & \frac{1}{8}x & 0 & 0 & 0 \\ \frac{3}{8}+\frac{1}{8}x & \frac{3}{8} & \frac{1}{8} & 0 & 0 & 0 \\ \frac{3}{8} & \frac{1}{8}+\frac{1}{8}x^{-1} & \frac{3}{8} & 0 & 0 & 0 \end{pmatrix}.$$

Now, the eigenvector associated with the eigenvalue $\frac{1+3x+x^2}{8x}$ for this extended matrix is the vector

$$\left\{1, \frac{3x}{1+x}, \frac{1+13x+x^2}{2+5x+2x^2}, \frac{x(2+32x+87x^2+55x^3+7x^4)}{(1+x)(2+11x+19x^2+11x^3+2x^4)}, \right.$$
$$\left. \frac{x(7+55x+87x^2+32x^3+2x^4)}{(1+x)(2+11x+19x^2+11x^3+2x^4)}, \frac{3x(5+23x+5x^2)}{2+11x+19x^2+11x^3+2x^4} \right\}.$$

Evaluating this eigenvector at $x = \omega_n$ yields a complex eigenvector of the form $\overline{z}_1 + i\overline{z}_2$, where $\overline{z}_1$ and $\overline{z}_2$ are the real eigenvectors associated with the double eigenvalue $\lambda[n]$. (Evaluating at $x = \omega_n^{-1}$ yields a complex conjugate eigenvector of the form $\overline{z}_1 - i\overline{z}_2$.) As observed previously, these real eigenvectors $\overline{z}_1$ and $\overline{z}_2$ define the behavior of the characteristic map on a single sector of M. To construct the characteristic map for all n sectors, these eigenvectors can be extended to eigenvectors of $\overline{S}$ by multiplying by powers of x^h for $0 < h < n$. Because $x^n == 1$, the action of this multiplication is to rotate the entries of $\overline{z}_1 + i\overline{z}_2$ in the complex plane and make n copies of a single sector. Figure 8.7 shows a plot of the mesh resulting from this rotation for valences ranging from 3 to 7.

To prove regularity of the corresponding characteristic maps ψ, we must show that ψ satisfies the Jacobian condition of equation 8.16 on the darkened annulus of Figure 8.7 corresponding to $1 \leq x+y \leq 2$ for $0 \leq h < n$. Given the n-fold rotational symmetry of ψ, we can restrict our regularity analysis to the two gray triangles in this annulus of Figure 8.7. The behavior of ψ on these two triangles is determined completely by the 4×4 submesh containing this pair of triangles. Figure 8.8 depicts these submeshes for extraordinary vertices of valences $3 \leq n \leq 7$.

Our final task is to show that the Jacobian condition of equation 8.16 holds over the gray regions in question. Due to the tensor product structure of these 4×4 submeshes, this verification can be done quite easily. Because the uniform subdivision

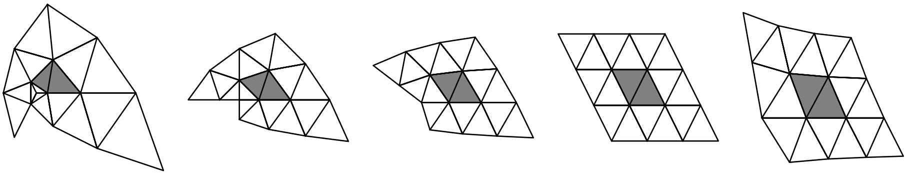


Figure 8.8 4×4 submeshes used in verifying regularity.

rules for our triangular schemes are based on box splines, we can compute the directional derivatives of ψ (i.e. $\psi^{(1,0)}$ and $\psi^{(0,1)}$) and represent these derivatives as box splines of lower order. The coefficients of these derivative schemes are formed by computing directional differences between adjacent control points with respect to the two coordinate directions. Because the basis functions associated with the directional derivative are non-negative, the maximum and minimum values of the four entries in the Jacobian matrix can be bounded using the convex hull property. Representing these bounds as an interval, we can have *Mathematica* compute the determinant of this matrix of intervals.

As a univariate example, consider the problem of proving regularity for the nonuniform curve scheme of our running example. The eigenvector z_1 associated with the subdominant eigenvalue $\lambda_1 == \frac{1}{2}$ had the form $\{\ldots, -3, -2, -1, \frac{1}{3}, 2, 4, 6, \ldots\}$. If $\psi[h, x]$ is the characteristic map for this scheme (i.e., the limit function associated with this eigenvector z_1), we must show that the Jacobian of $\psi[h, x]$, $\psi^{(1)}[h, x]$ has the same sign for all $x \geq 0$ where $h = 0, 1$. By applying equation 8.8, this condition can be reduced to showing that $\psi^{(1)}[h, x]$ has the same sign for $2 \leq x < 4$ where $h = 0, 1$. (Note that we use the annulus $[2, 4]$ in place of $[1, 2]$ because the subdivision rules for the curve scheme are uniform over this larger interval.) The subvectors $\{-5, -4, -3, -2, -1\}$ and $\{2, 4, 6, 8, 10\}$ of z_1 determine the behavior of $\psi[h, x]$ on the intervals $[2, 4]$ where $h = 0, 1$. To bound $\psi^{(1)}[x]$ on these intervals, we observe that the differences of consecutive coefficients in these subvectors are the coefficients of the uniform quadratic B-spline corresponding to $\psi^{(1)}[h, x]$. For $h == 0$, these differences are all -1, and therefore $\psi^{(1)}[0, x] == -1$ for $2 \leq x < 4$. Likewise, for $h == 1$, the maximum and minimum of these differences are 2, and therefore $\psi^{(1)}[1, x] == 2$ for $2 \leq x < 4$. Thus, $\psi[h, x]$ is regular for our curve example.

