Jonathan M. Borwein
Matthew P. Skerritt

An Introduction to Modern Mathematical Computing

With Maple™

Springer Undergraduate Texts
in Mathematics and Technology

Series Editors
Jonathan M. Borwein
Helge Holden

Jonathan M. Borwein • Matthew P. Skerritt

An Introduction to Modern Mathematical Computing

With Maple™

Jonathan M. Borwein
Director, Centre for Computer Assisted Research
Mathematics and its Applications (CARMA)
School of Mathematical and Physical Sciences
University of Newcastle
Callaghan, NSW 2308
Australia
jon.borwein@gmail.com

Matthew P. Skerritt
Centre for Computer Assisted Research
Mathematics and its Applications (CARMA)
School of Mathematical and Physical Sciences
University of Newcastle
Callaghan, NSW 2308
Australia
matt.skerritt@gmail.com

To my grandsons Jakob and Skye.
   Jonathan Borwein

To my late grandmother, Peggy, who ever urged me to hurry up with my PhD, lest she not be around to see it.
   Matthew Skerritt

Preface

Thirty years ago mathematical, as opposed to applied numerical, computation was difficult to perform and so relatively little used. Three threads changed that:

- The emergence of the personal computer, identified with the iconic Macintosh but made ubiquitous by the IBM PC.
- The discovery of fiber-optics and the consequent development of the modern internet culminating with the foundation of the World Wide Web in 1989 made possible by the invention of hypertext earlier in the decade.
- The building of the *Three Ms*: *Maple, Mathematica*, and MATLAB. Each of these is a complete mathematical computation workspace with a large and constantly expanding built-in "knowledge base". The first two are known as "computer algebra" or "symbolic computation" systems, sometimes written *CAS*. They aim to provide exact mathematical answers to mathematical questions such as what is

$$\int_{-\infty}^{\infty} e^{-x^2} \, dx,$$

what is the real root of $x^3 + x = 1$, or what is the next prime number after 1,000,000,000? The answers, respectively, are

$$\sqrt{\pi}, \quad \frac{\sqrt[3]{108 + 12\sqrt{93}}}{6} - \frac{2}{\sqrt[3]{108 + 12\sqrt{93}}}, \quad \text{and } 1{,}000{,}000{,}007.$$

The third M is primarily numerically based. The distinction, however, is not a simple one. Moreover, more and more modern mathematical computation requires a mixture of so-called *hybrid* numeric/symbolic computation and also relies on significant use of geometric, graphic and, visualization tools. It is even possible to mix these technologies, for example, to make use of MATLAB through a *Maple* interface; see also [6]. MATLAB is the preferred tool of many engineers and other scientists who need easy access to efficient numerical computation.

Of course each of these threads relies on earlier related events and projects, and there are many other open source and commercial software packages. For example, *Sage* is an open-source CAS, *GeoGebra* an open-source interactive geometry package, and *Octave* is an open-source counterpart of MATLAB. But this is not the place to discuss the merits and demerits of open source alternatives. For many purposes *Mathematica* and *Maple* are interchangeable as adjuncts to mathematical learning. We propose to use the latter.

After reading this book, you should find it easy to pick up the requisite skills to use *Mathematica* [14] or MATLAB.

Many introductions to computer packages aim to teach the *syntax* (rules and structure) and *semantics* (meaning) of the system as efficiently as possible [7, 8, 9, 13]. They assume one knows why one wishes to learn such things. By contrast, we intend to persuade that *Maple* and other like tools are worth knowing assuming only that one wishes to be a mathematician, a mathematics educator, a computer scientist, an engineer, or scientist, or anyone else who wishes/needs to use mathematics better. We also hope to explain how to become an experimental mathematician while learning to be better at proving things. To accomplish this our material is divided into three main chapters followed by a postscript. These cover the following topics:

- **Elementary number theory.** Using only mathematics that should be familiar from high school, we introduce most of the basic computational ideas behind *Maple*. By the end of this chapter the hope is that the reader can learn new features of *Maple* while also learning more mathematics.
- **Calculus of one and several variables.** In this chapter we revisit ideas met in first-year calculus and introduce the basic ways to plot and explore functions graphically in *Maple*. Many have been taught not to trust pictures in mathematics. This is bad advice. Rather, one has to learn how to draw trustworthy pictures.

> $plot\left(x\sin\left(\dfrac{1}{x}\right), x = -\dfrac{1}{2}..\dfrac{1}{2}\right)$
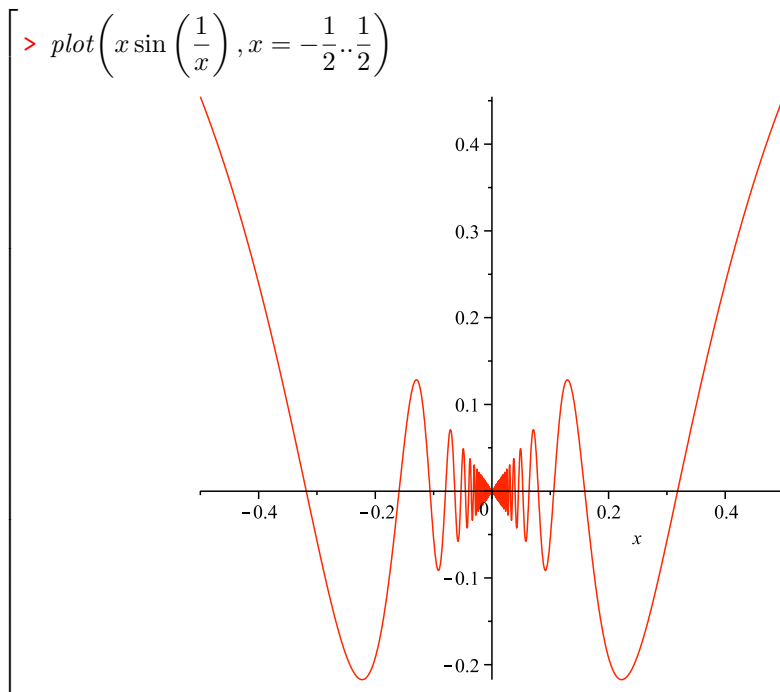
- **Introductory linear algebra.** In this chapter we show how much of linear algebra can be animated (i.e. brought to life) within a computer algebra system. We suppose the underlying concepts are familiar, but this is not necessary. One of the powerful attractions of computer-assisted mathematics is that it allows for a lot of "learning while doing" that may be achieved by using the help files in the system and also by consulting Internet mathematics resources such as *MathWorld, PlanetMath* or *Wikipedia*.

- **Visualization and interactive geometric computation.** Finally, we explore more carefully how visual computing [10, 11] can help build mathematical intuition and knowledge. This is a theme we will emphasize throughout the book.

Each chapter has three main sections forming that chapter's core content. The fourth section of each chapter has exercises and additional examples. The final section of each chapter is entitled "Further Explorations," and is intended to provide extra material for more mathematically advanced readers.

A more detailed discussion relating to many of these brief remarks may be followed up in [2, 3, 4] or [5], and in the references given therein.

The authors would like to thank Shoham Sabach and James Wan for their help proofreading preliminary versions of this book.

Additional Reading and References

We also supply a list of largely recent books at various levels that the reader may find useful or stimulating. Some are technical and some are more general.

1. George Boros and Victor Moll, *Irresistible Integrals*, Cambridge University Press, New York, 2004.
2. Jonathan M. Borwein and Peter B. Borwein, *Pi and the AGM: A Study in Analytic Number Theory and Computational Complexity*, John Wiley & Sons, New York, 1987 (Paperback, 1998).
3. Christian S. Calude, *Randomness and Complexity, from Leibniz To Chaitin*, World Scientific Press, Singapore, 2007.
4. Gregory Chaitin and Paul Davies, *Thinking About Gödel and Turing: Essays on Complexity, 1970-2007,* World Scientific, Singapore, 2007.
5. Richard Crandall and Carl Pomerance, *Prime Numbers: A Computational Perspective,* Springer, New York, 2001
6. Philip J. Davis, *Mathematics and Common Sense: A Case of Creative Tension,* A.K. Peters, Natick, MA, 2006.
7. Stephen R. Finch, *Mathematical Constants*, Cambridge University Press, Cambridge, UK, 2003.
8. Marius Giaguinto, *Visual Thinking in Mathematics,* Oxford University, Oxford, 2007.
9. Ronald L. Graham, Donald E. Knuth, and Oren Patashnik, *Concrete Mathematics*, Addison-Wesley, Boston, 1994.
10. Bonnie Gold and Roger Simons (Eds.), *Proof and Other Dilemmas: Mathematics and Philosophy*, Mathematical Association of America, Washington, DC, in press, 2008.
11. Richard K. Guy, *Unsolved Problems in Number Theory*, Springer-Verlag, Heidelberg, 1994.
12. Reuben Hersh, *What Is Mathematics Really?* Oxford University Press, Oxford, 1999.
13. J. Havil, *Gamma: Exploring Euler's Constant*, Princeton University Press, Princeton, NJ, 2003.
14. Steven G. Krantz, *The Proof Is in the Pudding: A Look at the Changing Nature of Mathematical Proof*, Springer, New York, 2010.
15. Marko Petkovsek, Herbert Wilf, and Doron Zeilberger, *A=B,* A.K. Peters, Natick, MA, 1996.

16. Nathalie Sinclair, David Pimm, and William Higginson (Eds.), *Mathematics and the Aesthetic. New Approaches to an Ancient Affinity*, CMS Books in Math, Springer-Verlag, New York, 2007.

17. J. M. Steele, *The Cauchy-Schwarz Master Class*, Mathematical Association of America, Washington, DC, 2004.

18. Karl R. Stromberg, *An Introduction to Classical Real Analysis*, Wadsworth, Belmont, CA, 1981.

19. Richard P. Stanley, *Enumerative Combinatorics*, Volumes 1 and 2, Cambridge University Press, New York, 1999.

20. Terence Tao, *Solving Mathematical Problems*, Oxford University Press, New York, 2006.

21. Nico M. Temme, *Special Functions, an Introduction to the Classical Functions of Mathematical Physics*, John Wiley, New York, 1996.

22. Fernando R. Villegas, *Experimental Number Theory*, Oxford University Press, New York, 2007.

Finally many useful links are maintained by the authors of *Mathematics by Experiment* [3] at www.experimentalmath.info.

<div align="right">

Jonathan Borwein
Matthew Skerritt
May 21, 2011

</div>

Contents

Conventions and Notation

Maple

Maple provides a wealth of different options for its interface, and mechanisms for entering commands. It is not within the scope of this book to deal with all of them. We present *Maple* input here as if it were entered in *Worksheet* mode with *2D Math* as both the input and output display settings. All three of these settings may be changed to be defaults (or not, of course) through *Maple*'s preferences.

This input scheme has the advantage that it looks a lot more like "regular" mathematics, making *Maple* worksheets easier to read than the historic text-only input scheme used in earlier versions. Using a worksheet mode instead of document mode also makes input, output, and explanatory text generally much clearer and is, in this author's opinion, superior when working on mathematical problems, although the document mode is probably preferable when trying to write results in a more human-readable format.

Maple examples in this book are formatted to look like they would in a *Maple* worksheet using the above assumptions, and look like the following

> *Input; Input*
> *Input*

$$\textit{Output}$$
$$\textit{Output}$$
$$\textit{Output}$$

<div align="center"><code>Warnings and Information</code></div>
<div align="center"><code>Errors</code></div>

Each input—and its associated output, warnings, and errors—are enclosed by a giant bracket (a "[", sometimes referred to as a "square bracket"), with a red > symbol acting as an input prompt. Multiple commands may be input together at the same prompt, and the input may even be spread over multiple lines. Input is colored black and is left aligned. Output is blue, and centered. Warnings are blue in a monospaced "typewriter" (Courier) font. Finally errors are red in a monospaced typewriter font. This is almost identical to *Maple* (given the above assumptions), with the only difference that *Maple*'s errors are a more purple-like color. For the sake of simplicity (and not too many colors) we have adopted red for this book.

Maple Input Basics

It is, unfortunately, not immediately obvious which keystrokes produce which input effects. As such we include Table P.1 which shows the non obvious keystrokes. Note that *Maple* will automatically format the text as mathematics for you at the moment it is typed. *Maple* will also attempt to make sensible decisions, based around order of operations, as to which parts of the input will be affected. If *Maple* makes an incorrect decision, then parentheses ((and)) are needed to make the expression unambiguous.

Operation	Keys	Example	Keystrokes
Multiplication	*	$a \cdot b$	a*b
	␣	$a\,b$	a␣b
Powers	^	a^b	a^b
Subscripts	_	a_b	a_b
Fractions	/	$\frac{a}{b}$	a/b
Not Equal	<>	$a \neq b$	a<>b
Less than or Equal to	<=	$a \leq b$	a<=b
Greater than or Equal to	>=	$a \geq b$	a>=b
Function	->	$a \to b$	a->b
Multiple Line Commands	shift-enter	$\begin{array}{c} a \\ b \end{array}$	a shift-enter b

Table P.1 *Maple* Input Keystrokes

Note that although multiplication may be produced using a space ($␣$) as well as the asterisk ($*$), the dot ($\cdot$) that *Maple* places instead of the asterisk leaves no ambiguity as to the nature of the calculation, whereas a space might easily be missed. As such the reader should feel free to use whichever method he or she prefers, but should be aware that the *Maple* examples in this book use the $\cdot$ notation to denote multiplication, unless there can be no ambiguity (for instance, with polynomial coefficients).

Useful Functions

We list here, for convenience, a small list of *Maple* functions which are arguably essential to know. These are functions to perform elementary mathematical operations (for example, square roots) or for simplification. The list is in Table P.2. Consult the help files on these functions for more specific details on their use. Note that many more functions are introduced and explained within the main text of this book.

Some comments:

- The **log** command may also be directed to perform logs to a base other than e if desired, but will perform natural logs unless specifically told otherwise. For example, the command log[2](3) will calculate the log of 3 to the base 2.
- The *Maple* names **pi** and **Pi** are both used for the lower case greek letter π, whereas **PI** is used for upper case Π. These are case sensitive, and all distinct. The most commonly used is probably **Pi** (capital "P" and small "i") which is used when the *value* of π is desired. The other two (**pi** and **PI**) are only symbols with no innate value.
- The commands for algebraic rearrangement work primarily as one would expect from their names. However, it should be noted that when faced with a complicated

Desired Effect	Command
Square root of a number ($\sqrt{x}$)	**sqrt**
nth root of a number ($\sqrt[n]{x}$)	**root**
	surd
Power of e (e^x)	**exp**
Natural logarithm ($\log x$)	**log**
	ln
Value of π	**Pi**
Value of ∞	**infinity**
Rearrange an algebraic expression	**simplify**
	factor
	expand
	combine
Conversion between forms	**convert**

Table P.2 Essential *Maple* Functions

answer from *Maple* it is sometimes the case that **factor**, **expand**, or even **combine** will produce a more simplified answer than the **simplify** command does. Furthermore, these commands used in conjunction with each other can produce superior results.

- The **convert** function is a multi faceted function that allows many different types of conversions to be performed (see Exercise 4). Of particular interest and utility, in conjunction with the algebraic rearrangement above, is the variant that allows conversion of an expression to one involving standard functions: *convert*(· , *StandardFunctions*).

Mathematics

Included in Table P.3 is a list of mathematical notation, and its meaning.

Notation	Meaning
$\mathbb{N}$	The set of natural numbers
$\mathbb{Z}$	The set of integers
$\mathbb{Q}$	The set of rational numbers
$\mathbb{R}$	The set of real numbers
$\mathbb{C}$	The set of complex numbers
$\log x$	The *natural* logarithm of x
$\log_b x$	The logarithm of x (base b)
$\sin x$	The sine function applied to x
$\sin^n x$	The nth power of the sine function applied to x. That is $(\sin x)^n$.
$\sin^{-1} x$	The inverse-sine function applied to x
$\arcsin x$	The inverse-sine function applied to x
$\cos x$	The cosine function applied to x
$\cos^n x$	The nth power of the cosine function applied to x. That is $(\cos x)^n$.
$\cos^{-1} x$	The inverse-cosine function applied to x
$\arccos x$	The inverse-cosine function applied to x

Table P.3 Mathematical Notation

Note: In the case of the log, sin, cos, and similar functions, if there is any ambiguity as to what the function in question is and is not to be applied to, then parentheses or brackets are used to make it clear. For example $2 + \sin(3x + 1)$.

Chapter 1
Number Theory

This chapter includes the basics of the use of *Maple*, illustrated by fairly simple examples mostly involving integers. For this chapter you need to know what a sequence is, an infinite sum, summation notation, what a function is, and what a polynomial is. By the end of the chapter you should be comfortable using *Maple* for moderately complex tasks, and should be ready to learn the new commands required for doing specific mathematics such as calculus or linear algebra.

1.1 Introduction to *Maple*

Before we can set about exploring mathematics with *Maple* we need to know how to input basic commands into it. This section will introduce *Maple* and its most basic commands.

1.1.1 Inputting Basic Maple Expressions

At its absolute most basic, *Maple* can be used as sort of an overblown pocket calculator. We give it an expression to calculate, and *Maple* performs the calculation.

> $1 + 2$

$$3$$

> $2 \cdot 3^5 + 12 - 2$

$$496$$

or even more complicated statements involving factorials, trigonometric functions, and a lot more besides

> $\dfrac{\left(\sin\left(\dfrac{\mathrm{Pi}}{2} \right) + 12! \cdot \mathrm{sqrt}(12) \right)}{\exp(4)}$

$$\dfrac{1 + 958003200\,\sqrt{3}}{\mathrm{e}^4}$$

Notice that in this last example that *Maple* didn't provide a decimal number as the answer to the input. This rather nicely illustrates a key difference between *Maple* and

your pocket calculator. *Maple* is a Computer Algebra System (CAS), and performs its calculations as exactly as possible. When no exact number occurs, *Maple* provides an exact expression. So e^4 and $\sqrt{3}$ are exact values, whereas 54.59815003 and 1.732050808 (respectively) are decimal approximations. *Maple* gives us the exact value unless we specifically ask it otherwise. To accomplish this we either use the **evalf** command, or put a decimal point next to a constant.

$$> \text{evalf}\left(\frac{\left(\sin\left(\frac{\text{Pi}}{2}\right) + 12! \cdot \text{sqrt}(12) \right)}{\exp(4)} \right)$$

$$3.039132676 \; 10^7$$

$$> \frac{\left(\sin\left(\frac{\text{Pi}}{2}\right) + 12! \cdot \text{sqrt}(12.) \right)}{\exp(4)}$$

$$\frac{1.659310217 \; 10^9}{\mathrm{e}^4}$$

Attention should be drawn, in the second example, to the period after the 12 in the square root. This is a shorthand for—in this case—12.0, and tells *Maple* that the value is a decimal. The inclusion of a decimal in an expression is one way to tell *Maple* that we wish a numeric (rather than symbolic) calculation to be performed. In this particular case, only the numerator was evaluated numerically, which suggests that *Maple* considers the numerator and denominators somehow to be separate expressions. If in doubt, it's usually easier just to put the entire expression in an **evalf** command.

Inasmuch as *Maple* is a CAS, we ought to expect that it can do some basic algebra. In point of fact it can, and a good start is to work with basic polynomials.

$$> 3x^2 + 2x^3 + 3$$

$$3x^2 + 2x^3 + 3$$

$$> 4x^2 + 9x^2$$

$$13x^2$$

Notice that *Maple* automatically adds the like terms together.

$$> 3x^4 \cdot 3y^2$$

$$9x^4 y^2$$

In general, *Maple* considers any word it does not otherwise know to be an algebraic variable. We could use the strings "alice" and "bob" in place of x and y and *Maple* will treat them just like any other algebraic variable.

$$> alice + bob; \; \% + 2; \% + 5$$

$$alice + bob$$
$$alice + bob + 2$$
$$alice + bob + 7$$

The previous example demonstrates a couple of things that we have not yet seen before, which we take some time to highlight.

First there were three commands on one line. Each command is ended by a semicolon (;), which tells *Maple* where a command ends. Alternatively, a command may be ended by a full colon (:) if we wish for the output of that command not to be displayed. Older versions of *Maple* required a semicolon (or colon as desired) after each and every

command—even if there were only a single command on a line—but version 12 and later only requires them between multiple commands.

Second there is the special character **%**. This character is used to refer to the value of the most recent calculation. The **%** is a very useful tool, but be careful when using it, as the most recent calculation performed may not be on the same, or even the previous line of your input (see Exercise 3).

1.1.2 Variables

In addition to providing an unknown quantity for working with algebra, *Maple* variables (any string of characters that *Maple* doesn't know to be something else) can also have values assigned to them. There are two primary reasons to do this. The first is to give a name to an expression you want to use later; the other is to store a value that might change. In reality, these are two sides to the same coin. To assign a value to a variable, we use the assignment operator (:=).

> $A := 2$

$$A := 2$$

Note that the assignment operator is often used in mathematics to denote a definition. This is no less true in *Maple*. We could happily view assigning a value to a variable name as defining the variable name to be that value. Once we have assigned a value to a variable, when we use the variable name, *Maple* automatically uses its value for the computation.

> $A; A + 2; A^{10}$

$$2$$
$$4$$
$$1024$$

> $Poly := 3x^3 + 2x^2 + 3x; Poly + 3x^2 + x$

$$Poly := 3x^3 + 2x^2 + 3x$$
$$3x^3 + 5x^2 + 4x$$

We can, if we wish, have *Maple* perform a calculation, and store the result of that calculation in the variable.

> $value := 4 \cdot 12 + 13^5$

$$value := 371341$$

We can even use variable names as input to functions, and *Maple* still uses the value of the variable. We look at exactly what a function is in the next subsection, but for now notice that in the example below, *Maple* has plotted the cubic $3x^3 + 2x^2 + 3x$ to which we gave the name *Poly* in an earlier example above.

> $plot(Poly, x = -1..1)$
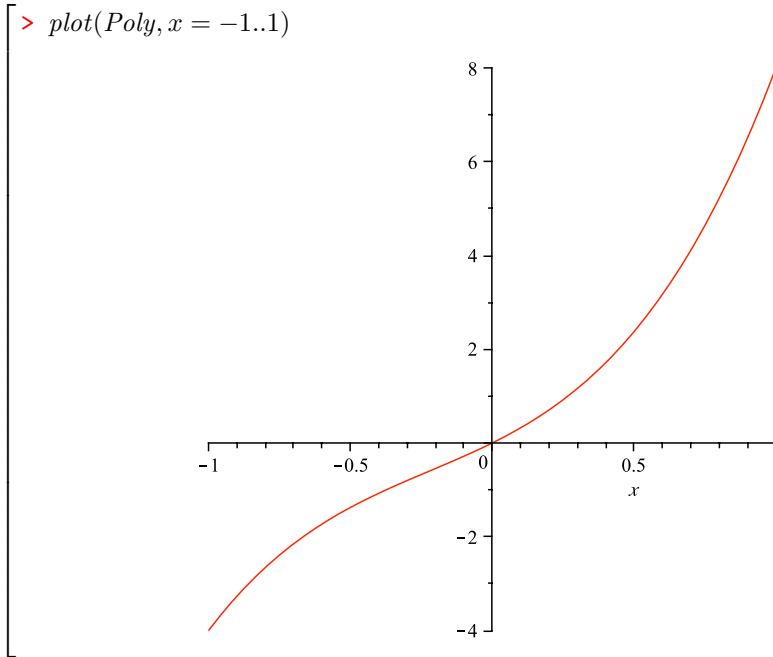
We may assign any valid *Maple* input to a variable, and *Maple* will always use the value of the variable wherever we use its name. This can be very useful, if occasionally confusing. A common theme throughout the course of the book is that one must remain aware of the subtleties of the system when utilizing it. With great power comes great responsibility.

So far we have used variables to give a name to something we wish to somehow utilize later on. However, as mentioned above, sometimes we want to use a variable as a storage box for values that can and will change. This is really no different from reassigning a variable name to a different expression.

> $a := 2; 2 \cdot a$

$$a := 2$$
$$4$$

> $a := 4; 2 \cdot a$

$$a := 4$$
$$8$$

Such a technique, however, lets us use a variable to store intermediate results of a calculation in progress.

> $total := 0$

$$total := 0$$

> $total := total + 12$

$$total := 12$$

> $total := total + 11$

$$total := 23$$

Note here that not only are we reassigning the value of the variable named *total* but we are also using its current value to calculate its new value. The line $total := total + 12$ means, in English, something along the line of "set the value of total to be its current value plus 12." What *Maple* actually does is evaluate the right-hand side of the definition

(which in this case is $total + 12$) and assign the result of that calculation back to the variable name on the left-hand side. These need not be the same variable, for instance:

```
> subtotal1 := 12; subtotal2 := 23
```
$$subtotal1 := 12$$
$$subtotal2 := 23$$

```
> total := subtotal1 + subtotal2
```
$$total := 35$$

The above code is perfectly valid, and is really just another case of the basic assignment of a variable we saw at the beginning of the subsection.

1.1.3 Functions

Having dealt with basic input and variables, we now move on to another key *Maple* concept, that of functions. A function, simplistically speaking, is just something that takes an input (perhaps several inputs) and produces output. We have already seen a couple of functions in our examples thus far such as **sin** and **evalf**. Functions are written in the form of $name(input, input, ...)$ where $name$ is the name of the function, and $input, input, ...$ is a comma separated list of the inputs, sometimes called parameters. To start with we deal mostly with functions that take a single input, and should look very similar to the functions seen in first-year calculus.

```
> ifactor(1573)
```
$$(11)^2 \, (13)$$

```
> factor(x^4 - 2x^3 - 13x^2 + 14x + 24)
```
$$(x - 2)(x - 4)(x + 3)(x + 1)$$

```
> simplify( (x^2 + x)/(x^3 + 2x) )
```
$$\frac{x + 1}{x^2 + 2}$$

```
> is(1 < 2)
```
$$true$$

```
> lhs(A = B)
```
$$A$$

A list of commonly used basic functions is included at the very beginning of this book, just after the preface. For more complicated functions, *Maple*'s own help files are always a good source of information. Every example thus far has been a function that comes built-in to *Maple*. In the next section we start creating our own functions, but for now we look at an example of a function that takes multiple inputs.

```
> convert(0.1234, rational)
```
$$\frac{617}{5000}$$

The astute reader will also recognize that the plot function we saw earlier was also a function that took multiple inputs.

1.1.4 Sequences, Lists, and Sets

In *Maple*, however, a sequence refers to a group of expressions separated by commas. For example, the following demonstrates two *Maple* sequences.

> $1, 2, 3; poly := x^3 + 3; poly, 5, bob$

$$1, 2, 3$$
$$poly := x^3 + 3$$
$$x^3 + 3, 5, bob$$

First-year calculus students should have studied infinite sequences and series (which were just a more intricately constructed sequence), and the difference between a *Maple* sequence and a mathematical (infinite) sequence could potentially be confusing. Although *Maple* sequences could well be thought of as finite sequences, it might be easier to just recognize the fact that the two are different. *Maple* can handle the usual calculus sequences, but does so using the **limit** function, which is dealt with in more detail in Chapter 2.

> $limit\left(\dfrac{1}{k^2}, k = infinity\right)$

$$0$$

So with that clarification out of the way, let's get back to *Maple* sequences. For the remainder of this chapter a sequence refers to a *Maple* sequence. Any valid *Maple* code may form an element of a sequence. The astute reader may have already noticed that the functions above, which took multiple variables, accepted their input as a sequence. For example, the **plot** function took as its input the sequence $poly, x = -1..1$ and the **convert** function had the sequence $0.1234, rational$ as its input.

Because sequences are just another valid *Maple* expression, they may be stored in a variable. If we have two sequences and put them into a single sequence, the result is one large sequence, not two nested sequences. This becomes clearer when we see how lists and sets "nest" later in the subsection.

> $A := a, b, c; B := 1, 2, 3; C := i, ii, iii$

$$A := a, b, c$$
$$B := 1, 2, 3$$
$$C := i, ii, iii$$

> A, B, C

$$a, b, c, 1, 2, 3, i, ii, iii$$

Sequences can also be a very convenient way of assigning multiple variables in a single command. For instance, if we wish to assign $A := 1$, $B := 2$, and $C := 3$ then we could use

> $A, B, C := 1, 2, 3$

$$A, B, C := 1, 2, 3$$

> $A; B; C;$

$$1$$
$$2$$
$$3$$

There is also a special sequence called the null sequence. This is a sequence with no elements, much as the empty set in set theory is the set with no elements. The null sequence in *Maple* is referenced by the keyword **NULL**. As such, putting the null sequence in any sequence doesn't change the sequence at all.

> $NULL, a, b, c; a, b, c, NULL; a, b, NULL, c$
$$a, b, c$$
$$a, b, c$$
$$a, b, c$$

Using a null sequence allows a list to be built up by parts within a variable, a little like the *total* variable was in the previous subsection.

> $S := NULL$
$$S :=$$

> $S := S, a, b, c$
$$S := a, b, c$$

> $S := S, d, e$
$$S := a, b, c, d, e$$

If we want to produce a sequence that follows a fairly predictable pattern, we have a handy command, **seq**. To print out the first ten squares we simply input

> $seq(k^2, k = 1..10)$
$$1, 4, 9, 16, 25, 36, 49, 64, 81, 100$$

or for something a little more complicated

> $seq\left(3 \cdot k^2 + \dfrac{k}{2}, k = 4..15\right)$
$$50, \frac{155}{2}, 111, \frac{301}{2}, 196, \frac{495}{2}, 305, \frac{737}{2}, 438, \frac{1027}{2}, 595, \frac{1365}{2}$$

In general, to have *Maple* print out the sequence $\{x_n\}_{n=a}^{b}$ we use the command $seq(x_n, n = a..b)$. There is a shortcut that can be applied if we wish to repeat the same term multiple times, which will come in handy when we do some calculus. To do this we use the sequence operator (**\$**).

> $x\$4$
$$x, x, x, x$$

This sequence operator can be used as a shortcut to the **seq** command, but be warned that it isn't quite as flexible as the **seq** command (see Exercises 5 and 6).

If we have a sequence, we may wish to use only a subsequence of it, or perhaps only a single element. *Maple* allows this through indexing using the index operator [], or through subscripting in the graphical editor. Be warned, subscripting is a shorthand for the square bracket operator. If unsure, use square brackets. For subsequences we also use the range (**..**) operator.

> $S := seq(k^2, k = 1..10);$
$$S := 1, 4, 9, 16, 25, 36, 49, 64, 81, 100$$

> $S[3]; S_4;$
$$9$$
$$16$$

> $S[5..8]; S_{3..7}$

$$25, 36, 49, 64$$
$$9, 16, 25, 36, 49$$

> $S[..4]; S_{..3}; S[6..]; S_{7..}$

$$1, 4, 9, 16$$
$$1, 4, 9$$
$$36, 49, 64, 81, 100$$
$$49, 64, 81, 100$$

In the latter example, the "half ranges" $..n$ and $m..$ mean "the first n elements" and "from the mth to the last element," respectively. Such notions make sense because we are indexing a sequence that contains only a finite number of elements. An alternate way of thinking of these ranges is to consider that *Maple* automatically inserts the beginning or the end index for the missing number as appropriate.

We now look at two related notions to sequences: lists and sets. Syntactically, a list or set is just a sequence enclosed in [] or {}, respectively. Sets are unsorted and ignore duplication, as should be expected by the reader familiar with elementary set theory. Lists are ordered and allow duplications. Both lists and sets can be nested, which makes them distinct in behavior from sequences. Lists and sets are also useful for removing ambiguity when trying to have a function recognize a sequence as a single input, an example of which can be seen in Exercise 14.

> $L := [1, 1, 2, 2, 3, 3, 4, 4]; S := \{1, 1, 2, 2, 3, 3, 4, 4\}$

$$L := [1, 1, 2, 2, 3, 3, 4, 4]$$
$$S := \{1, 2, 3, 4\}$$

> $[L, S]; \{L, S\}$

$$[[1, 1, 2, 2, 3, 3, 4, 4], \{1, 2, 3, 4\}]$$
$$\{[1, 1, 2, 2, 3, 3, 4, 4], \{1, 2, 3, 4\}\}$$

Lists and sets may be indexed exactly as with a sequence, with the difference that if a range (using the $..$ operator) is given for the index, then a list or set, respectively, is produced. Indexing a single element produces only the element as with a sequence

> $L[1], L_2, S[3], S_4$

$$1, 1, 2, 2$$

> $L[3..5], L_{..3}, S[3..], S_{2..3}$

$$[2, 2, 3], [1, 1, 2], \{3, 4\}, \{2, 3\}$$

Observe that in the above two examples we have used a sequence to display the results of the calculations on a single line.

If we wish to know whether a list or a set contains a particular element, we may use the **in** operator. This may be thought of as being the $\in$ operator, and is even printed as such in *Maple*.

> $1 \, \mathbf{in} \, S$

$$1 \in S$$

This command on its own does nothing but write itself out again with the element symbol. In order to have *Maple* actually tell us whether $1 \in S$ (in the above case) then we need to use the **is** or **evalb** commands. We saw **is** in Section 1.1.3. Both functions return a value of either *true* or *false*. In fact the function **evalb** is short for "evaluate as Boolean" where a Boolean value is either true or false.

```
> is(1 in S); evalb(9 in L)
```

$$true$$
$$false$$

And finally, the usual set operations of **union** and **intersect** work as expected with sets, and not at all with lists.

```
> {1, 2} union {2, 3, 4}; {1, 2} intersect {2, 3, 4}
```

$$\{1, 2, 3, 4\}$$
$$\{2\}$$

1.1.5 Sums and Products

Having dealt with the basics of *Maple*, we can now move onto some mathematics a little more like we might see at the beginning of a first-year math course. We begin by looking at how *Maple* can handle sums and products, both of the finite and the infinite variety.

For large additions (or, indeed, for anything too complicated to use the $+$ operator practically) *Maple* provides the **add** and **sum** commands. There are a couple of small differences between these two commands, but the most glaring difference is that **sum** is designed for infinite sums (sometimes known as series) and tries to be clever about the addition, whereas **add** simply adds together a bunch of terms, and so is generally a lot quicker than **sum**, but cannot handle infinite or indefinite sums. Both the sum and add commands look very much like the **seq** command.

```
> add(k^2, k = 1..10)
```

$$385$$

```
> sum(k^2, k = 1..10)
```

$$385$$

The sum command comes in two forms. If the command is written with a lower case "s", then *Maple* performs the summation as we have previously seen. This form is known as an the *active* form of the **sum** command. If the command is written with a capital "S", then *Maple* performs no immediate computation, and instead simply displays the sum in the sigma notation. This form is known as the *inert* form. In order to be able to ask *Maple* to perform the calculation of an inert sum, we have the **value** function. *Maple* displays an inert sum with a grey sigma, and an active sum (on the unusual occasions that it displays an active sum in sigma notation) with a blue sigma.

```
> Sum(k^2, k = 1..N)
```

$$\sum_{k=1}^{N} k^2$$

```
> factor(value(%))
```

$$\frac{1}{6}N(N+1)(2N+1)$$

As well as sums, *Maple* can also handle products. Much like **add** and **sum**, *Maple* has two functions to perform products; one that simply performs a multiplication, and the other which is smarter but slower and designed for evaluating infinite products. These commands are **mul** and **product**, respectively. Also like **sum**, the **product** command

comes in an inert and active variant which is indicated by a capital "P" for the inert, and a lower case "p" for the active.

> $mul(k^2, k = 1..10)$

$$13168189440000$$

> $product(k^2, k = 1..10)$

$$13168189440000$$

> $Product(k^2, k = 1..N); value(\%)$

$$\prod_{k=1}^{N} k^2$$

$$\Gamma(N + 1)^2$$

The result of the latter of these computations should make more sense if you know that $\Gamma(n) = (n - 1)!$

We show in later chapters more functions that have inert and active forms. It is a common enough occurrence in *Maple*. Be sure to remember that the function with a capital letter at the beginning of its name (e.g., **Sum** or **Product**) is the inert form that will not perform any calculation, and the function whose name is all lower case (e.g., **sum** or **product**) is the active form. Inert forms always require the use of the **value** function to have the calculation in question performed.

One should be made aware of a subtlety when evaluating inert sums and products. It may be tempting to use the **evalf** function instead of the **value** function to evaluate a sum or product. It may appear, superficially, that these two commands would do the same thing, however, they are quite different. The **evalf** functions causes *Maple* to obtain a *decimal approximation* of the calculation, whereas **value** causes the calculation to be performed symbolically (just as the active form of the function in question performs directly). Each method has its place, but it is important to understand the difference.

1.1.6 Packages

Not all of *Maple*'s functions are able to be used immediately. Instead some of them are stored away in packages. We have not, hitherto, come across any packaged functions yet, so we quickly look at an example now. Suppose we have a list, and we wish to reverse its order. There is a function named **Reverse** in a package named, appropriately enough, ListTools.

> $L := [1, 2, 3, 4, 5, 6, 7, 8, 9]$

$$L := [1, 2, 3, 4, 5, 6, 7, 8, 9]$$

If we try to use the **Reverse** function normally, nothing happens

> $Reverse(L)$

$$Reverse([1, 2, 3, 4, 5, 6, 7, 8, 9])$$

To use a function that is in a package we need to tell *Maple* exactly where to locate the function. That is, we need to tell *Maple* that the function is in a package, as well as in which package. This is done by using the so-called *long form* of the function, which

is of the form *package*[*function*]. In the particular case of our list reversal example, this would be *ListTools*[*Reverse*], which should be read as "the **Reverse** function from the ListTools package."

> $ListTools[Reverse](L)$

$$[9, 8, 7, 6, 5, 4, 3, 2, 1]$$

Being required to type *ListTools*[] every time we use a function might become tedious if we want to use the function more than just once or twice. Furthermore, we may wish to use other functions from the ListTools package. We may avoid the need to always use the long form of a function name by simply loading the package into *Maple* by using the **with** function. Doing this will load all the functions in the package, allowing their use directly, without need to specify to which package they belong.

> $with(ListTools)$

$$[BinaryPlace, BinarySearch, Categorize, DotProduct, Enumerate, FindRepetitions,$$
$$Flatten, FlattenOnce, Group, Interleave, Join, JoinSequence, LengthSplit,$$
$$MakeUnique, Occurrences, Pad, PartialSums, Reverse, Rotate, Search, SearchAll,$$
$$Sorted, Split, Transpose]$$

> $Reverse(L)$

$$[9, 8, 7, 6, 5, 4, 3, 2, 1]$$

Maple helpfully gives a list of all the new functions that are available when we use the **with** command. This may, of course, be suppressed in the usual way if we wish. Some small care must be taken here, as sometimes two (or more) packages may each contain a function (or functions) with identical names. In this case, the function from the most recently loaded package is the one that will be used, unless the long form is used to specifically and unambiguously refer to one or the other.

1.2 Putting It Together

In the previous section, we looked at inputting single commands into *Maple*. These may be thought of as building blocks. In this section we begin to put these building blocks together to produce more complicated calculations. We also begin to introduce some more serious mathematics in order to motivate or illustrate the particular *Maple* constructions with which we are dealing.

1.2.1 Creating Functions

In the previous section we have seen what a function is. In addition to the built-in functions, we may create our own. A function, using the arrow operator ($\rightarrow$) takes on the form *input* $\rightarrow$ *expression*, and is a perfectly valid *Maple* expression on its own. The $\rightarrow$ operator is produced by typing the text `->`

> $x \rightarrow 3x^2 + 4x - 2$

$$x \rightarrow 3x^2 + 4x - 2$$

The above function takes a single expression as input (which it calls x), and then uses that expression to perform the calculation $3x^2 + 4x - 2$. The name of the input

variable is entirely arbitrary and could be any valid *Maple* variable name. In order to be able to actually use this function, we need to assign it to a variable name.

> $f := x \rightarrow 3x^2 + 4x - 2$
$$f := x \rightarrow 3x^2 + 4x - 2$$

> $f(2); f(4); f(A); f(\Gamma(N)); f([1, 2])$
$$18$$
$$62$$
$$3A^2 + 4A - 2$$
$$3\,\Gamma(N)^2 + 4\,\Gamma(N) - 2$$
$$3[1, 2]^2 + 4[1, 2] - 2$$

It is interesting to see in the last example above that even though it makes apparently no sense mathematically, *Maple* will, nonetheless, accept $[1, 2]$ as input to the function f and produce the output string as if $[1, 2]$ were a variable.

How about something that might be a bit more familiar. Recall from first-year calculus that

$$\sum_{k=1}^{\infty} \frac{1}{k}$$

diverges, whereas

$$\sum_{k=1}^{\infty} \frac{1}{k^2}$$

converges. This is a good reminder that the divergence test is inconclusive in the case where the sequence has a limit of 0 because $1/k \rightarrow 0$ (as $k \rightarrow \infty$) yet the series diverges but $1/k^2 \rightarrow 0$ yet the series converges. To attempt to verify the convergence of the $1/k^2$ series, we can try the ratio test with *Maple*, but unfortunately this test also proves inconclusive.

> $a := k \rightarrow \dfrac{1}{k^2}; limit(a(N), N = infinity); abs\left(\dfrac{a(N+1)}{a(N)}\right); limit(\%, N = infinity)$

$$a := N \rightarrow \frac{1}{k^2}$$
$$0$$
$$\left|\frac{N^2}{(N+1)^2}\right|$$
$$1$$

We side-track for a moment and notice that for every N we have

$$\left|\frac{a(N+1)}{a(N)}\right| < 1$$

This is not hard to verify by hand, but we ask *Maple* for a verification anyway. Observe that the fraction is undefined for $N = -1$, but this isn't a concern for us as we're only interested in natural values of N. However, we may need to be a little specific in what we ask *Maple*.

> $is\left(\text{abs}\left(\dfrac{a(N+1)}{a(N)}\right)<1\right)$

$$\mathit{FAIL}$$

> $is\left(\text{abs}\left(\dfrac{a(N+1)}{a(N)}\right)<1\right)$ assuming $N::posint$

$$\mathit{true}$$

You should read $N::posint$ as "N is a positive integer." We could, in this particular case (but not always), have used $N \geq 0$ in place of $N::posint$ and would have obtained the same answer.

Now, returning to the question of convergence of $1/k^2$, we have yet to verify analytically whether the series converges or diverges. Recall, however, that an infinite sum is a limit of partial sums. That is,

$$\sum_{k=1}^{\infty} a(k) = \lim_{N\to\infty} \sum_{k=1}^{N} a(k)$$

As our next attempt we attempt to see how the partial sums behave. To do this we begin by creating the Nth partial sum of our series $1/k^2$ as a function in *Maple*.

> $f := N \rightarrow sum\left(\dfrac{1}{k^2}, k=1..N\right)$

$$f := N \rightarrow \sum_{k=1}^{N} \dfrac{1}{k^2}$$

We now have f as a function of N where N is the number of terms to sum. Let us now see how this sum behaves with increasing values

> $f(1), f(2), f(10), f(100), f(10000)$

$$1, \frac{5}{4}, \frac{1968329}{1270080},$$

$$\frac{15895086941330378731122979285175538597023834985437098598894328348038181310903699 01}{9721861444343810305896579766726231441619755839957462417827203547055179861652480 00},$$

$$-\Psi(1, 10001) + \frac{1}{6}\pi^2$$

Well that wasn't particularly helpful, although the term for $f(10000)$ looks promising. Let's try asking for decimal answers instead.

> $seq(evalf(f(10^i)), i=1..6)$

$1.549767731, 1.634983900, 1.643934568, 1.644834073, 1.644924068, 1.644933068$

Note that we needed to be careful here, as our function f consisted of a sum that used the dummy variable k, and so if we had also used k for the dummy variable in the **seq** command, they would have conflicted. See Exercise 13. We avoided this problem by using i as the dummy variable for the **seq** command.

We can see evidence here of convergence of the series to $1.644\ldots$, remembering that the sequence we asked for was

$$f(1), f(10), f(100), f(1000), f(10000), f(1000000)$$

As we calculate progressively larger and larger partial sums, the value of those partial sums seems to change less and less. At this point we may as well ask *Maple* if it can give us an answer for the limit.

> $limit(f(n), n = infinity); evalf(\%)$

$$\frac{1}{6}\pi^2$$

$$1.644934068$$

And there we have it. It looks very much as if the series converges to $1/6\ \pi^2$.

1.2.2 Loops

Until now if we wanted to perform something several times, we either typed it in multiple times at the command prompt, or we constructed a sequence. Sometimes these options aren't satisfactory. Let us revisit our example of the series $\sum(1/k^2)$. Earlier we used **seq** to print out the sequence

$$\left\{ \sum_{k=1}^{10^N} \frac{1}{k^2} \right\}_{N=1}^{6}$$

which quite conveniently demonstrated the convergence of the series. The sequence was easy to read. However, suppose we wanted to see more values of the sequence. Let's look at the values of the partial sums for values of N as the first 20 powers of 2. That is, $N = 2, 4, 8, 16, \ldots, 1048576$.

> $seq(evalf(f(2^i)), i = 1..20)$

```
1.250000000, 1.423611111, 1.527422052, 1.584346533, 1.614167263,
1.629430501, 1.637152005, 1.641035436, 1.642982848, 1.643957982,
1.644445906, 1.644689957, 1.644812005, 1.644873035, 1.644903551,
1.644918809, 1.644926439, 1.644930253, 1.644932161, 1.644933114
```

That's a bit of a mess, but not completely unreadable. Now, we would like to see which values of N produce which of those outputs. We can work it out by counting from the left and working out the power of 2, but it would be nicer to see it. We tell *Maple* to print $f(N) =$ before each answer. In order to do this without having *Maple* evaluate $f(N)$ as a function, we enclose the f in single quotes, which tells *Maple* to treat it as a symbol only, and not to evaluate it.

> $seq('f'(2^i) = evalf(f(2^i)), i = 1..20)$

```
 f(2) = 1.250000000, f(4) = 1.423611111, f(8) = 1.527422052, f(16) =
    1.584346533, f(32) = 1.614167263, f(64) = 1.629430501, f(128) =
 1.637152005, f(256) = 1.641035436, f(512) = 1.642982848, f(1024) =
1.643957982, f(2048) = 1.644445906, f(4096) = 1.644689957, f(8192) =
     1.644812005, f(16384) = 1.644873035, f(32768) = 1.644903551,
     f(65536) = 1.644918809, f(131072) = 1.644926439, f(262144) =
    1.644930253, f(524288) = 1.644932161, f(1048576) = 1.644933114
```

Ugh, now that's really a mess. We might improve matters if we could somehow put each equality on its own line, instead the one big sequence we currently have. We might achieve this by simply typing out all 20 expressions one after the other, but this would be slow and tedious, and would not work well if we wanted many computations to be performed. Fortunately, *Maple* provides a mechanism for such repeating calculations as these, called a **for** loop.

> **for** i **from** 1 **to** 20 **do** $'f'(2^i) = evalf(f(2^i))$ **od**

$$f(2) = 1.250000000$$
$$f(4) = 1.423611111$$
$$f(8) = 1.527422052$$
$$f(16) = 1.584346533$$
$$f(32) = 1.614167263$$
$$f(64) = 1.629430501$$
$$f(128) = 1.637152005$$
$$f(256) = 1.641035436$$
$$f(512) = 1.642982848$$
$$f(1024) = 1.643957982$$
$$f(2048) = 1.644445906$$
$$f(4096) = 1.644689957$$
$$f(8192) = 1.644812005$$
$$f(16384) = 1.644873035$$
$$f(32768) = 1.644903551$$
$$f(65536) = 1.644918809$$
$$f(131072) = 1.644926439$$
$$f(262144) = 1.644930253$$
$$f(524288) = 1.644932161$$
$$f(1048576) = 1.644933114$$

Now this is much easier to read. *Maple* has calculated the expression $'f'(2^i) = evalf(f(2^i))$ for us, and has done so 20 times, each time with the value of i increased by 1. After each calculation it has output the result of the calculation just as it would have if we had entered it manually at the command prompt.