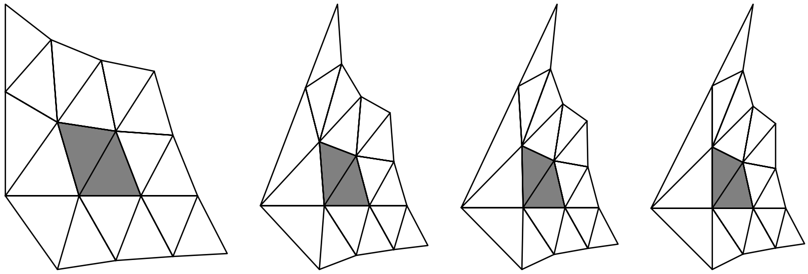
In the bivariate case, we store the 4×4 submesh in a positive orientation and compute an interval that bounds the value of the Jacobian using this method. If this interval is strictly positive, the Jacobian is positive over the entire gray region.

If the interval is strictly negative, the Jacobian test fails. If the interval contains zero, the gray region is recursively subdivided into four subregions using the subdivision rules associated with the schemes. The test is then applied to each subregion, with failure on any subregion implying violation of the Jacobian condition.

The associated implementation contains the *Mathematica* functions that implement this test (♔). The first function, jacobian, returns an interval bound on the value of the Jacobian for a given 4×4 mesh. The second function, regular, recursively applies jacobian using the uniform subdivision rule associated with the scheme. The user can test whether the characteristic map ψ satisfies regular for any valence n.

Ultimately, we would like to conclude that ψ is regular for all valences $n \geq 3$. One solution would be to take the limit of the 4×4 mesh associated with our gray region as $n \to \infty$ and then show that this limiting mesh satisfies the Jacobian condition. Unfortunately, the entries in the second eigenvector z_2 (i.e., the imaginary part of the complex eigenvector z) converge to zero as $n \to \infty$. The effect on the associated 4×4 mesh is to compress this mesh toward the s axis. The solution to this problem is to scale the t component of the characteristic map by a factor of $\frac{1}{\mathrm{Sin}[\frac{2\pi}{n}]}$ (i.e., divide z_2 by $\mathrm{Sin}[\frac{2\pi}{n}]$). This division does not affect the regularity of the map ψ and stretches the mesh in such a way that a unique limit mesh does exist as $n \to \infty$. Figure 8.9 shows this normalized mesh for $n = 8, 16, 32, \infty$.

Now, we can complete our proof of regularity by explicitly testing regularity for valences $3 \leq n \leq 31$. Then, using *Mathematica*, we can compute this normalized 4×4 mesh as a symbolic function of n. Finally, we can verify that all characteristic maps for valence $n \geq 32$ are regular by calling the function regular on this symbolic normalized mesh with $n == \mathrm{Interval}\ [\{32, \infty\}]$. The associated implementation performs these calculations and confirms that ψ is regular for all valences $n \geq 3$ (♔).



♔ **Figure 8.9** Nonuniform scaling to obtain a convergent submesh as $n \to \infty$.

8.4 Future Trends in Subdivision

This book has covered only the basics of subdivision. One of the exciting aspects of subdivision is that it is an area in which research is actively taking place. In this brief conclusion, we outline some of these research areas.

8.4.1 Solving Systems of Physical Equations

Chapters 4, 5, and 6 developed subdivision schemes that converge to the solutions of a range of physical problems. The basic approach in these chapters was to construct subdivision schemes that simulated the behavior of either a multilevel finite difference solver or a multilevel finite element solver. As the authors have shown in [162] and [163], these schemes are essentially a special type of multigrid method in which the prediction operator has been customized to the particular physical problem. For graphical applications, this predictor is accurate enough that the smoothing and coarse grid correction steps of traditional multigrid can be skipped. In this vein, one area that is ripe for further research is the connection between subdivision and multigrid. For example, techniques from traditional multigrid might lead to new, interesting types of subdivision schemes. Conversely, techniques from subdivision might lead to advances in multigrid solvers.

Another interesting possibility is the construction of finite element schemes whose associated bases are defined solely through subdivision. Chapter 6 includes two nonuniform examples of this approach: one for natural cubic splines and one for bounded harmonic splines. Although both of these examples are functional, a more interesting possibility is applying this idea to shapes defined as parametric meshes. Following this approach, the solution to the associated physical problem would be represented as an extra coordinate attached to the mesh. The advantage of this method is that it unifies the geometric representation of a shape and its finite element representation.

8.4.2 Adaptive Subdivision Schemes

Another area of subdivision that is ripe with promise is that of adaptive subdivision. For univariate spline schemes, the most powerful type of adaptive subdivision involves *knot insertion*. Given a set of knots M_0 and a vector of control points p_0, such spline schemes typically construct a function $p[x]$ that approximates the entries of p_0 with breakpoints at the knots of M_0. Given a new knot set M_1 that contains M_0, knot insertion schemes compute a new vector of control points p_1 that represent the same function $p[x]$. As in the case of subdivision, the vectors p_0 and p_1 are related via the matrix relation $p_1 = S_0 p_0$, where S_0 is a knot insertion matrix that depends on the knot sets M_0 and M_1. Notice that in this framework subdivision is a special instance of knot insertion in which the number of knots is doubled during each insertion step.

The knot insertion scheme for univariate B-splines is known as the Oslo algorithm [24]. Whereas variants of the Oslo algorithm are known for a range of univariate splines (e.g., exponential B-splines [89]), instances of knot insertion schemes for bivariate splines are much more difficult to construct. One fundamental difficulty is determining the correct analog of a knot in the bivariate case. Should the analog be a partition of the plane into polygonal pieces or a set of finite points? The B-patch approach of Dahmen et al. [35] uses the latter approach. Given a set of knots, it constructs a locally defined piecewise polynomial patch with vertices at the given knots. Unfortunately, it has proven difficult to generalize this method to spline surfaces consisting of multiple patches. In general, the problem of building spline surfaces with fully general knot insertion methods remains open.

Another approach to adaptive subdivision is to construct schemes that allow for smooth transitions between uniform meshes of different levels. For example, both the 4-8 scheme of Velho et al. [153] and the $\sqrt{3}$ scheme of Kobbelt [85] allow the user to subdivide only locally specified portions of a uniform mesh. The advantage of this approach is that the resulting meshes need be adapted only in areas of interest. The drawback of these approaches is that they abandon the piecewise functional representation that makes analyzing B-splines easier. For example, constructing some type of piecewise analytic representation for the scaling functions underlying adaptive $\sqrt{3}$ subdivision is very difficult. Ideally, a bivariate scheme for adaptive subdivision would capture all of the nice aspect of knot insertion for univariate B-splines: underlying function spaces, fully irregular knot geometry, local support, maximal smoothness, and so on.

8.4.3 Multiresolution Schemes

Another area that is intimately related to subdivision is multiresolution analysis. Given a fine mesh $\{M_k, p_k\}$, a multiresolution scheme typically computes a coarse mesh $\{M_{k-1}, p_{k-1}\}$ and a "detail" mesh $\{M_k, q_{k-1}\}$ via matrix equations

$$\begin{aligned} p_{k-1} &= A_{k-1}\, p_k, \\ q_{k-1} &= B_{k-1}\, p_k. \end{aligned}$$

The matrices A_{k-1} and B_{k-1}, known as *analysis filters*, typically depend on the topology of the fine mesh M_k. If these filters are chosen appropriately, the coarse mesh $\{M_{k-1}, p_{k-1}\}$ forms a "good" approximation to the fine mesh $\{M_k, p_k\}$. The detail mesh $\{M_k, q_{k-1}\}$ encodes the differences between the meshes $\{M_{k-1}, p_{k-1}\}$ and $\{M_k, p_k\}$. The process can be repeated k times to yield a base mesh $\{M_0, p_0\}$ plus k detail meshes $\{M_{j+1}, q_j\}$ where $0 \leq j < k$. The analysis process can be reversed via an inverse synthesis process. Expressed in matrix form, this relation is

$$p_k = S_{k-1} p_{k-1} + T_{k-1} q_{k-1}.$$

The matrices S_{k-1} and T_{k-1}, known as *synthesis filters*, typically depend on the topology of the coarse mesh M_{k-1}. The subdivision matrix S_{k-1} maps the vector p_{k-1} into the vector p_k, whereas the matrix T_{k-1} expands the detail coefficients onto the level k mesh. In the uniform case, there has been a massive amount of work on the relationship between subdivision and multiresolution analysis. Stollnitz et al. [148] and Strang et al. [150] give nice introductions to this topic.

One important limitation of most work on multiresolution analysis is that it assumes a functional domain. Subdivision schemes based on polyhedral meshes avoid this functional limitation. The existence of purely matrix schemes leads to the following problem: given a subdivision matrix S_{k-1}, choose the remaining synthesis matrix T_{k-1} such that the analysis matrices A_{k-1} and B_{k-1} are sparse. The sparsity restriction allows the filters to be applied efficiently even for large meshes. Both Lounsbery et al. [100] and Schröder et al. [136] consider this problem for the case of interpolatory schemes, whereas Dahmen and Micchelli [34] and Warren [159] consider the more general case of approximating schemes. Note that any filters that solve this problem can be generalized to a range of filters using the lifting scheme of Schröder et al. [136].

The drawback of these previous schemes is that they assume full subdivision of the meshes involved. A more elegant approach would be to combine adaptive subdivision with multiresolution analysis. Guskov et al. [69] propose a first step in this direction: a multiresolution scheme that incorporates adaptive subdivision based on minimizing a variational functional. Again, more work in this area is needed.

8.4.4 Methods for Traditional Modeling Operations

Currently, subdivision surfaces are very popular in computer graphics applications due to their ability to construct interesting shapes with a minimal amount of effort. On the other hand, NURBS (nonuniform rational B-splines) remain dominant in the traditional areas of geometric modeling, such as computer-aided design. In this area, computations such as Boolean set operations, surface/surface intersection, trimming, offset, and blending are very important. To facilitate the use of subdivision surfaces in this area, algorithms for computing these operations need to be developed.

One preliminary piece of work on the problem of surface/surface intersection is that of Litke et al. [98]. This paper describes a method based on the "combined" subdivision scheme of Levin [94] and exactly reproduces the intersection curve of two subdivision surfaces. Away from the curve, the original subdivision surfaces are approximated using a multiresolution scheme based on quasi-interpolation. More generally, developing methods for approximating existing geometry using subdivision remains a high priority.

8.4.5 $\mathcal{C}^2$ Subdivision Schemes for Polyhedral Meshes

Probably the biggest problem in subdivision is that of constructing a surface scheme whose limit surfaces are $\mathcal{C}^2$ at extraordinary vertices. Catmull-Clark and Loop subdivision yield limit surfaces that are $\mathcal{C}^2$ everywhere except at extraordinary vertices. At these vertices, these schemes deliver surfaces that are only $\mathcal{C}^1$. Although Prautzsch [122] and Reif [129] have proposed methods that are capable of producing limit surfaces that are $\mathcal{C}^2$ everywhere, the uniform subdivision rules associated with these schemes converge to piecewise polynomial surfaces with a high degree (i.e., quintic in the $\mathcal{C}^2$ case). Moreover, the resulting limit surfaces are relatively inflexible at the extraordinary vertex v because they generate a single degenerate polynomial patch in the neighborhood of v. (For Catmull-Clark and Loop subdivision, n distinct surface patches meet at an extraordinary vertex of valence n.)

Ideally, we desire a $\mathcal{C}^2$ scheme that has a flexibility similar to that of Catmull-Clark or Loop subdivision with subdivision rules of a similar support. Based on the negative results of Prautzsch and Reif [124], this generalization is most likely not a stationary scheme. One promising line of attack for this problem lies in constructing nonstationary schemes that are $\mathcal{C}^2$ at extraordinary vertices. For example, the nonstationary scheme of section 7.2.3 converges to surfaces of revolution that are $\mathcal{C}^2$ at their poles. Note that this approach is not a complete solution, in that the resulting surfaces of revolution are always spherical (or elliptical) at their poles. As usual, more work remains to be done in this area.