We have calculated some remarkably large partial sums here. It turns out, for this example, that we can blithely ask *Maple* to calculate some truly extraordinary large partial sums, and it will give us an answer almost instantaneously.

> $evalf(f(10^{12})); evalf(f(10^{100}))$

$$1.644934068$$
$$1.644934068$$

It is tempting to conclude that computer technology is just so fast nowadays that such a performance is simply to be expected. Unfortunately, this is not true as we show.

The previous example is taken from a slightly more general result. This result states that the p-series

$$\sum_{k=1}^{\infty} \frac{1}{k^p}$$

where $p \in \mathbb{R}$, converges only when $p > 1$. So let's see what happens when we choose $p = \pi$.

> $g := N \to sum\left(\dfrac{1}{k^{\mathrm{Pi}}}, k = 1..N\right)$

$$g := N \to \sum_{k=1}^{N} \frac{1}{k^{\pi}}$$

> $g(infinity); evalf(\%)$

$$\zeta(\pi)$$

$$1.176241738$$

We can see that the sum converges to the zeta (ζ) function evaluated at π. If we now go and look up the zeta function we find out that the zeta function is defined in terms of these infinite sums.

$$\zeta(p) := \sum_{k=1}^{\infty} \frac{1}{k^{p}}$$

Maple hasn't told us a great deal here. Nonetheless we have a decimal approximation to which we should see convergence. We can use the same **for** loop from above, but this time with the g function.

> **for** i **from** 1 **to** 20 **do** 'g'$(2^i) = evalf(g(2^i))$ **od**

$$g(2) = 1.113314732$$
$$g(4) = 1.157856429$$
$$g(8) = 1.171486718$$
$$g(16) = 1.175089712$$
$$g(32) = 1.175971771$$
$$g(64) = 1.176179520$$
$$g(128) = 1.176227516$$
$$g(256) = 1.176238498$$
$$g(512) = 1.176240996$$
$$g(1024) = 1.176241563$$
$$g(2048) = 1.176241563$$
$$g(4096) = 1.176241563$$
$$g(8192) = 1.176241563$$
$$g(16384) = 1.176241563$$
$$g(32768) = 1.176241563$$
$$g(65536) = 1.176241563$$
$$g(131072) = 1.176241563$$

Warning, computation interrupted

This time the computation was much slower, and seemed to get slower each time. If we think about this, we realize that *Maple* must perform twice as many computations each time, so it stands to reason that, say, $g(1000)$ should be quicker than $g(2000)$. If we try to calculate something only moderately outrageous (when compared to 10^{12} or 10^{100} from above), we find the computation takes too long to be useful to us.

> $evalf(g(10^6))$

Warning, computation interrupted

In point of fact, when one of the authors ran the above computations on his home laptop, it caused the system to collapse under its own weight (so to speak) as all its memory was eaten up (and hence the aborted computation). It is for this reason that the computations were interrupted.[1]

Remember here, that the reason we are doing this is to get a quick feel for the convergence of a system. There may very well be times when we should be happy to let a calculation run for perhaps even weeks or months if the value of the computation is sufficiently important. Computations of π to exceptionally large precision have been run that have taken months to perform. This is *not* such a case, however. It is important to remember our goals in order to ascertain how long a wait is too long.

The question now arises, why was the function f so fast? Even calculating a thousand billion (10^{12}) terms of the series was nearly instantaneous. The answer is found when we try to see what *Maple* thinks an arbitrary partial sum should look like

```
> f(N); g(N)
```

$$-\Psi\left(1, N+1\right) + \frac{1}{6}\,\pi^2$$

$$\sum_{k=1}^{N}\left(\frac{1}{k^\pi}\right)$$

We see here that our first function, $f(N) = \sum_{k=1}^{N} k^{-2}$, has a clear formula, whereas for our second function $g(N) = \sum_{k=1}^{N} k^{-\pi}$, *Maple* has just given us back the sum we gave it in the first place. We can infer from this that *Maple* cannot find a simple formula for this partial sum. It does, however, know a lot about the ζ-function and how to compute it quickly, as we saw when we asked for $g(\infty)$ earlier. This is in keeping with the behavior we seen thus far. Surely a single formula will be much faster to calculate than N terms of a sum.

Digging a little deeper we see more of the same idea.

```
> f(10); g(10)
```

$$\frac{1968329}{1270080}$$

$$1 + \frac{1}{2^\pi} + \frac{1}{3^\pi} + \frac{1}{4^\pi} + \frac{1}{5^\pi} + \frac{1}{6^\pi} + \frac{1}{7^\pi} + \frac{1}{8^\pi} + \frac{1}{9^\pi} + \frac{1}{10^\pi}$$

```
> f(12); g(12)
```

$$\frac{240505109}{153679680}$$

$$1 + \frac{1}{2^\pi} + \frac{1}{3^\pi} + \frac{1}{4^\pi} + \frac{1}{5^\pi} + \frac{1}{6^\pi} + \frac{1}{7^\pi} + \frac{1}{8^\pi} + \frac{1}{9^\pi} + \frac{1}{10^\pi} + \frac{1}{11^\pi} + \frac{1}{12^\pi}$$

```
> f(10^6); g(10^6)
```

$$-\Psi\left(1, 1000001\right) + \frac{1}{6}\,\pi^2$$

[Length of output exceeds limit of 1000000]

---

[1] For the more computer-knowledgeable readers, the computation caused excessive paging and large amounts of virtual memory to be assigned, as well as much of the current memory to be swapped out to disk. Even after aborting the calculation—no small feat on a system that is chronically unresponsive due to all the paging—the system would only page contents back into memory on demand; as such it was quite some time before ordinary use of the computer didn't result in a burst of disk activity, and corresponding delay.

What is particularly interesting here is that we were successfully able to calculate $g(10^6)$, and very quickly at that. Yet, when we tried (almost) the same calculation above it took a long time and nearly crashed the author's computer. It would seem, then, that the problem is not in the symbolic calculation of the partial sum, as much as in the evaluation of it to floating point.

If we modify our approach only slightly, we are able to calculate the millionth partial sum without the issues we saw earlier. We have already tried first computing $g(10^6)$ symbolically and then evaluating its decimal representation, and we did not fare well. Instead we calculate the decimal representation at each step, and add these decimals together as we go. We achieve this using the **add** function.

> $add(evalf(k^{-\,\mathrm{Pi}}), k = 1..10^6)$

$$1.176241737$$

This code completed in just under three minutes on the author's computer, and caused no discernible stress on the system.

It is very tempting to view the above numerical computation with some contempt. Decimal approximations are just that, approximations. Worse still, the above computation is adding approximations to approximations at every step, and doing it a million times. What's more, we are using a CAS, the whole point of which is to allow the computer to perform exact symbolic calculations. Surely it would seem reasonable, even preferable, to perform all computations symbolically, and to obtain decimal approximations from these exact mathematical constructs. This was certainly the opinion of one of the authors before the commencement of this book.

Of course, what we have seen in this previous example is a case where such an approach is simply not feasible. Sometimes we have to work purely numerically, and the reality is that numeric computations aren't as bad as all that. We should, of course, be quite aware of the fact that numerical approximation can introduce errors in our calculations, and be on the lookout for them, but this should not and does not detract from the usefulness of symbolic computation.

We look at this phenomenon again with a different example. This time we take the infinite product

$$\prod_{k=3}^{\infty} \cos\left(\frac{\pi}{k}\right)$$

If we sketch a picture of the cosine graph we can quickly see we are multiplying a bunch of numbers together, all of which are positive, and all of which are less than 1, but that are getting closer and closer to 1 as we progress. If we ask *Maple* for the answer, we are told it is 0.

> $P := Product\left(\cos\left(\dfrac{\mathrm{Pi}}{k}\right), k = 3..infinity\right)$

$$P := \prod_{k=3}^{\infty} \cos\left(\frac{\pi}{k}\right)$$

> $value(P)$

$$0$$

This is not entirely problematic, because we are talking about the limit of partial products. So it could well be the case that even though the number we multiply by at each stage is increasing, it may not be increasing quickly enough, and the whole process might still end up decreasing to a limit of 0. It is, however, incorrect. If we try a floating point evaluation instead, *Maple* gives us a different answer.

> $evalf(P)$

$$0.1149420449$$

Clearly, at least one of these answers must be incorrect. Note that we've been careful here to define P as an inert product, so that we can separately perform the **value** and **evalf** commands on the same product.

It is at this point that we would like to calculate some partial products, just as we did previously, and hope to see some sort of convergence. We find, unfortunately, the same problem as we had with our $k^{-\pi}$ sum. The large partial products just take too long to calculate.

> **for** i **from** 1 **to** 20 **do**$'$ $g'(2^i) = evalf(g(2^i))$ **od**

$$f(4) = 0.3535533905$$
$$f(8) = 0.2061903868$$
$$f(16) = 0.1550922784$$
$$f(32) = 0.1337985136$$
$$f(64) = 0.1240823134$$
$$f(128) = 0.1194421627$$
$$f(256) = 0.1171748402$$
$$f(512) = 0.1160541566$$
$$f(1024) = 0.1154970332$$
$$f(2048) = 0.1152192732$$
$$f(4096) = 0.1150805931$$
$$f(8192) = 0.1150113030$$
$$f(16384) = 0.1149766706$$
$$f(32768) = 0.1149593584$$

```
Warning, computation interrupted
```

It certainly looks, above, as if the answer of 0.1149420449 is probably the correct one, but we don't have the best data. If we adopt a numeric approach, and use a more naive loop, we can actually see the convergence much better

> $total := 1;$
> **for** i **from** 3 **to** 10^4 **do** $total := total \cdot \cos\left(\dfrac{\mathrm{Pi}}{i}\right)$ **od**

$$total := 1$$
$$total := 0.5000000000$$
$$total := 0.3535533905$$
$$total := 0.2860307013$$
$$\vdots$$
$$total := 0.1149987860$$
$$total := 0.1149987803$$

The actual output in *Maple* is far larger than we have shown above, and vertical dots have been placed in this book to shorten the space. In total the command takes around ten minutes to complete (and most of us will probably have stopped the computation before then). What it does, however, is give us a rather good intuitive feel for the way

the product changes as more terms are introduced, and the approximate rate at which it is changing.

We see fast change to start with, but as time wears on we see less and less change until only the last couple of decimal places seem to be changing at all. The rate of change is clearly decelerating, and when we recall that the value of $\cos(\pi/i) \to 1$ as i increases, it should seem very unlikely indeed that this will keep going all the way down to 0.

We might, now, try accelerating the same approach by calculating partial products in steps of 100 instead of in steps of 1.

> $total := 1;$
  for i **from** 0 **to** 10^4 **do**
    $total := total \cdot mul\left(evalf\left(\cos\left(\dfrac{\text{Pi}}{100 \cdot i + 3 + k}\right)\right), k = 0..99\right)$
  od

$$total := 1$$
$$total := 0.1206114938$$
$$total := 0.1177775543$$
$$total := 0.1168325277$$
$$\vdots$$
$$total := 0.1149436767$$
$$total := 0.1149436767$$
$$total := 0.1149436767$$

Ultimately, the same principle still applies, and this is really just a refinement of the technique. It is interesting that the last three entries are all the same number. This is probably best explained by numeric rounding. The terms at this stage are very likely so close to 1 by this stage, that to ten digits of accuracy, *Maple* is probably just calculating them as 1.

> $evalf\left(\cos\left(\dfrac{\text{Pi}}{10^6}\right)\right); evalf[20]\left(\cos\left(\dfrac{\text{Pi}}{10^6}\right)\right)$

$$1.000000000$$
$$0.99999999999506519780$$

We could increase the numerical accuracy and perform the computation again, but by now we should be fairly confident that the answer of 0 is incorrect, and that the answer of 0.1149420449 is correct. Now we should try something more analytical.

If we know a little about infinite products, we might try to check turning the product into a sum using the identity

$$\log\left(\prod_{k=a}^{\infty} f(k)\right) = \sum_{k=a}^{\infty} \log(f(k))$$

and so if the infinite product P really is zero, then the logarithm should be undefined, and hence the sum of the logarithms should diverge.

```
> sum(log(cos(Pi/k)), k = 3..infinity)
```

$$\sum_{k=3}^{\infty} \log\left(\cos\left(\frac{\pi}{k}\right)\right)$$

```
> evalf(%)
```

$$-2.163327235$$

```
> exp(%)
```

$$0.1149420449$$

We should be even more confident now in the answer of 0.1149420449. Working with the above sum (on paper or perhaps even with *Maple*) it is possible to calculate upper and lower bounds for the product, and to show conclusively that 0 is incorrect. We do not do so here, as we have strayed from both the discussion of numeric versus symbolic computation, as well as the topic of **for** loops. Further analysis of this product is left as an exercise for the reader.

1.2.3 Decision Structures

There are times when, as part of a computation we are performing, we need to make some sort of decision in order to proceed. To illustrate this idea, and how we implement decisions in *Maple* we look at the following problem.

Let us say we have a natural number n. Recall that if n can be divided by another natural number a evenly—that is, n/a is a natural number—we use the notation $a|n$ and say that a *divides* n or that a is a *divisor* of n. Furthermore, if $a|n$ then $n = ka$ for some $k \in \mathbb{N}$ and so, recalling modular arithmetic, $n \equiv 0 \mod a$.

The problem we now try to solve now with *Maple* is to find all the divisors of a number. To begin with, it is helpful to know that *Maple* can perform modular arithmetic using the **mod** operator. Simply put, entering $a \,\mathbf{mod}\, b$ will calculate the modulus of a (modulo b).

```
> 3 mod 4; 9 mod 7; 10 mod 5;
```

$$3$$
$$2$$
$$0$$

We start with a straightforward approach. Given our number n, whatever it happens to be, we recognize that no number bigger than n can possibly be a divisor of n, so we check every single number a less than n and see if $a|n$. This is just the thing for which a loop would be good. We'll start small with $n = 6$

```
> n := 6
```

$$n := 6$$

```
> for a from 1 to n do
      n
      ─
      a
  od
```

$$6$$

$$3$$

$$2$$

$$\frac{3}{2}$$

$$\frac{6}{5}$$

$$1$$

From this we can see that the divisors of 6 are $6, 3, 2, 1$. It would be nice if we could have *Maple* only show the divisors, and not the fractions that are clearly not divisors. To do this we have *Maple* make a decision using an **if** command.

Now, we need *Maple* to recognize which of the calculations are fractions, and which are whole numbers, but unfortunately we currently have no idea how we might do this. The answer is to use the modular arithmetic calculations from above, remembering that if $n/a \in \mathbb{N}$ then $n \equiv 0 \mod a$. This is something we already know how to express in *Maple*. In order to see if, say, 3 was a divisor of our n then we could issue the command

```
> if n mod 3 = 0 then  n
                       ─  fi
                       3
```

$$2$$

The above code should be read as "If n is equal to 0 modulo 3 then calculate $n/3$," and because n—which happens to be 6 at the moment—is most certainly equivalent to 0 modulo 3, *Maple* has correctly gone on to calculate $6/3 = 2$. Note that if n were some other number such that $n \not\equiv 0 \mod 3$ then *Maple* would have performed no calculation at all.

The code between the **if** and the **then** is a criterion for the code after the **then** to be carried out. If the criterion is met, the code is carried out, and if the criterion is not met, the code is not carried out. The **fi** simply tells *Maple* where the **if** statement ends. Note that in some cases (which we look at later), we may add a third piece of code to be carried out in the case that the criterion is not met. For the moment, let's look at an example with a single piece of code that does not execute

```
> if n mod 4 = 0 then  n
                       ─  fi
                       4
```

Of course, we don't want to have to type all of these in individually, so we now incorporate these decisions into our loop.

```
> for a from 1 to n do
      if n mod a = 0 then print( n/a ) fi
  od
```

$$6$$

$$3$$

$$2$$

$$1$$

In fact, it is usually quite rare that we use a decision manually when using *Maple*, where we can make these decisions for ourselves. It is much more common that a decision

would be part of a function or a loop where we do not have the luxury of being certain what values our variables contain.

We were forced to use the **print** function above, which just tells *Maple* to output an expression just as it would if we typed that expression in ourselves manually. This was required here because we had an **if** statement buried inside a **for** statement. We look more at this sort of thing later on, but for the time being be aware that once these things become buried inside each other (often referred to as "nesting") *Maple* will not produce output unless we specifically ask for it.

The astute reader may have noticed that each of our divisors—which were all of the form n/a—were, themselves, also values of a at some point in the loop. To see this a little more clearly, we modify our loop to print both n/a and a on the same line.

> **for** a **from** 1 **to** n **do**
   if $n \bmod a = 0$ **then** $print\left(a, \dfrac{n}{a}\right)$ **fi**
  od

$$1, 6$$
$$2, 3$$
$$3, 2$$
$$6, 1$$

We have, essentially, found each divisor twice. We checked every integer less than n (6 in our previous examples) to see if it was a divisor, but we need only check the numbers less than or equal to $\sqrt{n}$ (≈ 2.449 in our previous examples). We can see, by inspection above, that after our second repetition we had already found all the divisors of 6.

This is true in general because for any divisor a of a number, n say, we have a codivisor b such that $a \cdot b = n$. This is simply what it means to be a divisor. Now suppose that $a \leq \sqrt{n}$. It must, necessarily, be the case that $b \geq \sqrt{n}$, because

$$b \leq \sqrt{k} \implies ab < \sqrt{n}\sqrt{n} = n$$

which would contradict a and b being codivisors. Similarly, if $a \geq \sqrt{n}$ then $b \leq \sqrt{n}$. It follows, then, that once we've found all the divisors less than or equal to $\sqrt{n}$ then we have also found all the larger divisors in the codivisors.

We can use this fact to modify our loop and make it a little more efficient. There is a small problem here, however, because the square root of a number is not always a natural number. In order to avoid this problem we need to use the **while** clause of a loop, instead of the usual **to**. The loop will keep calculating (and the variable a will keep incrementing) until the condition after the **while** is met. See Exercise 15 and *Maple*'s help files for more information.

> **for** a **from** 1 **while** $is\left(a \leq \mathrm{sqrt}(n)\right)$ **do**
   if $n \bmod a = 0$ **then** $print\left(a, \dfrac{n}{a}\right)$ **fi**
  od

$$1, 6$$
$$2, 3$$

Such an improvement may not seem particularly worthwhile for the case of calculating the divisors of 6. However, for calculating the divisors of a large number, 1,000,000 say, then our modified loop would be performing only 1000 decisions, instead

of 1,000,000, which is a more significant difference. We now calculate the divisors of 999 (because the output is more manageable), requiring only 31 iterations of our loop.

> $n := 999$

$$n := 999$$

> **for** a **from** 1 **while** $is\,(a \leq \text{sqrt}(n))$ **do**
>   **if** $n \bmod a = 0$ **then** $print\left(a, \dfrac{n}{a}\right)$ **fi**
>   **od**

$$1, 999$$
$$3, 333$$
$$9, 111$$
$$27, 37$$

Note that due to a glitch in *Maple* at the time of writing, we needed to use the **is** function to evaluate the truth of the clause $a \leq \sqrt{n}$ in the loop. If we try the more readable loop without this function, *Maple* complains that it cannot ascertain whether a is less than $\sqrt{n}$.

> **for** a **from** 1 **while** $a \leq \text{sqrt}(n)$ **do**
>   **if** $n \bmod a = 0$ **then** $print\left(a, \dfrac{n}{a}\right)$ **fi**
>   **od**

```
Error, cannot determine if this expression is true or false: 4 <=
                        3*111^(1/2)
```

It might not surprise the reader to discover that *Maple* in fact has a built-in function that will calculate the divisors of a number. This function is the **divisors** function, in the numtheory package.

> $numtheory[divisors](6)$

$$\{1, 2, 3, 6\}$$

The reader may now find cause to wonder why we went through the above rigmarole of loops and decisions to calculate the divisors of a number, when we could have just used this **divisors** function right from the start. This is not an unfair question, and there are two primary reasons for this approach. The immediately obvious answer is we wished to introduce the reader to *Maple*'s loop and decision structures, and the divisor calculation seemed a natural example that lent itself nicely to demonstrating these structures. Furthermore we have demonstrated the ability to use the more basic building blocks of *Maple* to perform mathematics and solve problems when we don't know a more direct way of having *Maple* perform the calculation. By taking this longer route, we perhaps also allow ourselves to learn a little more about the mathematics than we might have if we had just asked for an answer directly.

This is illustrative of an approach that is used repeatedly by the authors in this book. We repeatedly construct mathematics and calculations from first (or, at least, earlier) principles before introducing the *Maple* command that would perform the calculation directly. The reader should be sure to understand both the *Maple* and the mathematics involved in these constructions, but should feel free to use the "direct" methods once they have been introduced. In fact, several exercises require the use of these direct methods seemingly blindly, and it is expected that knowledge of the underlying concepts will allow the reader to confirm the answers that *Maple* provides, even though the reader may not have the tools to produce the answer without the aid of the computer.

We now return to our discussion of divisors and of *Maple* decision structures, exploring both a little further. We introduce the notion of *proper divisors* and *perfect numbers*. When considering the divisors of some number, n say, it should be clear that n is always a divisor of itself. The set of proper divisors of n are simply all of its divisors, except for n itself. So the proper divisors of 6 are $\{1, 2, 3\}$. If we add the proper divisors together, we see that $3 + 2 + 1 = 6$, and that the sum of the proper divisors of 6 is 6 itself. This is an example of a *perfect number*, which is any number n whose proper divisors sum to the value of n.

We use *Maple* to calculate whether some numbers are perfect, starting with 6. We need to use a decision, but this time we have two options: either the number is perfect, or it is not. We want an answer in either case, so we need to use the **else** keyword of a decision. First we must find the proper divisors and add them.

```
> n := 6
```
$$n := 6$$

```
> div := numtheory[divisors](n)
```
$$div := \{1, 2, 3, 6\}$$

```
> N := add(k, k in div) − n
```
$$N := 6$$

Recall that we changed the value of n earlier, so we needed to set it back to 6. Also, we have used a slight variant of the **add** function. See Exercise 6 for a similar variant of the **seq** command. Finally, note that we only wanted to sum the *proper* divisors, but the **add** command variant we used will sum every divisor, so we needed to subtract out n to compensate for this unwanted addition.

We can clearly see that 6 is perfect (which we already knew), but we issue the **if** command regardless, to demonstrate the **else** keyword. Decisions with this keyword work almost identically to the decisions we have been using thus far. If the condition is met, then the code after the **then** keyword is executed. The **else** keyword signifies the end of the aforementioned code, much as the **fi** did previously. We now have a second block of code between the **then** and the **fi**, which is executed if the condition is not met.

```
> if N = n then n is perfect else n is imperfect fi
```
$$6 \; is \; perfect$$

Let's see if there are any perfect numbers less than 6. Currently if we want to know if a number is perfect, we have two or three calculations to perform; if we were to start at $n = 1$ and test up until $n = 6$ we would end up typing out ten commands, all of which would be a little tedious. One of the advantages of **for** loops is that they allow us to repeat not just single, but multiple calculations within the loop, so we use this technique instead.

> **for** n **from** 1 **to** 6 **do**
> $div := numtheory[divisors](n);$
> $N := add(k, k \textbf{ in } div) - n;$
>   **if** $N = n$ **then** $print(n \textit{ is perfect})$ **fi**;
> **od**

$$div := \{1\}$$
$$N := 0$$
$$div := \{1, 2\}$$
$$N := 1$$
$$div := \{1, 3\}$$
$$N := 1$$
$$div := \{1, 2, 4\}$$
$$N := 3$$
$$div := \{1, 5\}$$
$$N := 1$$
$$div := \{1\}$$
$$N := 6$$
$$6 \textit{ is perfect}$$

This is rather messy, but we can see the *div* and N variables being created at each repetition of the loop, and can also see (after perhaps a bit of thought) that 6 is the first perfect number. It would be better if we could suppress the output from the first two calculations, and only see the output of the **if**. In fact, we can do this, but it is in a round-about manner. Loops are an all-or-nothing affair; either we see every calculation within them, or we see none of them. We must completely suppress the output from the entire **for** loop (by ending it with a full colon after the **od**) and realize that the **print** command will always produce output, regardless of whether the command has been suppressed. An alternate solution is to create a procedure (see Section 1.2.4) to house the loop.

Let us now find all the perfect numbers less than or equal to 10,000.

> **for** n **from** 1 **to** 10000 **do**
> $div := numtheory[divisors](n);$
> $N := add(k, k \textbf{ in } div) - n;$
>   **if** $N = n$ **then** $print(n)$ **fi**;
> **od** :

$$6$$
$$28$$
$$496$$
$$8128$$

We see that there are only four perfect numbers less than 10,000.

Finally, we apply these same concepts directly to a new problem. Suppose we have a natural number, n, that is not perfect. Let m be the sum of the proper divisors of n. It is possible (but not necessarily likely) that n also happens to be equal to the sum of the proper divisors of m. If this happens, then n, m are called a pair of *amicable* numbers.

Our problem is to find all the amicable pairs where at least one of the pair is less than 10,000.

Applying some thought to the problem, we should realize that for any n there can only be one possibility for its amicable partner m, and that possibility is the sum of the proper divisors of n. If n is a perfect number, then our candidate for its amicable partner is itself, but remember we stipulated initially that n was not perfect, so we must make sure to somehow exclude perfect numbers in our computation.

Our approach, then, is clear. Given n, we calculate m directly, as well as the sum of the divisors of m. We then check to see if the sum of m's divisors is equal to n, and we also check that n is not perfect. The number if perfect so long as both of these checks are true. That is, n is perfect if and only if the sum of m's divisors is equal to n and n is not perfect. This leaves us with two criteria, which we are able to express in a single **if** command using the **and** operator. We implement this approach using a loop similar to that used above.

```
> for n from 1 to 10000 do
    ndiv := numtheory[divisors](n);
    m := add(k, k in ndiv) − n;
    mdiv := numtheory[divisors](m);
    N := add(k, k in mdiv) − m;
    if n ≠ m and n = N then print(n, m) fi;
  od :
```

$$220, 284$$
$$284, 220$$
$$1184, 1210$$
$$1210, 1184$$
$$2620, 2924$$
$$2924, 2620$$
$$5020, 5564$$
$$5564, 5020$$
$$6232, 6368$$
$$6368, 6232$$

Notice that we have found each pair twice, once starting with the smaller of the two, and once starting with the larger. We can see then that there are only five amicable pairs less than 10,000. It is left as an exercise for the reader to modify the loop so that it only prints each pair once.

1.2.4 Procedures

When we created our own functions earlier, we used the very handy arrow notation ($\rightarrow$). Using this method is very convenient, and allows us a lot of power when using *Maple*, but it doesn't take long to find that it does have some limitations.

For example, an earlier calculation required three commands to have *Maple* ascertain whether a number was perfect. It would be nice to have a function which performed the calculation for us, instead of having to type those three lines out every time we wish to see if a number were perfect or not. Unfortunately, the arrow notation only works for single line *Maple* commands, and so cannot help us here.

If we are clever, we can reduce the three commands into a single **if** statement. To do this, we perform the computation of the divisor set and the sum of these divisors in one

fell swoop within the condition of the **if** statement, as follows. Note that $n = 10{,}001$ after the most recent loop.

> **if** $n = add(k, k \textbf{ in } numtheory[divisors](n)) - n$ **then**
> $print(n \textit{ is perfect})$
> **else**
> $print(n \textit{ is imperfect})$
> **fi**

$$10001 \textit{ is imperfect}$$

This works, because we only need to use each of these intermediate computations a single time to establish an answer. All the same computations are performed here, but they are used in place and are never assigned to variable names and thus cannot be reused at a later date (without performing the computation again). Although we have broken this command up over several lines for ease of reading, this is nonetheless a single command.

Our desire now would be to turn this single command into a function using the arrow notation. It is unfortunate then to discover that we cannot use an **if** statement to make decisions in a function created using the arrow method.[2]

> $isperfect \quad := \quad n \quad \rightarrow \quad \textbf{if } n \quad = \quad add(k, k \textbf{ in } numtheory[divisors](n)) \quad - $
> $n \textbf{ then } print(n \textit{ is perfect}) \textbf{ else } print(n \textit{ is imperfect}) \textbf{ fi}$

$$\text{Error, invalid arrow procedure}$$

As it happens, the functions we have created up till now with the $\rightarrow$ operator have been special cases of a more general *Maple* construction known as a procedure. A procedure is, as its name suggests, a series of steps to be followed and behaves like the functions we have used and created so far in that it takes input and produces output. A procedure, however, allows multiple calculations to be performed as part of its processing, much as our loops and decisions above did.

We can see this if we use the **lprint** command with our already familiar arrow notation functions. We use a simple function that calculates the square of the input.

> $x \rightarrow x^2; lprint(\%)$

$$x \rightarrow x^2$$
$$\texttt{proc (x) options operator, arrow; x\^{}2 end proc}$$

A procedure, at its most basic, takes the form **proc**(*input*) *commands* **end proc**, although we may use **end** as a shorthand version of **end proc**. The commands, as do any *Maple* commands, must have semicolons between them if there are more than one. If we ignore the options in the example above, we see precisely that form. The result of the last calculation performed in a procedure is that which *Maple* outputs when the procedure is used.

In order to create the perfect number query function which we could not do earlier, we can use a procedure as follows.

---

[2] This is not absolutely true. It is possible to include an **if** statement within an arrow procedure when using the text-based (sometimes called 1D) *Maple* input scheme, but even in this case it is far from ideal.

```
>  isperfect := proc(n)
      if n = add(k, k in numtheory[divisors](n)) − n then
        print(n is perfect)
      else
        print(n is imperfect)
      fi
   end :
```

```
>  isperfect(6), isperfect(10), isperfect(12), isperfect(28)
          6  is  perfect, 10  is  imperfect, 12  is  imperfect, 28  is  imperfect
```

Our function works fairly well, although it gives some questionable results when we give it input that is not a natural number.

```
>  isperfect(−6), isperfect(x), isperfect([1, 2])
               −6  is  imperfect, x  is  imperfect, [1, 2]  is  imperfect
```

```
>  isperfect(1 + I)
Error, (in isperfect) invalid input: numtheory:-divisors expects its
   1st argument, n, to be of type Or(integer, Not(constant)), but
                         received 1+I
```

If we experiment a little we will probably find other things it doesn't handle very well. Nonetheless the procedure most certainly calculates whether a natural number is perfect.

Our procedures may consist of several lines and may even have variable assignment within them. However, if we assign variables, then we must tell *Maple* whether the variables are **local** to the procedure, or **global** to the worksheet. For more details on what these mean, see Exercise 16. We are only concerned with **local** procedure variables, which exist separately from any variables (even those with the same name) outside the procedure. Note that if we do not specify **local** or **global** for any variable inside a procedure, then *Maple* will report an error, and will assume the variable is local.

We rewrite our *isperfect* function using our earlier, and easier to follow, three-line calculation. Furthermore, we make some effort to ensure that the procedure will only work if we use a natural number as its input.

```
>  isperfect := proc(n :: posint)
      local divs, N;
      divs := numtheory[divisors](n);
      N := add(k, k in divs) − n;
      if n = N then
        true
      else
        false
      fi
   end :
```

```
>  isperfect(6), isperfect(10), isperfect(12), isperfect(28)
                         true, false, false, true
```

```
>  isperfect(−6);
   isperfect(x);
   isperfect([1, 2])
```

```
Error, invalid input: isperfect expects its 1st argument, n, to be
                    of type posint, but received -6
Error, invalid input: isperfect expects its 1st argument, n, to be
                    of type posint, but received x
Error, invalid input: isperfect expects its 1st argument, n, to be
                    of type posint, but received [1, 2]
```

First notice we have added the text *::posint* to the variable name. This is our way of telling *Maple* that we want the variable n in the procedure to be a positive integer. Recall that we used the same notation with the **assuming** keyword in Section 1.2.1. See the help file (**?types**) for more information on the different types that a variable might be. If we give *isperfect* anything other than a positive integer for the input parameter, we are given an error message as we saw.

Second, notice that we specified the variables *divs* and N as local to the procedure. It is very important that the **local** declaration ends with a semicolon. In fact any additional declarations (e.g., **description**, **global**, **option**) must end with a semicolon. See **?procedures** for more details on procedures in general and declarations in particular.

Finally, note that we changed the procedure ever so slightly to return either *true* or *false* instead of the previous *n is perfect*. There are two reasons for this. For one, *n is perfect* is actually an abuse of *Maple* notation, and although it looks quite good to our eye, to *Maple* it is actually a product of three unknown variables: $n \times is \times imperfect$. Perhaps more importantly, however, using *true* and *false* results in our procedure behaving properly with decision constructions, as the condition in a decision must evaluate to either *true* or *false*.

```
>  if isperfect(6) then print(6 is perfect) else print(6 is imperfect)
```
$$6 \;\; is \;\; perfect$$

What is important to notice above is that our *isperfect* function is able to be used as the condition to the **if** statement. The command above can be read as "if 6 is perfect then print '6 is perfect' otherwise print '6 is imperfect'."

Our loop to find the perfect numbers less than 10,000 can now become much more succinct, yet remain clear in its purpose.

```
>  for n from 1 to 10000 do
       if isperfect(n) then print(n) fi
   od
```
$$6$$
$$28$$
$$496$$
$$8128$$

Note that we didn't even need to suppress the output.

1.2.5 Nesting

We have looked at loops and decisions in the previous sections. Inside a loop, or a decision, we may have any valid *Maple* code, and even multiple valid *Maple* commands.

We have even seen that we may have a decision inside of a loop. One may ask if a loop may be placed inside a loop, or if a decision may be placed inside a decision, and the answer is yes they can. Doing such a thing is called *nesting*.

To illustrate this idea, let's look even more at our divisors and perfect numbers. Observe that if a number n is not perfect, then the sum of the proper divisors is either strictly greater, or strictly less than n itself. If the sum of the proper divisors is less than n we say the number is *deficient*, and if the sum of the proper divisors is greater than n then we say that n is *abundant*. We now have three mutually exclusive options for any natural number.

This idea lends itself nicely to a nested decision. We must first make a decision to see if the number is perfect, and if it is not then we must make a second decision as to whether it is abundant or deficient. We would implement this in *Maple* as follows (extending one of our previous procedures).

```
> classify := proc(n :: posint)
     local N;
     N := add(k, k in numtheory[divisors](n)) − n;
     if N = n then
       perfect
     else
       if N < n then
         deficient
       else
         abundant
       fi
     fi
  end :
```

```
> for n from 6 to 12 do n, classify(n) od
```

$$6, perfect$$
$$7, deficient$$
$$8, deficient$$
$$9, deficient$$
$$10, deficient$$
$$11, deficient$$
$$12, abundant$$

It is quite important here to notice the **if** within the **if** above; specifically it is within the **else** portion of the containing **if**. Also note that we need to use the sum of the proper divisors several times, so we are forced to store this value in a variable.

There is no necessary requirement that the innermost decision be within the **else** portion of the outermost decision, and furthermore we may nest potentially any number of decisions.

In this particular case, however, all our conditions are mutually exclusive and as a direct consequence we may use the rather simpler-to-read **elif** command, which behaves more or less as both an **else** and an **if** together.

```
>  classify := proc(n :: posint)
      local N;
      N := add(k, k in numtheory[divisors](n)) − n;
      if N = n then
        perfect
      elif N < n then
        deficient
      else
        abundant
      fi
   end :
```

```
>  for n from 6 to 12 do n, classify(n) od
```

$$6, \textit{perfect}$$
$$7, \textit{deficient}$$
$$8, \textit{deficient}$$
$$9, \textit{deficient}$$
$$10, \textit{deficient}$$
$$11, \textit{deficient}$$
$$12, \textit{abundant}$$

It is important to note in this case *Maple* considers this to be a *single* decision structure, and so only one **fi** is needed at the end. A decision structure consists of an **if** followed by a condition and a **then** followed by any number of **elif**s (each of which has a condition and a **then**), and finally an option **else** and the **fi** to end it. Maple simply starts at the top, and tests each condition until it finds the first that matches—or the **else** clause if nothing else matches—and then performs the appropriate calculation.

Let us now, on a whim, find all the abundant numbers less than or equal to 100. For a bit of variety, we collect these into a list by starting with a null list, and adding elements one by one as we find them.

```
>  A := NULL
```

$$A :=$$

```
>  for n from 1 to 100 do
      if classify(n) = abundant then A := A, n fi
   od :
```

```
>  A
```

$$12, 18, 20, 24, 30, 36, 40, 42, 48, 54, 56, 60, 66, 70, 72, 78, 80, 84, 88, 90, 96, 100$$

We leave our discussion of the properties of numbers and their divisors there for now. Before we move to nested loops, we show one more example of a nested decision. This time we use a nested decision to decide in which quadrant a point in the real plan lies. For the sake of simplicity, we consider any point that lies on the x-axis to be in the upper half-plane, and any point on the y-axis to be in the right half-plane. This means, in particular, that the point $(0, 0)$ is considered to be in the first quadrant.

We construct a procedure that takes two inputs: x and y. For our decision, we observe that when $y \geq 0$ we are in either the first ($x \geq 0$) or second ($x < 0$) quadrant. Otherwise

$y < 0$ and we are in either the third ($x < 0$) or fourth ($x \geq 0$) quadrant. Our nested decisions use this logic.

```
>  quadrant := proc(x, y)
       if y ≥ 0 then
          if x ≥ 0 then
             first
          else
             second
          fi;
       else
          if x < 0 then
             third
          else
             fourth
          fi;
       fi
   end :
```

```
>  quadrant(1, 1); quadrant(1, -2); quadrant(-4, -3); quadrant(-7, 12);
```
$$first$$
$$fourth$$
$$third$$
$$second$$

We now move on to nested loops. As the nested decisions above should suggest, a nested loop is simply a loop inside another loop. As with decisions, we may nest any number of loops, but we concern ourselves initially with just a pair of nested loops.

If we think of a double sum $\sum_{i=1}^{N} \sum_{j=1}^{M} f(i, j)$ then

$$\sum_{i=1}^{N} \sum_{j=1}^{M} f(i, j) = \sum_{i=1}^{N} \left(\sum_{j=1}^{M} f(i, j) \right)$$
$$= \sum_{i=1}^{N} \left(f(i, 1) + \cdots + f(i, M) \right)$$
$$= f(1, 1) + \cdots + f(1, M) + \cdots + f(N, 1) + \cdots + f(N, M)$$

The dummy variable, i in this case, assumes a set of values $1, \ldots, N$ and for each of these another sum is to be computed. For this second sum, the dummy variable j also assumes a set of values $1, \ldots, M$ and the i value stays (temporarily) fixed. The final result is the pattern we see above. A nested loop will behave in a very similar way. In fact, we may calculate a double sum with a nested loop, as we do below.

```
>  S := 0
```
$$S := 0$$

```
> for i from 1 to 3 do
      for j from 1 to 3 do
        S := S + f(i, j)
      od
   od
```

```
> S
```
$$f(1,1) + f(1,2) + f(1,3) + f(2,1) + f(2,2) + f(2,3) + f(3,1) + f(3,2) + f(3,3)$$

For each (temporarily) fixed value of i, the entirety of the innermost loop is calculated. Of course, we could achieve this same result using only the **sum** function, but the above nicely demonstrates the behavior of a nested loop.

```
> sum(sum(f(i, j), j = 1..3), i = 1..3)
```
$$f(1,1) + f(1,2) + f(1,3) + f(2,1) + f(2,2) + f(2,3) + f(3,1) + f(3,2) + f(3,3)$$

As an example of three nested loops, we construct all possible truth values for three variables A, B, and C. We also show a variant of the **for** loop in which the variables are not incremented, but instead are taken from the elements of a list, by the use of the **in** keyword.

```
> for A in [true, false] do
      for B in [true, false] do
        for C in [true, false] do
          print(A, B, C)
        od
      od
   od
```

$$true, true, true$$
$$true, true, false$$
$$true, false, true$$
$$true, false, false$$
$$false, true, true$$
$$false, true, false$$
$$false, false, true$$
$$false, false, false$$

1.2.6 Recursive Functions

For some calculations, it is mathematically convenient to have a function use itself as part of its own calculation. Such a technique is called *recursion*. As a natural example of this we look at the Fibonacci numbers.

Recall that the Fibonacci numbers are given by the relation $f_n = f_{n-1} + f_{n-2}$. This is an example of a *recurrence relation*, which we look at in more detail in Section 1.3.3. Nonetheless, it should be clear that in order to calculate any Fibonacci number, we need to know the previous two. Each of these two numbers is, itself, a Fibonacci number, they therefore may be calculated in turn by knowing the prior numbers. In order to prevent forever looking backwards we need a starting point, or some known Fibonacci numbers, and so we also stipulate that $f_1 = f_2 = 1$.

We may implement this in *Maple* fairly simply with a procedure that uses decision and recursion. If f_1 or f_2 are asked for (these will be $f(1)$ or $f(2)$ because of the way functions and procedures work in *Maple*) then we return the value 1, and otherwise we will use the same procedure again to calculate the previous two Fibonacci numbers. In this way we will eventually work our way back to either f_1 or f_2.

```
> f := proc(n :: posint)
     description "Calculate the nth Fibonacci number";
     if n = 1 or n = 2 then
       1;
     else
       f(n − 1) + f(n − 2)
     fi
  end :
```

```
> seq(f(n), n = 1..20)
```
$$1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765$$

We have used an optional declaration within this procedure, the **description** declaration, which is, as its name should suggest, just a description of what the procedure does. This is purely for our own benefit as a reminder of what the procedure is written to do. We may query this directly with *Maple*'s **Describe** command.

```
> Describe(f)
```
```
                 # Calculate the nth Fibonacci number
                          f( n::posint )
```

To make the code a little more intuitive to read we may, if we wish, eschew the decision completely and simply write only the recursive part of the procedure. If we do this, we also need to tell *Maple* directly what $f(1)$ and $f(2)$ are supposed to be. This is a lot more in line with the way we handle recurrence relations on paper, and so should be intuitively more familiar.

```
> f := proc(n :: posint)
     description "Calculate the nth Fibonacci number";
     f(n − 1) + f(n − 2)
  end :
```

```
> f(1) := 1; f(2) := 1;
```
$$f(1) := 1$$
$$f(2) := 1$$

```
> seq(f(n), n = 1..20)
```
$$1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765$$

We may apply this recursive function definition approach with manually defined initial conditions to the arrow notation for functions. If we do this, we lose the input type checking we get from a procedure using :: *posint*, but this is a small price to pay. We also lose the possibility of the **remember** option (see Section 1.2.7, below), which may be more significant. For much of the time, however, such concerns are likely nonexistent (at least to begin with), and the arrow notation is simple and quick to write. Using it below there is no mistaking that we most certainly are calculating the Fibonacci numbers.

> $f := n \to f(n-1) + f(n-2); f(1) := 1; f(2) := 1$
$$f := n \to f(n-1) + f(n-2)$$
$$f(1) := 1$$
$$f(2) := 1$$

> $seq(f(n), n = 1..20)$
$1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765$

As it was with divisors, so it is with the Fibonacci numbers; *Maple* contains an inbuilt function for their direct calculation. The function is the **fibonacci** function and is contained within the combinat package.

> $seq(combinat[fibonacci](n), n = 1..20)$
$1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765$

Using this function we do not need to write our own Fibonacci computing functions, no matter how simple they are. The reader, nonetheless, should take pains to understand the concepts in the Fibonacci computations we constructed above, for the Fibonacci numbers are not the only recurrence relation, and *Maple* will most certainly not always be so forthcoming with inbuilt functions for our convenience.

1.2.7 Computation Time

The previous section gives rise to the question of how long a computation will take to perform. It may not be obvious from the computation of the early terms of the Fibonacci sequence, but the above recursive function we wrote in Section 1.2.6 is actually quite slow. The problem stems from the fact that it does not remember previous calculations. For instance, suppose we ask for $f(5)$, the 5th Fibonacci number. *Maple*—acting in accord with our instructions—will first compute $f(4)$ which involves computing $f(3)$ and $f(2)$, and computing $f(3)$ involves, in turn, computing $f(2)$ and $f(1)$. *Maple* performs each of these computations, including computing $f(2)$ twice. Fortunately, we specified the value of $f(2)$ directly, so the extra computation is quick, just a simple matter of recalling the stored value. This is more easily seen with the tree diagram shown in Figure 1.1.
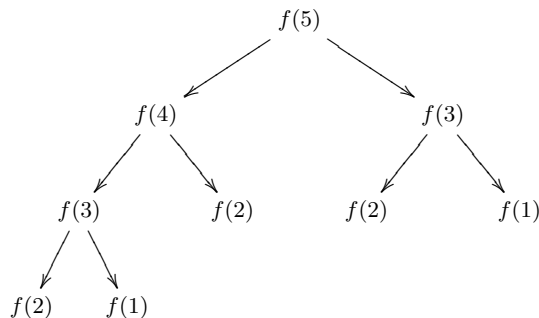
Fig. 1.1 Computation of the 5th Fibonacci number performed by the recursive f function.

In total, from a single request, *Maple* has performed 9 different computations (although 5 of these were simply looking up the specified initial values). If now we were to ask for $f(6)$, then our recursive function would calculate $f(5)$ in its entirety, as well as $f(4)$ also in its entirety for a total of 15 computations as shown in Figure 1.2. For large Fibonacci numbers, this recursive method will perform a staggering number of computations.
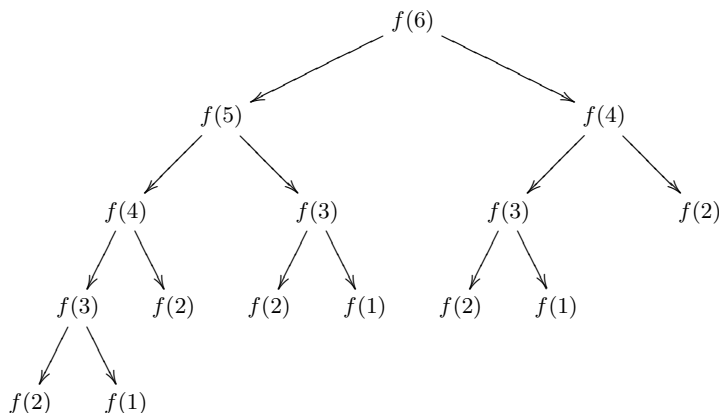

Fig. 1.2 Computation of the 6th Fibonacci number performed by the recursive f function.

This, however, turns out to be a small issue. We may tell *Maple* to remember previous computations of a procedure, so that if we ask for that computation again, *Maple* need only look up the answer from a table of remembered values, rather than performing the full computation. We do this by specifying the **remember** option when we write the function.

```
> f := proc(n :: posint)
      description "Calculate the nth Fibonacci number";
      option remember;
      f(n − 1) + f(n − 2)
   end :
```

```
> f(1) := 1; f(2) := 1;
```
$$f(1) := 1$$
$$f(2) := 1$$

```
> seq(f(n), n = 1..20)
```
$$1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765$$

This will greatly improve computation performance, at the expense of some extra memory usage. We show the tree diagram for the computation of $f(6)$ with the **remember** option in Figure 1.3. For this diagram, the left branch of any node represents the first computation. Once a node is computed for the first time, the value is remembered, and it need not be computed again.