References

[1] Abhyankar, S. and Bajaj, C.: *Automatic parameterization of rational curves and surfaces {III}: algebraic plane curves.* Computer-Aided Geometric Design 5, pp. 309–321, 1988.

[2] Abi-Ezzi, S.: *The graphical processing of B-splines in a highly dynamic environment*. Ph.D. thesis, Rensselaer Design Research Center, Rensselaer Polytechnic Institute, 1989.

[3] Alfeld, P.: *Scattered data interpolation in three or more variables.* In Lyche, T. and Schumaker, L. (eds.): *Mathematical methods in computer-aided geometric design*, pp. 1–33, New York: Academic Press, 1989.

[4] Alfhors, L.: *Complex analysis, second edition*. New York: McGraw-Hill, 1966.

[5] Bajaj, C. and Warren, J.: *A smooth subdivision scheme for hexahedral meshes.* To appear in *The Visual Computer*, 2001.

[6] Ball, A. and Storry, D.: *Conditions for tangent plane continuity over recursively generated B-spline surfaces.* ACM Transactions on Graphics 7(2), pp. 83–102, 1988.

[7] Ball, A. and Storry, D.: *An investigation of curvature variation over recursively generated B-spline surfaces.* ACM Transactions on Graphics 9(4), pp. 424–437, 1990.

[8] Barnsley, M.: *Fractals everywhere.* Academic Press, 1988.

[9] Bartels, R., Beatty, J. and Barsky, B.: *An introduction to splines for use in computer graphics and geometric modeling.* San Francisco: Morgan Kaufmann, 1987.

[10] Biermann, H., Levin, A. and Zorin, D.: *Piecewise smooth subdivision surfaces with normal control.* Proceedings of SIGGRAPH 2000, Annual Conference Series, pp. 113–120. New York: Addison Wesley Longman, 2000.

[11] Böhm, W.: *On the efficiency of knot insertion algorithms.* Computer-Aided Geometric Design 2(1–3), pp. 141–144, 1985.

[12] Bracewell, R.: *The Fourier transform and its applications, third edition*. New York: McGraw-Hill, 1999.

[13] Brandt, A.: *Multi-level adaptive solutions to boundary value problems.* Mathematics of Computations 31, pp. 333–390, 1977.

[14] Brezzi, F. and Fortin, M.: *Mixed and hybrid finite element methods.* Berlin: Springer-Verlag, 1991.

[15] Briggs, W.: *A multigrid tutorial*. SIAM, 1987.

[16] Catmull, E. and Clark, J.: *Recursively generated B-spline surfaces on arbitrary topological meshes*. Computer Aided Design 16(6), pp. 350–355, 1978.

[17] Cavaretta, A., Dahmen, W. and Micchelli, C.: *Stationary subdivision*. Memoirs of the AMS 453, AMS, 1991.

[18] Chaikin, G.: *An algorithm for high-speed curve generation*. Computer Graphics and Image Processing 3, pp. 346–349, 1974.

[19] Chen, J. and Lobo, N.: *Toward interactive-rate simulation of fluid with moving obstacles using Navier-Stokes equations*. Graphical Models and Image Processing 57(2), pp. 107–116, 1995.

[20] Chiba, N. et al.: *Visual simulation of water currents using a particle-based behavioral model*. Journal of Visualization and Computer Animation 6, pp. 155–171, 1995.

[21] Chopard, B. and Droz, M.: C*ellular automata modeling of physical systems*. New York: Cambridge University Press, 1998.

[22] Cirak, F., Ortiz, M. and Schröder, P.: *Integrated modeling, finite-element analysis, and design for thin-shell structures using subdivision*. To appear in Computer Aided Design, 2001.

[23] Cline, A. K.: *Scalar and planar-valued curve fitting using splines under tension*. Communications of the ACM 17, pp. 218–220, 1974.

[24] Cohen, E., Lyche, T. and Riesenfeld, R.: *Discrete B-splines and subdivision techniques in computer-aided geometric design and computer graphics*. Computer Graphics and Image Processing 14, pp. 87–111, 1980.

[25] Cohen, E., Lyche, T. and Riesenfeld, R.: *Discrete box splines and refinement algorithms*. Computer-Aided Geometric Design 1, pp. 131–148, 1984.

[26] Cohen, E., Lyche, T. and Riesenfeld, R.: C*ones and recurrence relations for simplex splines*. Constructive Approximation 3, pp. 131–141, 1987.

[27] Cormen, T., Leiserson, C. and Rivest, R.: *Introduction to algorithms*. New York: McGraw-Hill, 1990.

[28] Curry, H. and Schöenberg, I.: O*n Polya frequency functions IV: the fundamental spline functions and their limits*. Journal de Analyse Math. 17, pp. 71–107, 1966.

[29] Dahmen, W.: *Multivariate B-splines: recurrence relations and linear combinations of truncated powers*. In Schempp, W. and Zeller, K. (eds.): *Multivariate Approximation Theory*, Basel: Birkhaser, pp. 64–82, 1979.

[30] Dahmen, W.: O*n multivariate B-splines*. SIAM Journal of Numerical Analysis 17, pp. 179–191, 1980.

[31] Dahmen, W. and Micchelli, C.: C*omputation of inner products on multivariate B-splines*. Numerical and Functional Analysis and Optimization 3, pp. 357–375, 1981.

[32] Dahmen, W. and Micchelli, C.: *Recent progress in multivariate splines*. In Chui, C., Schumaker, L., and Ward, J. (eds.): *Approximation theory IV*, New York: Academic Press, pp. 27–121, 1983.

[33] Dahmen, W. and Micchelli, C.: *Subdivision algorithms for the generation of box spline surfaces.* Computer-Aided Geometric Design 1(2), pp. 115–129, 1984.

[34] Dahmen, W. and Micchelli, C.: *Banded matrices with banded inverses II: locally finite decomposition of spline spaces.* Constructive Approximation 9, pp. 263–281, 1993.

[35] Dahmen, W., Micchelli, C. and Seidel, H.: *Blossoming begets B-splines built better by B-patches.* Mathematics of Computation 59, pp. 265–287, 1993.

[36] Davis, P.: *Circulant matrices, second edition.* New York: Chelsea Publications, 1994.

[37] De Boor, C.: *On calculating with B-splines.* Journal of Approximation Theory 6, pp. 50–62, 1972.

[38] De Boor, C.: *A practical guide to splines.* New York: Springer-Verlag, 1978.

[39] De Boor, C. and Höllig, K.: *B-splines from parallelepipeds.* Journal d'Analyse Math. 42, pp. 99–115, 1983.

[40] De Boor, C., Höllig, K. and Riemenschneider, S.: *Box splines.* New York: Springer-Verlag, 1993.

[41] De Boor, C. and Lynch, R.: *On splines and their minimum properties.* Journal of Mathematics and Mechanics 15, pp. 953–968, 1966.

[42] DeRose, T., Kass, M. and Truong, T.: *Subdivision surfaces in character animation.* Proceedings of SIGGRAPH 98, Annual conference series, New York: ACM Press, pp. 85–94, 1998.

[43] Deslauriers, G. and Dubuc, S.: *Symmetric iterative interpolation processes.* Constructive Approximation 5, pp. 49–68, 1989.

[44] DoCarmo, M.: *Differential geometry of curves and surfaces.* New York: Prentice-Hall, 1976.

[45] Doo, D. and Sabin, M.: *Behavior of recursive division surfaces near extraordinary points.* Computer Aided Design 10(6), pp. 356–360, 1978.

[46] Duchon, J.: *Splines minimizing rotation invariant semi-norms in Sobolev spaces.* In Keller, M. (ed.): *Constructive theory of functions of several variables*, pp. 85–100, Berlin: Springer-Verlag, 1977.

[47] Dyn, N.: *Interpolation of scattered data by radial functions.* In Chui, C., Schumaker, L. and Utreras, F.: *Topics in multivariate approximation*, New York: Academic Press, pp. 47–61, 1987.

[48] Dyn, N.: *Interpolation and approximation by radial and related functions.* In Chui, C., Schumaker, L. and Ward, J. (eds.): *Approximation theory VI, volume 1*, New York: Academic Press, pp. 211–234, 1989.

[49] Dyn, N.: *Subdivision schemes in computer-aided geometric design.* In Light, W. (ed.): *Advances in numerical analysis II*, New York: Oxford University Press, pp. 36–104, 1992.

[50] Dyn, N., Gregory, J. and Levin, D.: *A four-point interpolatory subdivision scheme for curve design.* Computer-Aided Geometric Design 4, pp. 257–268, 1987.

[51] Dyn, N., Gregory, J. and Levin, D.: *Analysis of uniform binary subdivision schemes for curve design.* Constructive Approximation 7, pp. 127–147, 1991.

[52] Dyn, N., Hed, S. and Levin, D.: *Subdivision schemes for surface interpolation.* In Conte, A. et al. (eds.): *Workshop on computational geometry*, pp. 97–118, World Scientific Publishing, 1993.

[53] Dyn, N. and Levin, D.: *The subdivision experience.* In Laurent, P., LeMéhauté, A. and Schumaker, L. (eds.): *Curves and surfaces II*, Wellesley, MA: A. K. Peters, 1991.

[54] Dyn, N. and Levin, D.: *Analysis of asymptotically equivalent binary subdivision schemes.* Journal of Mathematical Analysis and Applications 193, pp. 594–621, 1995.

[55] Dyn, N., Levin, D. and Gregory, J.: *A butterfly subdivision scheme for surface interpolation with tension control.* ACM Transactions on Graphics 9(2), pp. 160–169, 1990.

[56] Dyn, N., Levin, D. and Rippa, S.: *Numerical procedures for surface fitting of scattered data by radial functions.* SIAM Journal of Statistical Computing 7, pp. 639–659, 1986.

[57] Dyn, N. and Ron, A.: *Multiresolution analysis by infinitely differentiable compactly supported functions.* CMS TSR #93-4, Madison, WI: University of Wisconsin Press, 1992.

[58] Evans, L.: *Partial differential equations.* New York: American Mathematical Society, 1998.

[59] Farin, G.: *Curves and surfaces for CAGD: a practical guide, third edition.* New York: Academic Press, 1992.

[60] Farin, G.: *NURBS: From projective geometry to practical use, second edition.* Wellesley, MA: A. K. Peters, 1999.

[61] Fields, J.: *Theory of the algebraic functions of a complex variable.* Berlin: Mayer & Müller, 1906.

[62] Firby, P. and Gardiner, C.: *Surface topology.* New York: John Wiley, 1982.

[63] Flemming, W.: *Functions of several variables.* New York: Springer-Verlag, 1977.

[64] Foster, N. and Metaxas, D.: *Realistic animation of liquids.* Graphical Models and Image Processing 58(5), pp. 471–483, 1996.

[65] Foster, N. and Metaxas, D.: *Modeling the motion of a hot, turbulent gas*. Proceedings of SIGGRAPH 97. Annual Conference Series, pp. 181–188, New York: Addison Wesley, 1997.

[66] Franke, R.: *Scattered data interpolation: test of some methods.* Mathematics of Computation 38, pp. 181–200, 1982.

[67] Golub, G. and Van Loan, C.: *Matrix computations, third edition.* Baltimore: Johns Hopkins University Press, 1996.

[68] Goodman, T.: *Polyhedral splines*. In Dahmen, W., Gasca, M. and Micchelli, C. (eds.): *Computation of curves and surfaces*, pp. 347–382, 1990.

[69] Guskov, I., Schröder, P. and Sweldens, W.: *Multiresolution signal processing for meshes.* Proceedings of SIGGRAPH 99, Annual Conference Series, pp. 325–334. New York: Addison Wesley Longman, 1999.