We may measure the time taken for a computation and thereby see the effect of this growing number of computations. *Maple* provides a function called **time** to measure computation time. The time is measured in seconds, but is a measure of CPU time
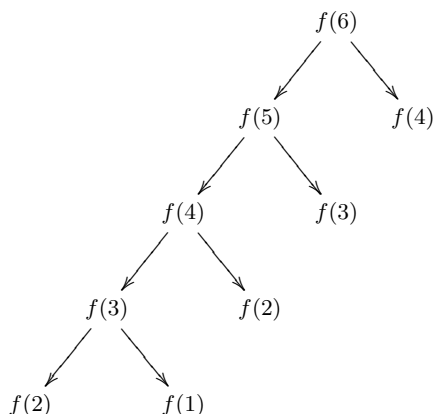
Fig. 1.3 Computations performed by the recursive f function with the **remember** option.

as opposed to actual elapsed time. The help documents aren't very specific, but it seems safe to presume that this measurement tries to take into account delays where no computation is being performed (idle time due to system multitasking, for example, ought not to count as CPU time). We do not worry ourselves with the technical specifics too much, and just consider this measurement as internally consistent and suitable as a comparison tool.

We start by measuring the time taken to calculate single Fibonacci numbers. We've already implemented the f function above as our faster (or so we claim) procedure, for Fibonacci number computation. To see the difference in speed, we need something to compare against. So we'll reimplement the arrow procedure, but call it F so we can test both variants together, and tell them apart.

> $F := n \rightarrow F(n-1) + F(n-2); F(1) := 1; F(2) := 1$

$$F := n \rightarrow F(n-1) + F(n-2)$$
$$F(1) := 1$$
$$F(2) := 1$$

> $time(F(30))$

$$1.003$$

The 30th Fibonacci number was chosen because it took about a second to compute, with the slow variant, on one of the authors computers. Anything smaller was a little too quick to be useful for illustrating the concepts illustrated in this section. The reader, when trying to replicate the above, might very well find that even this number is a little too quick if they are using a computer that is faster than the author's, which is likely. Some trial and error may be needed to find a Fibonacci number that takes, approximately, a second to calculate.

Notice with the above example that we did not see the output of the F function. We only saw the time it took to complete. We are confident that this function is correct, thus this is not such a problem. Nonetheless if we wish to see both the function output and the time taken in its computation we need to do something different. Fortunately **time** has an alternate usage where, if we give it no parameters, it reports back the current time (total elapsed CPU time for the worksheet) instead of the elapsed time for a single command. We simply record the current time before executing our procedure,

and record the time again once it is completed, and subtract the two. This technique is also given by the help files for the **time** function.

> $st := time(); \mathit{fib}_{30} = F(30); time() - st;$
$$st := 188.953$$
$$\mathit{fib}_{30} = 832040$$
$$0.987$$

This took slightly less time than the first execution. The execution time for a single command will always vary a little bit each time the command is executed. The important thing is that the times are always very close, "in the ballpark" if you like. In this case, the function takes very close to 1 second to compute. Let's see how long it takes for the 31st–35th Fibonacci numbers to compute (individually). Because we're more interested in the execution times than the Fibonacci numbers themselves for this discussion, we go back to the first use of the **time** function.

> $seq(time(F(k)), k = 31..35)$
$$1.616, 2.620, 4.225, 6.805, 11.010$$

Calculating $F(31)$ took more than half as long again as $F(30)$, and $F(32)$ took more than twice the computation time of $F(30)$. What is striking is that these times also exhibit a Fibonacci-like relationship.

$$time(F(32)) \approx time(F(30)) + time(F(31))$$
$$time(F(33)) \approx time(F(31)) + time(F(32))$$
$$\vdots$$

This is in keeping with, and explained by, our earlier observations. The computation for, say, $F(32)$ would involve calculating both $F(31)$ and $F(30)$ in their entirety. So it stands to reason that the computation times should sum in this Fibonacci-like way. We should expect, then, that $F(36)$ should take something in the vicinity of 18 seconds to compute.

> $time(F(36))$
$$17.653$$

Let us look at the faster variant now. This variant only ever calculates a previous Fibonacci number once, so if we calculate a Fibonacci number with it, then it will calculate every previous Fibonacci number only once as part of the computation. We should expect, then, that the speed should be roughly linear with the position of the Fibonacci number to be computed. That is, if it takes, say, 1 second to calculate the nth Fibonacci number, then we would expect around 2 seconds to calculate the $(2n)$th Fibonacci number.

> $seq(time(f(k)), k = 30..35)$
$$0., 0., 0., 0., 0., 0.$$

The function is clearly an improvement, however, the above does not really give us very much more information than that. Let's try something bigger

> $time(f(2000))$
$$0.010$$

That is exceptionally quick. Unfortunately, due of a limitation of this style of function, it is impractical to find a value that takes approximately a second to calculate.

Nonetheless, we can test our earlier expectation, that $f(2n)$ should take approximately twice as long as $f(n)$. In this case we expect $f(4000)$ to take approximately 0.02 seconds.

> $time(f(4000))$

$$0.013$$

This is not quite what we expected. If we think a little, however, about the consequences of remembering previous computations we should actually find the above result is not so surprising. When we calculated $f(2000)$, above, *Maple* saved, and remembered, the values of the first 2000 Fibonacci numbers. So when we then went on to calculate $f(4000)$, the first 2000 Fibonacci numbers would not have needed to be calculated again and so only 2000 more computations $(f(2001) - -f(4000))$ needed to be performed, and so the computation times ought to have been similar, which they were.

We perform two more computations in order to better support this idea. If the above claim is correct, then we should expect that a second calculation of $f(4000)$ should take almost no time at all, and neither should a calculation of $f(4001)$. If the claim is not correct, then $f(4000)$ and $f(4001)$ should both take similar amounts of time to calculate as the previous computation (0.013 seconds).

> $time(f(4000)), time(f(4000))$

$$0., 0.$$

This ability to remember and recall previous computations of a function with the **remember** option is very useful, but does make execution times less precise as we have seen. It is probably most useful to think of the time taken worst case scenario (that no values have been previously calculated and stored) as a baseline, with the idea that the **remember** option will often be better than this. In the case we have been dealing with, the Fibonacci number calculator with the **remember** option will, in the very worst case, only need to calculate each previous Fibonacci number once. This is significantly better than the earlier attempts that did not use the **remember** option.

We close this section with a slightly surprising result. Later, in Section 1.3.3 we look at the Fibonacci numbers again, and find a formula for calculating them. Some readers may already know of this formula. Such a formula would allow a function to perform only a single computation when computing any Fibonacci number. Contrast this with our best approach so far which involves (in the worse case) n computations when computing the nth Fibonacci number. It would seem reasonable to think that the inbuilt *Maple* function for Fibonacci numbers would exploit this. However, when we measure the time taken for each technique to compute the first 10,000 Fibonacci numbers we see something perhaps a little surprising.

> $restart :$

> $f := \mathbf{proc}(n :: posint)$
    $\mathbf{option}\ remember;$
    $f(n-1) + f(n-2)$
  $\mathbf{end} :$

> $f(1) := 1; f(2) := 1;$

$$f(1) := 1$$
$$f(2) := 1$$

> $time(seq(f(k), k = 1..10000));$
  $time(seq(combinat[fibonacci](k), k = 1..10000));$

$$0.051$$
$$1.322$$

We have been very careful here to **restart** and redefine our functions. We have done this to make sure our measurements are not skewed by previously performed and remembered computations. What we see is that our f function is drastically quicker than the inbuilt function. Upon first seeing this, one of the authors was quite surprised, however, further thought has lessened this surprise a little. It is left as an exercise for the reader to explore this. As a starting point, one thing to notice is that the f function is exceptionally well suited to sequential computation of Fibonacci numbers—reducing each subsequent computation to a single addition—but there may be other factors as well.

In closing this section, we note that there are many other small and subtle complexities to the detailed structures (loops, decisions and procedures) than we have been able to cover. What has been covered is, however, a very good introduction and should serve—along with the relevant exercises—as an excellent and solid starting point for the readers' own computations. Be sure to examine *Maple*'s help files for more information.

1.3 Enough Code, Already. Show Me Some Math!

We have spent the previous two sections learning *Maple* code mostly for its own sake. Even when we have tackled mathematical problems, we have done so to understand or illustrate particular *Maple* language concepts. By now we hope to be, at the very least, passably familiar with instructing *Maple* to perform calculations.

The point of using *Maple*, however, is to allow us to perform and explore mathematics, not the other way around. So, from this section onwards we move the emphasis away from *Maple* itself and onto mathematical concepts and problems. Our *Maple* skills learned in the previous sections are used to explore the mathematics, and new *Maple* concepts, functions, and so on are introduced as they are needed for the problems at hand.

1.3.1 Induction

In general, the work we do in *Maple* does not constitute a mathematical proof. *Maple* is more a tool for exploring mathematics that will often lead to greater understanding and perhaps the production of a proof by the usual (non-computer-related) means. However, we may use it to perform some basic induction for us.

Recall the formula for the sum of the first n squares is

$$\sum_{k=1}^{n} k^2 = \frac{n(n+1)(2n+1)}{6}$$

Suppose we only want to add the first n even squares. It's not hard to manipulate the sum to an expression involving the above sum.

$$\sum_{k=1}^{n}(2k)^2 = \sum_{k=1}^{n}4k^2 = 4\sum_{k=1}^{n}k^2 = \frac{2n(n+1)(2n+1)}{3}$$

and we can certainly check *Maple* to see if it provides the same answer.

> $sum((2k)^2, k = 1..n)$

$$\frac{4}{3}(n+1)^3 - 2(n+1)^2 + \frac{2}{3}n + \frac{2}{3}$$

> $factor(\%)$

$$\frac{2}{3}n(n+1)(2n+1)$$

That is all well and good, but it's not induction. How about we try the first n odd squares:

$$\sum_{k=1}^{n}(2k-1)^2$$

We could follow the same approach as above, and expand the summand into a quadratic, and apply the formulae we already know for $\sum_{k=1}^{n}k^2$, $\sum_{k=1}^{n}k$, and $\sum_{k=1}^{n}C$, respectively, and indeed a good number of first-year students would probably prefer this to induction. We do not do that this time, however. Instead we ask *Maple* what it thinks the answer is, and verify it using induction (also within *Maple*).

To begin with, we set up a function to more easily reuse the calculations.

> $f := N \to sum((2k-1)^2, k = 1..N);$

$$N \to \sum_{k=1}^{N}(2k-1)^2$$

Now we see what *Maple* thinks the function should look like for arbitrary N.

> $g := factor(f(N));$

$$g := \frac{1}{3}N(2N-1)(2N+1)$$

We have factorized the answer here to keep it neat, and we have assigned it to the variable g for future reference. So now we have a candidate for a formula. We can be pretty confident that it is correct because *Maple* provided it, but it never hurts to check, especially since we don't know yet exactly how *Maple*'s **sum** function handles an indefinite sum like that. This we now do. First we begin with a basis case.

> $f(1) = subs(N = 1, g)$

$$1 = 1$$

Note here the use of the substitution command **subs**. This command substitutes the value 1 for N in the expression g. Note here that we could have simply asked *Maple* $is(f(1) = subs(N = 1, g))$, but the answer or true or false is sometimes unreliable, and it is best to see the statement written out in its entirety before asking for an **is** or **evalb**. It is clear from the *Maple* output that the formula is correct for the basis case of $N = 1$.

Now we may complete the induction. Assuming that

$$\sum_{k=1}^{N}(2k-1)^2 = \frac{1}{3}N(2N-1)(2N+1)$$

we want to show that

$$\sum_{k=1}^{N+1} (2k-1)^2 = \frac{1}{3}(N+1)(2N+1)(2N+3)$$

which we do by showing that

$$\left(\sum_{k=1}^{N} (2k-1)^2\right) + (2N+1)^2 - \frac{1}{3}(N+1)(2N+1)(2N+3) = 0$$

> $f(N) + (2 \cdot (N+1) - 1)^2 - subs(N = N+1, g)$

$$\frac{1}{3}N(2N-1)(2N+1) + (2N+1)^2 - \frac{1}{3}(N+1)(2N+1)(2N+3)$$

> $simplify(\%)$

$$0$$

And we're done. We may not be sure how *Maple* handles an indefinite sum, but we can be extremely confident with its ability to do basic algebra. If we want to completely remove the question of the behavior of *Maple*'s **sum** function out of the equation—as it is technically in question here—we can do the same thing with g alone.

However, the substitution command is a little long, and we probably don't really want to be constantly typing it in whenever we want to assign a value to N. It would be much easier if g was a function itself. *Maple* provides a handy method for taking an arbitrary expression and turning it into a function. It is called **unapply** (and uses nifty ideas from modern logic involving the λ-calculus). We use it now to make g into a function of N, rather than just a fixed expression as it is now.

> $g := unapply(g, N)$

$$g := N \rightarrow \frac{1}{3}N(2N-1)(2N+1)$$

Now we can perform the inductive step using the functional notation for g

> $g(N) + (2 \cdot (N+1) - 1)^2 - g(N+1); simplify(\%)$

$$\frac{1}{3}N(2N-1)(2N+1) + (2N+1)^2 - \frac{1}{3}(N+1)(2N+1)(2N+3)$$
$$0$$

Be careful here. It might look awfully tempting to try something like the following as the inductive step.

> $f(N+1) - g(N+1)$

$$\frac{11}{3}N + \frac{19}{3} - 4(N+2)^2 + \frac{4}{3}(N+2)^3 - \frac{1}{3}(N+1)(2N+1)(2N+3)$$

> $simplify(\%)$

$$0$$

However, this is *not* induction, because we have not used the assumption that $f(N) = g$ in order to show that $f(N+1)$ has the required form. All we have done in this case is verify that *Maple*'s **sum** command produces the desired formula for inputs of N and $N+1$, but we have not proved the relation using induction.

We do one more induction with *Maple*. This time we verify a well known one, and because we won't be explaining every step of the way, the process is much shorter. We verify the formula

$$\sum_{k=1}^{N} k^3 = \frac{N^2(N+1)^2}{4}$$

We can verify in our heads that the relation is true when $N = 1$ so we may eschew the basis step in *Maple*, leaving us just the inductive step itself.

> $g := N \to \dfrac{N^2 \cdot (N+1)^2}{4}; g(N) + (N+1)^3 - g(N+1); simplify(\%)$

$$g := N \to \frac{1}{4} N^2 (N+1)^2$$

$$\frac{1}{4} N^2 (N+1)^2 + (N+1)^3 - \frac{1}{4}(N+1)^2(N+2)^2$$

$$0$$

1.3.2 Continued Fractions

Real numbers may, as we should already be aware, be expressed by decimals that either terminate or continue (countably) infinitely. The latter category may be further partitioned into recurring and nonrecurring infinite decimal representations. Rational numbers may be written as terminating or recurring decimals, and irrational numbers have only infinite nonrecurring decimal representations.

Another way to represent real numbers is with so-called *continued fractions*. A continued fraction is a, potentially infinite, fraction of the form

$$a_0 + \cfrac{b_0}{a_1 + \cfrac{b_1}{a_2 + \cfrac{b_2}{\ddots}}} \quad \text{where } a_i, b_i \in \mathbb{Z}$$

However, for the purposes of this section, we concentrate on simple continued fractions, where the b_i are all 1

$$a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\ddots}}}$$

often abbreviated to just $[a_0; a_1, a_2, \ldots]$.

The procedure for calculating the continued fraction of a number is quite straightforward. Let $x \in \mathbb{R}$. We let $x_0 = x$ and separate the integer part from the fractional part. That is, we take $a_0 = \lfloor x_0 \rfloor$,

$$x = a_0 + (x_0 - a_0) = a_0 + \cfrac{1}{\left(\cfrac{1}{x_0 - a_0}\right)}$$

We now invert the fractional part for $x_1 = (x_0 - a_0)^{-1}$ and set $a_1 = \lfloor x_1 \rfloor$, yielding

$$x = a_0 + \cfrac{1}{a_1 + (x_1 - a_1)} = a_0 + \cfrac{1}{a_1 + \cfrac{1}{\left(\cfrac{1}{(x_1 - a_1)}\right)}}$$

Inverting the fractional part again we end up with $x_2 = (x_1 - a_1)^{-1}$ and repeat the process.

We explore this idea in *Maple* with a terminating decimal $x = 1.23456789$. We use the **floor** command to extract the integer part. The reader is encouraged to read the help documentation regarding this function. We are careful to make sure we use the rational representation of our number, rather than the decimal representation, in order to allow *Maple* to keep the calculations exact. We also need to tell the **convert** function to perform an exact conversion to a rational number, otherwise it will try to make a rational approximation of the decimal number it is given. See the *Maple*'s help files on **?convert/rational** for more details.

```
> x := convert(1.23456789, rational, exact);
```
$$x := \frac{123456789}{100000000}$$

```
> x0 := x; L := floor(x0);
```
$$x_0 := \frac{123456789}{100000000}$$
$$L := 1$$

```
> x1 := 1/(x0 - floor(x0)); L := L, floor(x1)
```
$$x_1 := \frac{100000000}{23456789}$$
$$L := 1, 4$$

So we currently have $a_0 = 1$ and $a_1 = 4$ for our continued fraction. Next we need to calculate $x_2 = x_1 - 4$ and append $floor(x_2)$ to the end of the list, and repeat the process. We started off with a terminating decimal, therefore we should expect the continued fraction also to terminate. So instead of tiring our fingers typing the same two commands over and over again, we write a loop to finish the process for us.

> **for** i **from** 2 **while** $x_{i-1} - \text{floor}(x_{i-1}) \neq 0$ **do**
> $x_i := \dfrac{1}{x_{i-1} - \text{floor}(x_{i-1})}; L := L, \text{floor}(x_i);$
> **od**

$$x_2 := \frac{23456789}{6172844}$$
$$L := 1, 4, 3$$
$$x_3 := \frac{6172844}{4938257}$$
$$L := 1, 4, 5, 1$$
$$x_4 := \frac{4938257}{1234587}$$
$$L := 1, 4, 3, 1, 3$$
$$x_5 := \frac{1234587}{1234496}$$
$$L := 1, 4, 3, 1, 3, 1$$
$$x_6 := \frac{1234496}{91}$$
$$L := 1, 4, 3, 1, 3, 1, 13565$$
$$x_7 := \frac{91}{81}$$
$$L := 1, 4, 3, 1, 3, 1, 13565, 1$$
$$x_8 := \frac{81}{10}$$
$$L := 1, 4, 3, 1, 3, 1, 13565, 1, 8$$
$$x_9 := 10$$
$$L := 1, 4, 3, 1, 3, 1, 13565, 1, 8, 10$$

And so we have the continued fraction representation of 1.23456789 as

$$1 + \cfrac{1}{4 + \cfrac{1}{3 + \cfrac{1}{1 + \cfrac{1}{3 + \cfrac{1}{1 + \cfrac{1}{13565 + \cfrac{1}{1 + \cfrac{1}{8 + \cfrac{1}{10}}}}}}}}$$

or, more compactly, $[1; 4, 3, 1, 3, 1, 13565, 1, 8, 10]$. For the remainder of this book we only use this more compact notation when referring to continued fractions.

We pause here and talk briefly about the mysterious 13565 which appears in the middle of this continued fraction. It perhaps seems a little incongruous sitting there in the middle of a collection of predominantly single digit numbers. Or, at least, it

should look incongruous. Inasmuch as we've been working with rational numbers for the calculations in question, we can be quite confident that it is not a mistake. However, we should be aware that, in general, when we see such unexpected numbers pop up, that we may have a sign of numeric roundoff error (or some other mistake), and should be on our guard. Increasing the digit precision to see if the (apparent) anomaly persists is a good first step.

As the reader might very well have come to expect by now, *Maple* has functions inbuilt to allow for the conversion of a real or rational number to a continued fraction. There are, in fact, two methods. The first method is to use the highly flexible **convert** function, which is explored somewhat in Exercise 4.

> $convert\left(\dfrac{123456789}{100000000}, confrac\right)$

$$[1, 4, 3, 1, 3, 1, 13565, 1, 8, 10]$$

However, you should be aware that using this method may run afoul of *Maple*'s internal digit precision. For instance, with the default precision of 10 digits, *Maple* produces the following output.

> $convert(1.23456789, confrac)$

$$[1, 4, 3, 1, 3, 1, 13565]$$

which is missing the last three quotients. However, if we raise the internal precision to 20 digits then we obtain the correct result. For best results, one should avoid using decimals with the continued fraction conversion when using the **convert** function, if at all possible.

The alternative, and probably preferred, method is to use the **cfrac** function which is located within the numtheory package. This function seems to behave better with decimal numbers but note, however, that it will write the continued fraction out in the long form, unless we ask it otherwise.

> $x := convert(1.23456789, rational, exact):$
  $numtheory[cfrac](x); numtheory[cfrac](1.23456789, quotients)$

$$1 + \cfrac{1}{4 + \cfrac{1}{3 + \cfrac{1}{1 + \cfrac{1}{3 + \cfrac{1}{1 + \cfrac{1}{13565 + \cfrac{1}{1 + \cfrac{1}{8 + \cfrac{1}{10}}}}}}}}$$

$$[1, 4, 3, 1, 3, 1, 13565, 1, 8, 10]$$

The **cfrac** function also allows for converting a converted fraction in list form back to a rational number (see Exercise 23).

Let us explore continued fractions with some irrational numbers now. We start with the golden ratio $\phi = (1 + \sqrt{5})/2$. The name **phi** (or, equivalently, *phi*) is reserved as a function in the numtheory package, therefore we avoid using it, and use GR (short for golden ratio) instead.

> $GR = \dfrac{1 + \mathrm{sqrt}(5)}{2}; convert(GR, confrac)$

$$[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]$$

Well that's interesting. Let's see what the **cfrac** function makes of it

> $numtheory[cfrac](GR, quotients)$

$$[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, \dots]$$

> $numtheory[cfrac](GR, quotients, 20)$

$$[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, \dots]$$

Note here that the ellipses above do not mean "continuing on in the same fashion" as they usually would in printed mathematics, but instead mean something closer to "and some more stuff that wasn't calculated." So what we know is that the first twenty one quotients in the continued fraction of ϕ are all 1, and that there are some more quotients about which we don't know.

As it happens, being an irrational number, the continued fraction representation of ϕ is infinite, just as its decimal representation. However, unlike its decimal representation—which exhibits no particularly discernible pattern—the continued fraction representation does indeed exhibit a clear pattern. The pattern we have seen above continues infinitely for a continued fraction representation that is all 1.

So why is this? Well, first notice that $2 < \sqrt{5} < 3$ because $4 < 5 < 9$, and so $1 < \phi < 2$. This tells us straight away that the integer part of ϕ is 1. Subtracting this yields

$$\phi - 1 = \frac{1 + \sqrt{5}}{2} - \frac{2}{2} = \frac{\sqrt{5} - 1}{2}$$

Inverting this we get

$$\frac{2}{\sqrt{5} - 1} = \frac{2}{\sqrt{5} - 1} \cdot \frac{\sqrt{5} + 1}{\sqrt{5} + 1} = \frac{2(\sqrt{5} + 1)}{4} = \frac{1 + \sqrt{5}}{2} = \phi$$

which is back where we started from. It should be clear from this then that $\phi - 1 = 1/\phi$ and that we can continue the continued fraction process begun above indefinitely providing the continued fraction $[1; \overline{1, 1}]$ (where the bar above the $1, 1$ indicates that the pattern is repeated infinitely).

Finally, let us look at the continued fractions of some other irrationals.

> $with(numtheory) :$
> $cfrac(\mathrm{sqrt}(2), quotients, 20);$
> $cfrac(\mathrm{sqrt}(5), quotients, 20);$
> $cfrac(\exp(1), quotients, 20);$

$$[1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, \dots]$$
$$[2, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, \dots]$$
$$[2, 1, 2, 1, 1, 4, 1, 1, 6, 1, 1, 8, 1, 1, 10, 1, 1, 12, 1, 1, 14, \dots]$$

We can see, then, that continued fractions may give more information regarding the patterns behind a real number than its decimal expansion might. When exploring an unknown number, we should look at both a decimal approximation, as well as its continued fraction. However, not all numbers have a nice continued fraction representation, as we explore in Exercise 24.

1.3.3 Recurrence Relations

We should recall from Section 1.2.6 the Fibonacci numbers, and their formulation as $f(n) = f(n-1) + f(n-2)$ where $n \in \mathbb{N}$ and $f(1) = f(2) = 1$. This is an example of a recurrence relation, a sequence where the value of each element is dependent on one or more previous elements.

The requirement that n be natural is important here, because the recurrence relation describes a *sequence* and not a function, even though we have used functional notation so far. Recurrence relations are often written using the subscript notation more usually associated with a sequence, in which case the Fibonacci numbers are defined as a sequence $\{f_n\}_{n \in \mathbb{N}}$ where $f_n = f_{n-1} + f_{n-2}$ and $f_1 = f_2 = 1$. However, due to the way *Maple* handles recurrence relations, we continue to use the functional notation instead of the subscript notation.

The *order* of a recurrence relation is how far back one must look in order to calculate a term. The Fibonacci numbers are therefore of order two because one needs to know the two previous terms in order to calculate any term. A recurrence relation may have other properties, but for the sake of simplicity let us say that a *constant coefficient, linear, homogeneous* recurrence relation of order k is a recurrence relation that has the form

$$a(n) = c_1 \cdot a(n-1) + \cdots + c_k \cdot a(n-k)$$

where the c_i are constants.

To begin we look at some first-order recurrence relations. In fact, we look in full generality at first order linear recurrence relations with constant coefficients. Let $a(n) = c_1 \cdot a(n-1)$. This is the general form of a first-order linear homogeneous recurrence relation. If we wish to know what the 100th term in the sequence was—provided, of course that we know both the first term $a(0)$, and the coefficient (c_1), then we would have to calculate the second term before we could calculate the third term and so on. All in all we would have to perform 99 calculations. In fact, this is exactly what we have had to do when we wrote loops and procedures to calculate the Fibonacci numbers in Sections 1.2.2 and 1.2.4.

What we would ideally like is some formula or function of n that would always calculate the nth term in the sequence. The act of finding such a formula is called *solving* the recurrence relation. In the case of first-order relations, and especially first-order linear homogeneous recurrence relations with constant coefficients, doing so is quite straightforward. Observe that

$$
\begin{aligned}
a(0) &= 1 \cdot a(0) & &= (c_1)^0 \cdot a(0) \\
a(1) &= c_1 \cdot a(0) & &= (c_1)^1 \cdot a(0) \\
a(2) &= c_1 \cdot a(1) = c_1 \cdot (c_1 \cdot a(0)) & &= (c_1)^2 \cdot a(0) \\
a(3) &= c_1 \cdot a(2) = c_1 \cdot \left((c_1)^2 \cdot a(0)\right) &&= (c_1)^3 \cdot a(0)
\end{aligned}
$$

and so on. The pattern here is quite clear.

$$a(n) = (c_1)^n \cdot a(0)$$

We may easily now, if we wish, calculate the 100th element of the sequence as $(c_1)^{99} \cdot a(0)$.

A simple example of such a recurrence relation is a bank account that earns, say, 5% compound interest both calculated and paid annually. If we let $a(n)$ be the amount of money at the end of n years, then $a(0)$ is the initial deposit, and $a(n) = 1.05 \cdot a(0)$. If

we start with $5000 then we know, thanks to the analysis performed above, that after 5 years we will have $(1.05)^5 \cdot 5000 = \$6381.41$.

We can explore recurrence relations inside *Maple*, of course. *Maple* provides the **rsolve** command for solving recurrence relations. Happily, this function is not limited to only linear, homogeneous, constant coefficient recurrence relations. Unfortunately, not all recurrence relations are solvable. Let us see what *Maple* says about our earlier analysis.

> $rsolve(a(n) = c_1 \cdot a(n-1), a(n))$

$$a(0)c_1^n$$

Here we asked *Maple* to solve the recurrence relation $a(n) = c_1 \cdot a(n-1)$ and asked it to solve for $a(n)$. This means that the answer returned by the **rsolve** function should be the formula for $a(n)$. As we should hope, *Maple* gave us the answer we already knew. Let us now, quickly, drop the constant coefficient condition, and see what happens. For maximum generality, we assume that a function $f(n)$ multiplies the previous term in the recurrence.

> $rsolve(a(n) = f(n) \cdot a(n-1), a(n))$

$$\prod_{n0=0}^{n-1} f(n0+1)\, a(0)$$

It is not immediately clear which terms are part of the product. However, if *Maple* has multiple terms within a sum or product, it will enclose them in parentheses. With this in mind, the $a(0)$ term isn't inside the product, and the entire expression might be written less ambiguously as

$$\left(\prod_{n0=0}^{n-1} f(n0+1)\right) a(0) \ \text{ or } \ a(0) \prod_{n0=0}^{n-1} f(n0+1)$$

We may confirm the result using a very similar analysis to that performed for the constant coefficient cast

$$a(1) = f(1)\, a(0) \qquad\qquad = \left(\prod_{k=1}^{1} f(k)\right) a(0) = a(0) \prod_{k=0}^{1-1} f(k+1)$$

$$a(2) = f(2)\, a(1) = f(2)f(1)a(0) \quad = \left(\prod_{k=1}^{2} f(k)\right) a(0) = a(0) \prod_{k=0}^{2-1} f(k+1)$$

$$a(3) = f(3)\, a(2) = f(3)f(2)f(1)a(0) = \left(\prod_{k=1}^{3} f(k)\right) a(0) = a(0) \prod_{k=0}^{3-1} f(k+1)$$

$$\vdots$$

$$a(n) = f(n)f(n-1)\cdots f(1)\, a(0) \quad = \left(\prod_{k=1}^{n} f(k)\right) a(0) = a(0) \prod_{k=0}^{n-1} f(k+1)$$

Let us now look at second-order linear recurrence relations with constant coefficients. Performing an analysis like the ones above that we performed for first-order recurrence relations is of limited to no use with second-order relations. In short, there's too much going on to see a clear pattern. Fortunately there are some nice theorems that allow for easy solution.

Given a second order linear, homogeneous recurrence relation with constant coefficients,

$$a(n) = c_1 \cdot a(n-1) + c_2 \cdot a(n-2)$$

we construct the *characteristic polynomial* $x^2 - c_1 x - c_2$ and find the roots. Note that the recurrence may be re-written as

$$a(n) - c_1 \cdot a(n-1) + c_2 \cdot a(n-2) = 0$$

and so the transformation to the characteristic polynomial should be clearer now. Once we have the roots r_1, r_2 of the characteristic polynomial, then the recurrence relation has a formula depending on whether the roots are distinct. The general form of the solution is

$$a(n) = \begin{cases} A \cdot (r_1)^n + B \cdot (r_2)^n & \text{if } r_1 \neq r_2 \\ A \cdot (r_1)^n + n \cdot B \cdot (r_1)^n & \text{if } r_1 = r_2 \end{cases}$$

where A and B are constants. If we know some initial conditions, then we may substitute them into the general form of $a(n)$ in order to calculate exactly what the constants A and B are. What is interesting here is that the general form will always satisfy the recurrence relation no matter the choice of constants.

Let us explore this with the Fibonacci numbers. The characteristic polynomial we need to find the roots of is is $x^2 - x - 1$. Using the quadratic formula $x = (1/2a) \cdot (-b \pm \sqrt{b^2 - 4ac})$ we have $x = \frac{1}{2}(1 \pm \sqrt{5})$. So $r_1 = \frac{1}{2}(1 + \sqrt{5})$, which is the golden ratio that we looked at in Section 1.3.2, and $r_2 = \frac{1}{2}(1 - \sqrt{5})$. Turning to *Maple* now,

```
> f := n -> A · ((1 + sqrt(5))/2)^n + B · ((1 - sqrt(5))/2)^n;
```
$$f := n \rightarrow A\left(\frac{1}{2} + \frac{1}{2}\sqrt{5}\right)^n + B\left(-\frac{1}{2}\sqrt{5} + \frac{1}{2}\right)^n$$

```
> f(n - 1) + f(n - 2); simplify(%)
```
$$A\left(\frac{1}{2} + \frac{1}{2}\sqrt{5}\right)^{n-1} + B\left(-\frac{1}{2}\sqrt{5} + \frac{1}{2}\right)^{n-1} + A\left(\frac{1}{2} + \frac{1}{2}\sqrt{5}\right)^{n-2} + B\left(-\frac{1}{2}\sqrt{5} + \frac{1}{2}\right)^{n-2}$$
$$\frac{16\left(A\left(\frac{1}{2} + \frac{1}{2}\sqrt{5}\right)^n + B\left(-\frac{1}{2}\sqrt{5} + \frac{1}{2}\right)^n\right)}{\left(\sqrt{5} + 1\right)^2 \left(\sqrt{5} - 1\right)^2}$$

Unfortunately, for some reason, *Maple* doesn't simplify the denominator of the expression. It should take no time at all to calculate in our head that

$$\left(\sqrt{5} + 1\right)^2 \left(\sqrt{5} - 1\right)^2 = \left(\left(\sqrt{5} + 1\right)\left(\sqrt{5} - 1\right)\right)^2 = 4^2 = 16$$

and so the denominator and the multiple of 16 will cancel. We could, also, have asked *Maple* to specifically simplify only the denominator.

```
> denom(%) = simplify(denom(%))
```
$$\left(\sqrt{5} + 1\right)^2 \left(\sqrt{5} - 1\right)^2 = 16$$

In either case, we end up with the formula for $f(n)$ when we simplify $f(n-1) + f(n-2)$, showing that the choice of constants A and B does not matter.

In order to find the constants for the specific case of the Fibonacci numbers we could plug $f(1) = 1$ and $f(2) = 1$ into our equation above and solve for A and B. Instead of this, however, we just ask *Maple* to solve the recursion for us. We have already defined the variable f, above, so we make sure to use a different variable name this time.

> $rsolve(\{F(n) = F(n-1) + F(n-2), F(1) = 1, F(2) = 1\}, F(n))$

$$\frac{1}{5}\sqrt{5}\left(\frac{1}{2} + \frac{1}{2}\sqrt{5}\right)^n - \frac{1}{5}\sqrt{5}\left(\frac{1}{2} - \frac{1}{2}\sqrt{5}\right)^n$$

and so it would seem that $A = \frac{1}{5}\sqrt{5}$ and $B = -\frac{1}{5}\sqrt{5}$.

1.3.4 The Sieve of Eratosthenes

It would be hard to justify this chapter as being about number theory if we didn't mention prime numbers at some stage. Here we look at the problem of listing numbers that are prime, and use a technique known to the ancient Greeks. In particular, it was a man named Eratosthenes who is responsible for this technique.

Suppose we want to find all the prime numbers less than some number, 100 say. First we write the numbers in a square (or rectangle) thusly

$$
\begin{array}{cccccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\
11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\
21 & 22 & 23 & 24 & 25 & 26 & 27 & 28 & 29 & 30 \\
31 & 32 & 33 & 34 & 35 & 36 & 37 & 38 & 39 & 40 \\
41 & 42 & 43 & 44 & 45 & 46 & 47 & 48 & 49 & 50 \\
51 & 52 & 53 & 54 & 55 & 56 & 57 & 58 & 59 & 60 \\
61 & 62 & 63 & 64 & 65 & 66 & 67 & 68 & 69 & 70 \\
71 & 72 & 73 & 74 & 75 & 76 & 77 & 78 & 79 & 80 \\
81 & 82 & 83 & 84 & 85 & 86 & 87 & 88 & 89 & 90 \\
91 & 92 & 93 & 94 & 95 & 96 & 97 & 98 & 99 & 100
\end{array}
$$

We know that 1 is not prime, therefore we start by crossing it out. We now find the first uncrossed number, 2 in this case. This number is prime and so no multiple of is can be prime. So we go and cross out all the multiples of this number.

$$
\begin{array}{cccccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\
11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\
21 & 22 & 23 & 24 & 25 & 26 & 27 & 28 & 29 & 30 \\
31 & 32 & 33 & 34 & 35 & 36 & 37 & 38 & 39 & 40 \\
41 & 42 & 43 & 44 & 45 & 46 & 47 & 48 & 49 & 50 \\
51 & 52 & 53 & 54 & 55 & 56 & 57 & 58 & 59 & 60 \\
61 & 62 & 63 & 64 & 65 & 66 & 67 & 68 & 69 & 70 \\
71 & 72 & 73 & 74 & 75 & 76 & 77 & 78 & 79 & 80 \\
81 & 82 & 83 & 84 & 85 & 86 & 87 & 88 & 89 & 90 \\
91 & 92 & 93 & 94 & 95 & 96 & 97 & 98 & 99 & 100
\end{array}
$$

Having done that, we find the next number along (after 2, which was our previous number) which is not crossed out. In this case it's 3. This number must be prime because it is not a multiple of any prime number less than it, of which there was only one in

this case. We now cross out all multiples of 3. Note that some of the multiples of 3 are already crossed out, because they were also multiples of 2.

$$
\begin{array}{cccccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\
11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\
21 & 22 & 23 & 24 & 25 & 26 & 27 & 28 & 29 & 30 \\
31 & 32 & 33 & 34 & 35 & 36 & 37 & 38 & 39 & 40 \\
41 & 42 & 43 & 44 & 45 & 46 & 47 & 48 & 49 & 50 \\
51 & 52 & 53 & 54 & 55 & 56 & 57 & 58 & 59 & 60 \\
61 & 62 & 63 & 64 & 65 & 66 & 67 & 68 & 69 & 70 \\
71 & 72 & 73 & 74 & 75 & 76 & 77 & 78 & 79 & 80 \\
81 & 82 & 83 & 84 & 85 & 86 & 87 & 88 & 89 & 90 \\
91 & 92 & 93 & 94 & 95 & 96 & 97 & 98 & 99 & 100
\end{array}
$$

The next number that is not crossed out is 5, which must be prime because it is not a multiple of any prime smaller than it. We repeat this process and eventually there will be no new "next" uncrossed number, in which case we will have found all primes less than or equal to 100 (our chosen upper bound for this example). This yields the following final table.

$$
\begin{array}{cccccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\
11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\
21 & 22 & 23 & 24 & 25 & 26 & 27 & 28 & 29 & 30 \\
31 & 32 & 33 & 34 & 35 & 36 & 37 & 38 & 39 & 40 \\
41 & 42 & 43 & 44 & 45 & 46 & 47 & 48 & 49 & 50 \\
51 & 52 & 53 & 54 & 55 & 56 & 57 & 58 & 59 & 60 \\
61 & 62 & 63 & 64 & 65 & 66 & 67 & 68 & 69 & 70 \\
71 & 72 & 73 & 74 & 75 & 76 & 77 & 78 & 79 & 80 \\
81 & 82 & 83 & 84 & 85 & 86 & 87 & 88 & 89 & 90 \\
91 & 92 & 93 & 94 & 95 & 96 & 97 & 98 & 99 & 100
\end{array}
$$

We can see then that our list of prime numbers less than 100 is

$$2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97$$

When completing the previous example—and the reader is encouraged to do so by hand—we find that once we have crossed off all the multiples of 7, then all multiples of all later primes are already crossed off. In fact, this is a phenomenon that we always see when performing the Eratosthenes' sieve for any upper bound. The special property of 7 in our particular case, above, is that it is the largest prime less than or equal to $\sqrt{100}$. In general, when using Eratosthenes sieve to find prime numbers less than or equal to some bound, n say, we may always end the process when we've found all primes less than or equal to $\sqrt{n}$.

Recall from Section 1.2.3 that when finding the divisors of a number, n say, we need only to check the numbers less than or equal to $\sqrt{n}$. In a similar vein, when performing the Eratosthenes' sieve to find all primes less than n, when we find a prime, p say, we then proceed to cross off any number for which p is a divisor. We have effectively marked all numbers less than n that have p as a divisor. We only need a single divisor to decide a number is not prime, and it follows from our observations in Section 1.2.3 that if a number has a divisor, then it has a divisor less than its square root. Finally, the simple fact that

$$a \le b \implies \sqrt{a} \le \sqrt{b}$$

leads us to the conclusion that once we have crossed off all multiples of all primes less than $\sqrt{n}$ then we must have found a divisor for every number less than n, as long as there was one to find in the first place. Anything not crossed out must, therefore, be prime.

In our particular example, above, once we get past 10, we have crossed off every multiple of every prime less than or equal to $\sqrt{100}$ and so we must have found at least one divisor for every number less than or equal to 100, as long as there was a divisor to find. This is exactly the observation that led us to this line of inquiry.

Now we implement this technique in *Maple*. We make a procedure that we will name *Eratosthenes* that will return a sequence of all the primes less than some given number N which must be a positive integer.

In order to represent in *Maple* the "crossing out" of numbers as we performed in the sieve (above), we are going to create a list L that contains N elements. If L_i is true, then i is prime, and if L_i is $false$ then i is not prime. This list begins with all its values as $true$ and we "cross out" by setting the appropriate value in the list to $false$. We also create a sequence named *primeslist* which starts out empty, and to which we append the primes.

```
> Eratosthenes := proc(N :: posint)
    local L, primeslist, n, k;
    description "Calculate all primes less than or equal to N";
    L := Array(2..N, i → true);
    for n from 2 to trunc(sqrt(N)) do
      if L_n = true then
        for k from n by 1 while k · n ≤ N do
          L_{k·n} := false
        od
      fi
    od;
    # We computed the primes, now to collate them into a sequence
    primeslist := NULL;
    for n from 2 to N do
      if L_n = true then primeslist := primeslist, n fi
    od;
    primeslist;
  end :
```

```
> Eratosthenes(1000)
 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61,
 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137,
   139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199,
   211, 223, 227, 229, 233, 239, 241, 251, 257, 263, 269, 271, 277,
   281, 283, 293, 307, 311, 313, 317, 331, 337, 347, 349, 353, 359,
   367, 373, 379, 383, 389, 397, 401, 409, 419, 421, 431, 433, 439,
   443, 449, 457, 461, 463, 467, 479, 487, 491, 499, 503, 509, 521,
   523, 541, 547, 557, 563, 569, 571, 577, 587, 593, 599, 601, 607,
   613, 617, 619, 631, 641, 643, 647, 653, 659, 661, 673, 677, 683,
   691, 701, 709, 719, 727, 733, 739, 743, 751, 757, 761, 769, 773,
   787, 797, 809, 811, 821, 823, 827, 829, 839, 853, 857, 859, 863,
   877, 881, 883, 887, 907, 911, 919, 929, 937, 941, 947, 953, 967,
                   971, 977, 983, 991, 997
```

Instead of a list, we declared an **Array**. An **Array** is a quite powerful, and very general class in *Maple*, but in this example it behaves almost exactly as does a list. Using the **Array** gave us two advantages over a list. First we were able to specify an exact range of indices for indexing (2..N in this case). Second we were easily able to specify an initial value for each element, which is set when the **Array** was created. This initialization was in the form of a function mapping an index variable (which we called i) to the value we wanted L_i to have. Without the **Array** we would have needed to use something like $L := [seq(true, k = 1..N)]$ to achieve a similar result. In our case above we wanted the value at all indices to be *true* but this need not have been the case, and we could have had a more complicated initialization had we wanted.

Generating prime numbers is actually a fairly computationally intensive task. The sieve of Eratosthenes is quite interesting in its own right, and is reasonably efficient for such a simple algorithm. However, for primes larger than 10,000 it starts to get noticeably slow. Of course, as we should have well and truly come to expect by now, *Maple* provides inbuilt functions for finding primes, as well as testing primality. The big advantage to the inbuilt functions is that they perform all sorts of tricks to keep the execution times remarkably quick.

These functions are **ithprime** which calculates the ith prime number, **nextprime** which calculates the next prime number after a given number, **prevprime** which calculates the previous prime number before a given number, and finally **isprime** which calculates whether a given number is prime.

> $ithprime(5), nextprime(5), nextprime(6), prevprime(5), prevprime(4)$

$$11, 7, 7, 3, 3$$

1.4 Problems and Exercises

1. Enter the following expressions into *Maple*

   a. $\dfrac{1}{2}$

   b. e^2

   c. $x^2 + x - 1$

   d. $\dfrac{x^2 + 1}{x^2 - 1}$

   Perform the following calculations in *Maple*. Obtain a decimal approximation as well as simplified exact values.

   e. $\sin\left(\dfrac{3\pi}{5}\right)$

   f. $\dfrac{15!}{e^4}$

   g. $\dfrac{12! + 2^{\frac{1}{3}}}{\sin(2)^3}$

   h. $2^{2^{2^{2^2}}}$

   What do you notice about the floating point approximation of the final calculation, compared to the exact value?

2. We now explore *Maple*'s digit precision for decimal approximations. Enter the following commands into *Maple* and explain how the precision has been modified. Try some entries of your own if you are still unsure, or to test your explanation.

   a. **>** $evalf(\text{Pi}); evalf(\exp(1))$

   b. **>** $evalf[20](\text{Pi}); evalf[25](\exp(1))$

   c. **>** $evalf(\text{Pi}, 35); evalf(\exp(1), 30)$

   d. **>** $Digits := 50; evalf(\text{Pi}); evalf(\exp(1))$

   Obtain numeric approximations of the following expressions, to the indicated digit precision

   e. π^2 to 10 significant figures

   f. $\sin(1)$ to 10 significant figures

   g. e^π to 15 significant figures

   h. $\log(\pi)$ to 22 significant figures

   You can reset the digit precision to 10 decimal places using either $Digits := 10$ or the **restart** command. Beware that the **restart** command will completely undo anything you have done previously.

3. This exercise shows two cautionary scenarios.

   a. Some variables in *Maple* are protected, which means their values cannot be changed. Enter the following into *Maple*.

      i. **>** $NULL := 7$

      ii. **>** $\sin := \exp(2)$

      iii. **>** $\log := x^2 + 3x - 2$

      iv. **>** $sum := \dfrac{x + 1}{2x - 2}$

      Why might these names be protected?

   b. We look at why caution should be taken when using the **%** operator. Follow the instructions below.

      i. Enter the following *Maple* commands. Make sure to keep each entry in its own input bracket.

         > $(x + y)^3$

         > $expand(\%)$

> $subs(y = 1, \%)$

  ii. Edit the first command to instead read $(x + y)^4$. Be sure to remember to hit *enter* to perform the new calculation

  iii. Go back to the third calculation and press enter. What happened?

4. The **convert** function is a very useful function with very many uses. The following are just a tiny handful of examples of the convert function's capabilities. Enter the following into *Maple* and also read the help files regarding the **convert** function.

  a. > $convert\left(\dfrac{3 \cdot Pi}{2}, degrees\right)$

  b. > $convert(25\ degrees, radians)$

  c. > $convert\left(10, units, \dfrac{meters}{second}, \dfrac{kilometers}{hour}\right)$

  d. > $convert\left(60, units, \dfrac{km}{h}, \dfrac{m}{s}\right)$

  e. > $convert(\sinh(x), \exp)$

  f. > $convert\left(\dfrac{\exp(x) + \exp(-x)}{2}, trig\right)$

  g. > $convert([x, 2x, 3x, 4x], ` + `)$

  h. > $convert([x, 2x, 3x, 4x], ` * `)$

Now perform the following conversions.

  i. How many furlongs per fortnight is 60 kilometers per hour?

  j. What is the trigonometric identity for $(e^x - e^{-x})/(e^x + e^{-x})$

  k. What is the partial fraction representation of

$$\frac{x^2 + 3x + 2}{(x - 1)(x^2 + 2)^2}$$

5. We have seen the **$** operator used to create repeated sequences. It may also be used as a shortcut to the **seq** command. Enter the following commands, and explain how the **$** operator works. Try some entries of your own if you are still unsure, or to test your explanation.

  a. > $k^2\$k = 4..10$

  b. > $a_i\$i = 3..6$

  c. > $\dfrac{1}{k}\$k = 1..7$

  d. > $\$3..5$

Confirm your guess with *Maple*'s help: (**?$**)

6. *Maple*'s **seq** command is capable of creating sequences where the index variable increases by more than 1 at each step, or even where the index variable takes on arbitrary values entirely. Enter the following into *Maple* and explain what is happening. Note that the index variable can be any valid variable.

  a. > $seq(i^2, i = 1..10, 3)$

  b. > $seq(a^2 - a - 1, a = 1..10, 3)$

  c. > $seq\left(\dfrac{2}{n}, n\ \mathbf{in}\ [2, 3, 5, 7, 11, 12]\right)$

  d. > $seq\left(fib^2, fib\ \mathbf{in}\ [1, 1, 2, 3, 5, 8]\right)$

Now produce the following sequences using the **seq** command, and the ideas above.

e. $9, 25, 49, 81$ f. $1, \dfrac{1}{2}, \dfrac{1}{4}, \dfrac{1}{7}, \dfrac{1}{11}, \dfrac{1}{16}$

Note that neither of these techniques can be used with the **$** operator; see Exercise 5.