[70] Habib, A. and Warren, J.: *Edge and vertex insertion for a class of C^1 subdivision surfaces.* Computer-Aided Geometric Design 16(4), pp. 223–247, 1999.

[71] Halstead, M., Kass, M. and DeRose, T.: *Efficient, fair interpolation using Catmull-Clark surfaces.* Proceedings of SIGGRAPH 93, Annual Conference Series. New York: ACM Press, pp. 35–44, 1993.

[72] Harder, R. and Desmarias, R.: *Interpolation using surface splines.* Journal of Aircraft 9, pp. 189–197, 1972.

[73] Holt, F.: *Towards a curvature-continuous stationary subdivision algorithm.* Z. Angew. Math. Mech. 76, pp. 423–424, 1996.

[74] Hoppe, H., DeRose, T., Duchamp, T., Halstead, M., Jin, H., McDonald, J., Schweitzer, J. and Stuetzle, W.: *Piecewise smooth surface reconstruction.* Proceedings of SIGGRAPH 94, Annual Conference Series. New York: ACM Press, pp. 295–302, 1994.

[75] Horn, R. and Johnson, C.: *Topics in matrix analysis.* New York: Cambridge University Press, 1991.

[76] Hoschek, J. and Lasser, D.: *Fundamentals of computer-aided geometric design.* Boston: A. K. Peters, 1993.

[77] Jesperson, D.: *Multigrid methods for partial differential equations.* Studies in Numerical Analysis 24, 1984.

[78] Kass, M. and Miller, G.: *Rapid, stable fluid dynamics for computer graphics.* Proceedings of SIGGRAPH 89, Annual Conference Series, pp. 49–57, New York: Addison Wesley, 1989.

[79] Kelly, C. T.: *Iterative methods for linear and nonlinear systems.* Philadelphia: Society of Industrial and Applied Mathematics, 1995.

[80] Kobbelt, L.: *Interpolatory refinement by variational methods.* In Chui, C. and Schumaker, L. (eds.): *Approximation theory VIII, volume two: wavelets and multilevel approximation,* World Scientific Publishing, pp. 217–224, 1995.

[81] Kobbelt, L.: *Interpolating subdivision on open quadrilateral nets with arbitrary topology.* Computer Graphics Forum 15, pp. 409–420, 1996.

[82] Kobbelt, L.: *A variational approach to subdivision.* Computer-Aided Geometric Design 13(8), pp. 743–761, 1996.

[83] Kobbelt, L.: *Fairing by finite difference methods.* In Daehlen, M., Lyche, T. and Schumaker, L. (eds.): *Mathematical methods for curves and surfaces II,* Nashville: Vanderbilt University Press, 1998.

[84] Kobbelt, L.: *Discrete fairing and variational subdivision for freeform surface design.* The Visual Computer 16(3-4), pp. 142–150, 2000.

[85] Kobbelt, L.: *$\sqrt{3}$-subdivision.* Proceedings of SIGGRAPH 2000, Annual Conference Series, pp. 103–112, New York: Addison Wesley Longman, 2000.

[86] Kobbelt, L., Campagna, S., Vorsatz, J. and Seidel, H.: *Interactive multi-resolution modeling on arbitrary meshes.* Proceedings of SIGGRAPH 98, Annual Conference Series, pp. 105–114. New York: Addison Wesley, 1998.

[87] Kobbelt, L. and Schröder, P.: *A multiresolution framework for variational subdivision*. ACM Transactions on Graphics 17(4), pp. 209–237, 1998.

[88] Koch, P. and Lyche, T.: *Construction of exponential tension B-splines of arbitrary order.* In Laurent, P., LeMéhauté, A. and Schumaker, L. (eds.): *Curves and surfaces*, New York: Academic Press, 1991.

[89] Koch, P., Lyche, T., Neamtu, M. and Schumaker, L.: *Control curves and knot insertion for trigonometric splines.* Advances in Computational Mathematics 4, pp. 405–424, 1995.

[90] Kunz, E.: *Introduction to commutative algebra and algebraic geometry.* Boston: Birkhäuser, 1985.

[91] Lakatos, I.: *Proofs and refutations: the logic of mathematical discovery*. New York: Cambridge University Press, 1976.

[92] Lancaster, P. and Tismenetsky, M.: *Theory of matrices.* New York: Academic Press, 1985.

[93] Lane, J. and Riesenfeld, R.: *A theoretical development for the computer generation and display of piecewise polynomial functions.* Transactions on Pattern Analysis and Machine Intelligence 2(1), pp. 35–46, 1980.

[94] Levin, A.: *Combined subdivision schemes for the design of surfaces satisfying boundary conditions.* Computer-Aided Geometric Design 16(5), pp. 345–354, 1999.

[95] Levin, A.: *Interpolating nets of curves by smooth subdivision surfaces.* Computer Proceedings of SIGGRAPH 1999, Annual Conference Series, pp. 57–64, New York: Addison Wesley, 1999.

[96] Levy, H. and Lessman, F.: *Finite difference equations.* Dover Publications, 1992.

[97] Liggett, J.: *Fluid mechanics*. New York: McGraw-Hill, 1994.

[98] Litke, N., Levin, A. and Schröder, P.: *Trimming for subdivision surfaces.* Computer-Aided Geometric Design 18, pp. 463–481, 2001.

[99] Loop, C. T.: *Smooth subdivision surfaces based on triangles.* Master's thesis, Department of Mathematics, University of Utah, August 1987.

[100] Lounsbery, M., DeRose, T. and Warren, J.: *Multiresolution analysis for surfaces of arbitrary topological type.* ACM Transactions on Graphics 16(1), pp. 34–73, 1997.

[101] MacCracken, R. and Joy, K.: *Free-form deformations of solid primitives with constraints.* Proceedings of SIGGRAPH 1996, Annual Conference Series, pp. 181–188, New York: Addison Wesley Longman, 1996.

[102] Maz`ia, V. G.: *Sobolev spaces.* New York: Springer-Verlag, 1985.

[103] Micchelli, C.: *On a numerically efficient method for computing multivariate B-splines.* In Schempp, W. and Zeller, K. (eds.): *Multivariate approximation theory,* Basel: Birkhäuser, pp. 211–248, 1979.

[104] Micchelli, C. and Prautzsch, H.: *Computing curves invariant under halving.* Computer-Aided Geometric Design 4, pp. 113–140, 1987.

[105] Micchelli, C. and Prautzsch, H.: *Uniform refinement of curves.* Linear Algebra and Its Applications 114/115, pp. 841–870, 1989.

[106] Miller, G. and Pearce, A.: *Globular dynamics: a connected particle system for animating viscous fluids.* Computers and Graphics 13(3), pp. 305–309, 1989.

[107] Morin, G., Warren, J. and Weimer, H.: *A subdivision scheme for surfaces of revolution.* Computer-Aided Geometric Design 18, pp. 483–502, 2001.

[108] Mortenson, M.: *The computer graphics handbook: geometry and mathematics.* New York: Industrial Press Inc., 1990.

[109] Nasri, A.: *Surface interpolation on irregular networks with normal conditions.* Computer-Aided Geometric Design 8, pp. 89–96, 1991.

[110] Nasri, A.: *Curve interpolation in recursively generated B-spline surfaces over arbitrary topology.* Computer-Aided Geometric Design 14(1), pp. 13–30, 1997.

[111] Nasri, A.: *A 4-sided approach to curve interpolation by recursive subdivision surfaces.* The Visual Computer 14(7), pp. 343–353, 1998.

[112] Nasri, A.: *Interpolating meshes of boundary intersecting curves by subdivision surfaces.* The Visual Computer 16(1), pp. 3–14, 2000.

[113] Oden, J. and Reddy, J.: *An introduction to the mathematical theory of finite elements.* New York: John Wiley, 1976.

[114] Peters, J. and Nasri, A.: *Computing volumes of solids enclosed by recursive subdivision surfaces.* Computer Graphics Forum 16(3), pp. 89–94, 1997.

[115] Peters, J. and Reif, U.: *The simplest subdivision scheme for smoothing polyhedra.* ACM Transactions on Graphics 16(4), pp. 420–431, 1997.

[116] Peters, J. and Reif, U.: *Analysis of generalized B-spline subdivision algorithms.* SIAM Journal of Numerical Analysis 35, pp. 728–748, 1998.

[117] Peters, J. and Umlauf, G.: *Gaussian and mean curvature of subdivision surfaces.* In Cipolla, R. and Martin, R. (eds.): *The mathematics of surfaces IX*, IMA, pp. 59–69, 2000.

[118] Peters, J. and Umlauf, G.: *Computing curvature bounds for bounded curvature subdivision.* Computer-Aided Geometric Design 18, pp. 455–461, 2001.

[119] Pottmann, H. and Wagner, M.: *Helix splines as an example of affine Tchebycheffian splines.* Advances in Computational Mathematics 2, pp. 123–142, 1994.

[120] Powell, M.: *The theory of radial basis function approximation.* In Light, W. A., *Advances in numerical analysis II: wavelets, subdivision algorithms and radial functions*, Oxford: Clarendon Press, 1992.

[121] Powers, D.: *Boundary value problems.* New York: Harcourt Brace Jovanovich, 1987.

[122] Prautzsch, H.: *Freeform splines.* Computer-Aided Geometric Design 14, pp. 201–206, 1997.

[123] Prautzsch, H.: *Smoothness of subdivision surfaces at extraordinary points.* Advances in Computational Mathematics 9, pp. 377–389, 1998.

[124] Prautzsch, H. and Reif, U.: *Degree estimates of C^k piecewise polynomial subdivision surfaces.* Advances in Computational Mathematics 10, pp. 209–217, 1999.

[125] Prautzsch, H. and Umlauf, G.: *A $\mathcal{G}^2$ subdivision algorithm*. In Farin, G., Bieri, H., Brunnet, G. and DeRose, T. (eds.): *Geometric modeling*, Computing Supplements 13, pp. 217–224, New York: Springer-Verlag, 1998.

[126] Ramshaw, L.: Ph.D. thesis, Department of Computer Science, Stanford University, 1987.

[127] Ramshaw, L.: *Blossoms are polar forms.* Computer-Aided Geometric Design 6, pp. 323–358, 1989.

[128] Reif, U.: *A unified approach to subdivision algorithms near extraordinary points.* Computer-Aided Geometric Design 12, pp. 153–174, 1995.

[129] Reif, U.: *TURBS: topologically unrestricted B-splines.* Constructive Approximation 4, pp. 55–77, 1998.

[130] Reif, U. and Schröder, P. *Curvature integrability of subdivision surfaces.* To appear in Advances in Computational Mathematics, 2001.

[131] Requicha, A. and Voelker, H.: *Solid modeling: a historical summary and contemporary assessment*. IEEE Computer Graphics and Applications, pp. 9–24, 1982.

[132] Rockwood, A. and Chambers, P.: *Interactive curves and surfaces: a multimedia tutorial.* San Francisco: Morgan Kaufmann, 1996.

[133] Rothman, D. and Zaleski, S.: *Lattice-gas cellular automata*. New York: Cambridge University Press, 1997.

[134] Royden, H.: *Real analysis, third edition*. New York: Macmillan; London: Collier Macmillan, 1988.

[135] Sabin, M.: *Cubic recursive division with bounded curvature.* In Laurent, P., LeMehaute, A. and Schumaker, L. (eds.): *Curves and surfaces,* New York: Academic Press, pp. 411–414, 1991.

[136] Schröder, P. and Sweldens, W.: *Spherical wavelets: efficiently representing functions on the sphere.* Proceedings of SIGGRAPH 95, Annual Conference Series, pp. 161–172, New York: Addison Wesley, 1995.

[137] Schumaker, L.: *Spline functions: basic theory.* New York: John Wiley, 1981.