7. For lists and sequences—and many other *Maple* constructs that use natural indices—indexing may be performed by using a negative index that begins counting from the end, instead of the beginning. To illustrate this, create in *Maple* a list $L :=$ $[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$. Issue the following commands.

   a. `>` L_{-1} d. `>` $L[-7.. -5]$
   b. `>` $L[-3]$ e. `>` $L_{-4..}$
   c. `>` $L_{-4..-2}$ f. `>` $L[.. -6]$

Which of the following commands do you expect to fail? Enter them into *Maple* and see if you were correct. Explain why the failures occurred.

   g. $L_{-1..-3}$ i. $L[4.. -7]$
   h. $L[3.. -3]$ j. $L_{-7..2}$

Hint: Try to change the negative index into the usual positive index.

8. Enter the following commands into *Maple*. For the purposes of these commands $L := [x, y, z, x, y]$.

   a. `>` $op([1, 2, 3, 4, 5])$ d. `>` $op(L)$
   b. `>` $nops([1, 2, 3, 4, 5])$ e. `>` $nops(L)$
   c. `>` $numboccur([1, 2, 3, 4, 5], 3)$ f. `>` $numboccur(L, x)$

What does it look like the **op**, **nops**, and **numboccur** functions are doing? Now change L to be $L := [[1, 2], [2, 3], [3, 4]]$ and consider the following.

   g. `>` $op(L)$ j. `>` $op(x^2 + 2x - 3)$
   h. `>` $nops(L)$ k. `>` $nops(x^2 + 2x - 3)$
   i. `>` $numboccur(L, 3)$ l. `>` $numboccur(x^2 + 2x - 3, x)$

What do you now think **op**, **nops**, and **numboccur** are doing? Read *Maple*'s help files regarding these functions and see if they agree with your observations and guesses.

Finally, have a look at the ListTools package, and in particular the **Occurrences** function. Redefine L to be $L := [[1, 2], [3, 4], 5, 6, [1, 2]]$ and use this function to have *Maple* calculate the following. You should be able to verify the answers by eye.

   m. The number of occurrences of the element $[1, 2]$ in the list L.
   n. The number of occurrences of the element 5 in the list L.
   o. The number of occurrences of the element x in the list L.
   p. The number of occurrences of the element $[y, z]$ in the list L.

9. Create a list containing the first 1000 digits of π, in the correct order.
   In order to do this you need to manipulate the first 1000 digits of π into an integer and use the command $convert(n, base, 10)$ which converts an integer n into a list of digits in base 10.
   Hint: You need to use the **trunc** command at some stage.

10. Calculate the following sums and products.

a. $\displaystyle\sum_{i=1}^{100} \frac{1}{2^i}$

c. $\displaystyle\sum_{n=0}^{\infty} \frac{1}{n!}$

b. $\displaystyle\prod_{n=1}^{6} (1 + x^{2^n})$

d. $\displaystyle\prod_{k=1}^{\infty} \frac{4k^2}{4k^2 - 1}$

The following should look familiar from first-year calculus.

e. $\displaystyle\sum_{k=0}^{\infty} \frac{x^{2n}}{(2n)!}$

f. $\displaystyle\sum_{k=0}^{\infty} \frac{(-1)^n x^{2n+1}}{(2n+1)!}$

Convert the following sums to sigma notation, and then use that to implement them using the **sum** command.

g. $1 + 8 + 27 + \cdots + n^3$

h. $1 + \dfrac{1}{2} + \dfrac{1}{4} + \cdots + \dfrac{1}{2^i}$

i. $\dfrac{1}{1 \cdot 3} + \dfrac{1}{3 \cdot 5} + \dfrac{1}{5 \cdot 7} + \cdots + \dfrac{1}{(2k-1)(2k+1)}$

11. Use *Maple*'s **sum** command to explore the binomial formula. Recall that the binomial theorem states that

$$(x + y)^n = \sum_{k=0}^{n} \binom{n}{k} x^{n-k} y^k, \ \text{where} \ \binom{n}{k} = \frac{n!}{(n-k)!k!}$$

You should also use *Maple*'s help to find a built-in function that will calculate the binomial coefficients $\binom{n}{k}$.

   a. Create a function, f say, using the arrow notation ($\rightarrow$) that expands $(x + y)^n$.
   b. Create another function, g say, using the arrow notation that uses *Maple*'s **sum** command to evaluate the sum in the binomial formula.
   c. Choose some values for n and check that $f(n) = g(n)$.
   d. Ask *Maple* to evaluate $f(N)$. To what does it evaluate?

Perform these tasks twice: once using *Maple*'s built-in function for binomial coefficients and once using the formula for the coefficients.

12. An arithmetic sequence is a sequence where each term differs from the previous term by a fixed amount. This "fixed amount" is called the *common difference* of the sequence. For example,

$$a = \{1, 2, 3, 4, \ldots\}, \quad b = \{1, 3, 5, 7, \ldots\}, \ \text{and} \ c = \{1, 4, 7, 10, \ldots\}$$

are arithmetic sequences with a common difference of 1, 2, and 3, respectively.

   a. Determine a formula for the nth element of an arbitrary arithmetic sequence starting at 1, with common difference d.
   b. Write functions that will calculate the first n terms in the arithmetic sequences a, b, and c (above).
      Hint: Use part (a) in combination with a **seq** command.

Arithmetic sequences may have any first element, but those beginning at 1 generate the so-called *polygonal numbers*. The sequence a generates the triangular numbers;

the nth triangular number is the sum of the first n terms in the arithmetic progression a (above). That is,

$$T := \left\{ \sum_{k=1}^{N} a_k \right\}_{N=1}^{\infty}$$

Similarly the nth square number is the sum of the first n terms in the arithmetic progression b and the nth pentagonal number is the sum of the first n terms in the arithmetic sequence c.

Note: The square numbers are precisely the numbers that are perfect squares.

   c. Write functions that will calculate the nth triangular, square, and pentagonal numbers.

   d. Repeat (b) and (c) for the hexagonal numbers. (You will need to extrapolate which arithmetic sequence generates the hexagonal numbers).

13. Enter the following into *Maple*. What is causing the errors? Can you modify the commands so that they work?

   a. `>` $k := 1; sum\left(\dfrac{1}{k}, k = 1..10\right)$

   b. `>` $n := 3; Sum(n^3, n = 1..5); value(\%)$

   c. `>` $f := N \rightarrow sum(k, k = 1..N); seq(f(k), k = 1..10)$

Note that the **seq** command does not exhibit this problem.

   d. `>` $k := 1; seq\left(\dfrac{1}{k}, k = 1..10\right);' k' = k;$

   e. `>` $n := 3; seq(n^3, n = 1..5);' n' := n;$

14. If we wish to apply a function to every item in a list, *Maple* provides a function named **map**. Enter the following commands, and read *Maple*'s help on the **map** function. What is the **map** function doing?

   a. `>` $map(\exp, [1, 2, 3, 4, 5])$

   b. `>` $L := [0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}, 2\pi]; map(\sin, L)$

   c. `>` $f := x \rightarrow 2 \cdot x^2; map(f, [0, 2, 4, 6, 8])$

   d. `>` $f := x \rightarrow \frac{x}{2}; L := [0, \frac{1}{2}, 1, \frac{3}{2}, 2]; map(f, L)$

   e. `>` $map(N \rightarrow \frac{1}{N}, [2, Pi, \exp(1), \log(Pi)])$

   f. `>` $C := [seq(k + 1 + k \cdot I, k = 1..10)]; map(z \rightarrow \text{abs}(z), C)$

Create a list P that contains the first five prime numbers $[2, 3, 5, 7, 11]$. Use **map** and the list P to create the following lists.

   g. $[3, 4, 6, 8, 12]$

   h. $[5, 10, 26, 50, 122]$

   i. $[\ln(2\pi), \ln(3\pi), \ln(5\pi), \ln(7\pi), \ln(11\pi)]$

15. Just as with sequences (see Exercise 6) *Maple* **for** loops are capable of having their counter increase by more than 1 at each step, or even have the counter take on arbitrary values entirely (from a list or a set). Enter the following into *Maple* and explain what is happening.

   a. `>` **for** i **from** 1 **to** 10 **by** 3 **do** i^2 **od**

   b. `>` **for** a **from** 1 **to** 10 **by** 3 **do** $a^2 - a - 1$ **od**

   c. `>` **for** n **in** $[2, 3, 5, 7, 11, 12]$ **do** $\dfrac{2}{n}$ **od**

  d. **>** **for** fib **in** $[1, 1, 2, 3, 5, 8]$ **do** fib^2 **od**

Loops in *Maple* may also continue indefinitely until some criterion is met. Enter the following, and explain what the loop is doing.

  e. **>** **for** i **from** 1 **while** $i^2 < 1000$ **do** i **od**
  f. **>** **for** N **from** 1 **by** 2 **while** $N^3 < 10000$ **do** $N, N^3, N^3 - N^2$ **od**

Create loops to calculate the following

  g. Decimal approximations of e, e^6, e^{11} and so on where the power of e is less than 100.
  h. Decimal approximations of $\dfrac{\pi}{2}, \dfrac{\pi}{3}, \dfrac{\pi}{5}, \dfrac{\pi}{7}, \dfrac{\pi}{11}$.
  i. The Fibonacci numbers less than 1200.

Note: Recall from Section 1.3.3 that the nth Fibonacci number can be calculated using the **fibonacci** function from the combinat package.

16. Implement the following procedure. What does it appear to do?

  > $f := $ **proc**$()$
       local $a, b;$
       global $G, H;$
       $a, b := 1, 2;$
       $G, H := 3, 4;$
     end

Now perform the following in *Maple*.

  a. **>** $a :=$ Pi; a; $f()$; a;   c. **>** $G := \log(Pi)$; G; $f()$; G;
  b. **>** $b := \exp(2)$; b; $f()$; b;   d. **>** $H := \exp(Pi)$; H; $f()$; H;

What is happening to the **local** and **global** variables. What do you think is the difference between a local and a global variable for a procedure. Perform some tests in *Maple* and refer to the help files on procedures (**?procedure**) to confirm your guess.

17. Write procedures to perform the following. Your procedures may have two or more input variables by having a sequence of variable names inside the parentheses after the **proc** declaration. For example **proc**(x, y) or **proc**(a, b, c).

  a. Add all multiples of 5 and 7 less than an arbitrary number.
  b. Generate a sequence using the Fibonacci equation, but from an arbitrary pair of initial conditions.

Be sure to test your procedures with small cases that you can verify by hand.

18. Use nesting to perform the following.

  a. Print out the first 5 rows of Pascal's triangle.
  b. Create a function or procedure that prints out the first N rows of Pascal's triangle

Hint: Try asking *Maple* what it knows about Pascal's triangle. You might need to not use punctuation. Also recall that Exercise 11 dealt with the binomial formula, which is related to Pascal's triangle.

19. Implement the Fibonacci number calculation procedure what used the **remember** option from Section 1.2.7. Make sure that there are no previously computed and remembered values (issue a **restart** command before implementing the function if necessary).

a. Compute the following (in order), taking note of any error messages.

  i. $f(6000)$ iv. $f(2000)$
 ii. $f(4000)$ v. $f(4000)$
iii. $f(3000)$ vi. $f(6000)$

b. Why was it impractical in Section 1.2.7 to find a Fibonacci number that took approximately a second to compute using this procedure?

20. One should be careful when using the **%** operator for creating functions. Enter the following into *Maple*.

a. **>** $sum(k, k = 1..n);$ b. **>** $sum(k, k = 1..n);$
     $f := n \rightarrow \%;$ $f := unapply(\%, n);$
     $'f'(2) = f(2);$ $'f'(2) = f(2);$

What appears to have happened here?

21. Use the **sum** command to find a formula for

a. An arbitrary polygonal number, that is, the sum of the first n terms of an arbitrary arithmetic sequence beginning at 1 with common difference d

b. The sum of the first n terms of an arbitrary arithmetic sequence beginning at a with common difference d

Use induction to prove these formulae.

See Exercise 12 for the definition of an arithmetic sequence and polygonal numbers.

22. Perform the manual calculation of the continued fraction of 1.23456789 (the one involving the **for** loop) from Section 1.3.2 without starting with a rational number. What happens? Why do you think this is happening?

WARNING: Save your worksheet before attempting this. You will probably want to use the "stop sign" icon to cancel computations, when things seem to be going on for too long.

23. Use the **cfrac** function to change the following continued fractions into rational numbers.

a. $[1, 2]$ c. $[1, 1, 1, 2]$
b. $[1, 1, 2]$ d. $[1, 1, 1, 1, 2]$

Do you notice an interesting pattern with these examples? Formulate a conjecture regarding this pattern of continued fractions and the rational numbers they are equal to, and test your conjecture.

Hint: This behavior is related to the Fibonacci numbers and their relation to the golden ratio.

24. Have *Maple* calculate the continued fractions for the following numbers.

a. $\sqrt[3]{2}$ b. $(\sqrt[5]{2})^3$ c. π

Can you see any pattern? Be sure to try several computations, computing different amounts of quotients each time.

25. What is the solution to the general first order recurrence relation

$$a(n) = f(n)\, a(n-1)^p$$

where $p > 1$ is a positive power?

Test this answer by picking some simple functions $f(n)$ (polynomials are recommended) and powers of p and seeing if the first however-many terms in the sequences agree when you calculate the terms both using recursion, and the formula given by **rsolve**.

26. Find general solutions to the following recurrence relations. Verify the solutions.

   a. $a(n) = 5a(n - 1) - 6a(n - 2)$
   b. $s(n) = 2s(n - 2)$

Now find the solutions to the following recurrence relations with initial values. Test the solutions both by testing that the formula satisfies the recurrence, and by having *Maple* calculate the first 10 or 20 terms of the sequence from both the recurrence and the solution.

   c. $f(n) = f(n - 1) + 2f(n - 2)$ where $f(0) = 1, f(1) = -2$
   d. $a(n) = a(n - 1) \cdot a(n - 2)^2$ where $a(0) = 2, a(1) = 3$

27. Find a formula for the number of ways to climb a flight of steps of height n if 1 or 2 steps may be taken at a time.
   Hint: Formulate the problem as a recurrence relation.

28. Use the techniques from Exercise 1.2.7 to measure the time taken for the sieve of Eratosthenes from Section 1.3.4

   a. Measure the time taken to calculate the first $10,000, 20,000, 30,000$, and so on up to the first $90,000$ primes. Use this information to estimate the time required to calculate the first $100,000$ primes.
   b. Remove the part of the procedure that collects the primes into a list. Repeat the previous part with this modified procedure.
   c. Attempt to modify the procedure to more efficiently collate the primes into a list or sequence.

Produce the same lists of primes using *Maple*'s **seq** command and the inbuilt prime number functions, and measure how much time these take. Were these quicker or slower than the Eratosthenes' sieve procedure?

1.5 Further Explorations

This section presents open ended problems for the interested reader. The idea is to introduce mathematics that may be explored with no real constraints. Each topic may ask specific questions, or state particular calculations to be performed, however, these should be considered to be a *beginning* to exploration, and not an exhaustive set of steps to be performed. It is expected and encouraged that one explore these topics using one's own means.

1. The field of rational numbers may be extended to incorporate irrational numbers in a way similar to the way in which the real numbers are extended to the complex numbers. If R is the solution of the equation $x^2 = 2$, then it must be that $R^2 = 2$, and we know that R must be an irrational number.

   If we include our new number R into the rationals (i.e., consider $Q' = \mathbb{Q} \cup \{R\}$) then our new set Q' will no longer be a field. In particular, what is $1 + R$? In order

to have a field, we need to add every rational multiple of R as well as addition of every rational number with every multiple of R.

We end up with what is known as a *field extension*

$$\mathbb{Q}(R) := \{a + bR \mid a, b \in \mathbb{Q}\}$$

with the operations

$$a + bR + c + dR = (a + c) + (b + d)R$$
$$(a + bR)(c + dR) = ac + (ad + bc)R + bdR^2 = (ac + 2bd) + (ad + bc)R$$

where $a, b, c, d \in \mathbb{Q}$.

Maple allows exploration of these ideas with the **evala** (evaluate in an algebraic number, or function, field) and **RootOf** functions. For instance,

```
> R := RootOf(x² = 2)
```
$$R := RootOf(\_Z^2 - 2)$$

```
> R²; evala(R²)
```
$$RootOf(\_Z^2 - 2)^2$$
$$2$$

2. **The $3n + 1$ Problem**. This problem has many other names: Collatz's problem, the Syracuse problem, Kakutani's problem, Hase's algorithm, and Ulam's problem.

We begin with the following simple algorithm that we apply recursively, starting with an arbitrary natural number, n say.

- If n is even then halve it.
- If n is odd then multiply it by 3 and then add 1.

We continue this process with each new number obtained until we end up at 1. The actual problem is does this procedure always terminate? An equivalent formulation of the problem is, if we think of the numbers produced as an infinite sequence, does the sequence always end with a repetition of the subsequence $4, 2, 1$?

For example, if we start with the number 13, then we get

$$13 \to 40 \to 20 \to 10 \to 5 \to 16 \to 8 \to 4 \to 2 \to 1$$

or, equivalently, the infinite sequence

$$\{13, 40, 20, 10, 5, 16, 8, 4, 2, 1, 4, 2, 1, \dots\}$$

For some starting values it will take a large number of steps, and on the way very large numbers might be encountered before the sequence finally begins to drop back to 1. Such sequences are sometimes called hailstone or juggler sequences.

Implement this algorithm in *Maple*. See what happens when you start with 7 (you can even check this particular one without the help of *Maple*). Then try some other starting values. The sequence starting at 27 takes one hundred and eleven steps to reach 1. You might experiment with the rule a little: for example, what happens if you change the 3 to a 5, thus making a "$5n + 1$" rule?

You should ask yourself how (i.e., under what circumstances) the system could possibly fail to terminate. For this to happen there must be a starting value that either diverges or settles into an infinite loop (other than $4, 2, 1$).

3. A real number is said to be *normal* if its numeric expansion in any base has a
normal distribution. This is to say that each single digit is equally likely to appear,
as is each pair of digits, each triple of digits, and so on. There are not many known
normal numbers, and none are known that were not constructed to be normal in
the first place. It has been conjectured, however, that the number π is normal.

We may obtain a "feel" for whether a number may be normal by counting its digits.
If we count, say, 100 digits of a number, and find that there are, give or take, 10
of each digit then it's possible the number is normal. If we find this general idea
holds for 1000, and even 10,000 digits (with approximately 100 or 1000 of each
digit, respectively) the hypothesis looks even stronger. We could carry on and then
count each pair of digits and see if there's roughly an even number of those. We
could continue in this fashion, and even perform the same sort of analysis using a
different base.

Note that a number may be normally distributed in a particular base representation,
but not be a normal number. Such a number, if normally distributed in base b is
said to be b-normal.

Some good numbers to begin looking at for possible normality are

$$\frac{1+\sqrt{5}}{2}, \sqrt{\frac{1+\sqrt{5}}{2}}, \pi, e, \log 2, \sqrt[3]{2}$$

Generate 100, 1000, and 10000 decimal digits of these numbers (more, if you wish),
and count the digits, and pairs of digits. If you're feeling adventurous try triples
and even quadruples.

Try with another base, binary say. You'll need to ask yourself, with what probability
should each binary digit occur for normal distribution. With what probability should
each double (as well as triple or quadruple if you test those) occur?

There is a similar analysis that can be performed on the continued fraction of a real
number. The Gauss–Kuz'min distribution states that the probability that $a_n = k$
(in the continued fraction expansion $[a_1, a_2, \dots]$ of a random real number) is

$$\text{Prob}(a_n = k) = \log_2 \left[1 - \frac{1}{(k+1)^2} \right]$$

In other words, approximately 42% of numbers in a continued fraction expansion
will be 1, approximately 17% will be 2, around 9.4% will be 3, and so on. Note
that because this is a continued fraction expansion, then it is possible for numbers
greater than 9 to be in the expansion; in fact any natural number may be included
(you should find that the infinite sum of these probabilities is equal to 1 as you
would expect). As such, the distribution also tells us that approximately 1.4% of
numbers in the expansion will be 10, 0.55% will be 50, and 0.14% will be 100.

Chapter 2
Calculus

In this chapter we explore calculus with the help of *Maple*. Much of what is seen in this chapter should be familiar (or at least recognized) from first-year study. We aim to revise this material, as well as visualize it in ways that are, one hopes, easily accessible and in addition provide new insight even to the more capable reader. We also attempt to introduce some newer (or, at least, less familiar) material. In all cases here the aim is to use *Maple* to complement and improve our own calculus skills; the goal is one of a human/machine collaboration, not that of an electronic replacement for calculus skills.

2.1 Revision and Introduction

In this section we introduce the *Maple* commands best suited for studying and performing calculus. In addition we recall key concepts from typical first-year calculus courses. It is, however, expected that the reader is familiar with the underlying concepts and is able to perform such first-year calculus including (but not limited to) differentiation and integration of single variable functions, evaluation of limits, and curve sketching, among others. The reader is encouraged to review his or her favorite (or most readily available) calculus text.

2.1.1 Plotting

In much first-year calculus the ability to be able to visualize the functions and concepts being studied is quite valuable, but usually not readily available. Indeed in almost anything involving calculus visualization is a powerful tool. So we begin this calculus chapter by looking at how to have *Maple* plot functions.

Briefly in Section 1.1.2 we saw a plot of a cubic. The *Maple* command to plot a function is, unsurprisingly, **plot**. The most basic use of the **plot** function is to simply give it an expression involving a single variable, usually x but any valid *Maple* variable name will work just as well.

> $plot(x^2)$
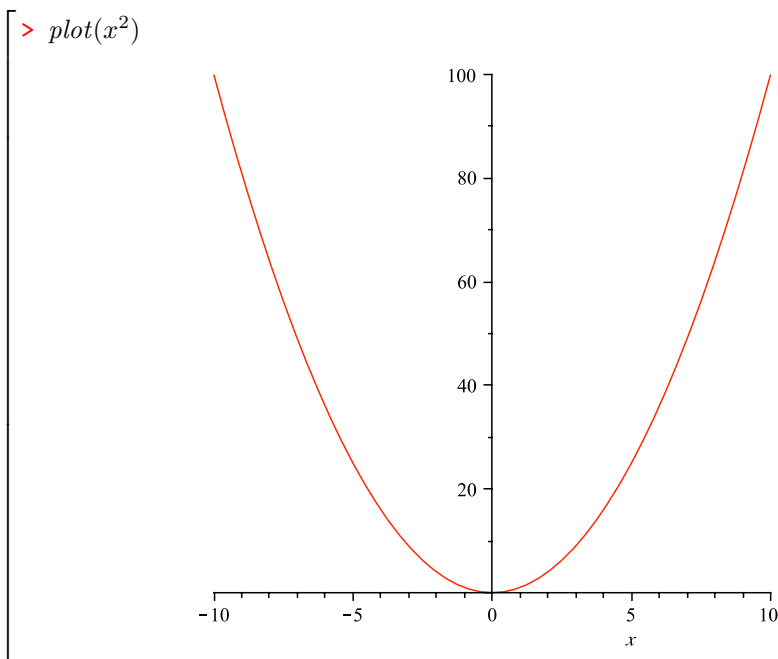
Maple automatically assigns the horizontal axis to be the axis for the independent variable (x in the previous example) and labels it as such. The vertical axis is unnamed, but depicts—as it always does—the values of the function (or the dependent variable). Also, unless we tell it otherwise, the **plot** command will plot over the horizontal interval $(-10, 10)$. If we wish to plot over a different interval, we must provide a second argument.

> $plot(\sin(t), t = 0..2 \cdot Pi)$
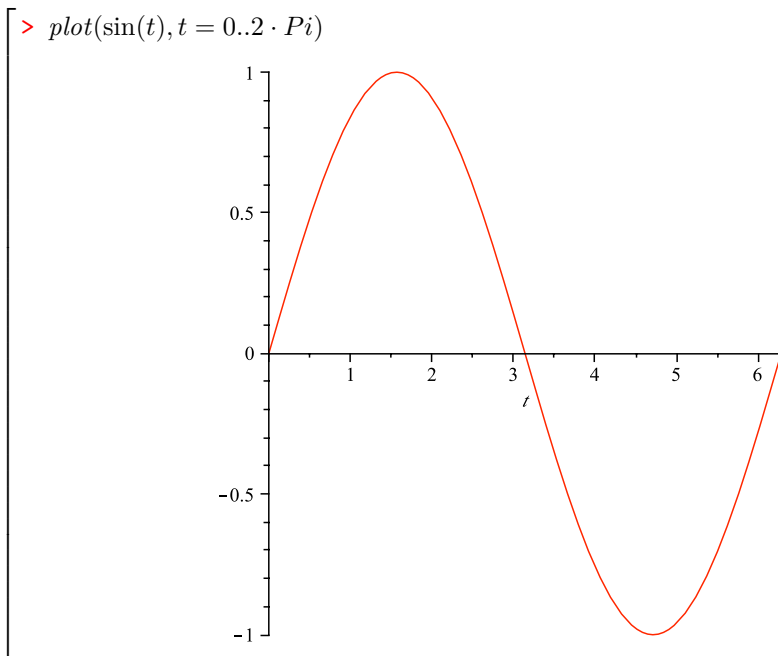
Notice here that our horizontal axis is now t, where it was x for the parabola example. Notice the vertical axis in these two examples. The values of the vertical axis automatically adjust to suit the function we are trying to plot. For the parabola the vertical axis was between 0 and 100, corresponding to the values the parabola would

have for $-10 \le x \le 10$, and the sine curve used vertical range $(-1, 1)$ just as we would have hoped it did. It should be interesting to note that the actual screen space taken up by the two plots is the same in both cases, showing that the vertical scale is different in both cases. In fact, the vertical scale and horizontal scale may be different even in the same plot, as is the case in both of these examples.

If we have a function to plot—instead of an expression—there is another way we may ask for a plot. Note above that although sin is definitely a function, $\sin(t)$ is actually a *Maple* expression. The former will take an input and produce an output, but the latter will not take any input at all. If we have a function or procedure we wish to plot, we may omit the input from the function/procedure, and omit the variable name from the second input parameter. For example, to plot the tan function between its asymptotes we can use the following.
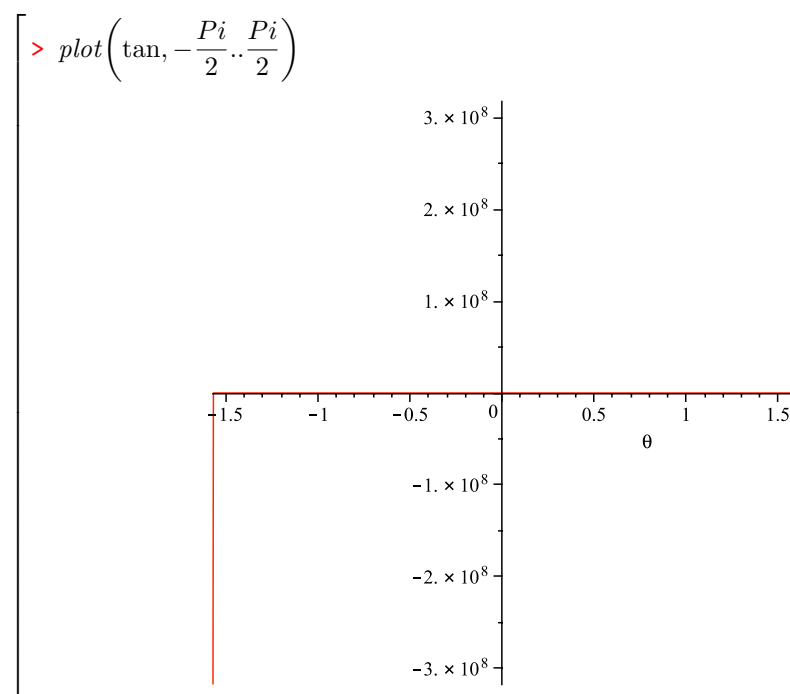
> $plot\left(\tan, -\dfrac{Pi}{2} .. \dfrac{Pi}{2}\right)$

This is equivalent to the command

$$plot\left(\tan(t), t = -\dfrac{Pi}{2} .. \dfrac{Pi}{2}\right)$$

Unfortunately the plot we see certainly doesn't look like the tan function that we all know and love. If we have a look at the first plot we produced, we should notice the scale of the vertical axis is exceptionally large, and that we have two seemingly vertical lines at the end of the graph. If we were to somehow limit the vertical interval (and thus the vertical scale), perhaps we'd get a better picture.

As it happens, we may do just this with a third argument to the plot function. This third argument must be a range. It may be in the form of an interval $(a..b)$ or may be assigned to a variable $(var = a..b)$. In the latter case, the vertical axis is labeled with the variable name. In either case, however, the vertical axis range must be present after the horizontal axis range. For example, to refine our plot of the tan function we might do the following.

> $plot\left(\tan, -\dfrac{Pi}{2}..\dfrac{Pi}{2}, -5..5\right)$
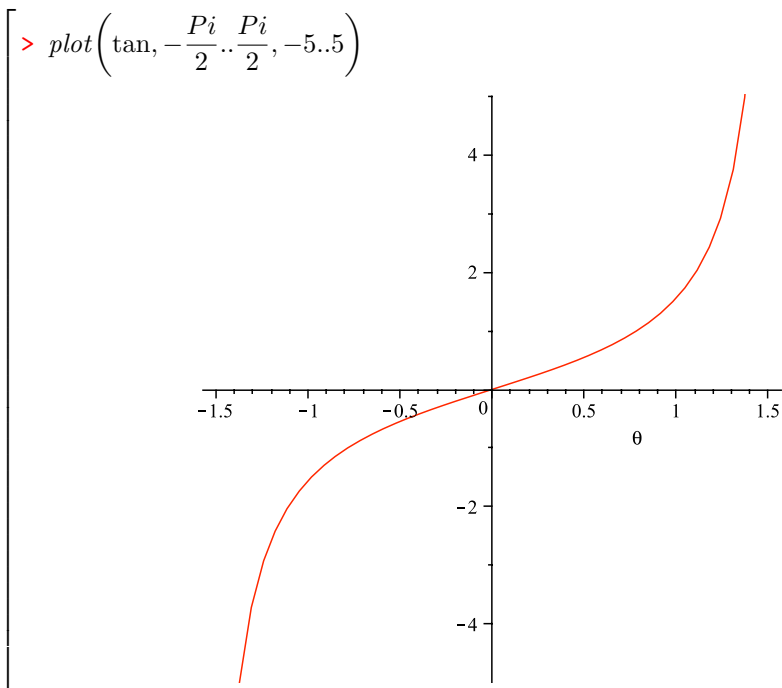
We now have a much more familiar plot. To better understand why this helped, we need to know how *Maple* plots a function.

When *Maple* plots a function it evaluates that function at various points along the horizontal interval and fits a curve to the sampled points. We see this sampling behavior by asking *Maple* to plot points, instead of a line, as follows.

> $plot(\sin, 0..2 \cdot Pi, style = point)$
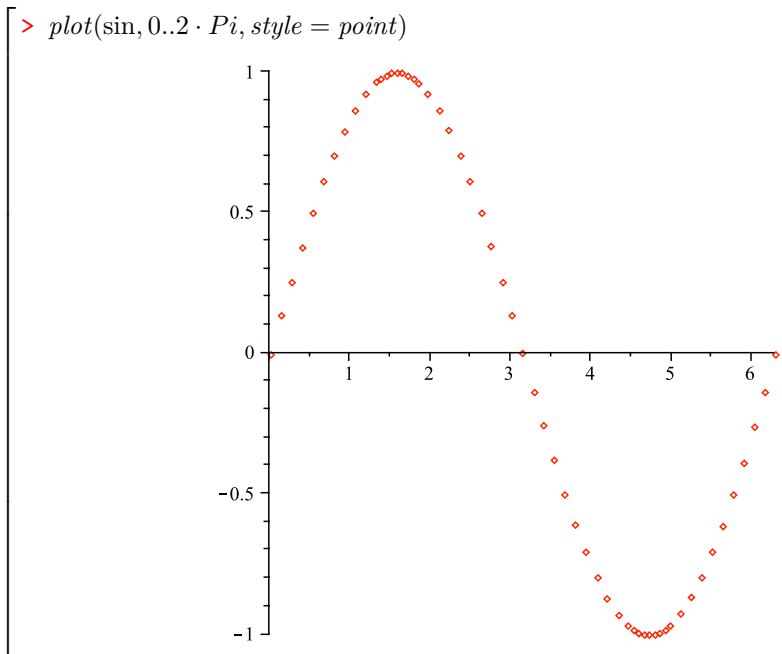
By default, *Maple* uses 50 sample points, but the number of points may be specified by the **numpoints** input parameter to the **plot** function.

Usually the sample points are evenly spaced along the horizontal range, however, *Maple* is rather clever and if it detects that the values of the function are changing too quickly between sample points, it will sample the function at an extra point between them to try to obtain more and better information with which to plot the function. We observe this happening at the peaks of the curve plotted above.

This extra sampling—known as subdivision—can, by default, happen up to 6 times between any two sample points and so one should be aware that *Maple* could potentially end up evaluating a function at as many as 6 times the number of sampling points requested. This behavior is, of course, able to be controlled and modified using input parameters (see **?plot/options**).

Now, if we have a look at the tan plot in point mode, we see two extreme points far to the top and bottom of the graph, with the remainder of the sampled points on or very near the x-axis.

> $plot\left(\tan, -\dfrac{Pi}{2}..\dfrac{Pi}{2}, style = point\right)$
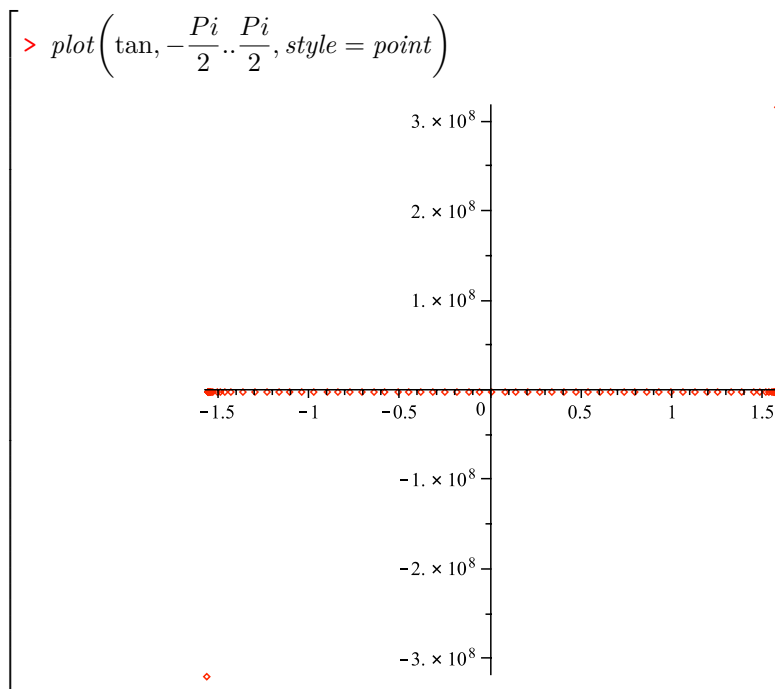
Of course, with the vertical range so large, the scale is such that the points seemingly on the x-axis could have values varying anywhere between $\pm 1{,}000{,}000$ or more and we wouldn't be able to tell the difference.

Without explicitly specifying the vertical range, *Maple* adjusts it accordingly between the largest and smallest sampled values. It should be clear then that by specifying the vertical range we see only the plot generated from the sampled points inside that range. It is worth noting that even with a specified vertical range *Maple* still samples all the same points as it would have without the specified range.

2.1.2 Multiple Plots

An interesting example comes to us from Borwein and Devlin [5]. Suppose we have two expressions, $y - y^2$ and $-y^2 \log(y)$, and wish to know (and eventually prove) which (if either) is always larger when $y \in (0, 1)$. A good first step would be to plot the two

expressions over the unit interval, to see if the curves cross each other. However, up until now we have only plotted single expressions. Two separate plots are of limited (if any) use to us. We need a way to plot the two expressions on the same pair of axes. This is made possible in one of two ways.

The first method is far and away the simplest. *Maple*'s **plot** command will happily plot a list of expressions (or functions). In place of a single expression we simply provide a list, and any parameters that modify the plots must also be provided in a list. For example, to plot the above two functions, with the first one being colored the usual red, and the second colored blue (so we may identify which is which) we would enter the following.

> $plot([y - y^2, -y^2 \cdot \log(y)], y = 0..1, color = [red, blue])$
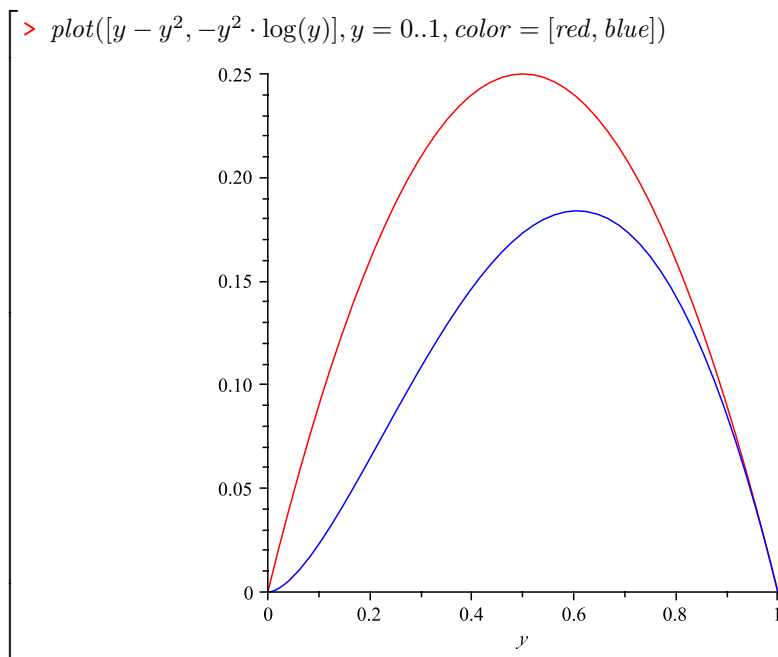
For the second method, let us now consider a modified version of our example. This time we compare $y^2 - y^4$ and $-y^2 \log(y)$ (also from [5]). First we will make a simple observation: up until now, any valid *Maple* expression could be assigned to a variable name. It should, then, be a natural question to ask whether the same can be done with the plot function. The answer is that yes it can, although the logistics of such an assignment are probably not obvious. Let's try this.

> $plot1 := plot(y^2 - y^4, y = 0..1, color = red);$
$plot2 := plot(-y^2 \cdot \log(y), y = 0..1, color = blue)$

$$plot1 := PLOT(\ldots)$$
$$plot2 := PLOT(\ldots)$$

Maple stores what is known as a plot structure in the variable. If we wish to see the actual plot, then we may either just ask *Maple* for the contents of the variable in the usual way, or we may use the **display** function from the plots package.

> *plot1*
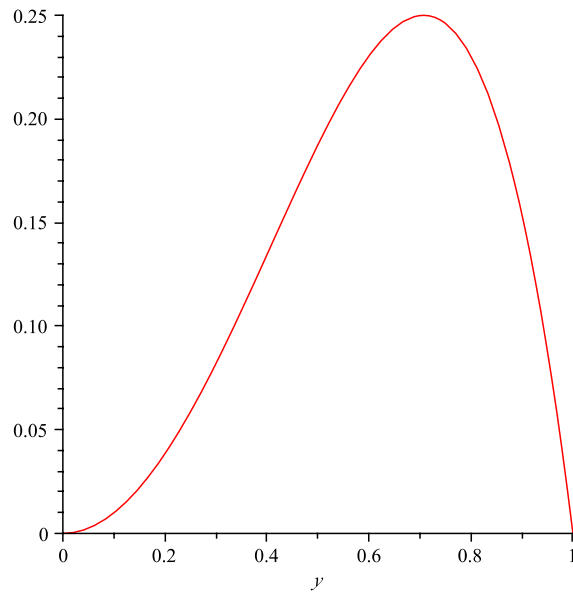
> *plots*[*display*](*plot2*)
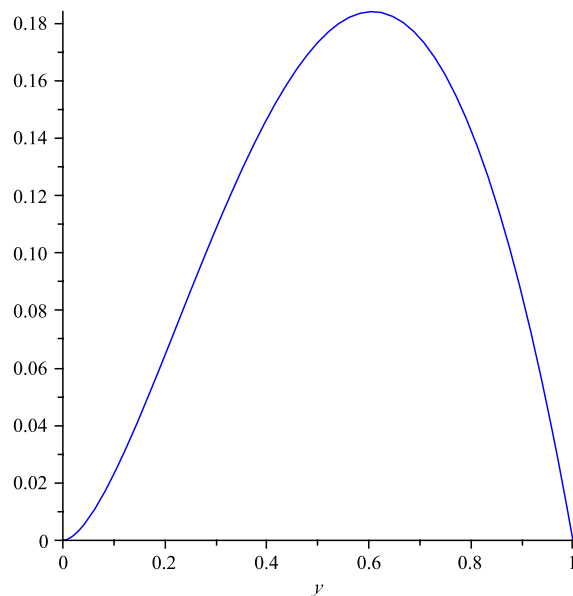
This is all well and good, but it doesn't really help us show multiple plots on the same axes, at least not in any different way from the method we have already used. The key to this lies in the **display** function, whose entire purpose is not so much the display of single stored plots (which, as we have seen can be displayed easily without the use of this function), but multiple plots. Just as the plot function will accept a list of expressions and plot them on the same axes, so will **display** accept a list (or set) of plot structures, and display them all on the same axes.

> $plots[display]([plot1, plot2])$
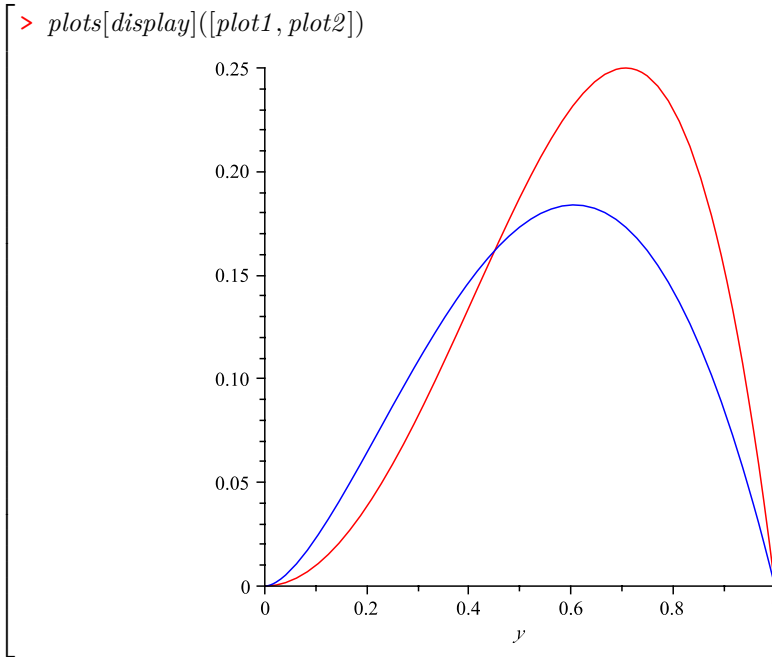
The use of **display** for the above example above may seem unnecessarily long and complicated, when we could more easily use just a single **plot** as we did in the first example. Such an observation is quite well founded. However, there are situations where the easier method is either impossible, or impractical to use. It is these cases where **display** really shines.

To illustrate this utility, we show the previous example and a plot of its sample points at the same time. In fact, we use both of the above techniques at the same time. We use a single **plot** command to produce the two curves, then a second single **plot** command two produce the two curves as points (using the $style = point$ option).

It is possible to produce the same result with a single **plot** command, but is long and difficult to read, whereas the following commands are quite descriptive, and easy to follow.

> $f := y^2 - y^4; g := -y^2 \cdot \log(y)$

$$f := y^2 - y^4$$

$$g := -y^2 \cdot \log(y)$$

> $curves := plot([f, g], y = 0..1, color = [red, blue]) :$
  $points := plot([f, g], y = 0..1, color = [red, blue], style = point) :$

We now combine these two plots using the **display** function, and have the plot we wished to see. We use this technique again later in the book when demonstrating sequences whose points lie on continuous functions in Section 2.1.3.

> $plots[display]([curves, points])$
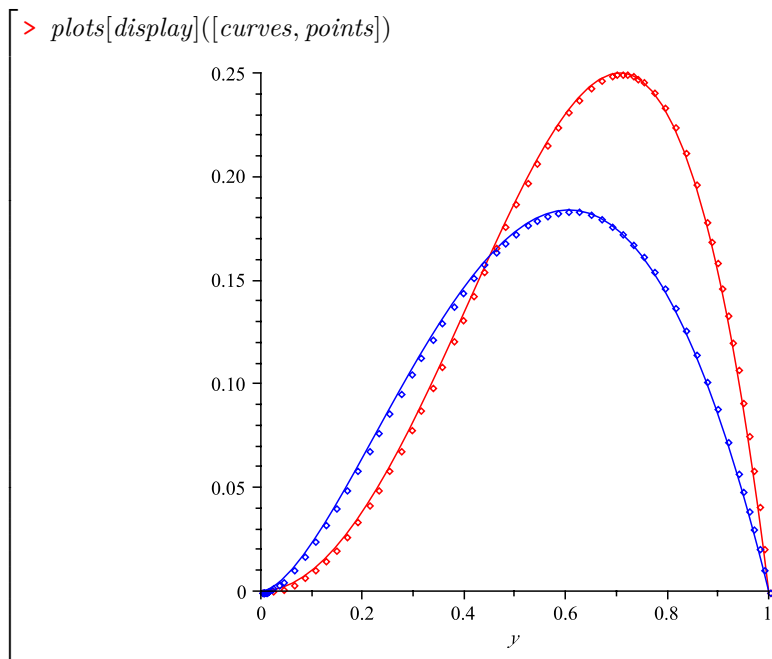
Display is also a very valuable tool when a function plot is difficult and time consuming and we want to reuse it quickly and with certainty that it is what we wanted to draw.

2.1.3 Limits

Calculus, ultimately, all comes down to limits. So it is with limits that we begin our exploration of calculus proper. We have already seen and used very briefly in the previous chapter, the *Maple* **limit** command. We look at it in more detail here, and recall quickly the math behind limits.

We may take a limit of a sequence (of the infinite variety) or of a function. The intuitive (and mathematically imprecise) notion of a limit is a value that we may approach as closely as we could ever wish, just by traveling sufficiently far along the sequence or function.

We start with sequences. Recall that the limit, L say, of some sequence

$$\{x_n\}_{n=1}^{\infty} = x_1, x_2, x_3, \cdots$$

written

$$\lim_{n \to \infty} x_n = L$$

is a number such that for every $\epsilon > 0$ we can find a natural number N so that whenever we have any other number $n \geq N$ it will be the case that $|x_n - L| \leq \epsilon$.