[138] Schweikert, D.: *An interpolation curve using a spline in tension*. J. Math. A. Physics 45, pp. 312–317, 1966.

[139] Sederberg, T., Zheng, J., Sewell, D. and Sabin, M.: *Non-uniform recursive subdivision surfaces.* Proceedings of SIGGRAPH 98, Annual Conference Series, pp. 387–394, New York: Addison Wesley, 1998.

[140] Seidel, H.: *An introduction to polar forms.* IEEE Computer Graphics and Applications 13, 1993.

[141] Singer, I. and Thorpe, J.: *Lecture notes on elementary topology and geometry.* New York: Springer-Verlag, 1996.

[142] Sommerfeld, A.: *Eine besondere auschauliche Ableitung des Gaussischen Fehlergesetzes.* Berlin: Festschrift Ludwig Boltzmann, pp. 848–859, 1904.

[143] Spanier, J. and Oldham, K.: *An atlas of functions.* Washington, DC: Hemisphere Publications, 1987.

[144] Spiegel, M.: *Schaum's outline of theory and problems of calculus of finite differences and difference equations.* New York: McGraw-Hill, 1971.

[145] Stam, J.: *Exact evaluation of Catmull-Clark subdivision surfaces at arbitrary parameter values*. Proceedings of SIGGRAPH 98, Annual Conference Series, pp. 395–404, New York: Addison Wesley, 1998.

[146] Stam, J.: *On subdivision schemes generalizing B-spline surfaces of arbitrary degree*. Computer-Aided Geometric Design 18, pp. 397–427, 2001.

[147] Stam, J. and Fiume, E.: *Turbulent wind fields for gaseous phenomena*. Proceedings of SIGGRAPH 93. Annual Conference Series, pp. 369–376, New York: Addison Wesley, 1993.

[148] Stollnitz, E., DeRose, T. and Salesin, D.: *Wavelets for computer graphics*. San Francisco: Morgan Kaufmann, 1996.

[149] Strang, G.: *Linear algebra and its applications*. New York: Academic Press, 1980.

[150] Strang, G. and Nguyen, T.: *Wavelets and filter banks*. Wellesley, MA: Wellesley-Cambridge Press, 1997.

[151] Taylor, A.: *Advanced calculus*. Waltham, MA: Ginn and Company, 1955.

[152] Varga, R.: *Matrix iterative analysis, second edition*. New York: Springer-Verlag, 2000.

[153] Velho, L. and Zorin, D.: *4-8 subdivision*. Computer-Aided Geometric Design 18, pp. 397–427, 2001.

[154] Wagner, M. and Pottmann, H.: *Symmetric Tchebycheffian B-spline schemes*. In Laurent, P., Le Mehaúte, A. and Schumaker, L. (eds.): *Curves and Surfaces in Geometric Design*, Wellesley, MA: A. K. Peters, pp. 483–490, 1994.

[155] Wahba, G. and Wendelberger, J.: *Some new mathematical methods for variational objective analysis using splines and cross validation*. Monthly Weather Review 108, pp. 1122–1143, 1980.

[156] Warren, J.: *On algebraic surfaces meeting with geometric continuity*. Ph.D. thesis, Department of Computer Science, Cornell University, 1986.

[157] Warren, J.: *Binary subdivision schemes for functions over irregular knot sequences*. In Daehlen, M., Lyche, T. and Schumaker, L. (eds.): *Mathematical methods in CAGD III*, New York: Academic Press, 1995.

[158] Warren, J.: *Subdivision methods for geometric design*. Manuscript available online at *http://www.cs.rice.edu/~jwarren/*, 1995.

[159] Warren, J.: *Sparse filter banks for binary subdivision schemes*. In Goodman, T. (ed.): *Mathematics of surface VII*, Dundee: IMA, 1996.

[160] Warren, J. and Weimer, H.: *Variational subdivision for natural cubic splines*. In Chui, C. and Schumaker, L. (eds.): *Approximation theory IX*, Nashville: Vanderbilt University Press, pp. 345–352, 1998.

[161] Warren, J. and Weimer, H.: *Subdivision schemes for variational splines*. Rice University, Department of Computer Science Technical Report, CS-TR 00-354, available online at *http://cs-tr.cs.rice.edu/*, 2000.

[162] Weimer, H. and Warren, J.: *Subdivision schemes for thin-plate splines*. Computer Graphics Forum 17(3), pp. 303–313 and 392, 1998.

[163] Weimer, H. and Warren, J.: *Subdivision schemes for fluid flow*. Proceedings of SIGGRAPH 1999, Annual Conference Series. New York: ACM Press, pp. 111–120, 1999.

[164] Weisstein, E.: C*RC concise encyclopedia of mathematics.* Boca Raton, FL: CRC Press, 1999.

[165] Wejchert, J. and Haumann, D.: *Animation aerodynamics*. Proceedings of SIGGRAPH 1991, Annual Conference Series, pp. 19–22, New York: Addison Wesley, 1991.

[166] Young, D.: *Iterative solution of large linear systems.* New York: Academic Press, 1971.

[167] Zariski, O. and Samuel, P.: C*ommutative algebra*. New York: Springer-Verlag, 1975.

[168] Zhang, J.: C-*curves: an extension of cubic curves.* Computer-Aided Geometric Design 13, pp. 199–217, 1996.

[169] Zorin, D.: *Smoothness of stationary subdivision on irregular meshes.* Constructive Approximation 16, pp. 359–398, 2000.

[170] Zorin, D. and Schröder, P.: *A unified framework for primal/dual quadrilateral subdivision schemes.* Computer-Aided Geometric Design 18, pp. 483–502, 2001.

[171] Zorin, D., Schröder, P. and Sweldens, W.: *Interpolating subdivision for meshes with arbitrary topology.* Proceedings of SIGGRAPH 1996, Annual Conference Series. pp. 189–192, New York: ACM Press, 1996.

This Page Intentionally Left Blank

Index