The sequence

$$\left\{\frac{1}{k}\right\}_{k=1}^{\infty} = 1, \frac{1}{2}, \frac{1}{3}, \cdots$$

should be familiar and has, of course, limit 0. We ask *Maple* to verify this. The *Maple* **limit** command, just like **sum** and **prod** has both an inert and active form, with the inert form being the one with the capital "L".

> $f := k \rightarrow \dfrac{1}{k}; Limit(f(k), k = infinity) = limit\left(\dfrac{1}{k}, k = infinity\right)$

$$f := k \rightarrow \frac{1}{k}$$

$$\lim_{k \to \infty} f(k) = 0$$

As usual, the **value** command will perform the active calculation on the inert form.

We may also, if we wish a more visual clarification of the convergence, plot this. We could simply just plot the continuous function $1/k$ to see the convergence (using the fact that if the function converges, then the sequence evaluated only at integer points also converges). Instead, however, we use *Maple*'s point plot option and see only the points of the sequence in our visualization.

To do this we construct a sequence (*Maple* sequence, that is) of 2-element lists $[x, y]$ each of which represent a point in the Cartesian plane. Because we are plotting a sequence, we choose the x-axis to be our index, and the y-axis to be the sequence element. As such the points are $[k, 1/k]$. We look at the first 100 points. This sequence of lists is then put into a containing list, so the **plot** function does not get confused.

> $plot\left(\left[seq\left(\left[k, \dfrac{1}{k}\right], k = 1..100\right)\right], style = point\right)$
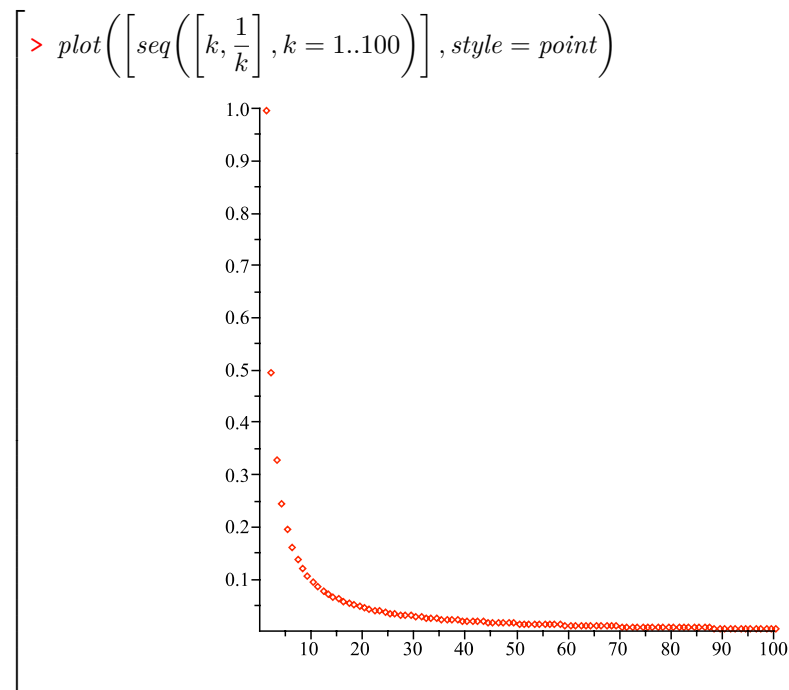
The convergence is visually pretty clear. It is worth stressing at this point, however, that these plots give an *indication* of convergence, not a proof of convergence. There is always the possibility that the sequence does something odd after the interval we have plotted. So we must still perform regular mathematics to verify the limits, or at the very least ask *Maple* to evaluate the limit.

Recall now that an infinite sum is defined to be a limit of its partial sums. Mathematically, that is,

$$\sum_{k=1}^{\infty} f(k) = \lim_{N \to \infty} \sum_{k=1}^{N} f(k)$$

We have already used *Maple*'s **sum** command to calculate infinite sums for us in Section 1.1.5, but we do this again now, and demonstrate the limit property. Let us use the series $1/k^2$ again, which we know from the previous section converges to $\pi^2/6$.

> $sum\left(\dfrac{1}{k^2}, k = 1..N\right);$
> $Limit(\%, N = infinity) = limit(\%, N = infinity)$

$$-\Psi(1, N+1) + \frac{1}{6}\pi^2$$

$$\lim_{N \to \infty} \left(-\Psi(1, N+1) + \frac{1}{6}\pi^2\right) = \frac{1}{6}\pi^2$$

We have not seen the Ψ function before, although we may ask *Maple* about it by using the command **?Psi**. We should expect, due to the algebra of limits, that $\lim_{N \to \infty} -\Psi(1, N+1) = 0$. It is always good to cross check answers we are unfamiliar with, we therefore do so.

> $limit(-\Psi(1, N+1))$

$$0$$

> $plot(-\Psi(1, N+1), N = 1..100)$

The convergence in the picture seems pretty clear, and combined with the results *Maple* gave us for the infinite sum as well as the limit of the partial sums, and the limit of the Ψ function—not to mention the numeric approximations we saw in the previous section—we may be quite confident of the validity of the answer.

Now let us look at limits of continuous functions. For this purpose we consider the function $f(x) = 1/(x^3 - 2x^2)$. Our first impulse should be to plot the function to see what it looks like, but doing so produces something that is not so useful.

> $plot\left(\dfrac{1}{x^3 - 2x^2}\right)$
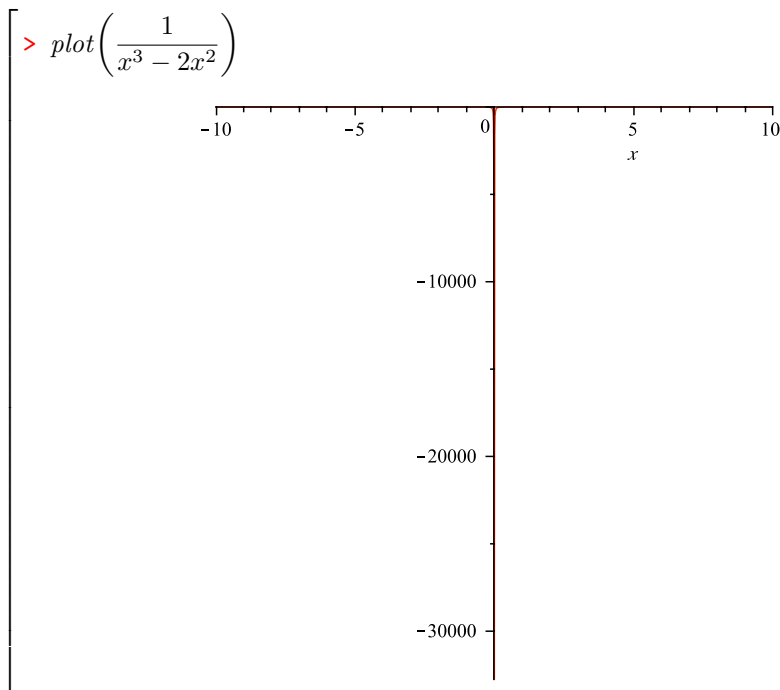
We could attempt to use trial and error to find a good interval to plot over, but instead we perform some calculus. First note that the denominator is equal to $x^2(x-2)$, which tells us that the function is not defined at $x = 0$ or $x = 2$ and that we should probably expect vertical asymptotes at those points. It should also be clear, using the algebra of limits, that

$$\lim_{x\to a}\frac{1}{x^3 - 2x^2} = \lim_{x\to a}\frac{\frac{1}{x^3}}{1 - \frac{2}{x}} = \frac{\lim_{x\to a}\frac{1}{x^3}}{1 - \lim_{x\to a}\frac{2}{x}}$$

and so for $a = \pm\infty$ the limit will be 0.

The limits at $a = 0$ and $a = 2$ are only a little bit trickier to work out. We again look at the factored denominator $x^2(x-2)$ and observe that

$$\lim_{x\to 0} x^2(x-2) = 0 \text{ and } \lim_{x\to 2} x^2(x-2) = 0$$

Furthermore we can see that x^2 will always be ≥ 0, and that for all points near $a = 0$ it is the case that $x - 2 < 0$ so the denominator near $a = 0$ must always be negative. We conclude therefore that the limit at $a = 0$ is $-\infty$. Finally observe that for $x > 2$ we have $x - 2 > 0$ and that for $x < 2$ we have $x - 2 < 0$ which tells us that $f(x) \to -\infty$ as $x \to 2^-$ and that $f(x) \to \infty$ as $x \to 2^+$

We now have plenty of information for us to choose appropriate plotting ranges. In order to put the undefined points (with the vertical asymptotes) evenly spread across the x-axis we plot across the interval $x \in (-2, 4)$. However, because we know that the function has vertical asymptotes, we should limit the y-axis range somewhat. A quick substitution of $x = 1$ into the function shows that the midpoint between the vertical asymptotes attains the value -1, and so we take a reasonable guess that $y \in (-5, 5)$ will suffice.

> $plot\left(\dfrac{1}{x^3 - 2x^2}, x = -2..4, y = -5..5, discont = true\right)$
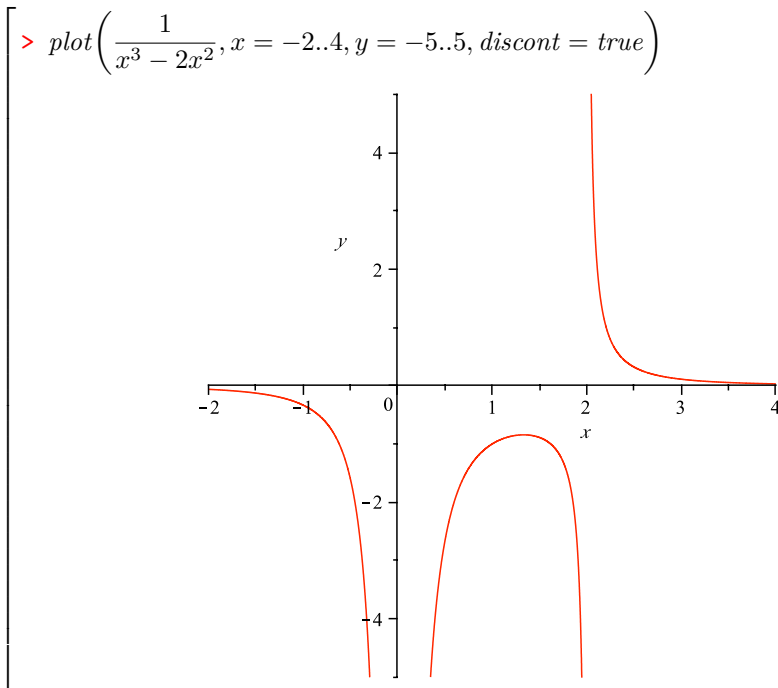
The $discont = true$ argument to the plot command simply tells **plot** that we are plotting a discontinuous function allowing for a better plotting. Without that argument, **plot** assumes it is plotting a continuous function and ends up putting vertical lines at points of discontinuity, or at asymptotes.

We now verify the limits with *Maple*.

> $f := \dfrac{1}{x^3 - 2x^2};$
> $limit(f, x = -infinity), limit(f, x = 0), limit(f, x = 2), limit(f, x = infinity)$

$$f := \frac{1}{x^3 - 2x^2}$$

$$0, -\infty, undefined, 0$$

The undefined limit at $a = 2$ should be completely unsurprising thanks to the plot we performed above. The limit from below and the limit from above are not the same. Recall that $\lim_{x \to a^-} = \lim_{x \to a^+} = L$ if and only if $\lim_{x \to a} = L$. Fortunately for us *Maple* can handle single directional limits by allowing the **limit** command to take a third input variable for this purpose, which may be one of *left* or *right*. As might be expected, the left limit at a is when $x \to a^-$ and the right limit at a is when $x \to a^+$.

> $Limit(f, x = 2, left) = limit(f, x = 2, left);$
> $Limit(f, x = 2, right) = limit(f, x = 2, right);$

$$\lim_{x \to 2^-} \frac{1}{x^3 - 2x^2} = -\infty$$

$$\lim_{x \to 2^+} \frac{1}{x^3 - 2x^2} = \infty$$

We now return to sequence limits, and a cautionary example. *Maple*'s **limit** function calculates real- or complex-valued (function) limits. When we used it to calculate sequence limits above, what we were really doing was evaluating the real-valued limit at infinity of the functions in question, and using the theorem which states that if

$f(x) \to L$ as $x \to \infty$ exists for a real-valued function f, then the sequence $\{f(n)\}_{n\in\mathbb{N}}$ converges to the same limit as $n \to \infty$.

However, caution is advised. This relationship between limits of functions and limits of sequences does not always work the other way around. Consider the function

$$f(x) = x\sin(Pi \cdot x) + 2\tanh(x - 5)$$

and the corresponding sequence $\{f(n)\}_{n\in\mathbb{N}}$. If we evaluate limits or plot the function we might very well be tempted to conclude that the sequence does not converge.

> $f := x \to x \cdot \sin(\text{Pi} \cdot x) + 2 \cdot \tanh(x - 5);$
> $limit(f(x), x = infinity)$

$$f := x \to x \cdot \sin(\pi \cdot x) + 2 \cdot \tanh(x - 5)$$
$$undefined$$

> $plot(f)$
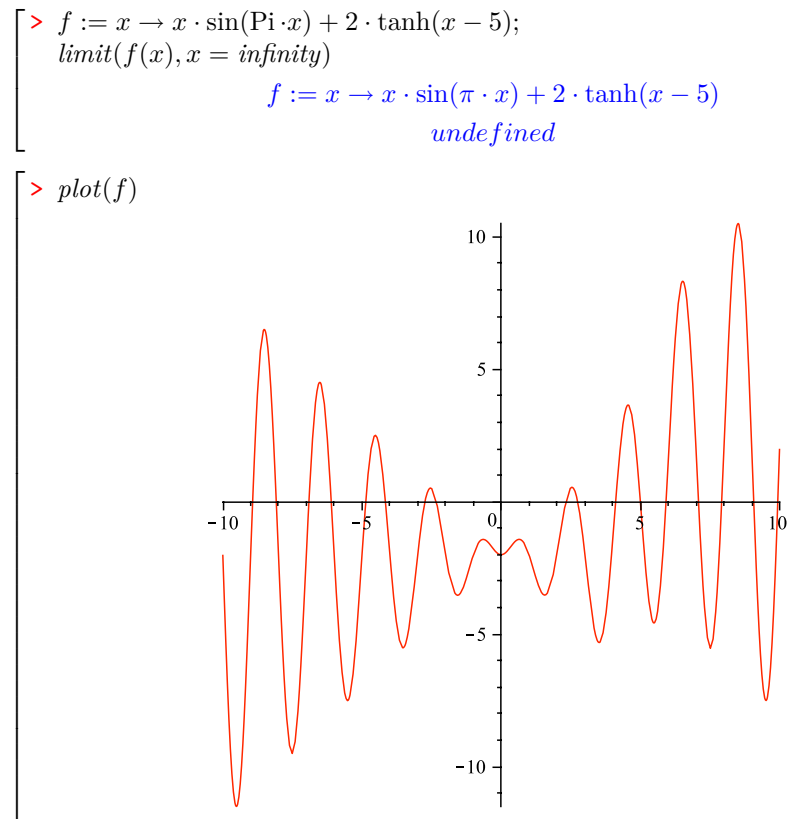
We clearly see a function with no limit at infinity. However, plotting only the sequence points shows an entirely different picture.[1]

> $plot([seq([n, f(n)], n = 1..100)], style = point)$

---
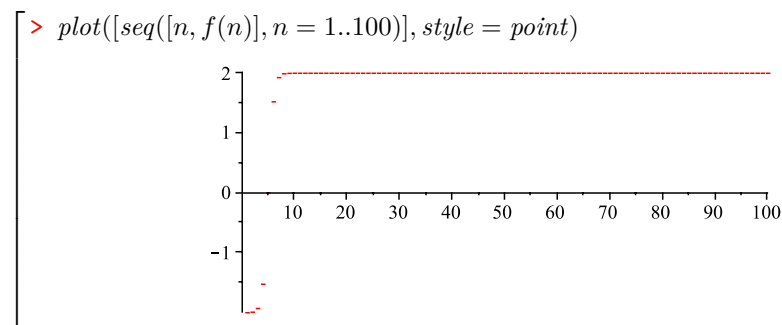
[1] The astute reader will notice that this plot is not square, as the others are. This is because, due to the information displayed in the plot and aesthetic reasons, the author manually shrunk the height (using the mouse and a drag action).

Our sequence has what looks very much like a limit. Applying our calculus knowledge, we see that $n \sin(\pi n) = 0$ for $n \in \mathbb{N}$, and so that part of the function will not affect the sequence at all. Furthermore, because $\tanh(x) \to 1$ as $x \to \infty$ it must be the case that $2 \tanh(x - 5) \to 2$ as $x \to \infty$. In short, the sequence $\{f(n)\}_{n \in \mathbb{N}}$ ought to converge to the value 2, and the plot of the sequence points exhibited precisely this behavior.

If we plot the real-valued function and the sequence on the same axes, we can see the convergence a little better.

```
> p1 := plot(f, 0..20, color = grey) :
  p2 := plot([seq([n, f(n)], n = 1..100)], style = point) :
  plots[display]([p1, p2])
```
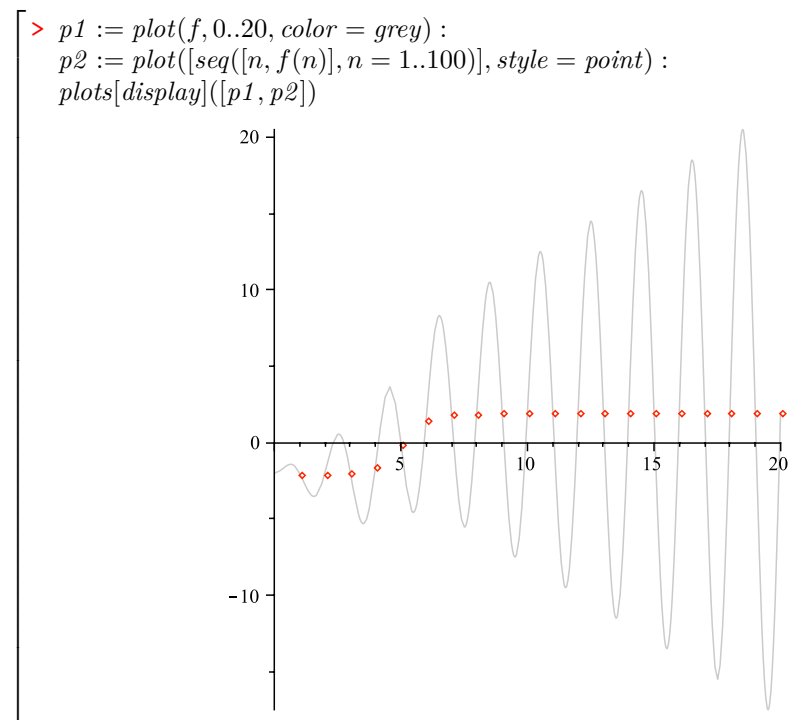

All that remains is to see if we may convince *Maple* to provide us with the correct answer for the sequence limit. Because we are evaluating a limit at infinity we are already evaluating a left limit and cannot possibly try to change to a right limit or a bidirectional limit. A good attempt would be to use the **assuming** keyword.

```
> limit(f(n), n = infinity) assuming n :: posint
```
$$undefined$$

Unfortunately, as it turns out, due to n being a dummy variable inside the **limit** function, the assumptions we requested for n do not quite seem to be being applied. See **?assuming** for more details. The solution is to use the **assume** command, which works in a similar way to **assuming** except that the assumptions become globally accepted for the entire worksheet (as opposed to just the current command). So we make sure to explicitly reset the n variable after performing our calculation.

```
> assume(n :: posint); limit(f(n), n = infinity); n := 'n' :
```
$$2$$

To reiterate the key points here, the lack of a limit of a real valued function does not imply the lack of a limit of a sequence of evaluations of that function, and—more important—one must always keep one's wits about one when using a CAS (of course this is true when reading a book or taking a bus too).

2.1.4 Differentiation

Differentiation is, fundamentally, all about calculating rates of change. Recall that the derivative of a function, $f(x)$ say, at any point a is the slope of the tangent line to f at the point a. Recall, also, that the tangent at a is defined to be the line through a with slope equal to the limit of the slopes of lines drawn between a and points near a, as depicted in Figure 2.1. So it is that we come to the limit definition of the derivative
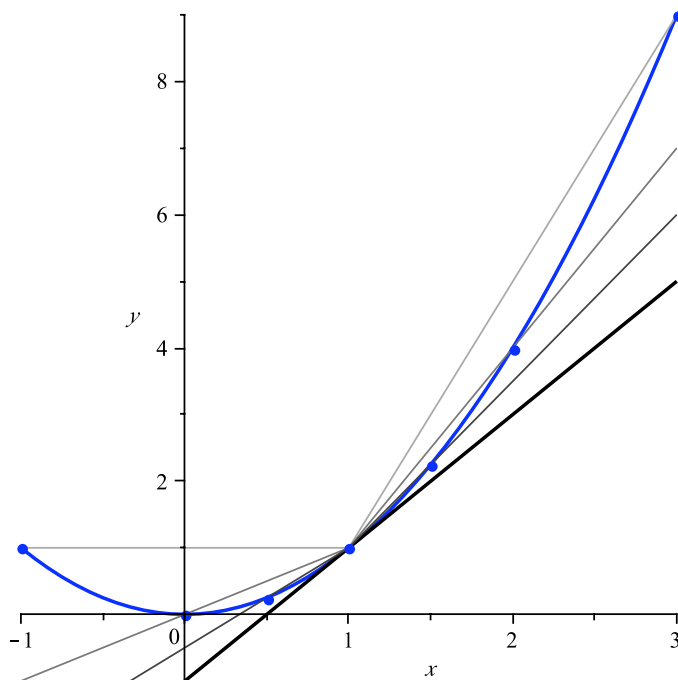
Fig. 2.1 Depiction of the convergence of lines to the tangent.

at a point a as

$$f'(a) = \lim_{x \to a} \frac{f(x) - f(a)}{x - a} = \lim_{h \to 0} \frac{f(a + h) - f(a)}{h}$$

Let us explore this a little before we introduce *Maple*'s native differentiation commands. We start with a parabola $f(x) = x^2$ and so, of course, we know that $f'(x) = 2x$ meaning that the tangent to the parabola at any point a has slope $2a$. We write a small procedure to output the limit and its value.

```
> d := proc(f :: procedure, a)
    Limit( (f(a + h) - f(a))/h, h = 0 ) = limit( (f(a + h) - f(a))/h, h = 0 )
  end :
```

> $d(x \rightarrow x^2, 1); d(x \rightarrow x^2, 2); d(x \rightarrow x^2, x)$

$$\lim_{h \to 0} \frac{(1+h)^2 - 1}{h} = 2$$

$$\lim_{h \to 2} \frac{(2+h)^2 - 4}{h} = 4$$

$$\lim_{h \to 0} \frac{(x+h)^2 - x^2}{h} = 2x$$

If we look a little closer at the final limit, we should see that

$$(x + h)^2 - x^2 = x^2 - 2hx - h^2 - x^2 = h(2x - h)$$

and so the limit then becomes

$$\lim_{h \to 0} \frac{h(2x - h)}{h} = \lim_{h \to 0} (2x - h) = 2x$$

thus verifying both *Maple*'s limit calculation and our regular differentiation technique (for the parabola, at least).

Turning our eye now to trigonometric functions, we choose the sin function and the point $\pi/2$. This produces a limit that is a little trickier to work out on paper.

> $d\left(\sin, \dfrac{Pi}{2}\right);$

$$\lim_{h \to 0} \frac{\cos(h) - 1}{h} = 0$$

As ever, our instinct should be to plot the function. However, to be on the safe side, we ask for a lot of sample points, to try to get a good idea of the behavior near the $h = 0$ point which we know is undefined.

> $plot\left(\dfrac{\cos(h) - 1}{h}, h = -1..1, numpoints = 1000\right)$
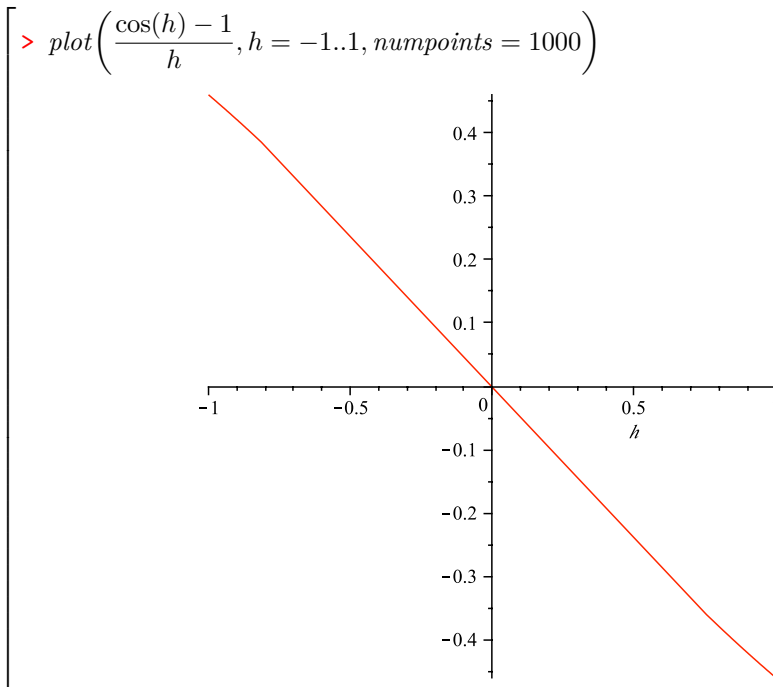
Of course, in order to perform differentiation in *Maple* we need not perform limit calculations each and every time. There are two functions, named **diff** and **D**, that obtain the derivative of a function. The difference between them is that **diff** takes an expression as input and produces an expression as output, and **D** takes a function as an input and produces a function as an output. It should also be mentioned that **diff** also has an inert form **Diff**.

> $diff(x^2 + x + 1, x)$; $diff(\exp(2x), x)$; $Diff(\sin(x), x) = diff(\sin(x), x)$

$$2x + 1$$

$$2e^{2x}$$

$$\frac{d}{dx}\sin(x) = \cos(x)$$

Notice that we need to tell **diff** with respect to which variable we are taking the derivative. For similar differentiation using the **D** function, we have the following.

> $D(x \to x^2 + x + 1)$; $D(\exp)$; $D(\sin)(\text{Pi})$

$$x \to 2x + 1$$

$$\exp$$

$$-1$$

Note that inasmuch as **D** produces a function, we may immediately pass input to it for evaluation, as we did in the last calculation above (which quite correctly calculated $\cos(\pi) = -1$).

If we wish to calculate a second derivative, we do the following.

> $p := x \to x^2 + x + 1$; $diff(p(x), x, x)$; $D^{(2)}(p)$;

$$p := x \to x^2 + x + 1$$

$$2$$

$$2$$

Note that the parentheses around the 2 in $D^{(2)}$ are very important, and their omission will cause the command to fail. In general, for the kth derivative we use $diff(f(x), x, x, \ldots, x)$ where the sequence of xs is of length k (which we may abbreviate $x\$k$), or we use $D^{(k)}(f)$.

> $p := x \to 4x^5 - 3x^2 + x - 2$:
  for k **from** 1 **while** $D^{(k-1)}(p)(x) \neq 0$ **do**
      $Diff(p(x), x\$k) = D^{(k)}(p)(x)$
  od

$$\frac{d}{dx}(4x^5 - 3x^2 + x - 2) = 20x^4 - 6x + 1$$

$$\frac{d^2}{dx^2}(4x^5 - 3x^2 + x - 2) = 80x^3 - 6$$

$$\frac{d^3}{dx^3}(4x^5 - 3x^2 + x - 2) = 240x^2$$

$$\frac{d^4}{dx^4}(4x^5 - 3x^2 + x - 2) = 480x$$

$$\frac{d^5}{dx^5}(4x^5 - 3x^2 + x - 2) = 480$$

$$\frac{d^6}{dx^6}(4x^5 - 3x^2 + x - 2) = 0$$

2.1.5 Integration

Integration grows out of the problem of calculating area underneath a curve, although its applications are far more wide and varied than that. Recall that a definite integral of a continuous function f between two points a and b may be approximated by a limit of rectangles.

$$\int_a^b f(x)\,dx = \lim_{n\to\infty}\left(\Delta x \sum_{k=1}^n f(a + k\Delta x)\right) \ \text{ where } \Delta x := \frac{b - a}{n}$$

In fact, the approximation can be made from rectangles coming from an arbitrary partitioning of the interval (a, b) with the heights of the rectangles being taken from arbitrary points within each of the partition elements. See [12] or any elementary calculus text for more details.

> $integral := \mathbf{proc}(f :: procedure, r :: range)\,\mathbf{local}\,\text{Delta}, a, b;$
> $\quad a, b := lhs(r), rhs(r);\text{Delta} := \frac{b-a}{n};$
> $\quad Limit(\text{Delta} \cdot Sum(f(a + k \cdot \text{Delta}), k = 1..n), n = infinity) =$
> $limit(\text{Delta} \cdot sum(f(a + k \cdot Delta), k = 1..n), n = infinity)$
> $\mathbf{end}:$

> $integral(x \to x^2, 0..2); integral(\sin, 0..\text{Pi}); integral(\sin, 0..2 \cdot \text{Pi})$

$$\lim_{n\to\infty}\left(\frac{2\left(\sum_{k=1}^n \frac{4k^2}{n^2}\right)}{n}\right) = \frac{8}{3}$$

$$\lim_{n\to\infty}\left(\frac{\pi\left(\sum_{k=1}^n \sin\left(\frac{k\pi}{n}\right)\right)}{n}\right) = 2$$

$$\lim_{n\to\infty}\left(\frac{2\pi\left(\sum_{k=1}^n \sin\left(\frac{2k\pi}{n}\right)\right)}{n}\right) = 0$$

Once this limit is established, then we are presented with the fundamental theorem of calculus which states that

$$\int_a^b f(x)\,dx = F(b) - F(a) \ \text{ where } F' = f$$

and nicely links differentiation and definite integrals. We also, by convention, denote the indefinite integral

$$\int f(x)\,dx = F(x) \Leftrightarrow F' = f$$

We may use this to check the limits found above. First we have $x^3/3$ as an antiderivative of x^2, and evaluating

$$\frac{2^3}{3} - \frac{0}{3} = \frac{8}{3}$$

verifies the integral. Similarly we have $-\cos$ as an antiderivative of $\sin$ leading us to $-\cos(\pi) - (-\cos(0)) = 1 + 1 = 2$ and $(-\cos(2\pi) - (-\cos(0))) = -1 + 1 = 0$. In fact, the final integral can be verified by the symmetric nature of the cosine graph between 0 and 2π.

Again, as with differentiation, we need not perform limit calculations within *Maple* if we wish to calculate an integral, as there is a handy function named **int** (and its inert form **Int**). The **int** function can handle both definite and indefinite integrals and takes an expression as its first argument, and a range as its second argument in the form $var = a..b$ where var is the variable over which to integrate. In the case of an indefinite integral, the second parameter is just var with no range.

> $Int(x^2, x = 1..3) = int(x^2, x = 1..3);$
> $Int(\sin(x), x) = int(\sin(x), x)$

$$\int_1^3 x^2\ dx = \frac{26}{3}$$

$$\int \sin(x)\ dx = -\cos(x)$$

The **int** command will also accept function (as opposed to expression) for its first parameter. This works in precisely the same way as the **plot** function does, also requiring the omission of $var =$ from the second parameter. However, it can only do this for definite integrals, and the inert form does not behave very well this way.

> $Int(\sin, 0..\mathrm{Pi}) = int(\sin, 0..\mathrm{Pi})$

$$Int(\sin, 0..\pi) = 2$$

> $value(lhs(\%))$

$$2$$

To close this section, we mention that the Student and IntegrationTools packages each contain many useful tools to help manipulate integrals, for instance to extract the integrand quickly without learning a lot about operands in *Maple* or resorting to cutting and pasting. The reader is, as always, actively encouraged to utilize *Maple*'s help files to learn more.

2.2 Univariate Calculus

2.2.1 Optimization

Suppose we wish to calculate the longest ladder that we may carry around a corner with one corridor 2 meters in width, and the other 1 meter in width. This is an example of an Optimization problem. We may use calculus (specifically, differentiation) to solve problems along these lines.

Given any angle θ we know that

$$x = \frac{1}{\cos(\theta)}\ \text{ and }\ y = \frac{2}{\sin(\theta)}$$

Therefore, the length L of a ladder that touches the corner, and the opposite walls of each corridor at an angle of θ to the corner, is given by the formula

$$L := \frac{1}{\cos(\theta)} + \frac{2}{\sin(\theta)}$$

We can see straight away that $0 < \theta < \pi/2$ and that $1/\cos(\theta) \to \infty$ as $\theta \to (\pi/2)^-$ as well as $2/\sin(\theta) \to \infty$ as $\theta \to 0^+$ which tells us that our L function will tend toward infinity at its endpoints. We also know that $L > 0$ for all values of θ, just as we would expect for the length of a ladder. We plot the function, being sure to limit the y-axis.

> $L := theta \to \dfrac{1}{\cos(theta)} + \dfrac{2}{\sin(theta)};$

$plot\left(L(theta), theta = 0..\dfrac{\text{Pi}}{2}, y = 0..100 \right)$

$$L := \theta \to \frac{1}{\cos(\theta)} + \frac{2}{\sin(\theta)}$$
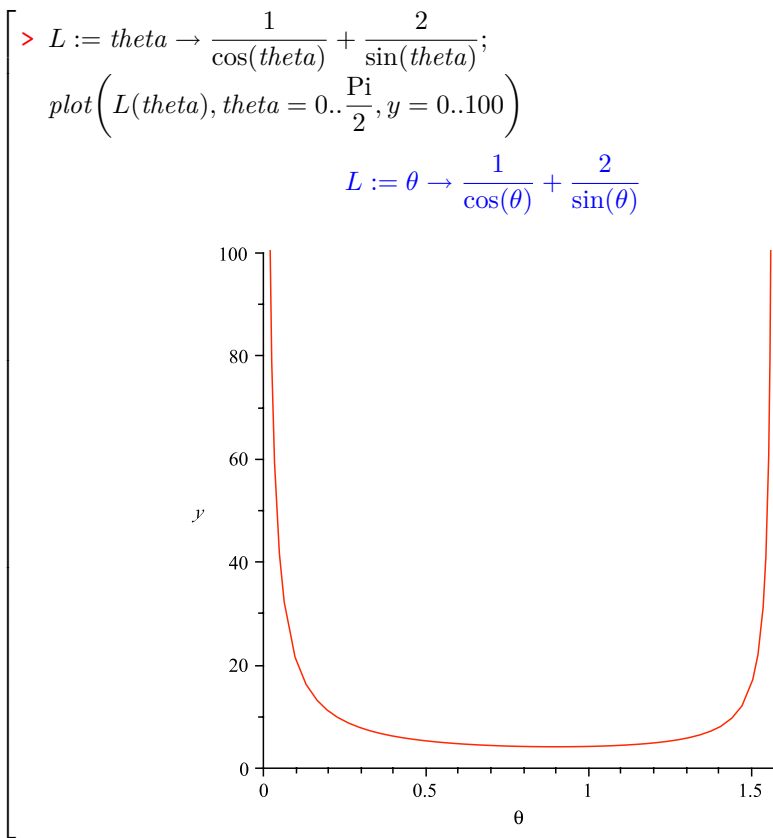
What is interesting here is that the function is concave up and has no maximum values, although it does have a minimum value. What's going on here? Well, our function L, as we know, calculates the length of a ladder at an angle of θ that touches both the far walls of the two corridors as well as the corner. Such a ladder is the longest possible ladder that can rest at that particular angle. However, if a ladder is to be carried around the corner, then it must go through all angles from $\theta = 0$ to $\theta = \pi/2$ and not become stuck. What we are, in essence, looking for is the shortest of all such longest ladders, and hence a minimum of the function L.

With a little trial and error, we may find the following plot and see that the minimum value lies somewhere in the range $0.5 \leq \theta \leq 1$. We can, in fact, do a little better with these bounds. Looking at the plot (below) we can clearly see the the minimum lies to the right of the midpoint of the θ-axis. Even though it is not labelled, we know this midpoint must be $\pi/4$ because our plot is for theta between 0 and $\pi/2$. Our minimum, therefore, must lie in the range $\pi/4 \leq \theta \leq 1$.

> $plot\left(L(\text{theta}), \text{theta} = 0..\dfrac{\text{Pi}}{2}, y = 4..10\right)$
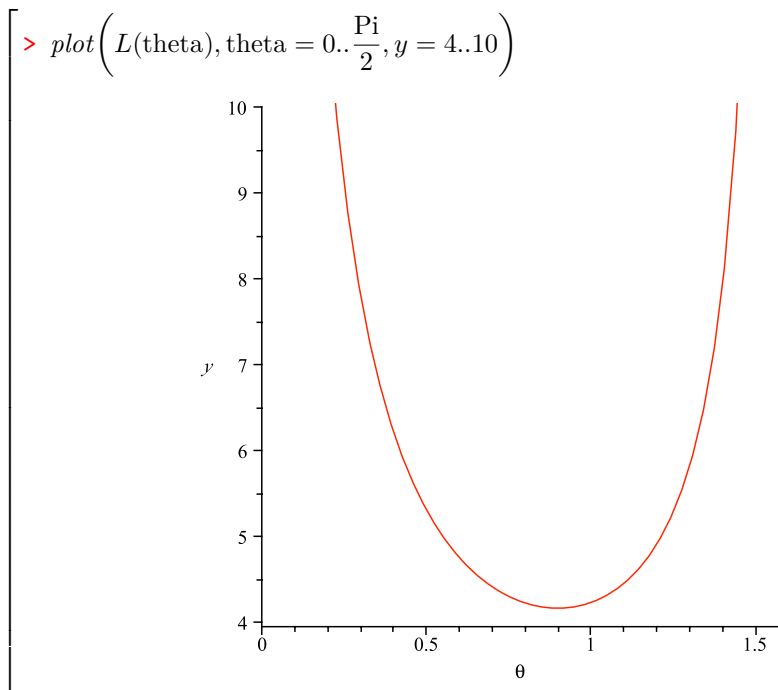
We could, if we wished "zoom in" by plotting the region $\pi/4 \le \theta \le 1$ and $4 \le L \le 5$ in the hopes of obtaining better bounds. We could even continue along these lines for some time. There is little point in doing so, at least in this case. Instead we proceed to find the minimum symbolically. The minimum is a turning point, and so will have a derivative of 0. We therefore solve $L'(\theta) = 0$.

> $D(L)(\text{theta}); solve(\% = 0)$

$$\frac{\sin(\theta)}{\cos(\theta)^2} - \frac{2\cos(\theta)}{\sin(\theta)^2}$$

$$\arctan(2^{1/3}), -\arctan\left(\frac{1}{2}\,2^{1/3} - \frac{1}{2}I\sqrt{3}\,2^{1/3}\right), -\arctan\left(\frac{1}{2}\,2^{1/3} + \frac{1}{2}I\sqrt{3}\,2^{1/3}\right)$$

Well, that's a bit of a mess. *Maple* seems to have given us three solutions, two of which appear to be complex. Nonetheless, there is clearly a real solution there, and evaluating it numerically shows it to be in the range we were expecting.

> $evalf\left(\arctan\left(2^{1/3}\right)\right); L = L\left(\arctan\left(2^{1/3}\right)\right); evalf(\%)$

$$0.8999083481$$

$$L = \sqrt{2^{2/3} + 1} + 2^{2/3}\sqrt{2^{2/3} + 1}$$

$$L = 4.161938184$$

And so there we have it. A ladder of approximately 4.16 meters in length is the longest we may carry around the corner. We can see quite well from the plots above that this is clearly a local minimum, however, a quick second derivative test won't hurt.

> $D^{(2)}(L)\left(\arctan\left(2^{1/3}\right)\right); is(\% > 0); evalf(\%\%)$

$$3\,2^{2/3}\sqrt{2^{2/3} + 1} + 3\sqrt{2^{2/3} + 1}$$

$$true$$

$$12.48581455$$

2.2.2 Integral Evaluation

Integral evaluation can, at times, be quite tricky. A tool such as *Maple* can indeed be an asset, however, even it may be unable to perform certain integrals symbolically. For example, suppose we ask *Maple* to integrate xe^{x^3} between 0 and 1.

> $int\big(x \cdot \exp\big(x^3\big), x = 0..1\big)$

$$\int_0^1 xe^{x^3}\, dx$$

Notice here that we definitely used the active form of the integral command, and yet *Maple* still returned only the integral back again. It seems that *Maple* cannot provide a better symbolic answer than the integral itself. The use of hand-methods should prove equally frustrating. We note that if the integrand were xe^{x^2}, then the task would be trifling easy. At times like this, all one can really do is ask for a numerical answer

> $evalf[50]\big(int\big(x \cdot \exp\big(x^3\big), x = 0..1\big)\big)$; $identify(\%)$;

  0.78119703110865591510743281434829950577669739096218

  0.78119703110865591510743281434829950577669739096218

The use of the **identify** is a request to *Maple* to take a good guess at a likely symbolic representation of the given decimal number. In this case **identify** simply gives us back the decimal number, meaning that it could not find any likely symbolic representation. Similarly, neither Sloane's On-line Encyclopedia of Integer Sequences[2], nor the Inverse Symbolic Calculator[3] turn up anything. It looks like we're stuck with just a numerical approximation of the integral.

Suppose we wish to evaluate the integral

$$\int_0^\pi \frac{x\sin(x)}{1 + \cos^2(x)}\, dx$$

Well, our first attempt should be to see what *Maple* thinks.

> $simplify\bigg(int\bigg(\dfrac{x \cdot \sin(x)}{1 + \cos(x)^2}, x = 0..\,\mathrm{Pi}\bigg)\bigg)$

$$\frac{1}{4}\pi^2 - dilog\bigg(\frac{\sqrt{2} - 1 - I}{\sqrt{2} - 1}\bigg) + dilog\bigg(\frac{1 - I + \sqrt{2}}{1 + \sqrt{2}}\bigg)$$

$$- dilog\bigg(\frac{1 + I + \sqrt{2}}{1 + \sqrt{2}}\bigg) + dilog\bigg(\frac{\sqrt{2} - 1 + I}{\sqrt{2} - 1}\bigg) + I\pi \ln\big(1 + \sqrt{2}\big)$$

That's not really very useful. Let's try to evaluate that mess as a decimal number.

> $evalf[50](\%)$

  2.4674011002723396547086227499690377838284248518102 $- 3.\,10^{-49}I$

Hmmm, that's a complex number even if the complex part is infinitesimal. Let's try again, and see if asking *Maple* to evaluate the integral numerically works any better. We even throw in an **identify** for good measure.

---

[2] http://oeis.org/ at the time of writing

[3] http://isc.carma.newcastle.edu.au/ at the time of writing

> $int\left(\dfrac{x \cdot \sin(x)}{1+\cos(x)^2}, x = (0.).. \operatorname{Pi}\right); identify(\%)$

$$2.467401100$$

$$\frac{1}{4}\pi^2$$

Numerically evaluating the integral, and asking *Maple* to guess a symbolic value, gives us that the integral is very probably equal to $\frac{1}{4}\pi^2$. At this point we would start looking for a more formal proof. Fortunately, in this case, it can be shown (see Exercise 5) that

$$\int_0^\pi x f(\sin(x))\, dx = \frac{\pi}{2}\int_0^\pi f(\sin(x))\, dx$$

and it should be pretty clear that

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

which leaves us with the rather simpler integral

$$\int_0^\pi \frac{x\sin(x)}{1+\cos^2(x)}\, dx = \frac{\pi}{2}\int_0^\pi \frac{\sin(x)}{1+\cos^2(x)}\, dx$$

As it happens, this new integral is easier for *Maple* to handle.

> $\dfrac{\operatorname{Pi}}{2}\cdot int\left(\dfrac{\sin(x)}{1+\cos^2(x)}, x = 0.. \operatorname{Pi}\right)$

$$\frac{1}{4}\pi^2$$

Let us perform another integral. This time we evaluate

$$\int_0^\infty \frac{x^2}{\sqrt{e^x-1}}\, dx$$

As a first attempt, we see what *Maple* makes of the integral.

> $int\left(\dfrac{x^2}{\operatorname{sqrt}(\exp(x)-1)}, x = 0..infinity\right)$

$$\int_0^\infty \frac{x^2}{\sqrt{e^x-1}}\, dx$$

Unfortunately, *Maple* apparently cannot evaluate this integral. Our next course of action is to attempt numeric evaluation.

> $evalf(\%); identify(\%)$

$$16.37297620$$

$$\frac{9}{2}\, 2^{3/5}\, \sqrt{3}\, \zeta(5)^9$$

We now have a possible value for the integral. Remember, however, that the **identify** function gives a *probable* symbolic expression but not a certain one. We return shortly to the question of how reliable this symbolic guess is, but it should be clear that more work is needed to be performed in order to verify the identity with any certainty.

We have a closer look at the integral. Looking at the denominator, $\sqrt{e^x-1}$, we see that a substitution of $x = \log(u)$ will change the denominator to $\sqrt{u-1}$ thus

eliminating the exponent term. Performing this substitution, we see that

$$\frac{dx}{du} = \frac{1}{u} \Rightarrow dx = \frac{du}{u}$$

It is also the case that $\log(0) = 1$ and $\log(x) \to \infty$ as $x \to \infty$ so that

$$\int_0^\infty \frac{x^2}{\sqrt{e^x - 1}}\, dx = \int_1^\infty \frac{\log(u)^2}{u\sqrt{u-1}}\, du$$

Turning again to *Maple* with this new integral we find altogether better success than we did previously.

> $int\left(\dfrac{\log(u)^2}{u \cdot \text{sqrt}(u - 1)}, x = 1..infinity\right)$

$$\frac{1}{3}\pi^3 + 4\pi\ln(2)^2$$

Interestingly, however, we have an altogether different symbolic answer to the one produced by **identify**, above. It is prudent then, to perform a sanity check

> $evalf(\%)$

$$16.37297620$$

which is precisely the same 10-digit value we obtained from our first attempt. This is absolutely expected, of course, because substitutions do not alter the value of an integral.

As an epilogue to this example, we return to the question of the symbolic answer that **identify** gave us earlier. It turns out that if we obtain even one extra digit of precision for this integral, then *Maple* is completely unable to identify the decimal approximation.

> $evalf[11]\left(int\left(\dfrac{x^2}{\text{sqrt}(\exp(x) - 1)}, x = 0..infinity\right)\right); identify(\%)$

$$16.372976196$$
$$16.372976196$$

Indeed, for any precision greater than 11 digits, it seems that *Maple* is unable to identify the number.

Exploring this a little more closely, we find that the two differ after, approximately, eight decimal places.

> $evalf\left(int\left(\dfrac{x^2}{\text{sqrt}(\exp(x) - 1)}, x = 0..infinity\right) - \dfrac{9}{2} \cdot 2^{3/5} \cdot \text{sqrt}(3) \cdot \text{Zeta}(5)^9\right)$

$$-4.\,10^{-8}$$

> $evalf[50]\left(int\left(\dfrac{x^2}{\text{sqrt}(\exp(x) - 1)}, x = 0..infinity\right) - \dfrac{9}{2} \cdot 2^{3/5} \cdot \text{sqrt}(3) \cdot \text{Zeta}(5)^9\right)$

$$-5.7359902850345135770849920305260620836853\,10^{-8}$$

It would seem that **identify** was confused by not having enough decimal digits of the number in question with which to work.

At least two things should be readily apparent from these examples. First and foremost is that *Maple* is not a substitution for poor calculus skills (or, at least, is a poor one). Second, answers—especially from the **identify** function—should be checked from a number of avenues before being accepted. Human/machine collaboration is the name of the game here. *Maple* can be wonderful for performing tedious calculations quickly,

and is remarkably adept at performing integral calculus, but a correct substitution, or other mathematical insight on our part can mean the difference between successfully obtaining a symbolic answer, or not.

2.2.3 Symbolic Integrals

In the previous section we took pains to calculate the value of certain definite integrals, but did not always find a symbolic answer. Our attempts to find a symbolic answer, however, began and ended with issuing an **int** command. In one case we performed a substitution, but this amounted to a second (and successful, as it happened) attempt at issuing an **int** command. The reader may wonder if it is possible to have *Maple* perform such a substitution. The astute reader may realize that an integral itself is just another mathematical expression that is subject to symbolic manipulation, and may further wonder whether *Maple* can treat one as such. In fact, these two questions are related, and the answer to both is yes and involves the Integration Tools package.

> $with(IntegrationTools)$

$[Change, Combine, Expand, Flip, GetIntegrand, GetRange, GetVariable, Parts,$
$Split]$

There are not many functions here, but what is there should look familiar. The reader should spend some time looking at the help files for this package (**?Integration Tools**). It should be mentioned that the Integration Tools functions seem to work better with inert integrals, and can be a little unpredictable with active integrals. Let us look first at the integral from the previous example.

> $A := Int\left(\dfrac{x^2}{\mathrm{sqrt}(\exp(x) - 1)}, x = 0..infinity\right)$

$$A := \int_0^\infty \frac{x^2}{\sqrt{e^x - 1}}\, dx$$

We know, from above, that *Maple* cannot evaluate this integral directly. We also know, from above, exactly what the substitution we need to perform is; $x = \log(u)$. It should be little surprise that the function we need to use to have *Maple* perform this substitution is the **Change** function (for a change of variable).

> $Change(A, x = \log(u))$

$$\int_1^\infty \frac{\ln(u)^2}{\sqrt{u - 1}\, u}\, du$$

We could just as easily have phrased the change of variable as $u = e^x$

> $Change(A, u = \exp(x))$

$$\int_1^\infty \frac{\ln(u)^2}{\sqrt{u - 1}\, u}\, du$$

Either way we have the integral that we calculated by hand in the previous section, and we know that *Maple* can compute this integral

> $value(\%)$

$$\frac{1}{3}\,\pi^3 + 4\,\pi\,\ln(2)^2$$

We look now at a different example. We calculate the integral of $e^x \cos(x)$. As it happens, *Maple* can calculate this directly.

> $B := Int(\exp(x) \cdot \cos(x), x)$

$$B := \int e^x \cos(x)\,dx$$

> $value(B)$

$$\frac{1}{2}\,e^x \cos(x) + \frac{1}{2}\,e^x \sin(x)$$

We explore this integral a little. If we were to try this by hand, our first instinct should be to try to use integration by parts. Setting $u = e^x$ and $dv = \cos(x)$ we have that $du = e^x$ and $v = \sin(x)$. We use the **Parts** function from the IntegrationTools package to perform this integration by parts. Much like the **Change** function, we first specify the integral on which we wish to work. The second parameter is the value of u for the integration by parts. An optional third parameter may also be used to specify the v parameter directly, but we do not use this here. The interested reader should consult the help files.

> $B = Parts(B, \exp(x))$

$$\int e^x \cos(x)dx = e^x \sin(x) - \int e^x \sin(x)\,dx$$

In order to make any use of this, we need to perform the same integration by parts on this new integral. One useful feature of the IntegrationTools functions is that they only operate on integrals, and tend to ignore anything else. This may sometimes be inconvenient in the case of a change of variables, but in this case we exploit the feature.

> $C := Parts(B, \exp(x))$

$$C := e^x \sin(x) - \int e^x \sin(x)\,dx$$

> $C := Parts(C, \exp(x))$

$$C := e^x \sin(x) + e^x \cos(x) + \int -e^x \cos(x)\,dx$$

We now have the integral of $e^x \cos(x)$ again, which is what we started with, although there's a negative in this newest one. We can pull the negative outside the integrand with the **simplify** command.

> $C := simplify(C)$

$$C := e^x \sin(x) + e^x \cos(x) - \int e^x \cos(x)\,dx$$

We now have the following identity.

> $B = C$

$$\int e^x \cos(x)dx = e^x \sin(x) + e^x \cos(x) - \int e^x \cos(x)\,dx$$

We can see the result easily enough from this identity, however, we have *Maple* produce it for us in order to demonstrate how we may perform arithmetic on equalities. The first step is to add the integral to both sides; then we divide by 2.

> $\% + B$

$$2 \int e^x \cos(x)dx = e^x \sin(x) + e^x \cos(x)$$

> $\dfrac{\%}{2}$

$$\int e^x \cos(x)\,dx = \frac{1}{2}\,e^x \sin(x) + \frac{1}{2}\,e^x \cos(x)$$

2.2.4 Differential Equations

Differential equations are equations that relate a function to its derivatives. A solution to a differential equation is a function that has the required relationship with its derivatives. The simplest differential equation is $y' = y$ which has the solution $y = Ce^x$ (where C is an arbitrary constant). This should be nothing new to anybody who has studied first-year calculus.

A first-order linear differential equation can always be re-written to have the form

$$y' + P(x)\,y = Q(x)$$

and can be solved by use of an *integrating factor*

$$I(x) := e^{\int P(x)\,dx}$$

which has the property that

$$\int \Big(I(x)\,y' + I(x)\,P(x)\,y\Big)\,dx = I(x)\,y$$

It is fairly simple to verify the claimed property of the integrating factor by hand. We check it in *Maple*.

> $IF := \exp(int(P(x), x));$

$$IF := e^{\int P(x)\,dx}$$

> $int(IF \cdot diff(y(x), x) + IF \cdot P(x) \cdot y(x), x)$

$$e^{\int P(x)\,dx}\,y(x)$$

> $diff(IF \cdot y(x), x)$

$$e^{\int P(x)\,dx}\left(\frac{d}{dx}\,y(x)\right) + e^{\int P(x)\,dx}\,P(x)\,y(x)$$

Of course, we only needed to check one of these identities, the other one we get for free with the fundamental theorem of calculus.

With this identity available to us, we may solve the equation by multiplying the left-hand and right-hand sides by the integrating factor, integrating both sides, and solving for y. The solution, therefore, to the differential equation is

$$y' + P(x)\,y = Q(x) \implies \int I(x)(y' + P(x)\,y)\,dx = \int I(x)Q(x)\,dx$$

$$\implies I(x)\,y = \int I(x)Q(x)\,dx$$

and so the solution is

$$y = \frac{\int I(x)\, Q(x)\, dx}{I(x)}$$

Let us now consider the linear differential equation $xy' + y = 3x^3$, which may be rewritten as $y' + x^{-1}y = 3x^2$; then we may use *Maple*

> $P := x \to \dfrac{1}{x}; Q := x \to 3 \cdot x^2; IF := \exp(int(P(x), x))$

$$P := x \to \frac{1}{x}$$
$$Q := x \to 3\,x^2$$
$$IF := x$$

> $\dfrac{int(IF \cdot Q(x), x) + C}{IF}$

$$\frac{\frac{3}{4}\,x^4 + C}{x}$$

Note, however, that for a general solution we needed to add manually the constant of integration when calculating the answer, because *Maple*'s **int** function does not include this constant.

We may cross-check this using *Maple*'s inbuilt differential equation solving function **dsolve**.

> $dsolve(x \cdot diff(y(x), x) + y(x) = 3 \cdot x^3)$

$$y(x) = \frac{\frac{3}{4}\,x^4 + \_C1}{x}$$

Moving on now to second-order differential equations. A *second-order linear differential equation* is a differential equation of the form

$$P(x)\,y'' + Q(x)\,y' + R(x)\,y = G(x)$$

and is furthermore said to be *homogeneous* if $G(x) = 0$. Finally, if the functions $P(x)$, $Q(x)$, and $R(x)$ are constant functions then the differential equation is said to have *constant coefficients*; however, if the differential equation is still to be second-order then $P(x) \neq 0$.

Homogeneous second-order linear differential equations with constant coefficients may be solved in a manner almost identical to that used to solve homogeneous second-order linear recurrence relations with constant coefficients, which we looked at in Section 1.3.3.

Given the equation $ay'' + by' + c = 0$ we construct the *characteristic equation* $at^2 + bt + c = 0$ and solve for t. There are only three possibilities for the roots r_1, r_2 of the equation. The general formula is as follows.

$$y(x) = \begin{cases} Ae^{r_1 x} + Be^{r_2 x} & r_1 \neq r_2 \\ Ae^{r_1 x} + Bxe^{r_2 x} & r_1 = r_2 \\ e^{\alpha x}\left(A\sin(\beta x) + B\cos(\beta x)\right) & r_1, r_2 = \alpha \pm i\beta \end{cases}$$

where A and B are arbitrary constants.

Let's look at some examples. We start with a differential equation that has the same characteristic equation as the Fibonacci numbers, $y'' - y' - y = 0$. Because this has characteristic equation $t^2 - t - 1$, we know from Section 1.3.2 that the roots are

$t = \frac{1}{2}(1 \pm \sqrt{5})$. As such, the solution to the differential equation should be

$$y = Ae^{x \cdot (1+\sqrt{5})/2} + Be^{x \cdot (1-\sqrt{5})/2}$$

We check this in *Maple*.

> $y := A \cdot \exp\left(\dfrac{x \cdot (1 + \mathrm{sqrt}(5))}{2}\right) + B \cdot \exp\left(\dfrac{x \cdot (1 - \mathrm{sqrt}(5))}{2}\right)$

$$Ae^{\frac{1}{2}x(1+\sqrt{5})} + Be^{\frac{1}{2}x(1-\sqrt{5})}$$

> $\mathit{diff}(y, x); \mathit{diff}(y, x, x)$

$$A\left(\frac{1}{2} + \frac{1}{2}\sqrt{5}\right)e^{\frac{1}{2}x(1+\sqrt{5})} + B\left(\frac{1}{2} - \frac{1}{2}\sqrt{5}\right)e^{\frac{1}{2}x(1-\sqrt{5})}$$

$$A\left(\frac{1}{2} + \frac{1}{2}\sqrt{5}\right)^2 e^{\frac{1}{2}x(1+\sqrt{5})} + B\left(\frac{1}{2} - \frac{1}{2}\sqrt{5}\right)^2 e^{\frac{1}{2}x(1-\sqrt{5})}$$

> $\mathit{simplify}(\mathit{diff}(y, x, x) - \mathit{diff}(y, x) - y)$

$$0$$

And asking *Maple* for the solution directly also gives the expected result (which is always nice).

> $y :=' y'; \mathit{dsolve}\left(\mathrm{D}^{(2)}(y)(x) - \mathrm{D}(y)(x) - y(x) = 0\right)$

$$y(x) = \_C1\, e^{\frac{1}{2}x(1+\sqrt{5})} + \_C2\, e^{-\frac{1}{2}x(-1+\sqrt{5})}$$

In the case of nonhomogeneous second-order linear differential equations with constant coefficients the solution is obtained by first calculating the solution to the equivalent homogeneous differential equation (essentially just pretending that $G(x) = 0$) and adding to that a *particular* solution.

To be more mathematically rigorous here, given the equation

$$ay'' + by' + cy = G(x)$$

the general solution is

$$y = y_p + y_c$$

where y_p is some (any) solution to the equation, and y_c is the solution to the complementary homogeneous equation

$$ay'' + by' + cy = 0$$

which is henceforth referred to only as the *complementary equation*.

Ordinarily in first-year courses the particular solution is obtained by a guess-and-check approach. A more systematic method named *variation of parameters* is described in [12], but tends to be trickier and slower to implement in practice, at least for the sorts of problems studied in first-year calculus.

We eschew the entire business, and go straight to the "just ask *Maple*" method. We extend our previous example to a nonhomogeneous problem where $G(x) = \sin x$.

> $\mathit{dsolve}\left(\mathrm{D}^{(2)}(y)(x) - \mathrm{D}(y)(x) - y(x) = \sin(x)\right)$

$$y(x) = e^{\frac{1}{2}x(1+\sqrt{5})}\_C2 + e^{-\frac{1}{2}x(-1+\sqrt{5})}\_C1 + \frac{1}{5}\cos(x) - \frac{2}{5}\sin(x)$$

We can clearly see our homogeneous solution there (even though the arbitrary constants have moved from the front to the back of each term), and an extra bit at the end which is, presumably, the particular solution. We have a closer look at that.

> $y := x \rightarrow \dfrac{1}{5}\cos(x) - \dfrac{2}{5}\sin(x);$
> $\mathrm{D}^{(2)}(y)(x) - \mathrm{D}(y)(x) - y(x)$

$$y := \frac{1}{5}\,\cos(x) - \frac{2}{5}\,\sin(x)$$

$$\sin(x)$$

So it is clear that $\frac{1}{5}\cos x - \frac{2}{5}\sin x$ is a solution to the original nonhomogeneous problem. It should be clear, using the fact that

$$\frac{d}{dx}(f+g) = \frac{d}{dx}(f) + \frac{d}{dx}(g)$$

that the entire function returned by *Maple* is also a solution to the equation as well.

One might well ask why, when we have a perfectly good and easy-to-write solution such as $\frac{1}{5}\cos x - \frac{2}{5}\sin x$ would we ever want to bother with adding in the extra mess of the solution to the complementary equation as well. The answer is that $y_c + y_p$ is the *general* solution to the equation. That is to say that any and every solution to the equation can be written in that form. Indeed, the particular solution, above, was the same as the general solution with $\_C1 = \_C2 = 0$. The proof of this fact is quite elementary, and can be found in [12] if desired.

2.2.5 Parametric Equations, Alternative co-ordinates, and Other Esoteric Plotting Fun

Recall that when a graph is plotted, what we are seeing is a graphical representation of pairs of points that satisfy some relationship. The parabola, for example, is the collection of all points (x, y) in $\mathbb{R} \times \mathbb{R}$ where $y = x^2$.

We may also plot functions from parametric equations, where a parameter, t say, varies, and the points in the plot are of the form $(x, y) = (x(t), y(t))$. There is not necessarily a relationship between the x and y co-ordinates in a parametric equation, apart from the fact that they share the same t-value. Of course, any function $f(x)$ may be turned into a parametric equation $(x, y) = (t, f(t))$.

Maple's **plot** function will allow us to plot parametric equations, if we so wish. To do so, we must pass a three-element list as the first argument to the **plot** function, instead of the usual expression. This list must be in the form of $[x(t), y(t), t = a..b]$ where $a..b$ is a range, t is any valid variable name, and $x(t)$ and $y(t)$ are arbitrary expressions involving that variable. Note that *Maple* will automatically scale the horizontal and vertical axes to fit the plot, based on the values of $x(t)$ and $y(t)$ as t varies through its range.

However, one should be careful; the axis ranges are independent of the parameter range, and all three may be modified separately. In the case of our parabola, there is a very direct relationship between the parameter t, and the horizontal and vertical ranges. This need not be so.

For example, to plot our parabola using parametric equations, we would input the following.

> $plot([t, t^2, t = -10..10])$
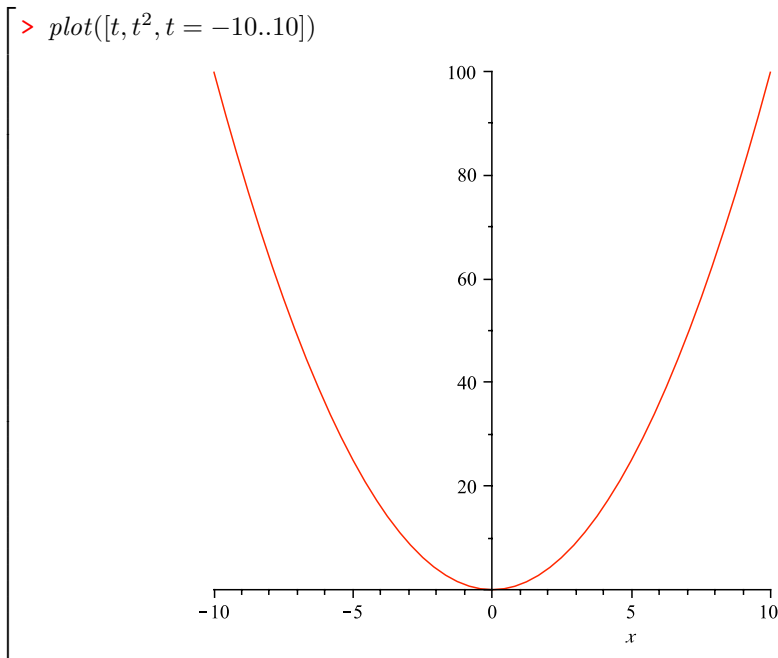
To see the independence of the axis and parameter variables, recall the parametric equations of a circle centered at an arbitrary point, (x_0, y_0) say, are

$$(x, y) = (x_0, y_0) + (r \cos \theta, r \sin \theta) = (x_0 + r \cos \theta, y_0 + r \sin \theta)$$

This is easiest to see by treating these parametric equations as vector equations, and recalling that the parametric equations of a circle centered at the origin are $(x, y) = (r \cos \theta, r \sin \theta)$.

We plot a circle, centered at $(1, 2)$, with radius 3. We should expect the horizontal range to be $-2..4$ and the vertical range to be $-1..5$.

> $plot([1 + 3 \cdot \cos(t), 2 + 3 \cdot \sin(t), t = 0..2 \cdot \text{Pi}], -3..5, -2..6)$
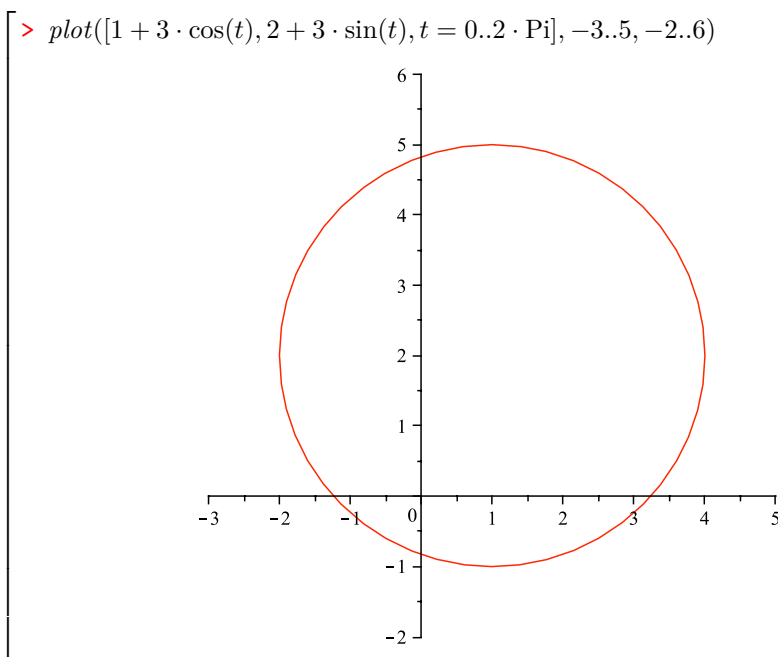
In order to demonstrate the independent nature of the axis and parameter ranges, we specifically instructed *Maple* to plot a larger axis range. We can clearly see the circle neatly within the ranges as defined by the parameter, and yet *Maple* has happily plotted with the extra range we requested.

In a lot of (if not most) cases separately modifying the horizontal, vertical, and parameter ranges will not be necessary. Nonetheless, it is worth being aware of the fact that they may be independently specified.

The circle example demonstrates a very useful feature of parametric plots, which is that they may be used to plot curves that are not the result of functions. The circle is clearly not a function, as it violates the vertical line test. Recall that a function always associates a single value in the range with each single value in the domain. This is not the case with a circle; we cannot assign a function $y = f(x)$ that will produce all the points in a circle. We can, of course, simply plot $y = \pm\sqrt{1 - x^2}$ and display them together, but doing this is often neither easy nor even possible.

To further illustrate this advantage of parametric equations, we plot a spiral using parametric equations. We use the parametric equations $(x, y) = (t \cos t, t \sin t)$. This varies from the circle in that the radius is no longer fixed. If we think of the points (x, y) as vectors, then each point on the line is a vector of length t and angle t. As t increases, then the angle will cycle, but the length will continue increasing. We plot four full revolutions of this spiral.

> $plot([t \cdot \cos(t), t \cdot \sin(t), t = 0..8\,\mathrm{Pi}])$
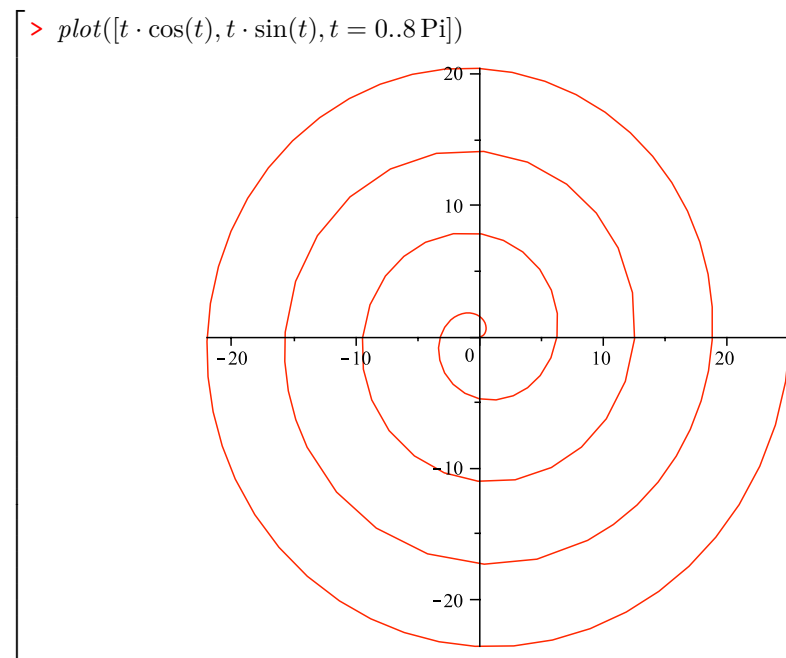
Returning to our circle plots, the reader may well recall that although there is no explicit function for a circle, there most certainly is an equation that gives an implicit function for the circle. That equation is, of course, $(x - x_0)^2 + (y - y_0)^2 = r^2$ where (x_0, y_0) is the center of the circle and r is the radius. One may well wonder if *Maple* can plot such implicit equations. The answer is that yes it can, although we need to use a special function in the plots package, which goes by the name of **implicitplot**.

> $plots[implicitplot]\big((x-1)^2 + (y-2)^2 = 9, x = -3..5, y = -2..6\big)$
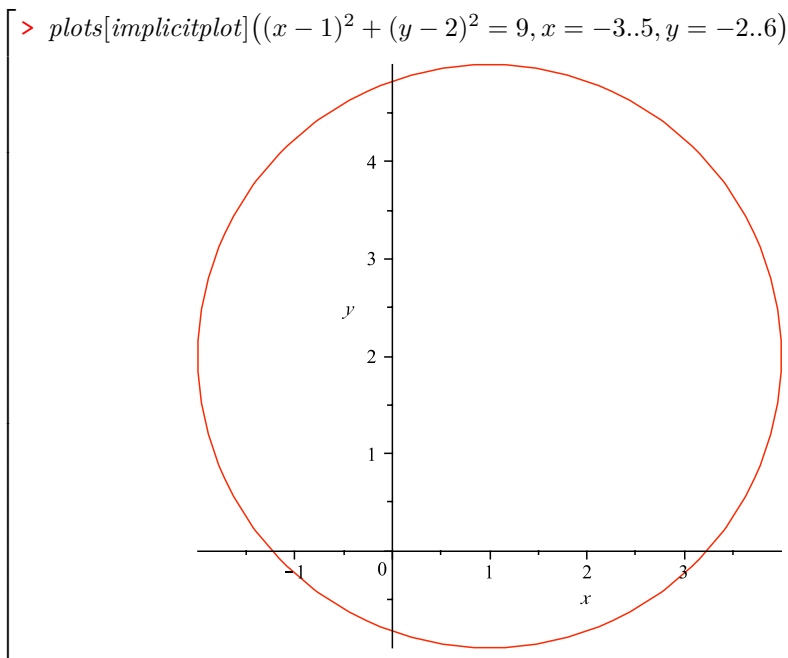
Notice that even though we asked for the plot to be in the extra range, just as we did with the parametric plot above, the **implicitplot** function plotted the circle to its extremities, and no more. It would seem that when evaluating implicit equations, *Maple* evaluates all the pairs (x, y) within the range that satisfy the equation, and then works out the required scale for the axes.

So far we have considered only Cartesian co-ordinates (of the form (x, y) in relation to two axes) when plotting. Another common co-ordinate system is the so-called *polar* co-ordinates, which are co-ordinates of the form (r, θ) where θ is the angle, and r is the distance traveled in that direction. The polar co-ordinates $(\sqrt{2}, \frac{1}{4}\pi)$, for example, correspond to the Cartesian co-ordinates $(1, 1)$. We may freely convert between polar and Cartesian co-ordinates with the identities $r = \sqrt{x^2 + y^2}$, $x = r\cos\theta$, and $y = r\sin\theta$. These identities may easily be confirmed by drawing up a trigonometric triangle.

Maple will happily plot polar equations (i.e., equations given in terms of polar co-ordinates) for us either on a regular Cartesian pair of axes, or on a special background more suited to the polar co-ordinates. The latter requires a special function in the plots package. Polar plots expect an expression for r to be a function of θ (just as regular plots expect an expression for y as a function of x). This should be unsurprising, given that polar equations are usually written as $r = r(\theta)$.

Let us start with a circle, as it is quite simple. A circle contains points that are equidistant from its center, hence r is the constant radius, and θ may take any value. So our polar equation is simply $r = C$ where C is a constant. Such a circle, however, will always be centered at the origin. Plotting, in polar co-ordinates, a circle not centered at the origin is trickier. See Exercise 10b.

Let us look at the spiral again. The spiral construction with parametric equations $(x, y) = (t\cos t, t\sin t)$ was constructed in a way that is very amenable to a polar equation. Recall that our t parameter varied as both an angle and a distance. This sounds very much like a polar equation. It should come as no surprise then that the polar equation of the spiral is simply $r = \theta$. To plot this, we simply pass the parameter *coords=polar* to the **plot** function.

> $plot(\text{theta}, \text{theta} = 0..8 \cdot \text{Pi}, coords = \text{polar})$
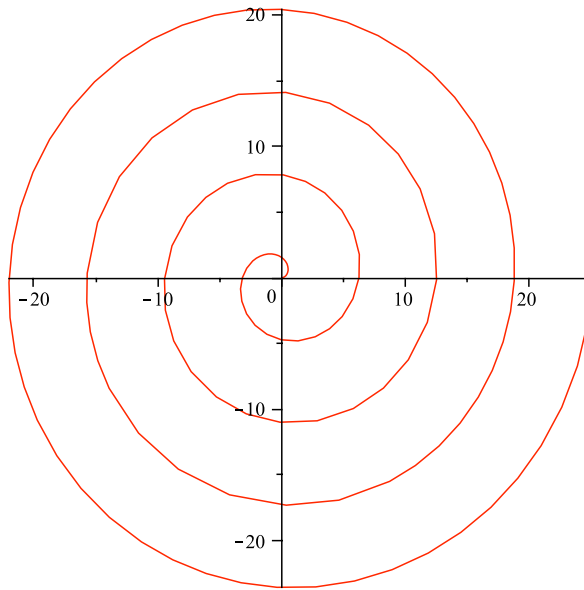
Alternatively we may use the **polarplot** function from the plots package. This has the advantage that it much better labels the r and θ parameters of the polar coordinate system, allowing us to much more easily read these values directly from the plot in much the same way we might read (x, y) value pairs from a plot on the plane.

> $plots[polarplot](\text{theta}, \text{theta} = 0..8 \cdot \text{Pi})$
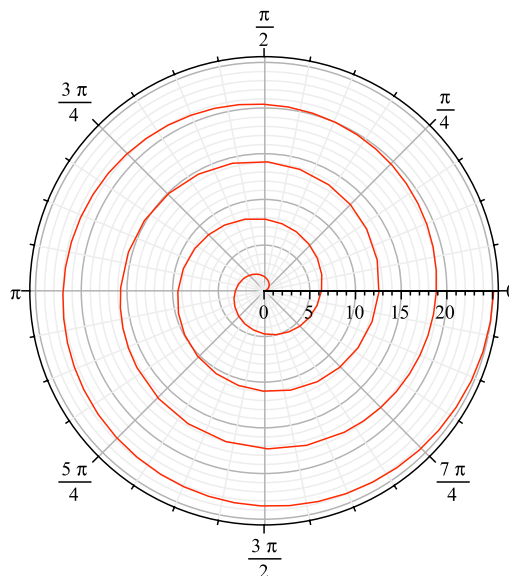
For a more interesting and complicated example, we now look at a more complicated polar plot; $r = \sin(8\theta/5)$. This is clearly a cyclic equation, however the precise nature of the cycling needs some thought. It is clear that r will cycle every time θ changes through $5\pi/4$ radians. In order to cycle fully through all possible pairs of (r, θ) we need

to find integers a, b such that $a \cdot 2\pi = b \cdot 5\pi/4$ and the smallest such value for a is $a = 5$, corresponding to $b = 8$. We therefore plot this function for $0 \le \theta \le 10\pi$.

> $plot\left(\sin\left(\dfrac{8 \cdot \text{theta}}{5}\right), \text{theta} = 0..10 \cdot \text{Pi}, coords = \text{polar}\right)$
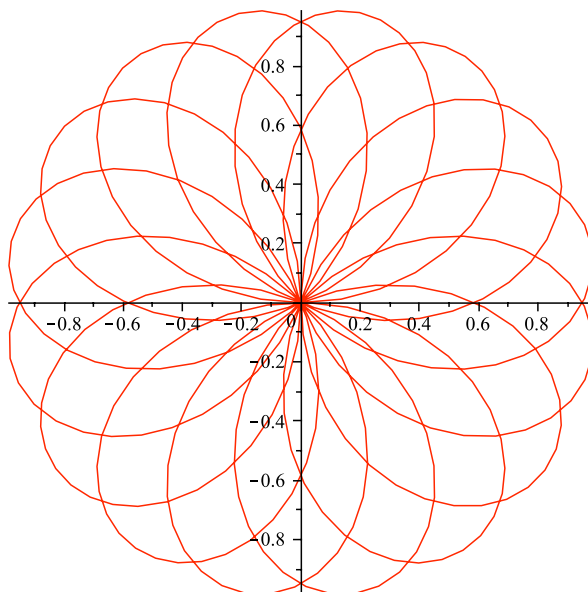
   We may, using **implicitplot**, plot ranges of inequalities, with filled-in regions. We look at a couple of simple examples to close this section. The general approach is basically as one would expect; we replace the equation to be plotted with the inequality. For instance, if we wish to see the inequality $4x - 2y < 3$, we would try the following.

> $plots[implicitplot](4x - 2y < 3, x = -1..1, y = -1..1)$
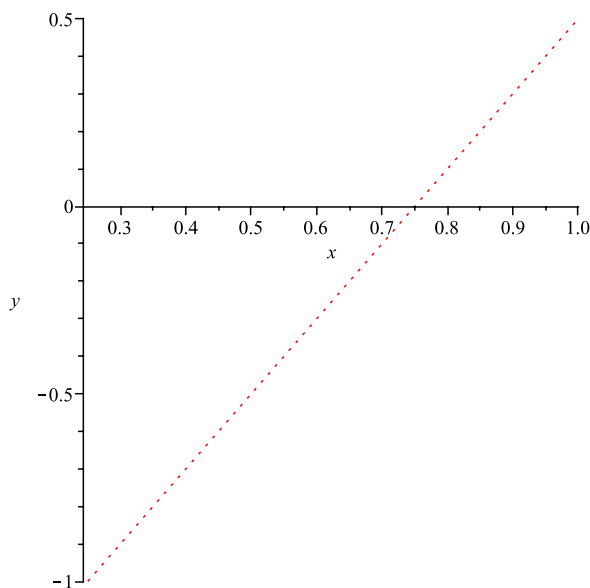
The only catch is that in the above plot we have no idea on which side of the line the region we want lies, even though it is just a simple matter of substituting $x = y = 0$

into the inequality. In order to have *Maple* shade the relevant area, we need to use the
filledregions option.

> $plots[implicitplot](4x - 2y < 3, x = -1..1, y = -1..1, filledregions = true)$
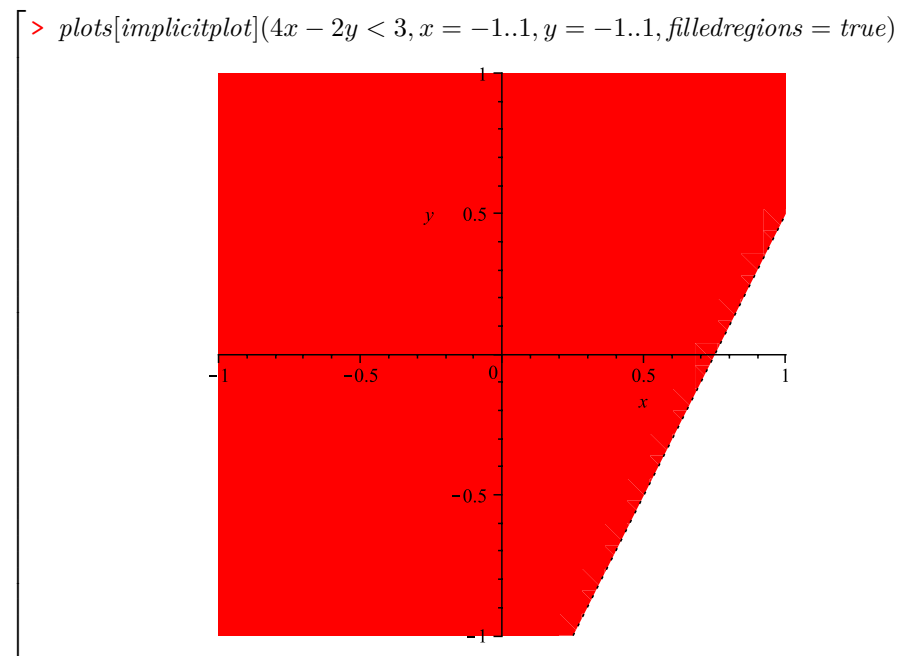
The option to change colors is a little different here as well. If we use the **color** option
we will only change the color of the implicit curve itself, and not the regions. To color
the regions we need to use **coloring**[4] option, and we must use a list with this option,
even for only one color. To demonstrate this, and to close off this section, we plot our
familiar radius-3 circle, centered at $(1, 2)$ with a black border and blue interior.

---

[4] Note that, unlike other *Maple* options and keywords, the British spelling of "colouring" will not
be recognised as an alternative spelling of **coloring**.

> $plots[implicitplot]\big((x-1)^2 + (y-2)^2 \le 9, x = -3..5, y = -2..6, color = black, coloring = [blue], filledregions = true\big)$
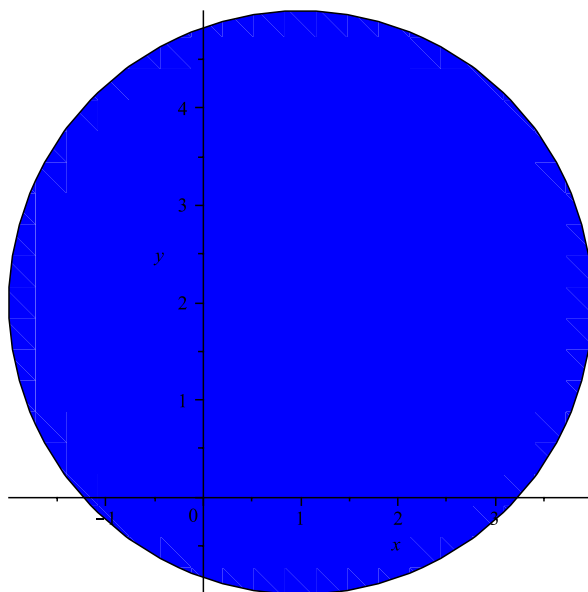
2.3 Multivariate Calculus

2.3.1 Three-Dimensional Plotting

The **plot** function, which is for two-dimensional plots, has a counterpart named **plot3d** which is rather unsurprisingly for three-dimensional plots. Ordinarily, **plot3d** will plot a function $z = f(x,y)$. To plot, for example, the paraboloid $z = x^2 + y^2$ we simply input the following.

> $plot3d(x^2 + y^2, x = -1..1, y = -1..1)$
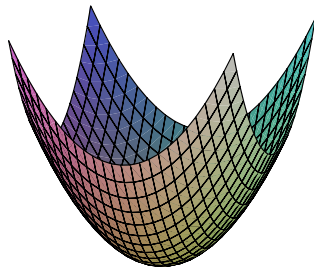
Be aware that unlike the **plot** function, **plot3d** has no default values for the input range and so we must explicitly state ranges for both independent variables. Also be aware that 3-D plots default to having the z-axis pointing "up" (which is to say up with respect to the screen), unlike 2-D plots which have the y-axis pointing up. An advantage to these plots is that they may be rotated by dragging them with the mouse, which allows a better appreciation of the overall three-dimensional shape, even though we ultimately only have a two-dimensional projection on our computer screen. Of course,

such rotation will almost always change which, if any, axis is pointing up with respect to the screen.

Just as with the **plot** command and two-dimensional plots, we may parameterize surfaces and may have **plot3d** plot them for us. We do this by asking **plot3d** to plot a list of exactly three elements, instead of an expression. The three-element list is interpreted to be the parametric values for points $[x, y, z]$. Note that this is slightly different from the two-dimensional case, in that the range for the parameter(s) is not contained within the list.

> $plot3d([v \cdot \cos(u), v \cdot \sin(u), v], u = 0..2 \cdot \text{Pi}, v = -1..1)$
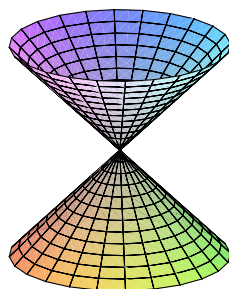
Three-dimensional plots are usually—although not always—parameterized with two parameters. The trivial parameterization, for a function $f(x, y)$ is simply $[u, v, f(u, v)]$, and should look quite similar to the trivial parameterization of a function of one variable.

We look quickly at two alternative co-ordinate systems for three-dimensional surfaces. These are *cylindrical* and *spherical* co-ordinates. Both may be invoked with an appropriate **coords** option in the **plot3d** function.

Cylindrical co-ordinates are an extension of polar co-ordinates. Any point in 3-space may be located by specifying an angle on the xy-plane, and a distance from the origin at that angle (which is identical to polar co-ordinates), and finally by a height above (or below) the xy-plane. As such a point in cylindrical co-ordinates is of the form $[r, \theta, z]$, and as the name might suggest, it is quite easy to plot a cylinder using such co-ordinates.

A word of warning. *Maple*, by default, expects cylindrical co-ordinates to be expressed in the form of r as a function of θ and z. That is, $r = f(\theta, z)$. This is probably counterintuitive, as we were most probably expecting the form $z = f(r, \theta)$ just as for Cartesian co-ordinates, where we plot functions of the form $z = f(x, y)$. The easiest way around this problem is to plot cylindrical functions as parameterized plots of the form $[r, \theta, z]$, and is precisely what we do.

> $plot3d([1, \text{theta}, z], \text{theta} = 0..2 \cdot \text{Pi}, z = -1..1, coords = cylindrical)$
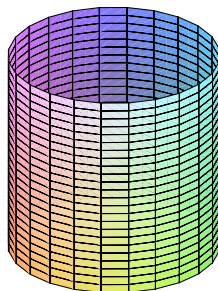
> $plot3d([r, \text{theta}, r], r = -1..1, \text{theta} = 0..2 \cdot \text{Pi}, coords = cylindrical)$
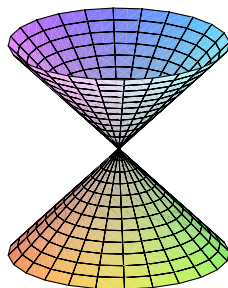
We show more cylindrical plots in Section 2.3.2.

Spherical co-ordinates are similar to cylindrical, differing only in the final co-ordinate. As with cylindrical co-ordinates, we have an r, θ pair that locates a point on the xy-plane. We then rotate that point *vertically* by an angle $\phi \in [0..\pi]$ where the angle is measured against the z-axis, with 0 pointing upwards, and π pointing downward. Thus a point in 3-space in spherical co-ordinates has the form $[r, \theta, \phi]$. Our r co-ordinate in this system will always be the distance of a point from the origin (note that this is not the case with cylindrical co-ordinates). As the name might suggest, this co-ordinate system is very well suited to locating points on a sphere. Indeed, anybody familiar with longitude and latitude on maps of earth will have seen this concept before.

As with cylindrical co-ordinates, it is usually easier to plot spherical co-ordinate plots as parametric plots, as *Maple*'s default is to expect r to be as a function of θ and ϕ.

> $plot3d([1, \text{theta}, \text{phi}], \text{theta} = 0..2 \cdot \text{Pi}, \text{phi} = -0.. \text{Pi } coords = spherical)$
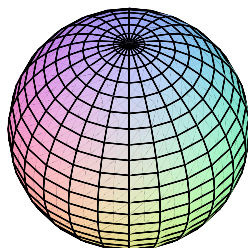
We may also plot this sphere using the **implicitplot3d** function (from the plots package), which is a three-dimensional analogue of the **implicitplot** function from Section 2.2.5.

> $plots[implicitplot3d](x^2 + y^2 + z^2 = 1, x = -1..1, y = -1..1, z = -1..1)$
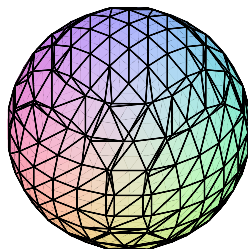
The interested reader is encouraged to check out the *Maple* help files on co-ordinates. See **?coords**, **?plot[coords]**, and **?plot3d[coords]**.

2.3.2 Surfaces and Volumes of Rotation

In general, calculating volumes is usually a job for iterated integrals (see Section 2.3.4). However, volumes of solids of revolution may be calculated with a single integral. A solid of revolution is produced by rotating a curve in the plane about a line. Usually, but not always, one of the axes is chosen. As an example, let us consider the sine curve between 0 and π.

> $plot(\sin(x), x = 0..\,\mathrm{Pi}, scaling = constrained)$
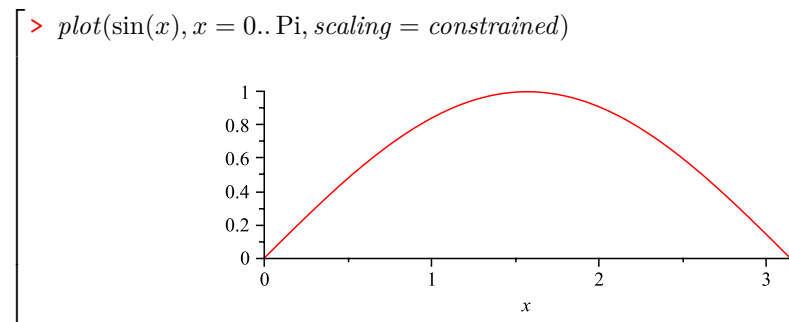
If we rotate the above sine curve about the x-axis we can imagine a sort of symmetrical teardrop shape. In fact, we can do better than imagine. We can have *Maple* draw us a picture. In order to plot our rotated sine curve, however, it is simplest to use a parametric equation. To this end, notice that at any point $x \in [0, \pi]$, our rotated surface will have a cross-section, parallel to the yz-plane which is a circle or radius $\sin(x)$. We may, therefore, parameterize the surface of the rotated surface by

$$[x, y, z] = [t, \sin(t)\sin(\theta), \sin(t)\cos(\theta)]$$

where $t \in [0, \pi]$ and $\theta \in [0, 2\pi)$.

This we may now plot.

> $plot3d([t, \sin(t) \cdot \sin(theta), \sin(t) \cdot \cos(theta)], t = 0..\,\mathrm{Pi}, theta = 0..2 \cdot \mathrm{Pi},$
   $scaling = constrained)$
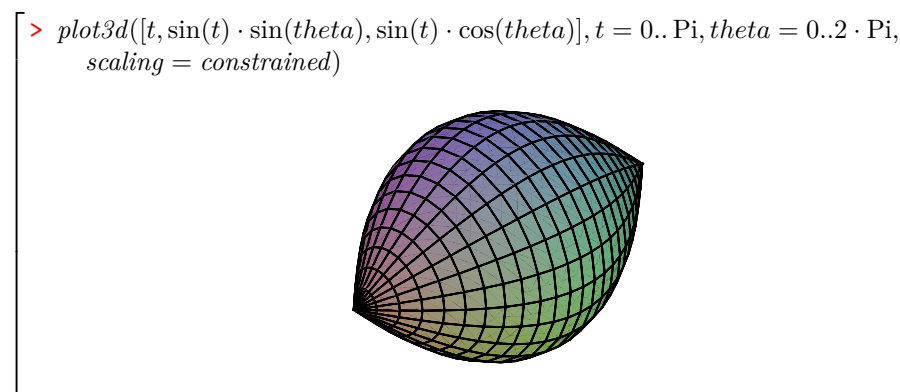
The parameterization of this surface gives us a hint as to how to use integration to calculate the volume. We think of the volume as an infinite number of infinitely small disks (otherwise known as "circles"), and add up the area of each circle over an interval, the interval being $[0..\pi]$ in this case. The accumulated area is the volume. This is, incidentally, identical to a Riemann sum where we (more or less) add up the height of an infinite number of infinitely small boxes (otherwise known as "lines") and accumulate these heights over an interval to obtain an area. The disks we are calculating are perpendicular to the axis of rotation.

We know that the area of any particular disk is πr^2, and because our radius is $\sin(x)$, each disk in our particular example will have area $A(x) = \pi \sin(x)^2$. So the area,

remembering we have only rotated the portion of the sine curve between 0 and π, is

$$\int_0^\pi A(x)\,dx = \int_0^\pi \pi \sin(x)^2\,dx = \pi \int_0^\pi \sin(x)^2\,dx$$

This we may now calculate in *Maple* or manually, as we see fit.

> $\mathrm{Pi} \cdot int(\sin(x)^2, x = 0..Pi)$

$$\frac{1}{2}\,\pi^2$$

In general, the volume of a solid produced by rotating a function $f(x)$ inside an interval $[a, b]$ around the x-axis is given by the integral

$$\pi \int_a^b f(x)^2\,dx$$

If we wish to rotate a function $f(x)$ around the y-axis, then we need to rewrite the function as a function of y instead. Note that this is more complicated than just replacing every x in the function with a y. For example, let us rotate the parabola $y = x^2$ around the y-axis for $x \in [0, 2]$.

We start by plotting what we want to see. First, however, remember that with a three-dimensional plot it is the z-axis which is pointing upwards, but for a two-dimensional plot it is the y-axis that is pointing upwards. This is easily fixed by just renaming the y-axis to be the z-axis instead, giving us our "new" function of $z = x^2$, being rotated around the z-axis. For the remainder of this section, and any time we discuss volumes of revolution, we consider the 2-D plots to be in the xz-plane.

Once this is done, we notice that when rotating a curve around the z-axis, we have a natural cylindrical co-ordinate representation, r representing the distance from the origin (between 0 and 2), θ as the angle, and $z = r^2$ as the height above that point.

> $plot3d([r, theta, r^2], r = 0..2, theta = 0..2 \cdot \mathrm{Pi}, coords = cylindrical)$
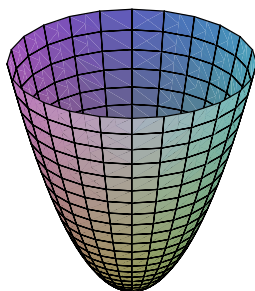
Straight away we have an issue (although it is not immediately obvious). It is not clear from this plot whether we mean the volume between the paraboloid and the z-axis, or the volume "below" the paraboloid, down to the xy-plane. In fact we meant the latter, which is troublesome to visualize on that plot.

We remedy matters here. It is useful to be explicit about precisely which area we are rotating around the z-axis. In this case, we wish to take the volume under the curve $z = x^2$, but above the x-axis, and between the values $x = 0$ and $x = 2$. By plotting these bounds in two dimensions we should make it clearer exactly which area will be rotated.

> $plot([[x, x^2, x = 0..2], [2, z, z = 0..4], [x, 0, x = 0..2]], x = 0..2, z, color = red)$
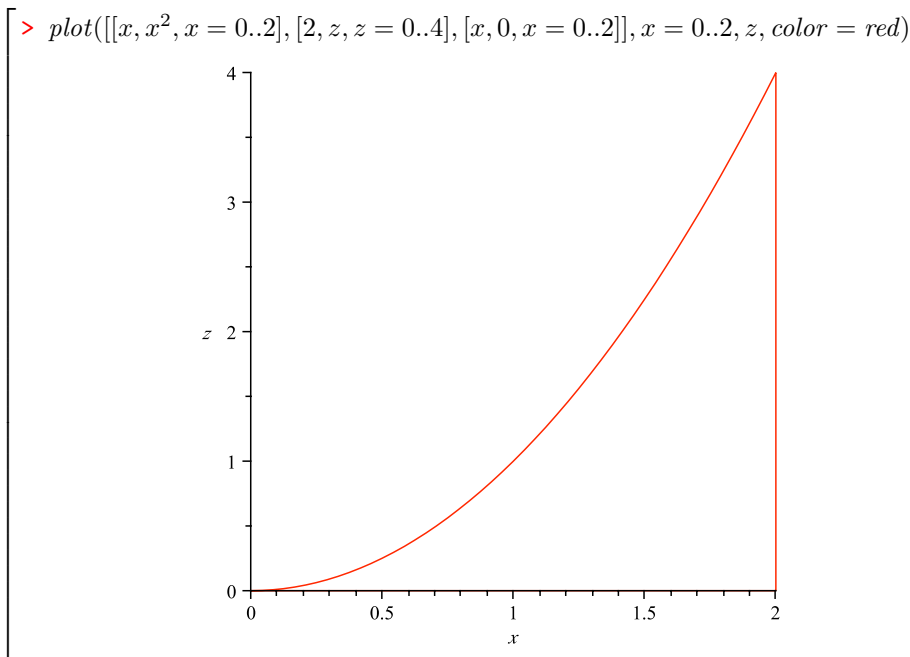
We can't see the line that forms the lower bound, because the x-axis is obscuring it, nonetheless it is quite clear now exactly which area is being rotated. We plot the volume of revolution now, and again—just as with the 2-D plot above—we include the bounds as well. The line where $x = 2$ becomes a cylinder when rotated, and the line at the bottom of the plot becomes a disk. We are also cunning and leave a part of the solid open, so that we can see inside for a better impression of the solid in question.

> $plot3d \left(\left[\left[x, \text{theta}, x^2\right], \left[2, \text{theta}, x^2\right], [x, \text{theta}, 0] \right], x = 0..2, \text{theta} = \right.$
>
> $\left. 0..\dfrac{5 \cdot \text{Pi}}{3}, coords = cylindrical \right)$
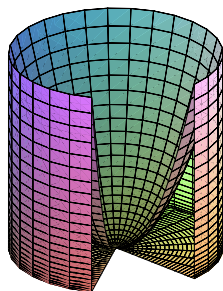
Now in order to use our integral, above, we need to be rotating around the same axis as the independent variable of our function. Here, however, we are rotating a function of x around the z-axis (still using the z-axis as the upward pointing one). We need to rewrite our function as $x = f(z)$. Rearranging $z = x^2$ in this manner produces $x = \sqrt{z}$. We also notice that $x = 0 \implies z = 0$ and $x = 2 \implies z = 4$, so we may equivalently consider our rotation as rotating the function $x = \sqrt{z}$ for $z \in [0, 4]$ around the z-axis. However, it is the area of the function above $x = \sqrt{z}$ that we are rotating around the z-axis. The upper bound for this function is the line $x = 2$.

> $plot([\text{sqrt}(z), 2], z = 0..4, x, color = red)$
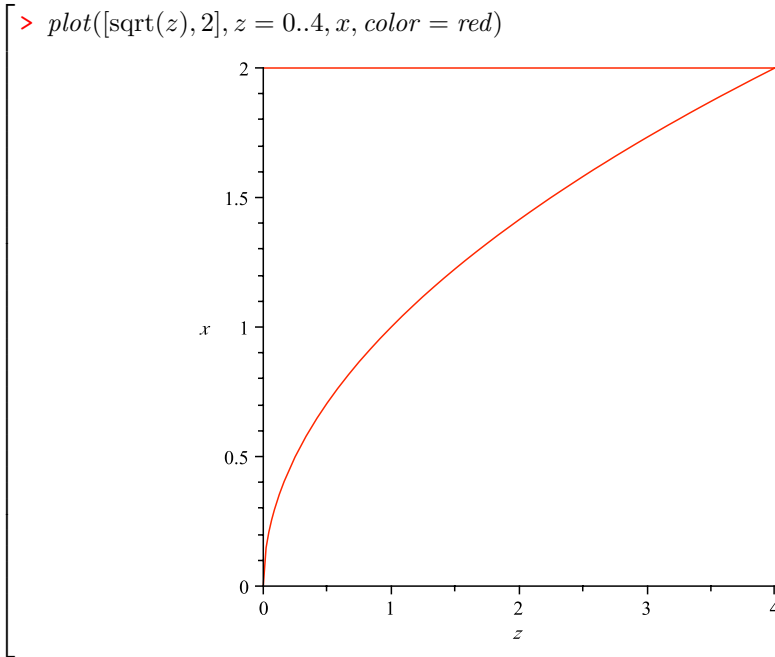
So, what we actually want is the area between the two curves $x = 2$ and $x = \sqrt{z}$ for $0 \leq z \leq 4$ to be rotated. That's easy. Observe that $x = 2$ is always larger than $z = \sqrt{z}$ on the interval in question. For the area itself, we could happily calculate the integral of the larger minus the integral of the smaller. We may take the same approach with the volume integral. If we calculate the volume of the cylinder we obtain by rotating $x = 2$ around the z-axis, and subtract from that the volume of the paraboloid (obtained by rotating the volume under the function $x = \sqrt{z}$) then we have calculated the precise area we wanted. This is demonstrated in Figure 2.2.
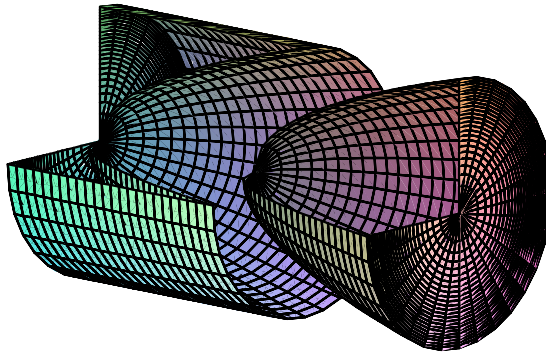
Fig. 2.2 Cylinder with paraboloid removed.

In short, we wish to calculate

$$\pi \int_0^4 2^2 \, dz - \pi \int_0^4 \sqrt{z}^2 \, dz = \pi \int_0^4 4 - z \, dz$$

> $\text{Pi} \cdot int(4 - z, z = 0..4)$

$$8\,\pi$$

Let us return now to our sine function, plotted between $x = 0$ and $x = \pi$. We now rotate this around the z-axis (treating z as the axis pointing "up" again, as we did with the last example). We start as we have each other time, with plotting the surface to see with what we're dealing. We add a **scaling** parameter to make the plot clearer in this case.

> $plot3d([x \cdot \sin(theta), x \cdot \cos(theta), \sin(x)], x = 0..\text{Pi}, theta = 0..2 \cdot \text{Pi},$
    $scaling = constrained)$
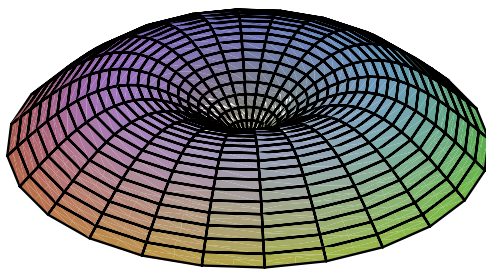
Now we have a problem here, inasmuch as there's no particularly easy or, at least, obvious way to rewrite $z = \sin(x)$ as a function of z. The z-interval is clearly $[0, 1]$, however, for any given value of z there are two values of x. This is easily seen with a plot.

> $plot([sin(x), 0.9], x = 0..\text{Pi}, z, color = [red, blue], scaling = constrained)$
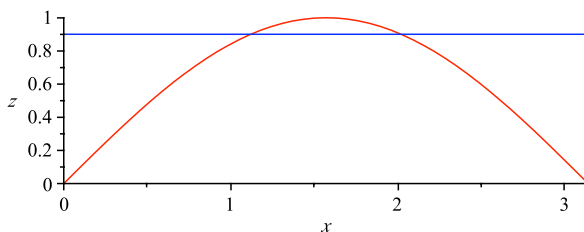
The line $z = 0.9$ clearly cuts the sine function in two places.

Now, we may try and express the solid of rotation as an area between two functions, $f(z), g(z)$ and calculate the integral accordingly. However finding these two functions is tedious and time consuming. Instead, we use a slightly different integral to calculate the area. Our previous method used disks that were perpendicular to the axis of rotation. This time we use a method known as "shells". To do this, we approximate the area as infinitely many cylinders, centered at the origin, and with radius x and height $\sin(x)$. The surface area of each cylinder is the circumference multiplied by the height. As such, the area function is $A(x) = 2\pi x \sin(x)$, and our volume may now be calculated as

$$\int_0^\pi A(x)\,dx = \int_0^\pi 2\pi x \sin(x)\,dx = 2\pi \int_0^\pi x \sin(x)\,dx$$

> $2 \cdot \text{Pi} \cdot int(x \cdot \sin(x), x = 0..\text{Pi})$

$$2\,\pi^2$$

And, in general, if we rotate a function $y = f(x)$ between $x = a$ and $x = b$ around the y-axis then the volume of the solid of revolution is given by the integral

$$2\pi \int_a^b x f(x)\, dx$$

And we always have the possibility of changing between these two integrals, by re-writing the function and interchanging the dependent and independent variables if the integration in one method proves too troublesome.

We may check our paraboloid volume calculation using this method.

> $2 \cdot \mathrm{Pi} \cdot int(x \cdot x^2, x = 0..2)$

$$8\,\pi$$

2.3.3 Partial and Directional Derivatives

Recall that for a function of two or more variables, the derivatives are taken with respect to one of the variables at a time. These are known as *partial derivatives*. There are several ways to denote partial derivatives, of which we use the following two. Let $f : \mathbb{R}^2 \to \mathbb{R}$. Then the first partial derivative of f with respect to x is denoted by

$$f_x \;\; \text{or} \;\; \frac{\partial f}{\partial x}$$

and the first partial derivative of f with respect to y is denoted by

$$f_y \;\; \text{or} \;\; \frac{\partial f}{\partial y}$$

Both of these are functions ($\mathbb{R}^2 \to \mathbb{R}$) in their own right.

Note that, if we do not have a name for our function, we may write the function in brackets after the ∂ notation. For example,

$$\frac{\partial}{\partial x} \left(\frac{x^2 + y^2}{x^2 - y^2} \right)$$

would be the first partial derivative with respect to x of the rational polynomial $(x^2 + y^2)/(x^2 - y^2)$.

If we take the derivative of one of these derivatives, then we again may take a partial derivative. By doing so we obtain a *second partial derivative*. The variable with respect to which the second derivative is taken may be different from that with respect to which the first derivative was taken (because a first partial derivative is still a valid function in its own right). The four possibilities for a second partial derivative are as follows.

$$f_{x,x} \;\; \text{or} \;\; \frac{\partial^2 f}{\partial x^2} \qquad\qquad\qquad f_{x,y} \;\; \text{or} \;\; \frac{\partial^2 f}{\partial y \partial x}$$

$$f_{y,x} \;\; \text{or} \;\; \frac{\partial^2 f}{\partial x \partial y} \qquad\qquad\qquad f_{y,y} \;\; \text{or} \;\; \frac{\partial^2 f}{\partial y^2}$$

In general if we have a function that is an nth partial derivative, which we then take yet another partial derivative of, then we have the following scenario.

$$(f_{x_1,\ldots,x_n})_{x_{n+1}} = f_{x_1,\ldots,x_{n+1}} \ \text{ or } \ \frac{\partial}{\partial x_{n+1}} \left(\frac{\partial^n}{\partial x_n \cdots \partial x_1} \right) = \frac{\partial^{n+1}}{\partial x_{n+1} \cdots \partial x_1}$$

Note that the above also demonstrates why the ∂ notation has the variables written backwards.

Like standard derivatives, partial derivatives are defined in terms of the limit of the slopes of a series of lines between the point at which we wish to calculate the derivative, and another point near to it, as follows.

$$f_x(x,y) := \lim_{h \to 0} \frac{f(x+h, y) - f(x,y)}{h} \ \text{ or } \ f_y(x,y) := \lim_{h \to 0} \frac{f(x, y+h) - f(x,y)}{h}$$

Maple may perform partial derivatives. Indeed, the **diff** function we have already used for regular differentiation will happily perform partial differentiation. For a first partial derivative, we tell **diff** with respect to which variable we want to take the derivative.

```
> p := sum(sum(x^i · y^j, j = 0..2), i = 0..2);
  diff(p, x); diff(p, y)
```
$$p := 1 + y + y^2 + x + xy + xy^2 + x^2 + x^2 y + x^2 y^2$$
$$1 + y + y^2 + 2x + 2xy + 2xy^2$$
$$1 + 2y + x + 2xy + x^2 + 2x^2 y$$

And for second and later partial derivatives, we simply list the derivatives in the order they are to be taken. Note that this is exactly what we did for single variable derivatives, only there was only one variable in that case. So $\partial^2 f / \partial x^2$ was calculated with $diff(f(x), x, x)$ which told *Maple* to take the first derivative with respect to x and then the second derivative with respect to x. Multivariable derivatives are different only in that we may take the derivative with respect to a different variable at each step.

```
> diff(p, x, x); diff(p, x, y);
  diff(p, y, x); diff(p, y, y)
```
$$2 + 2y + 2y^2$$
$$1 + 2y + 2x + 4xy$$
$$1 + 2y + 2x + 4xy$$
$$2 + 2x + 2x^2$$

Of course, the inert form of the function still works, as do—as suggested above—higher-order derivatives.

```
> Diff(p, x, x, y) = diff(p, x, x, y)
```
$$\frac{\partial^3}{\partial y \partial x^2} \left(1 + y + y^2 + x + xy + xy^2 + x^2 + x^2 y + x^2 y^2 \right) = 2 + 4y$$

The **D** function may also be used. For a function of multiple variables, to compute the first partial derivative with respect to the first variable, we use the subscript 1 with the D. To compute the first partial derivative with respect to the second variable, we use a subscript of 2, and so on. For example,

> $f := (x, y) \rightarrow x^2 - y^2 + x \cdot y; \mathrm{D}_1(f), \mathrm{D}_2(f)$

$$f := (x, y) \rightarrow x^2 - y^2 + xy$$
$$(x, y) \rightarrow 2x + y$$
$$(x, y) \rightarrow -2y + x$$

To compute second partial derivatives, we simply subscript **D** with a sequence of numbers describing the variable numbers, in the order that the derivatives are taken. The following examples may be quickly checked by hand.

> $\mathit{Diff}(f(x, y), x\$2) = \mathrm{D}_{1,1}(f)(x, y)$
$\mathit{Diff}(f(x, y), y, x) = \mathrm{D}_{2,1}(f)(x, y);$

$$\frac{\partial^2}{\partial x^2}\left(x^2 - y^2 + xy\right) = 2$$
$$\frac{\partial^2}{\partial x \partial y}\left(x^2 - y^2 + xy\right) = 1$$

Note that although this is different from the method we used to take derivatives of single variable functions with the **D** command, this method also works for single variable functions.

Recall that for a function with continuous second partial derivatives, then the second partial derivatives, $f_{y,x}$ and $f_{x,y}$ will be equal. This is Clairaut's theorem (see [12]). More precisely:

Theorem 1 (Clairaut's). *Let f be a function defined on a disk $D \subset \mathbb{R}^2$, such that all the second partial derivatives are continuous on D. Then $f_{y,x}(a, b) = f_{x,y}(a, b)$ for every $(a, b) \in D$.*

It follows, of course, that if the function f is defined for all of $\mathbb{R}^2$ and its partial derivatives are continuous on all of $\mathbb{R}^2$ then it will certainly be the case that $f_{y,x} = f_{x,y}$.

We have seen an example of this above, with the second partial derivatives of our polynomial p. A quick check of our first partial derivatives of our function f shows that the second partial derivatives $f_{x,y}$ and $f_{y,x}$ are clearly equal. Let's look at another example.

> $f := \sin\left(x^2 + y^2\right); \mathit{diff}(f, x, y) = \mathit{diff}(f, y, x);$

$$f := \sin\left(x^2 + y^2\right)$$
$$-4 \sin\left(x^2 + y^2\right) yx = -4 \sin\left(x^2 + y^2\right) yx$$

And this even extends to higher partial derivatives.

> $\mathit{diff}(f, x, x, y); \mathit{diff}(f, x, y, x); \mathit{diff}(f, y, x, x)$

$$-8 \cos\left(x^2 + y^2\right) yx^2 - 4 \sin\left(x^2 + y^2\right) y$$
$$-8 \cos\left(x^2 + y^2\right) yx^2 - 4 \sin\left(x^2 + y^2\right) y$$
$$-8 \cos\left(x^2 + y^2\right) yx^2 - 4 \sin\left(x^2 + y^2\right) y$$

For an example where Clairaut's theorem does not hold, see Exercise 14.

The partial derivatives, just as regular derivatives, allow calculation of a line that is tangent to a surface, $f(x, y)$ say. However, there are potentially many such tangent lines in a three-dimensional space. As such, the partial derivatives f_x and f_y give the slope of a tangent line parallel to the x- or y-axes, respectively.

Lines in three-dimensional space are tricky, but may be plotted with a parametric equation. In the case of our directional derivatives, we know a point on the line, the direction each line is traveling, and the rate at which the height of each line changes (the slope). In short, we have all the information we need to plot the tangent lines.

For a surface $f(x, y)$ we may use the following parameterization of the tangent lines.

$$[x_0 + u, y_0, f(x_0, y_0) + u \cdot f_x(x_0, y_0)] \quad \text{in the } x \text{ direction}$$
$$[x_0, y_0 + v, f(x_0, y_0) + v \cdot f_y(x_0, y_0)] \quad \text{in the } y \text{ direction}$$

and, of course, the obvious parameterization of the surface itself as $[u, v, f(u, v)]$.

```
> tl_x := proc(f :: procedure)
     (x0, y0) → [x0 + u, y0, f(x0, y0) + u · D_1(f)(x0, y0)]
  end :
  tl_y := proc(f :: procedure)
     (x0, y0) → [x0, y0 + v, f(x0, y0) + v · D_2(f)(x0, y0)]
  end :
```

```
> f := (x, y) → x² + y² :
  plot3d([[u, v, f(u, v)], tl_x(f)(0, 1), tl_y(f)(0, 1)], u = −1..1, v = −1..1,
     axes = framed)
```
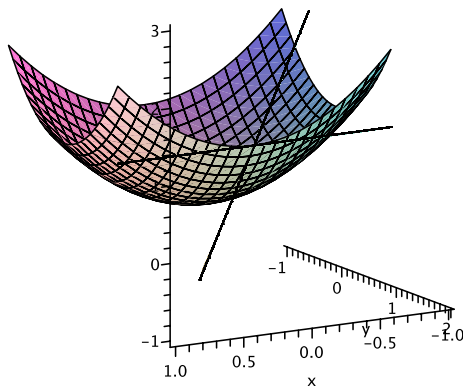

The directional derivatives allow us to calculate the *tangent plane* to a surface in 3-D. Given a function, $f(x, y)$ say, with continuous derivatives the equation of the tangent plane to the surface of f at a point (x_0, y_0, z_0) is

$$z - z_0 = \frac{\partial f}{\partial x}(x_0, y_0) \cdot (x - x_0) + \frac{\partial f}{\partial y}(x_0, y_0) \cdot (y - y_0)$$

which may be rewritten as

$$z = \frac{\partial f}{\partial x}(x_0, y_0) \cdot (x - x_0) + \frac{\partial f}{\partial y}(x_0, y_0) \cdot (y - y_0) + z_0$$
$$= \frac{\partial f}{\partial x}(x_0, y_0) \cdot (x - x_0) + \frac{\partial f}{\partial y}(x_0, y_0) \cdot (y - y_0) + f(x_0, y_0)$$

because if (x_0, y_0, z_0) is a point on the surface of $f(x, y)$, then it must be the case that $z_0 = f(x_0, y_0)$.

We may (and, indeed, do) explore this in *Maple*. First we create a procedure that returns an arrow-notation function for the tangent plane equation.

```
> tp := proc(f :: procedure)
     (x0, y0) → f(x0, y0) + D_1(f)(x0, y0) · (x − x0) + D_2(f)(x0, y0) · (y − y0)
  end :
```

Now we have a function *tp* which, when given a procedure as input, will return a new function that calculates the tangent plane at a given point. Because the *tp* creates a function, we expect to be able to use the expression $tp(f)(a, b)$ to calculate the tangent plane to the function f at the point (a, b).

> $f := (x, y) \rightarrow x^2 + y^2; tp(f)(a, b)$

$$f := (x, y) \rightarrow x^2 + y^2$$

$$a^2 + b^2 + 2\,a\,(x - a) + 2\,b\,(y - b)$$

Well, that's all well and good, but it's about time we produced some plots.

> $plot3d([f(x, y), tp(f)(0, 0)], x = -2..2, y = -2..2);$
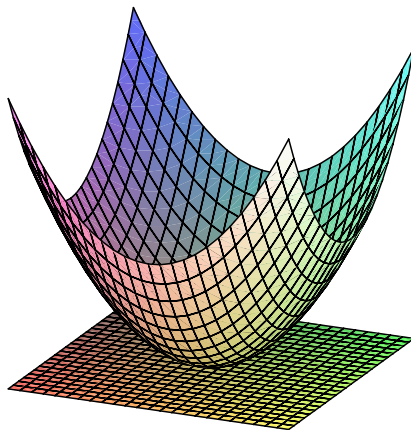
> $plot3d([f(x, y), tp(f)(1, 1)], x = -2..2, y = -2..2);$
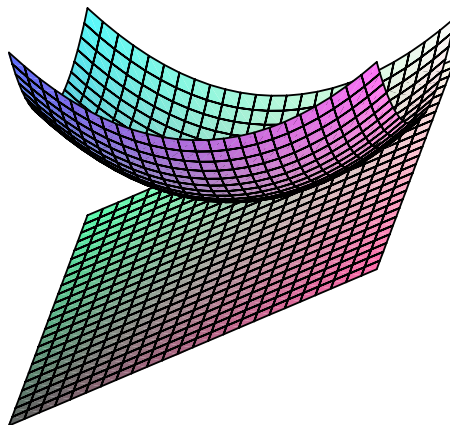
We may calculate tangent lines (and so instantaneous rate of change) at a point in both the x- and y-directions as we wish. Furthermore, we may use this information to calculate the tangent plane at the same point. It should follow, therefore, that we ought to be able to calculate the slope of a tangent line in any direction, and also that it should lie on the tangent plane we have already calculated.

To do this, we need a vector of unit length pointing in the direction we wish to calculate the slope. Call this vector $u = (a, b)$. Then we may calculate the directional derivative in the direction of u at any point (x, y)

$$D_u f(x,y) := f_x(x,y) \cdot a + f_y(x,y) \cdot b$$

We show that this vector is on the tangent plane. If we ignore the z-axis for the moment, we can parameterize the line from a point (x_0, y_0) in the direction of u as $[x_0 + ta, y_0 + tb]$ (think of the vector equation $(x_0, y_0) + t \cdot (a, b)$). Getting back to thinking three-dimensionally, we know that this line is climbing at the rate of $D_u f(x_0, y_0)$ in the z axis for each unit moved in the direction of u, which is supposed to be of unit length anyway.

This leads us to the parameterization of the line as

$$[x_0 + t \cdot a, y_0 + t \cdot b, f(x_0, y_0) + t \cdot D_u f(x_0, y_0)]$$

and so if a point on this line is on the plane, which we should recall has equation

$$z = f(x_0, y_0) + f_x(x_0, y_0) \cdot (x - x_0) + f_y(x_0, y_0) \cdot (y - y_0)$$

then it must be the case that every point on the line as parameterized above, satisfies the equation. Over to *Maple* now.

> $f :=' f' :$
  $tp := (x0, y0) \rightarrow f(x0, y0) + D_1(f)(x0, y0) \cdot (x - x0) + D_2(f)(x0, y0)$
      $\cdot (y - y0)$;
  $d_u := (x0, y0) \rightarrow D_1(f)(x0, y0) \cdot a + D_2(f)(x0, y0) \cdot b;$

$$tp := (x0, y0) \rightarrow f(x0, y0) + D_1(f)(x0, y0)(x - x0) + D_2(f)(x0, y0)(y - y0)$$
$$d_u := (x0, y0) \rightarrow D_1(f)(x0, y0)a + D_2(f)(x0, y0)b$$

> $subs(\{x = x0 - t \cdot a, y = y0 - t \cdot b\}, tp(x0, y0))$
  $f(x0, y0) + t \cdot (d_u(x0, y0))$;

$$f(x0, y0) + D_1(f)(x0, y0)ta + D_2(f)(x0, y0)tb$$
$$f(x0, y0) + t(D_1(f)(x0, y0)a + D_2(f)(x0, y0)b)$$

> $simplify(\% - \%\%)$

$$0$$

And there we have it. Notice that we needed to reset the f variable in order to maintain proper generality. We could probably have done without the final subtraction, inasmuch as it was clear that the two calculated expressions were the same, but it never hurts to check and we didn't have to go out of our way to do so. In any event, it is clear that the directional derivative in the direction of a unit vector u will give us the slope of a tangent line that lies on the tangent plane and travels parallel to the direction of u.

2.3.4 Double Integrals

Just as the space under a curve, but above the x-axis, may be calculated as an area via integration, so too can the space under a surface (i.e., a function $z = f(x,y)$), and above the x-y plane, may be calculated as a volume using integration.

The same basic approach applies. Given a range for x and y we have a rectangle that is a subsection of the xy-plane. We partition the x and y ranges, resulting in the area under the surface being partitioned into rectangles. We then create rectangular prisms

by choosing a point inside every rectangle, and setting the height of that rectangular prism to be the height of the function evaluated at that point. This should sound awfully similar to the method of approximating the area under a curve with rectangles.

If we let A_{ij} be the area of the (i, j)th subrectangle (i.e., the rectangle that is at the intersection of the ith x-partition and jth y-partition), and (x_{ij}^*, y_{ij}^*) be a point inside that subrectangle, then the area of the resulting rectangular prism is $f(x_{ij}^*, y_{ij}^*)A_{ij}$. The volume underneath the surface (and above the xy-plane) may be approximated by adding up the volumes of all the rectangular prisms.

This is well demonstrated in *Maple* using the **ApproximateInt** command from the Student[MultivariateCalculus] package. Note that in the *Maple* worksheet itself, the following two commands produce animations, and not just static pictures. Such animation would be long and tedious to produce by hand (although quite possible with the **animate** command from the plots package). Note that the text above the images is automatically generated and packaged part of the output of **ApproximateInt**.

We may, as it happens, use a more interactive manner of exploration. *Maple* provides a number of interactive demonstration screens, refereed to as *tutors*. In particular is the **ApproximateIntTutor**, which may also be used to produce the following animations. The tutor has the advantage of being interactive, allowing experimentation with quickly seen results.

The tutors are best used for exploration. Unfortunately due to their interactive nature, they are troublesome to explain in print, and so fall out of the scope of this book. They may be accessed through the *Tools* menu, which has a *Tutors* submenu, and cover many areas of university mathematics appropriate to students. The reader is encouraged to explore these for themselves (see also Exercise 1).

> $Student[MultivariateCalculus][ApproximateInt](x^2 + y^2, x = -2..2, y = -2..2,$
> $method = midpoint, coordinates = cartesian, partition = [25, 25],$
> $output = animation)$

An animation of approximations to the integral of
f(x, y) = x^2+y^2
over the region [–2 .. 2, –2 .. 2]
using a midpoint Riemann sum
Actual value: 42.667
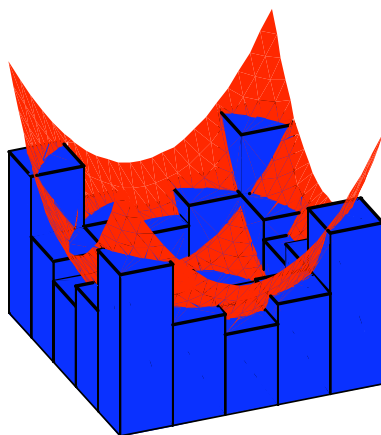
The above is using midpoints as the sample points (x_{ij}^*, y_{ij}^*). Random points work just as well, as it happens (and, incidentally, also work for Riemann sums of single variable functions).

> $Student[MultivariateCalculus][ApproximateInt](x^2 + y^2, x = -2..2, y = -2..2,$
> $method = random, coordinates = cartesian, partition = [25, 25],$
> $output = animation)$

An animation of approximations to the integral of
f(x, y) = x^2+y^2
over the region [-2 .. 2, -2 .. 2]
using a random Riemann sum
Actual value: 42.667
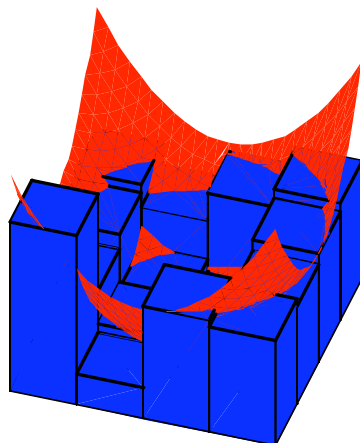
We obtain the volume under the curve by taking progressively more and more rectangular prisms (which are, in turn, smaller and smaller). As such the volume under the surface may be approximated by the double sum

$$\sum_{i=1}^{n} \sum_{j=1}^{m} f(x_{ij}^*, y_{ij}^*) A_i j$$

and the volume as the limit of this sum as both n and m approach ∞.

$$V = \lim_{n,m \to \infty} \sum_{i=1}^{n} \sum_{j=1}^{m} f(x_{ij}^*, y_{ij}^*) A_i j$$

The question now is, how do we turn this into an integral? The key lies in recognizing that the two sums (above) may be taken to infinity independently. In integration terms, fix y as a constant, and calculate the integral of the function as if it were just a function of the single variable x. The result will be a function of y that tells us the *area* under the surface for any given y value. Integrating this function with respect to y will give us the volume. This is, in fact, very similar to our volumes of revolution from Section 2.3.2. If we think of this as accumulation, we accumulate an infinite amount of areas to obtain a volume, just as we did with the disks or cylinders of the solids of revolution.

To make this more rigorous, let $f(x, y)$ be a function of two variables. We wish to find the volume under the surface of f for $x \in [x_1, x_2]$ and $y \in [y_1, y_2]$. In other words we want to integrate over the rectangle $[x_1, x_2] \times [y_1, y_2]$. Furthermore, suppose that $f(x, y)$ is *continuous* over that rectangle. The integral $\int_{x_1}^{x_2} f(x, y) \, dx$ is used to denote integrating with respect to x while holding y fixed. Similarly $\int_{y_1}^{y_2} f(x, y) \, dy$ denotes integrating with respect to y while holding x to be fixed. These are called *partial integrals* (compare to partial derivatives).

Then the volume can be calculated as

$$V = \int_{y_1}^{y_2} \left(\int_{x_1}^{x_2} f(x, y)\, dx \right) dy$$

which, for simplicity's sake is usually just written without the bracketing, as it is understood that the innermost integral needs to be performed first, before the outermost integral may be performed

$$V = \int_{y_1}^{y_2} \int_{x_1}^{x_2} f(x, y)\, dx\, dy$$

Let's look at this in *Maple* using our favorite paraboloid, over the rectangle $[-2, 2] \times [-2, 2]$.

> $f := x^2 + y^2;$
> $Int(f, x = -2..2) = int(f, x = -2..2)$

$$f := x^2 + y^2$$

$$\int_{-2}^{2} \left(x^2 + y^2\right) dx = \frac{16}{3} + 4\,y^2$$

> $Int(rhs(\%), y = -2..2) = int(rhs(\%), y = -2..2)$

$$\int_{-2}^{2} \left(\frac{16}{3} + 4\,y^2\right) dy = \frac{128}{3}$$

We see from the above that the first integration did indeed leave us with a function of y. We also see that, apparently, the volume is 128/3 cubic units. We should hope, given the double sum definition, and the basic idea that we are calculating a volume, that if the order of integration is reversed, we should obtain the same answer. As it happens, we do indeed.

> $Int(f, y = -2..2) = int(f, y = -2..2)$

$$\int_{-2}^{2} \left(x^2 + y^2\right) dy = 4\,x^2 + \frac{16}{3}$$

> $Int(rhs(\%), x = -2..2) = int(rhs(\%), x = -2..2)$

$$\int_{-2}^{2} \left(4\,x^2 + \frac{16}{3}\right) dx = \frac{128}{3}$$

This guarantee that the double integral will always give the same answer, no matter the order the integrals are performed in, is given by Fubini's theorem. This guarantee is dependent on the function f being continuous over the rectangle in question. In fact, it is even true sometimes when f is not continuous over the rectangle, but we do not concern ourselves with the generalization.

Maple is often capable, it should come as no surprise to find, of handling double (even multiple) integrals without the need to manually perform each single integral. This is achieved by simply providing the ranges for x and y in a list, in the order they are to be performed.

> $Int(f, [x = -2..2, y = -2..2]) = int(f, [x = -2..2, y = -2..2])$
> $Int(f, [y = -2..2, x = -2..2]) = int(f, [y = -2..2, x = -2..2])$

$$\int_{-2}^{2} \int_{-2}^{2} \left(x^2 + y^2 \right) dxdy = \frac{128}{3}$$

$$\int_{-2}^{2} \int_{-2}^{2} \left(x^2 + y^2 \right) dydx = \frac{128}{3}$$

The reader may have noticed that the volume we get for this paraboloid is somewhat larger than the 8π cubic units of the paraboloid we obtained from revolving $z = x^2$ around the z-axis. This is explained by remembering that the volume of revolution was calculated with a circular base, whereas our paraboloid has a square base. The extra area under the corners (where, incidentally, the paraboloid realizes its largest values) explains the discrepancy.

To see this, we can adopt a couple of approaches. The first (and least satisfying) is to plot both surfaces together and see if we can see one completely contained in the other.

> $P1 := plot3d(x^2 + y^2, x = -2..2, y = -2..2, color = blue) :$
> $P2 := plot3d([r, \text{theta}, r^2], r = 0..2, \text{theta} = 0..2 \cdot \text{Pi}, coords = cylindrical,$
> $color = red) :$

> $plots[display]([P1, P2])$
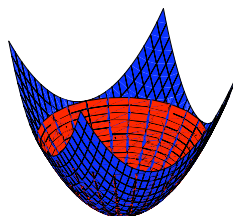
We can certainly see where the revolved paraboloid stops, and the other continues. However, a more compelling way of convincing ourselves of this would be to make a new function that is $x^2 + y^2$ as long as $\sqrt{x^2 + y^2} \leq 2$ and 0 otherwise. That is, we want

$$F(x, y) = \begin{cases} x^2 + y^2 & \text{if } \sqrt{x^2 + y^2} \leq 2 \\ 0 & \text{otherwise} \end{cases}$$

Integrating this function then should only give us the area of the paraboloid under the circle of radius 2 centered at the origin, and thus our 8π volume.

> $F := (x, y) \rightarrow piecewise(\text{sqrt}(x^2 + y^2) \leq 2, x^2 + y^2, 0);$
> $int(F(x, y), [x = -2..2, y = -2..2])$

$$F := (x, y) \rightarrow piecewise\left(\sqrt{x^2 + y^2} \leq 2, x^2 + y^2, 0 \right)$$

$$8\pi$$

Readers familiar with iterated integrals are be able to verify this using integration techniques for type I or type II regions (see [12]). We content ourselves with the above.

Let us return to the idea of partial integration. We should expect the notions of integrals as antiderivatives to extend to partial integrals and partial differentiation. We explore this in *Maple*.

> $int(f, x); diff(\%, x)$

$$\frac{1}{3}\,x^3 + y^2 x$$

$$x^2 + y^2$$

> $int(f, y); diff(\%, y)$

$$x^2 y + \frac{1}{3}\,y^3$$

$$x^2 + y^2$$

That certainly looks promising. Partial integration with respect to x or y is undone by partial differentiation with respect to x or y as appropriate. Also interesting is that if we follow the usual substitution done by hand in integration, we can probably obtain the definite integrals *Maple* calculated for us earlier.

> $int(f, x); subs(x = 2, \%) - subs(x = -2, \%);$

$$\frac{1}{3}\,x^3 + y^2 x$$

$$\frac{16}{3} + 4\,y^2$$

> $int(f, y); subs(y = 2, \%) - subs(y = -2, \%);$

$$x^2 y + \frac{1}{3}\,y^3$$

$$4\,x^2 + \frac{16}{3}$$

That is, we have shown that

$$\left[\frac{1}{3}\,x^3 + y^2 x\right]_{x=-2}^{x=2} = \frac{16}{3} + 4\,y^2 \;\; \text{and} \;\; \left[x^2 y + \frac{1}{3}\,y^3\right]_{y=-2}^{y=2} = 4\,x^2 + \frac{16}{3}$$

which is all as it should be, but is nonetheless nice to have verified.

Finally, we try a truly general function, and see if *Maple*'s partial differentiation and partial integration still undo each other.

> $Diff(Int(g(x, y), x), x) = diff(int(g(x, y), x), x);$
> $Int(Diff(g(x, y), x), x) = int(diff(g(x, y), x), x)$

$$\frac{\partial}{\partial x} \int g(x, y)\, dx = g(x, y)$$

$$\int \frac{\partial}{\partial x} g(x, y)\, dx = g(x, y)$$

> $Diff(Int(g(x, y), y), y) = diff(int(g(x, y), y), y);$
> $Int(Diff(g(x, y), y), y) = int(diff(g(x, y), y), y)$

$$\frac{\partial}{\partial y} \int g(x, y)\, dy = g(x, y)$$

$$\int \frac{\partial}{\partial y} g(x, y)\, dy = g(x, y)$$

Maple certainly seems to think that partial differentiation and partial integration are inverse procedures.

We have only just scraped the surface here with double integrals in particular, and multivariate calculus in general. The interested reader is encouraged to read [12] and/or

any other good calculus texts. Exploration and confirmation of such material should be possible with the tools used and discussed in this section.

2.4 Exercises

The exercises for this, and subsequent, chapters are less numerous and slightly longer in duration when compared to the exercises from Chapter 1. An effort has been made to keep the amount of work each question requires roughly the same for each question, and also for each question to be more or less self-contained. However, no guarantees to this effect are made.

1. *Maple* provides a group of packages all collected into a super-package named Student. The goal of these packages is to "assist with the teaching and learning of standard undergraduate mathematics." In particular is the Student[Calculus1] package that deals with single variable calculus. Read the help information on this package (**?Student[Calculus1]**), giving particular (but not exclusive) attention to the *visualization* and *interractive* sections. Experiment with some of these commands and tutors.
   Note: The tutors may alternatively be accessed directly through the *Tools* menu.
2. Plot the following functions. Make sure that, where possible, you plot all important information to the plot; turning points, zeroes, and so on.

   a. $x^5 - 7x^4 - 162x^3 + 878x^2 + 3937x - 15015$
   b. $\dfrac{\sin x}{x}$
   c. $\dfrac{\cos x - 1}{x}$
   d. $x^5 - 3x^4 + x^2 - x - 5$

   The following plots produce slightly unexpected results. Plot the functions, and identify the unexpected behavior. Modify the **plot** parameters to produce a more correct plot. Also, have *Maple* plot the functions with identical scale for the vertical and horizontal axes.

   e. $2 + \sin x$ f. $\sin(x)^2 + \cos(x)^2$

3. Evaluate the limits of the following functions for $x = \pm\infty$ as well as any undefined points. Produce appropriate plots to demonstrate these limits.

   a. $\tanh x$ c. $\sin\dfrac{1}{x}$
   b. $\dfrac{x^2 + 1}{x^2 - 1}$ d. $\cos\dfrac{1}{x}$

4. Find all critical points, maxima and minima, and inflection points for the function $y = x^4 + x$.
5. Using the substitution $u = \pi - x$ and the IntegrationTools package check the identity

$$\int_0^\pi x f(\sin x)\, dx = \frac{\pi}{2} \int_0^\pi f(\sin x)\, dx$$

   Note: Recall that the IntegrationTools functions seem to work best with inert integrals.

6. A cone may be constructed by cutting a sector out of a circle, and then joining the two straight lines CA and CB (from the diagram to the right) which are created by the removal of arc. If the circle has radius R, then find a formula for the maximum volume that such a cone may have.
   Hint: Try finding the circumference of the circle at the top of the cone.
   Hint2: You will probably need to tell *Maple* some assumptions about your variables.
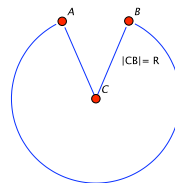
7. Evaluate the following integrals. In each case the answer is a combination (i.e., sums or products) of constants such as e, $\sqrt{2}$, $\sqrt{3}$, π, $\zeta(3)$, $\log 2$, and γ (Euler's gamma constant). At least one even involves $\log \pi$. If *Maple* cannot calculate the integral directly, evaluate it numerically and try to **identify** it.

   a. $\displaystyle\int_0^{\frac{\pi}{2}} \frac{x^2}{\sin^2 x}\, dx$

   b. $\displaystyle\int_0^{\frac{\pi}{2}} \frac{x^4}{\sin^4 x}\, dx$

   c. $\displaystyle\int_0^{\frac{\pi}{2}} \frac{\arcsin\left(\frac{\sqrt{2}}{2}\sin x\right)\sin x}{\sqrt{4-2\sin^2 x}}\, dx$

   d. $\displaystyle\int_0^{\infty} \frac{\log x}{\cosh^2 x}\, dx$

   The following two integrals arise from mathematical physics, but neither had a known closed form as of 2009. This may have changed.

   e. $\displaystyle\int_0^1 \frac{\log\left(\sqrt{3+y^2}+1\right) - \log\left(\sqrt{3+y^2}-1\right)}{1+y^2}\, dy$

   f. $\displaystyle\int_3^4 \frac{\operatorname{arcsec}(x)}{\sqrt{x^2-4x-3}}\, dx$

8. Solve the following linear differential equations, verify the solution, and plot it for some values of the constant.

   a. $xy' + y = x\cos x$ (for $x > 0$)   b. $y' + (\cos x)y = \cos x$

   How do the solution curves change as the constant changes?
   If you are feeling adventurous, try and plot the solutions using the **DEplot** function from the DEtools package.

9. The differential equation

$$x^2 y'' + xy' + (x^2 - \alpha^2)y = 0$$

   is known as the Bessel equation. The solutions to this equation give rise to the so-called *Bessel functions of the first and second kind*, $J_\alpha(x)$ and $Y_\alpha(x)$, respectively.

   a. Solve the Bessel equation for the special cases of $\alpha = \frac{1}{2}$ and $\alpha = \frac{3}{2}$. Verify the solutions.
   b. Solve the Bessel equation for the general case. Verify the solution.
   c. Plot the Bessel functions J_α and Y_α for some values of α of your choosing.

   The modified Bessel functions of the first and second kind—$I_\alpha(x)$ and $K_\alpha(x)$, respectively—are solutions of the modified Bessel equation,

$$x^2 y'' + xy' - (x^2 + \alpha^2)y = 0$$

   d. Solve the modified Bessel equation for the general case. Verify the solution.

e. Plot the modified Bessel functions I_α and K_α for some values of α of your choosing.

10. a. Plot Pac-Man[5] on a set of Cartesian axes. You need only produce the basic outline.
      Hint: Use multiple polar co-ordinate plots and the **display** function.
      b. Find a polar equation for the circle of radius 3 with center $(1, 2)$. Plot the circle using this equation.

11. a. Find the volume obtained by rotating the area between by the curve $z = x^2$ and the z-axis for $x \in [0, 1]$ around the z-axis. Notice anything interesting?
      b. Plot and calculate the volumes obtained by rotating the following areas around the z-axis. In all cases the curve is $z = \log x$ with $x \in [0, 1]$.
      i. The area underneath the curve (between the curve and the x-axis)
      ii. The area between the curve and the z-axis.
      Check your answers by calculating both the disks and the shells method.

12. Find and classify the critical points of the following functions of two variables. Recall that critical points occur where $f_x(a, b) = f_y(a, b) = 0$. A critical point may be a maximum, a minimum, or a saddle point.

a. $f(x, y) := 3x^2 y + y^3 - 3x^2 - 3y^2 + 2$. b. $f(x, y) := xy e^{-x^2 - y^2}$

13. Verify that the following functions are solutions to the partial differential equation $f_x + f_y = \sin(x) + \cos(y)$. A solution to a partial differential equation is a function whose partial derivatives satisfy the equation (compare to an ordinary differential equation).

a. $\sin(y) - \cos(x) + C \cdot (y - x)$ c. $\sin(y) - \cos(x) + \sqrt{y - x}$
b. $\sin(y) - \cos(x) + (y - x)^n$ d. $\sin(y) - \cos(x) + e^{C \cdot (y - x)^n}$

What do you think the general solution might be?

14. Let f be the function defined below.

$$f(x, y) := \begin{cases} \dfrac{xy(x^2 - y^2)}{x^2 + y^2} & \text{if } (x, y) \neq (0, 0) \\ 0 & \text{if } (x, y) = (0, 0) \end{cases}$$

a. Verify—visually or otherwise—that f is continuous at the origin.
b. Show that $f_{xy}(0, 0) \neq f_{yx}(0, 0)$ and so Clairaut's theorem does not apply to this function.

Note: You need to use the limit definition on all functions to establish the value of their partial derivatives at the point $(0, 0)$.

15. Plot the following, and calculate their area using iterated integrals.

a. The area underneath $z = x^5 y^3 e^{xy}$ for $0 \le x \le 1$ and $0 \le y \le 1$
b. The area between $z = e^{-x^2} \cos(x^2 + y^2)$ and $z = 2 - x^2 - y^2$ for $|x| \le 1$ and $|y| \le 1$

The following iterated integrals are volumes underneath surfaces ($z = f(x, y)$) for points (x, y) that do not lie inside a rectangle. Calculate the integrals. Can you work out the bounding curves for the points (x, y)?

---

[5] Pac-Man is a video game character from the early 1980s. A quick Internet search should be all that is needed in order to know what the plot must look like.

c. $\displaystyle\int_1^2 \int_0^y x^2 y^2 \ dx \ dy$

d. $\displaystyle\int_0^2 \int_0^{\frac{x^2}{2}} \frac{x}{\sqrt{1+x^2+y^2}} \ dy \ dx$

2.5 Further Explorations

1. Finding limits.

   a. Let $a_0 = 0, a_1 = \frac{1}{2}$ and define

   $$a_{n+1} := \frac{1 + a_n + a_{n-1}^3}{3}$$

   Determine the limit as $n \to \infty$, and find out what happens when $a_1 = a$ is allowed to vary.

   b. Let $a_1 = 1$ and define

   $$a_{n+1} := \frac{3 + 2a_n}{3 + a_n}$$

   Determine the limit and find out what happens when $a_1 = a$ is allowed to vary.

   The above two limits are easy enough to find and (depending on what you know) to prove.

   c. Let $a_1 \geq 1$ be given and determine the limit of the iteration

   $$a_{n+1} := a_n - \frac{a_n}{\sqrt{1 + a_n^2}} + \sin(\theta)$$

   for arbitrary θ.

2. A *(strict) mean* $M(a, b)$ is a continuous function of two positive numbers that calculates a number c lying between a and b (strictly between them as long as $a \neq b$). The arithmetic and geometric means are clearly such objects.
   A mean iteration takes two means, M and N say, and iterates by setting $a_0 = a$, $b_0 = b$ and

   $$a_{n+1} := M(a_n, b_n), \quad b_{n+1} := N(a_n, b_n)$$

   The limit of such a strict mean iteration always exists and is denoted by $M \otimes N(a, b)$. Identify the limits of the mean iteration defined by the means in the following.

   a. $M(a, b) := \dfrac{a + b}{2}$, $N(a, b) := \dfrac{2ab}{a + b}$

   b. $M(a, b) := \dfrac{a + \sqrt{ab}}{2}$, $N(a, b) := \dfrac{b + \sqrt{ab}}{2}$

3. Define

   $$\text{sinc}(x) := \frac{\sin x}{x}$$

   and explore the following integrals. Calculate them for all (natural) values of N from 1 to 8 at least (more if you wish), and measure the time each calculation takes to perform. Can you work out what is going on?

a. $\displaystyle\int_0^\infty \prod_{n=0}^N \mathrm{sinc}\left(\frac{x}{2n+1}\right)\, dx$ b. $\displaystyle\int_0^\infty \prod_{n=0}^N \mathrm{sinc}\left(\frac{x}{3n+2}\right)\, dx$

Chapter 3
Linear Algebra

3.1 Introduction and Review

In this section we introduce *Maple*'s linear algebra capabilities, and examine some of the basic linear algebra that should already be familiar (or, at least, have been seen before now). We presume that the reader is proficient, at least, in Gaussian and Gauss–Jordan elimination of matrices.

Maple provides two packages for working with linear algebra. The first is the linalg package, which has since been superseded by the LinearAlgebra package. The former package is still available, and may be needed for older *Maple* files. We deal with the newer LinearAlgebra package for the entirety of this book.

3.1.1 Vectors and Matrices in Maple

Before we can start to explore much linear algebra in *Maple*, we must first know how to create the basic building blocks required. As such, this section is essentially devoted solely to *Maple* syntax and semantics for the basic building blocks of linear algebra; matrices and vectors. This is unfortunate, but necessary, and we return to more predominant mathematical endeavors as quickly as possible.

As previous stated, linear algebra more or less boils down to vectors and matrices (although the distinction is not all that clear and students of second-year or later linear algebra should see that matrices may, themselves, be the vectors of a vector space). *Maple* can handle creation, and basic arithmetic (addition, scaling, and non-commutative multiplication) without the need to load any external packages.

There are a number of ways to declare a vector. Note, however, that a *Maple* list is not a vector, even though it might behave a little like one. The simplest way to define a vector is to use the $\langle \rangle$ shortcut notation.

```
> ⟨1, 2, 3⟩; ⟨a|b|c|d|e⟩
```

$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

$$\begin{bmatrix} a & b & c & d & e \end{bmatrix}$$

This method provides, at a glance, an easy way to see that something is a vector. The only trick is remembering that commas are used for the delimiter when one wants a column vector, and vertical bars are used when one wants a row vector.

Maple also provides a **Vector** command (note the capital "V") for creation of vectors. The simplest form of this command takes a list as input, and produces a column vector, or a row vector if the correct parameters are used

> $Vector([a, b, c, d]); Vector_{row}([1, 2, 3, 4])$

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 3 & 4 \end{bmatrix}$$

The **Vector** command may, instead, be used to specify the number of elements in the vector, as well as a function (or value) that each element is to take. If no second parameter is given, then the default value is 0.

> $Vector(5); Vector[row](4, a); Vector(3, i \rightarrow 3 \cdot i)$

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} a\,(1) & a\,(2) & a\,(3) & a\,(4) \end{bmatrix}$$

$$\begin{bmatrix} 2 \\ 4 \\ 6 \end{bmatrix}$$

For additional usage of the **Vector** command, consult *Maple*'s help files (**?Vector**).

Matrices may be declared in a similar fashion. We may think of a matrix as a vector where each row is also a vector (or, equivalently, a row vector where each column is also a vector). As such, our $\langle \rangle$ shortcut also allows us to declare matrices.

> $\langle \langle 1|2|3 \rangle, \langle 4|5|6 \rangle, \langle 7|8|9 \rangle \rangle; \langle \langle a, b, c \rangle | \langle d, e, f \rangle | \langle g, h, i \rangle \rangle$

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

$$\begin{bmatrix} a & d & g \\ b & e & h \\ c & f & i \end{bmatrix}$$

Just as with vectors, there is also a **Matrix** function for declaration of matrices. The basic form of this command requires a list of lists, where the innermost lists must all be the same length, and represent the row vectors.

> $Matrix([[1, 2, 3], [a, b, c], [i, ii, iii]])$

$$\begin{bmatrix} 1 & 2 & 3 \\ a & b & c \\ i & ii & iii \end{bmatrix}$$

There remains, just as with vectors, the capacity to directly specify the size of the matrix, and a function (or value) to populate it, with the default being 0. Note that when specifying the size of a matrix, both the number of rows, and the number of columns must be specified, in that order. If only one size is given then the matrix is presumed to be square, but one should be careful using this method, as the second parameter must unmistakably be a function or a variable. If unsure, it is always best to explicitly state both width and height.

> $Matrix(2, 3); Matrix(2, f); Matrix(4, (i, j) \rightarrow 10 \cdot i + j)$

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} f(1,1) & f(1,2) \\ f(2,1) & f(2,2) \end{bmatrix}$$

$$\begin{bmatrix} 11 & 12 & 13 & 14 \\ 21 & 22 & 23 & 24 \\ 31 & 32 & 33 & 34 \\ 41 & 42 & 43 & 44 \end{bmatrix}$$

Now that we know how to produce vectors and matrices in *Maple*, we should expect to be able to perform the usual arithmetic with them. Most of this is possible still without the need for an extra package, but we begin to run into some situations where it might be better to begin using some of the LinearAlgebra functions.

Matrices and vectors may both be added and subtracted with the usual *Maple* operators for those purposes. They may be scaled using the multiplication and division operators (note that division by a is the same as scaling the vector by a^{-1}). Note that vectors must both either be row or column vectors in order to be added. Similarly, matrices must be of the same size to be able to added.

> $\langle u_1, u_2 \rangle + \langle v_1, v_2 \rangle; \langle u_1 | u_2 \rangle + \langle v_1 | v_2 \rangle; a \cdot \langle u_1 | u_2 \rangle; \langle u_1, u_2 \rangle + \langle v_1 | v_2 \rangle$

$$\begin{bmatrix} u_1 + v_1 \\ u_2 + v_2 \end{bmatrix}$$

$$\begin{bmatrix} u_1 + v_1 & u_2 + v_2 \end{bmatrix}$$

$$\begin{bmatrix} au_1 & au_2 \end{bmatrix}$$

Error, (in rtable/Sum) invalid arguments

> $M_1 := \langle\langle 1|2|3\rangle, \langle 4|5|6\rangle, \langle 7|8|9\rangle\rangle; M_2 := \langle\langle a,b,c\rangle|\langle d,e,f\rangle|\langle g,h,i\rangle\rangle$

$$M_1 := \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

$$M_2 := \begin{bmatrix} a & d & g \\ b & e & h \\ c & 2 & i \end{bmatrix}$$

> $M_1 + M_2; 3 \cdot M_1$

$$\begin{bmatrix} a+1 & d+2 & 3+g \\ b+4 & e+5 & 6+h \\ c+7 & 10 & 9+i \end{bmatrix}$$

$$\begin{bmatrix} 3 & 6 & 9 \\ 12 & 15 & 18 \\ 21 & 24 & 27 \end{bmatrix}$$

Matrices may also be multiplied using the **.** (dot) operator. Be very careful here, this is a "full stop" (a period) and not the $\cdot$ that we have hitherto used for multiplication. We can also take powers of matrices using the usual superscript method.

> $M_1 \,.\, M_2; M_1 \,.\, M_1; M_1^2$

$$\begin{bmatrix} a+2b+3c & d+2e+3f & g+2h+3i \\ 4a+5b+6c & 4d+5e+6f & 4g+5h+6i \\ 7a+8b+9c & 7d+8e+9f & 7g+8h+9i \end{bmatrix}$$

$$\begin{bmatrix} 30 & 36 & 42 \\ 66 & 81 & 96 \\ 102 & 126 & 150 \end{bmatrix}$$

$$\begin{bmatrix} 30 & 36 & 42 \\ 66 & 81 & 96 \\ 102 & 126 & 150 \end{bmatrix}$$

The **.** operator is a *Maple* operator for performing noncommutative multiplication; see **?dot**. Be careful, though, as it can be difficult to tell the difference between the two multiplication symbols. The **.** (dot) operator can also, as the name may make us suspect, be used to calculate the dot product of two vectors.

> $\langle u_1, u_2\rangle \,.\, \langle v_1, v_2\rangle$

$$\overline{u_1}v_1 + \overline{u_2}v_2$$

The answer here is the more general dot product as defined for a complex vector space. This, of course, gives the same answer in the real case when we remember that the conjugate or a real number is itself (i.e., $\bar{x} = x$ if $x \in \mathbb{R}$). We should be careful to make sure that the vectors are both the same kind of vector (row or column), otherwise *Maple* may produce some unexpected results.

> $\langle u_1 | u_2 \rangle \cdot \langle v_1, v_2 \rangle$
> $\langle u_1, u_2 \rangle \cdot \langle v_1 | v_2 \rangle$

$$v_1 u_1 + v_2 u_2$$

$$\begin{bmatrix} v_1 u_1 & u_1 v_2 \\ u_2 v_1 & v_2 u_2 \end{bmatrix}$$

The first case looks like the regular dot product (although without the complex conjugation). The second case is a 2×2 matrix, and is precisely the matrix we would get if we considered the u vector to be a single column matrix, and the v vector to be a single row matrix.

The confusion above may be alleviated somewhat with some functions from the LinearAlgebra package. However, this comes at a cost of extra typing, and a less mathematical look to the input. The **DotProduct** and **Multiply** commands perform the dot product, and multiplication, respectively.

The advantage of **DotProduct** is that we don't need to make sure that the two vectors are of the same type and, in addition, we can't accidentally perform matrix multiplication.

> $with(LinearAlgebra):$

> $DotProduct(< u_1, u_2 >, < v_1, v_2 >)$
> $DotProduct(< u_1 | u_2 >, < v_1, v_2 >)$
> $DotProduct(< u_1, u_2 >, < v_1 | v_2 >)$
> $DotProduct(< u_1 | u_2 >, < v_1 | v_2 >)$

$$\overline{u_1} v_1 + \overline{u_2} v_2$$

$$\overline{v_1} u_1 + \overline{v_2} u_2$$

$$\overline{u_1} v_1 + \overline{u_2} v_2$$

$$\overline{v_1} u_1 + \overline{v_2} u_2$$

The **Multiply** command more or less does everything that the **.** operator as well as the usual multiplication ($\cdot$) operator do, with the exception of the vector dot product. Consulting the help files on this function explains the unusual behavior when we try to multiply two types of vectors together (which should, in turn, explain why the **.** operator behaves unusually in this case as well).

> $Multiply(M_1, M_2); Multiply(\langle u_1 | u_2 \rangle, \langle v_1, v_2 \rangle)$

$$\begin{bmatrix} a + 2b + 3c & d + 2e + 3f & g + 2h + 3i \\ 4a + 5b + 6c & 4d + 5e + 6f & 4g + 5h + 6i \\ 7a + 8b + 9c & 7d + 8e + 9f & 7g + 8h + 9i \end{bmatrix}$$

$$v_1 u_1 + v_2 u_2$$

Multiplying a row vector by a column vector (in that order) results in a scalar. The scalar in question happens to be the single element inside the 1×1 matrix we would have if we considered the vectors to be single row/column matrices, respectively. This is different from the dot product because we are only multiplying the individual elements of each vector together. If the vectors contained complex numbers we would not end up with the complex conjugation in this case that we do with the dot product.

Finally, before we move on to something more mathematical, we demonstrate two more shortcuts. The inverse of a matrix (if it exists), may be calculated by using exponentiation notation, as one might well expect. That is, if we raise a matrix to the

power -1 *Maple* will calculate the inverse matrix. Recall that for a square matrix A the inverse matrix is the unique matrix (usually written A^{-1}) with the property that $A^{-1}A = AA^{-1} = I$.

> $A := \langle\langle 1, 2\rangle | \langle 3, 4\rangle\rangle;$
> A^{-1}
> $A^{-1}.A, A.A^{-1}$

$$A := \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$$

$$\begin{bmatrix} -2 & \frac{3}{2} \\ 1 & -\frac{1}{2} \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

> $\langle\langle 1, 2, 3\rangle | \langle 4, 5, 6\rangle\rangle^{-1}$

$\qquad$ Error, (in rtable/Power) power -1 not defined for non-square
$\qquad\qquad$ Matrices; try LinearAlgebra[MatrixInverse] to obtain a
$\qquad\qquad\qquad\qquad$ pseudo-inverse

The *transpose* of a matrix (the matrix created when column vectors are changed to be row vectors, or—equivalently—vice versa, usually written A^T) may be obtained by raising a matrix (in *Maple*, that is) to the power %T.

> $A := \langle\langle a, b, c\rangle | \langle d, e, f\rangle\rangle; A^{\%T}; \%^{\%T}$

$$A := \begin{bmatrix} a & d \\ b & e \\ c & f \end{bmatrix}$$

$$\begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix}$$

$$\begin{bmatrix} a & d \\ b & e \\ c & f \end{bmatrix}$$

Naturally, there are additional functions in the LinearAlgebra package that will also perform both the matrix inverse and the transpose.

The other functions in the LinearAlgebra package may (and even should) be examined using *Maple*'s help files on the package (**?LinearAlgebra**). Further discussion on functions from this package (and *Maple* approaches to linear algebra in general) takes place as and when they are needed in the course of the mathematics that follows.

3.1.2 Simultaneous Linear Equations

One of the first parts of linear algebra a student sees is the method of solving systems of linear equations using matrices and Gauss–Jordan elimination. We re-examine this topic now.

Let's start with a simple example. We want to find values for x and y that satisfy both the equations $x + y = 2$ and $2x + y = 3$ simultaneously. This we should be able to do quickly in our heads, or on paper. If we were to use *Maple* we would probably use the **solve** command.

> $solve(\{x + y = 2, 2 \cdot x + y = 3\})$

$$\{x = 1, y = 1\}$$

However, we can also attack this problem using our tools from linear algebra. Let's start by constructing a vector where the elements inside the vector are the equations we are trying to simultaneously solve

$$\begin{bmatrix} x + y = 2 \\ 2x + y = 3 \end{bmatrix}$$

This can be thought of as

$$\begin{bmatrix} x + y \\ 2x + y \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

which in turn can be thought of as

$$\begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

which is an equation of the form $Ax = b$ where A is a matrix, and x and b are matrices.

The solution to this problem is to construct the *augmented matrix*

$$\begin{bmatrix} 1 & 1 & 2 \\ 2 & 1 & 3 \end{bmatrix}$$

and to perform *Gaussian elimination* in order that the matrix be in *row echelon form*, or *Gauss–Jordan elimination* in order that the matrix be in *reduced row echelon form*.

Reduced row echelon is the easier form to see the solutions directly, but is usually a little tricky (or, at least fiddly) to produce by hand. Fortunately, we do not have such problems when using a CAS, and the LinearAlgebra package includes a function that will calculate the reduced row echelon form of a matrix and is named, probably unsurprisingly, **ReducedRowEchelonForm**.

> $A := \langle\langle 1|1|2 \rangle, \langle 2|1|3 \rangle\rangle; ReducedRowEchelonForm(A)$

$$A := \begin{bmatrix} 1 & 1 & 2 \\ 2 & 1 & 3 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

We can read the answer that $x = 1$ and $y = 1$ directly from this matrix, remembering that it is an augmented matrix representing the vector equation

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

which is equivalent to our original equations (due to the fact that elementary row operations, as performed in Gauss–Jordan elimination do not change the solution of a system of equations).

In general, for a system of m simultaneous linear equations in n unknowns, we construct the $m \times n$ coefficient matrix, A say, and the vector of constants (the right-hand sides of the equations) b. The system may then be considered as the vector equation

$$Ax = b$$

where x is the vector of the unknown values. We then calculate the solution by reducing the augmented matrix $[A \,|\, b]$ into reduced row echelon form. (Note here that $[A \,|\, b]$ is the matrix A with b added in as a final column vector). This should not be anything new to any student who has studied first-year university mathematics.

In the case of systems of equations with two or three unknowns, there is a clear geometric interpretation of these systems of linear equations. The equation $ax + by = c$ describes a unique line in the Euclidean plane. This line can be thought of as the collection of all points (x, y) in the plane that satisfy the equation $ax + by = c$. When evaluating two such equations simultaneously, we are looking for the collection of all points (x, y) that satisfy both equations at the same time, or—geometrically—the collection of all points that lie on both lines (the intersection of the two lines).

With the plane, there are not very many different things that can happen. Either the equations all describe the same line, in which case we should see infinitely many solutions, or the lines will meet at a point somewhere, in which case there will be one solution, or the lines are parallel in which case there will be no solutions.

Let's have a look at the example from above, $x + y = 2$ and $2x + y = 3$. We already know that the only solution to this system of equations is $x = 1, y = 1$. When we plot these lines we should expect to see them crossing at the point $(1, 1)$. Note that in order to plot these lines, we needed to manipulate the equations into the equivalent forms of $y = 2 - x$ and $y = 3 - 2x$.

> $plot([2 - x, 3 - 2x], x = -1..2, color = [red, blue])$
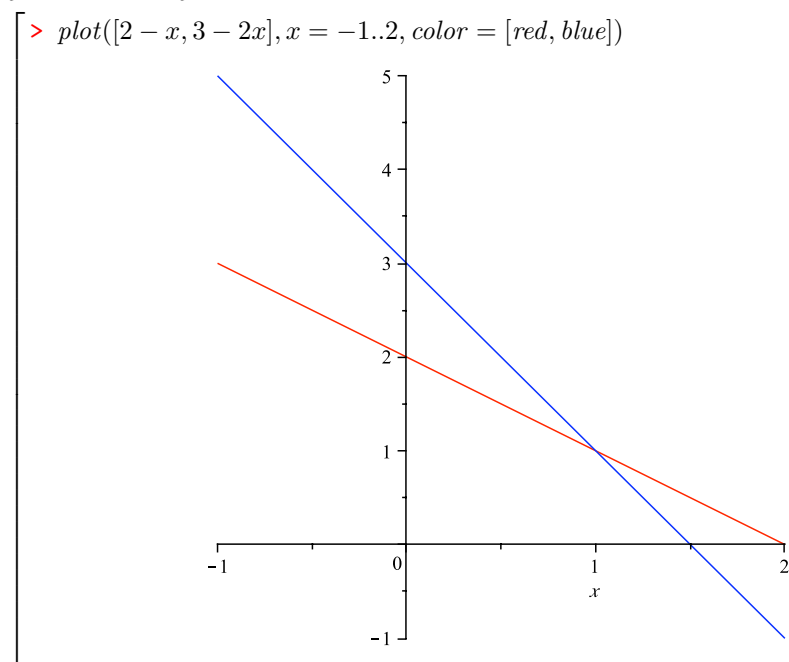
When more than two lines are introduced, the chances that there is any point at which they all intersect becomes smaller (it is quite likely that any two lines will intersect; the point here is that a solution to the simultaneous equations must be a point where all the lines intersect at the same point). For instance, if we extend the system above to

also include the equations $-3x + y = 4$ and $-x + y = -2$, plotting all four lines shows clearly that there is no common intersection point for the four lines

> $plot([2 - x, 3 - 2x, 4 + 3x, x - 2], color = [red, red, blue, black])$
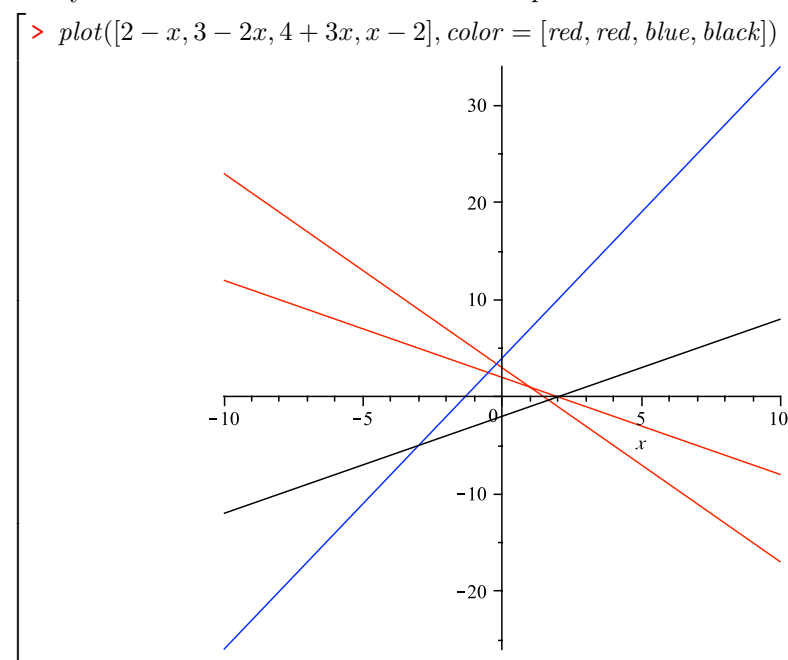
And the linear algebra produces the expected result

> $M, b := \langle\langle 1|1\rangle, \langle 2|1\rangle, \langle -3|1\rangle, \langle -1|1\rangle\rangle, \langle 2, 3, 4, -2\rangle$

$$M, b := \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ -3 & 1 \\ -1 & 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 3 \\ 4 \\ -2 \end{bmatrix}$$

> $M.\langle x, y\rangle = b;$
> $LinearSolve(M, b)$

$$\begin{bmatrix} x + y \\ 2x + y \\ -3x + y \\ -x + y \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \\ 4 \\ -2 \end{bmatrix}$$

Error, (in LinearAlgebra:-LA_Main:-LinearSolve) inconsistent system

It doesn't really come much clearer than that error message, so long as we know that an inconsistent system of linear equations has no solution. Even if we didn't know this, it should be quite clear form the context.

We may also, if we wish, read this directly off the augmented matrix in reduced row echelon form. Note the second row from the bottom which should be read $0x + 0y = 1$.

> $ReducedRowEchelonForm(\langle M|b\rangle)$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

Let's now have a look at this in three dimensions. Recall that in real 3-space, the equation $ax + by + cz = d$ describes a plane. The intersection of any three or more different planes will either be a line, a point, or nonexistent. In the case of only two distinct planes then the intersection can only be a line or nonexistent. We start with the following set of equations.

$$x + y + z = 3$$
$$-x - y + z = -2$$
$$2x + y + z = 0$$

Our aim here is to see how the planes intersect (if they do at all). To do this most effectively we need to know if they intersect, and if so then where. So we first solve the system of equations.

To solve these equations we now use the **LinearSolve** function from the LinearAlgebra package. This function has two basic forms. If A is a matrix, and b is a vector, then the command $LinearSolve(A, b)$ will solve the linear system $Ax = b$. However, if we use the form $LinearSolve(A)$ (with no b) then *Maple* will presume that A is an augmented matrix, and solve the linear system for that augmented matrix. We use the first form here.

> $M, b := \langle\langle 1|1|1\rangle, \langle -1|1|1\rangle, \langle 2|1|1\rangle\rangle, \langle 3, -2, 0\rangle$

$$M, b := \begin{bmatrix} 1 & 1 & 1 \\ -1 & -1 & 1 \\ 2 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 3 \\ -2 \\ 0 \end{bmatrix}$$

> $LinearSolve(M, b)$

$$\begin{bmatrix} -3 \\ \dfrac{11}{2} \\ \dfrac{1}{2} \end{bmatrix}$$

And so we now see that there is but a single point, $[-3, \frac{11}{2}, \frac{1}{2}]$ at which the three planes intersect. We can now plot the planes in such a way as to see this intersection clearly. In order to have the intersection point close to the middle of the plot (so as, we hope, to see the most of the interrelations of the planes) we plot over the x-range $(0, 6)$ and the y-range $(3, 9)$.
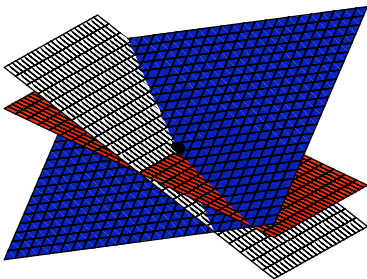
We have a small problem, however. If we ask *Maple* to plot these functions with a single plot command, we will have a list of three elements for our plot, which **plot3d** will interpret as a parametric equation. To avoid this problem we parameterize the three planes as

$$\left[x, y, \frac{1}{c}(-ax - by + d) \right]$$

> $pt := convert(LinearSolve(M, b), list)$

$$pt := \left[-3, \frac{11}{2}, \frac{1}{2} \right]$$

> $planes := plot3d([[x, y, 3 - x - y], [x, y, x + y - 2], [x, y, -2 * x - y]], x = -6..0,$
> $\quad y = 3..9, color = [red, blue, white]):$

> $intpoint := plot3d([pt], x = -6..0, y = 3..9, style = point, symbol =$
> $\quad solidsphere, symbolsize = 30, color = black):$

> $plots[display]([planes, intpoint])$

3.1.3 Elementary Row Operations

We now look more closely at the elementary row (and, equivalently, column) operations. There are three basic operations that may be performed:

- Swap two rows.
- Multiply a row by a constant.
- Add a multiple of one row to another row.

It is expected that none of this is new to the reader.

What may be new, however, is that it is possible to construct matrices that perform these row operations when they are multiplied (on the left) with another matrix. It is this idea that we explore. It is important to note at this point that there is no practical purpose to performing row operations with matrix multiplication and, indeed, directly performing the operations is quicker and easier. However, the mere fact that we can do this allows us to prove some facts about invertible matrices.

To construct these row-operation matrices, called *elementary matrices*, we simply perform a row operation manually on the identity matrix. The resulting matrix will then perform that same row operation when it is multiplied with another matrix. We demonstrate this for the case of 2×2 matrices.

> $ID := IdentityMatrix(2)$
> $G := \langle\langle a|b\rangle, \langle c|d\rangle\rangle$

$$ID := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$G := \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

> $S := RowOperation(ID, [1, 2])$

$$S := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Maple's LinearAlgebra package provides us with the **RowOperation** function (as we have used above) in order to perform row operations. There are (unsurprisingly) three ways of using this function. The method used above, $RowOperation(M, [a, b])$, will swap rows a and b in the matrix M. This is precisely what we saw happen to the identity matrix above.

> $S.G$

$$\begin{bmatrix} c & d \\ a & b \end{bmatrix}$$

> $G.S$

$$\begin{bmatrix} b & a \\ d & c \end{bmatrix}$$

When we multiply our matrix S on the left against an arbitrary matrix, we can see that it swaps the rows as claimed. This behavior can be verified on paper, and doing so should demonstrate just why the matrix does this. Furthermore, when we multiply on the right by our matrix, it performs the swap as a column operation. This may also be confirmed on paper.

Continuing on in this manner, we now create the matrices for multiplying a given row by a constant. The command form for this operation is $RowOperation(M, r, k)$ which multiplies row r in matrix M by constant k.

> $M_1 := RowOperation(ID, 1, k);$
> $M_2 := RowOperation(ID, 2, k);$

$$M_1 := \begin{bmatrix} k & 0 \\ 0 & 1 \end{bmatrix}$$

$$M_2 := \begin{bmatrix} 1 & 0 \\ 0 & k \end{bmatrix}$$

> $M_1.G, M_2.G$

$$\begin{bmatrix} ka & kb \\ c & d \end{bmatrix}, \begin{bmatrix} a & b \\ kc & kd \end{bmatrix}$$

> $G.M_1, G.M_2$

$$\begin{bmatrix} ka & b \\ kc & d \end{bmatrix}, \begin{bmatrix} a & kb \\ c & kd \end{bmatrix}$$

Sure enough these two matrices behave just as we should have expected. Multiplying on the left against an arbitrary matrix performs the row operation, and multiplying on the right performs the equivalent column operation.

Finally, we create the matrices for adding a multiple of one row to another row. The command form for this operation is $RowOperation(M, [a, b], k)$ which adds row b multiplied by k to row a. We use a possibly overly complicated naming scheme for these matrices, but it leaves little to no doubt as to what the matrix does

> $A_{r1+k \cdot r2} := RowOperation(ID, [1, 2], k);$
> $A_{r2+k \cdot r1} := RowOperation(ID, [2, 1], k);$

$$A_{r1+k \cdot r2} := \begin{bmatrix} 1 & k \\ 0 & 1 \end{bmatrix}$$

$$A_{r2+k \cdot r1} := \begin{bmatrix} 1 & 0 \\ k & 1 \end{bmatrix}$$

> $A_{r1+k \cdot r2}.G, A_{r2+k \cdot r1}.G$

$$\begin{bmatrix} a + ck & b + dk \\ c & d \end{bmatrix}, \begin{bmatrix} a & b \\ ka + c & kb + d \end{bmatrix}$$

> $G.A_{r1+k \cdot r2}, G.A_{r2+k \cdot r1}$

$$\begin{bmatrix} a & ka + b \\ c & ck + d \end{bmatrix}, \begin{bmatrix} a + kb & b \\ c + dk & d \end{bmatrix}$$

It should be mentioned at this point that these elementary matrices need not be multiplied with only square matrices. The elementary matrices themselves will always be square (because they are obtained by performing a row operation on an identity matrix, which is always square). However, just as row operations may be performed on any size matrix, so will the elementary matrices perform their appropriate row operation on any size matrix. The only stipulation is the usual one for matrix multiplication, which is that when multiplying matrices A and B together, A must have the same number of columns as B has rows.

What this means for our elementary matrices is that an $n \times n$ elementary matrix will perform its row operation when multiplying on the left any matrix with exactly n rows. Similarly, when multiplying on the right said matrix will perform column operations on any matrix with n columns

> $B := \langle \langle 11|12|13|14|15 \rangle, \langle 21|22|23|24|25 \rangle \rangle$

$$B := \begin{bmatrix} 11 & 12 & 13 & 14 & 15 \\ 21 & 22 & 23 & 24 & 25 \end{bmatrix}$$

> $S.B \, ; M_1.B \, ; A_{r2+k \cdot r1}.B$

$$\begin{bmatrix} 21 & 22 & 23 & 24 & 25 \\ 11 & 12 & 13 & 14 & 15 \end{bmatrix}$$

$$\begin{bmatrix} 11\,k & 12\,k & 13\,k & 14\,k & 15\,k \\ 21 & 22 & 23 & 24 & 25 \end{bmatrix}$$

$$\begin{bmatrix} 11 & 12 & 13 & 14 & 15 \\ 11\,k + 21 & 12\,k + 22 & 13\,k + 23 & 14\,k + 24 & 15\,k + 25 \end{bmatrix}$$

> $B.S$

Error, (in LinearAlgebra:-MatrixMatrixMultiply) first matrix column
          dimension (5) <> second matrix row dimension (2)

Notice that the final thing we tried was to multiply on the right instead of on the left, and *Maple* issued an error to tell us that the matrices could not be multiplied in that order, just as we already knew they couldn't.

> $B := B^{\%T}$

$$B := \begin{bmatrix} 11 & 21 \\ 12 & 22 \\ 13 & 23 \\ 14 & 24 \\ 15 & 25 \end{bmatrix}$$

> $B.S\,,\ B.M_1\,,\ B.A_{r2+kr1}$

$$\begin{bmatrix} 21 & 11 \\ 22 & 12 \\ 23 & 13 \\ 24 & 14 \\ 25 & 15 \end{bmatrix},\ \begin{bmatrix} 11 & k\,21 \\ 12 & k\,22 \\ 13 & k\,23 \\ 14 & k\,24 \\ 15 & k\,25 \end{bmatrix},\ \begin{bmatrix} 11+21\,k & 21 \\ 12+22\,k & 22 \\ 13+23\,k & 23 \\ 14+24\,k & 24 \\ 15+25\,k & 25 \end{bmatrix}$$

> $S.B$

Error, (in LinearAlgebra:-MatrixMatrixMultiply) first matrix column
          dimension (2) <> second matrix row dimension (5)

We now put all this together with a real (pun intended) matrix. In order to allow us to properly use the M and A matrices correctly, we create functions that allow us to choose the particular k value we want.

> $R := \langle\langle 1,2\rangle | \langle 3,4\rangle\rangle$

$$R := \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$$

> $M_1 := unapply(M_1):$
  $M_2 := unapply(M_2):$
  $M_1(x), M_2(x)$

$$\begin{bmatrix} x & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & x \end{bmatrix}$$

> $A_{r1+k\cdot r2} := unapply(A_{r1+k\cdot r2}):$
  $A_{r2+k\cdot r1} := unapply(A_{r2+k\cdot r1}):$
  $A_{r1+k\cdot r2}(x), A_{r2+k\cdot r1}(x)$

$$\begin{bmatrix} 1 & x \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ x & 1 \end{bmatrix}$$

Now we perform row operation—via multiplication with elementary matrices—to reduce our matrix R into reduced row echelon form.

> R

$$\begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$$

> $A_{r2+k\cdot r1}(-2).\%$

$$\begin{bmatrix} 1 & 3 \\ 0 & -2 \end{bmatrix}$$

> $M_2\left(-\dfrac{1}{2}\right).\%$

$$\begin{bmatrix} 1 & 3 \\ 0 & 1 \end{bmatrix}$$

> $A_{r1+k\cdot r2}(-3).\%$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

We should note here that we have inadvertently found an inverse matrix for R. That inverse is the product of the three elementary matrices we used in the correct order. Because we were multiplying on the left the entire way, the correct order for multiplication is to start with the final elementary matrix, and work our way backwards.

> $A_{r1+k\cdot r2}(-3).M_2\left(-\dfrac{1}{2}\right).A_{r2+k\cdot r1}(-2).G$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

> $A_{r1+k\cdot r2}(-3).M_2\left(-\dfrac{1}{2}\right).A_{r2+k\cdot r1}(-2)$

$$\begin{bmatrix} -2 & \dfrac{3}{2} \\ 1 & -\dfrac{1}{2} \end{bmatrix}$$

> $R.\%$, $\%.R$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Sure enough, if we ask *Maple* directly for the inverse of our matrix R then it gives us the matrix we just calculated.

> R^{-1}

$$\begin{bmatrix} -2 & \dfrac{3}{2} \\ 1 & -\dfrac{1}{2} \end{bmatrix}$$

This is no coincidence. It is necessarily the case that an invertible matrix will always become the identity matrix when changed into reduced row echelon form. Furthermore any invertible matrix can always be created by multiplying some sequence of elementary matrices together. We look at this idea now.

First observe that every elementary matrix is invertible. This is a direct consequence of the fact that the row operations themselves are invertible processes. We can, very simply, undo any row operation we perform as follows:

Operation	**Inverse**
Swap rows a and b	Swap rows a and b
Multiply a row a by a constant k	Multiply a row a by the constant $\frac{1}{k}$
Add k times row a to row b	Add $-k$ times row a to row b

We look at three-dimensional elementary matrices this time. First we should unassign the variables we used for the various flavors of elementary matrices earlier, as we will use them again.

> $A, M, S :=' A', ' M', ' S'$
> $ID := IdentityMatrix(3)$

$$A, M, S := A, M, S$$

$$ID := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

We simplify the creation of the matrices with the use of some **for** loops. Observe that for swapping rows, swapping rows a and b is exactly the same as swapping rows b and a, so we avoid creating duplicate row swapping matrices, and stipulate, when swapping rows, that we will always specify the lowest numbered row first.

> **for** i **from** 1 **to** 3 **do**
>   **for** j **from** $i + 1$ **to** 3 **do**
> $S_{i,j} := RowOperation(ID, [i, j])$
>   **od**
> **od** :

> **for** i **from** 1 **to** 3 **do**
> $RowOperation(ID, i, k);$
> $M_i := unapply(\%, k);$
> **od** :

> **for** i **from** 1 **to** 3 **do**
>   **for** j **in** $\{1, 2, 3\}$ **minus** $\{i\}$ **do**
> $RowOperation(ID, [i, j], k);$
> $A_{r\|i+k\cdot r\|j} := unapply(\%, k);$
>   **od**
> **od** :

Here, $\|$ is the *Maple* operator for concatenation. the string $r\|i + k \cdot r\|j$ is converted by *Maple* to $ri + k \cdot rj$, with i and j being correctly substituted for their correct values. Conversely, merely inputting $ri + k \cdot rj$ directly would not perform the relevant substitution because *Maple* would consider ri and rj to be complete variable names in their own right.

Let us have a look at our fresh new elementary matrices now.

> $S_{1,2}, S_{1,3}, S_{2,3};$
> $M_1(k), M_2(k), M_3(k);$
> $A_{r1+k\cdot r2}(k), A_{r1+k\cdot r3}(k), A_{r2+k\cdot r1}(k), A_{r2+k\cdot r3}, A_{r3+k\cdot r1}, A_{r3+k\cdot r2}(k);$

$$
\begin{bmatrix} 1\ 0\ 0 \\ 0\ 0\ 1 \\ 0\ 1\ 0 \end{bmatrix}, \begin{bmatrix} 0\ 0\ 1 \\ 0\ 1\ 0 \\ 1\ 0\ 0 \end{bmatrix}, \begin{bmatrix} 0\ 1\ 0 \\ 1\ 0\ 0 \\ 0\ 0\ 1 \end{bmatrix}
$$

$$
\begin{bmatrix} k\ 0\ 0 \\ 0\ 1\ 0 \\ 0\ 0\ 1 \end{bmatrix}, \begin{bmatrix} 1\ 0\ 0 \\ 0\ k\ 0 \\ 0\ 0\ 1 \end{bmatrix}, \begin{bmatrix} 1\ 0\ 0 \\ 0\ 1\ 0 \\ 0\ 0\ k \end{bmatrix}
$$

$$
\begin{bmatrix} 1\ k\ 0 \\ 0\ 1\ 0 \\ 0\ 0\ 1 \end{bmatrix}, \begin{bmatrix} 1\ 0\ k \\ 0\ 1\ 0 \\ 0\ 0\ 1 \end{bmatrix}, \begin{bmatrix} 1\ 0\ 0 \\ k\ 1\ 0 \\ 0\ 0\ 1 \end{bmatrix}, \begin{bmatrix} 1\ 0\ 0 \\ 0\ 1\ k \\ 0\ 0\ 1 \end{bmatrix}, \begin{bmatrix} 1\ 0\ 0 \\ 0\ 1\ 0 \\ k\ 0\ 1 \end{bmatrix}, \begin{bmatrix} 1\ 0\ 0 \\ 0\ 1\ 0 \\ 0\ k\ 1 \end{bmatrix}
$$

We now perform row reduction again on a real 3×3 matrix, just as we did above for our 2×2 matrix R. In fact, we even reuse the name. However, we proceed a little more quickly, and take multiple steps at a time, where it is obvious to do so.

> $R := \langle\langle 1, -2, 3\rangle | \langle 3, 5, -6\rangle | \langle -7, 8, 9\rangle\rangle$

$$
\begin{bmatrix} 1 & 4 & -7 \\ -2 & 5 & 8 \\ 3 & -6 & 9 \end{bmatrix}
$$

> $A_{r2+k\cdot r1}(2).A_{r3+k\cdot r1}(-3).\%$

$$
\begin{bmatrix} 1 & 4 & -7 \\ 0 & 13 & -6 \\ 0 & -18 & 30 \end{bmatrix}
$$

> $M_2\left(\dfrac{1}{13}\right).\%$

$$
\begin{bmatrix} 1 & 4 & -7 \\ 0 & 1 & -\dfrac{6}{13} \\ 0 & -18 & 30 \end{bmatrix}
$$

> $A_{r1+k\cdot r2}(-4).A_{r3+k\cdot r2}(18).\%$

$$
\begin{bmatrix} 1 & 0 & -\dfrac{67}{13} \\ 0 & 1 & -\dfrac{6}{13} \\ 0 & 0 & \dfrac{282}{13} \end{bmatrix}
$$

> $M_3\left(\dfrac{13}{282}\right).\%$

$$\begin{bmatrix} 1 & 0 & -\dfrac{67}{13} \\[2mm] 0 & 1 & -\dfrac{6}{13} \\[2mm] 0 & 0 & 1 \end{bmatrix}$$

> $A_{r1+k\cdot r3}\left(\dfrac{67}{13}\right).A_{r2+k\cdot r3}\left(\dfrac{6}{13}\right).\%$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

And so it is that we may replay this sequence of moves to find the inverse of R (observing that the order of the A matrices at each step doesn't matter).

> $A_{r1+k\cdot r3}\left(\dfrac{67}{13}\right).A_{r2+k\cdot r3}\left(\dfrac{6}{13}\right).M_3\left(\dfrac{13}{282}\right).A_{r1+k\cdot r2}(-4).A_{r3+k\cdot r2}(18).M_2\left(\dfrac{1}{13}\right)$
> $.A_{r2+k\cdot r1}(2).A_{r3+k\cdot r1}(-3)$

$$\begin{bmatrix} \dfrac{31}{94} & \dfrac{1}{47} & \dfrac{67}{282} \\[2mm] \dfrac{7}{47} & \dfrac{5}{47} & \dfrac{1}{47} \\[2mm] -\dfrac{1}{94} & \dfrac{3}{47} & \dfrac{13}{282} \end{bmatrix}$$

> $R.\%\,,\ \%.R$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

For simplicity's sake, let's call this sequence of elementary matrices

$$E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8$$

(because it's easier to write and type). We know that

$$E_1 E_2 E_3 E_4 E_5 E_6 E_7 E_8 R = I \ \text{ or equivalently } \ E_1 E_2 E_3 E_4 E_5 E_6 E_7 E_8 = R^{-1}$$

and we also know that all of the E_i matrices as well as the R matrix are invertible. Recalling that $(AB)^{-1} = B^{-1}A^{-1}$, then it must be the case that

$$R = E_8^{-1} E_7^{-1} E_6^{-1} E_5^{-1} E_4^{-1} E_3^{-1} E_2^{-1} E_1^{-1}$$

Checking this in *Maple* for our above example:

$$> A_{r3+k\cdot r1}(3) \cdot A_{r2+k\cdot r1}(-2) \cdot M_2(13) \cdot A_{r3+k\cdot r2}(-18) \cdot A_{r1+k\cdot r2}(4) \cdot M_3\left(\frac{282}{13}\right)$$

$$\cdot A_{r2+k\cdot r3}\left(\frac{-6}{13}\right) \cdot A_{r1+k\cdot r3}\left(\frac{-67}{13}\right)$$

$$\begin{bmatrix} 1 & 4 & -7 \\ -2 & 5 & 8 \\ 3 & -6 & 9 \end{bmatrix}$$

In fact, this will always happen for an invertible matrix. We have the following theorem.

Theorem 2. *Let A be a square matrix. The following are equivalent.*

1. *A is invertible.*
2. *The linear system $Ax = 0$ has only the trivial solution (i.e., $x = 0$).*
3. *The reduced row echelon form of A is the identity matrix.*
4. *A may be expressed as a product of elementary matrices.*

Proof. Observe that if A is invertible, then

$$Ax = 0 \implies A^{-1}(Ax) = A^{-1}0 \implies (A^{-1}A)x = 0 \implies x = 0$$

showing that the x vector must be the zero vector.

If $x = 0$ is the only solution to the linear sy stem $Ax = 0$ then the augmented matrix $[A|0]$ must have reduced row echelon form $[I|0]$ (corresponding to the solution $x = 0$). And so the reduced row echelon form of A is I.

If the I is the reduced row echelon form of A then it must be the case that there is some set of elementary matrices such that

$$E_1 E_2 \cdots E_i A = I \implies A = E_i^{-1} \cdots E_2^{-1} E_1^{-1}$$

showing that A is expressible as the product of elementary matrices.

And finally if A is expressible as the product of elementary matrices, then A must be invertible, inasmuch as the elementary matrices are all invertible, so

$$A^{-1} = E_1 E_2 \cdots E_i$$

It is precisely this theorem that allows us to find the inverse of an invertible matrix A by using row operations on the augmented matrix $[A|I]$ where I is the identity matrix. Remember, however, that in practice there's no point to performing the row operations with the elementary matrices, as directly performing row operations on a matrix is far faster (both for a computer and for a human). However, the existence of the elementary matrices allows us to prove the above theorem, whose utility is quite significant indeed.

$$> \langle R|ID \rangle$$

$$\begin{bmatrix} 1 & 4 & -7 & 1 & 0 & 0 \\ -2 & 5 & 8 & 0 & 1 & 0 \\ 3 & -6 & 9 & 0 & 0 & 1 \end{bmatrix}$$

> *ReducedRowEchelonForm*(%)

$$\begin{bmatrix} 1 \ 0 \ 0 & \dfrac{31}{94} & \dfrac{1}{47} & \dfrac{67}{282} \\[2ex] 0 \ 1 \ 0 & \dfrac{7}{47} & \dfrac{5}{47} & \dfrac{1}{47} \\[2ex] 0 \ 0 \ 1 & -\dfrac{1}{94} & \dfrac{3}{47} & \dfrac{13}{282} \end{bmatrix}$$

3.2 Vector Spaces

3.2.1 Vector Spaces

Until now we have been using vectors without really saying what they are. However. before we can do much more with linear algebra, we need to define exactly what a vector, or rather a *vector space* is.

If V is a set of objects, and $\mathbb{F}$ is a field, then we call V a vector space—or, more accurately, a vector space over $\mathbb{F}$—if the elements of V and $\mathbb{F}$ interact as described below. Note that fields for our purposes are nearly always real numbers $\mathbb{R}$ and sometimes complex numbers $\mathbb{C}$, although other fields do exist and may be used in place of $\mathbb{F}$. The elements of V are, not surprisingly, the vectors, and the elements of $\mathbb{F}$ are referred to as *scalars*.

For V to be a vector space (over the field $\mathbb{F}$, remember), then there must be an addition operator on V (i.e., a way of adding any two vectors in such a way as to always result in a vector) and a scalar multiplication operation (i.e., a way of multiplying any scalar and any vector in such a way as to always result in a vector). In addition, the following axioms must hold for $u, v, w \in V$ and $a, b \in \mathbb{F}$

1. $u + v = v + u$ (Commutativity of vector addition).
2. $(u + v) + w = u + (v + w)$ (Associativity of vector addition).
3. There is an element $\mathbf{0} \in V$ such that $\mathbf{0} + u = u$ (Additive identity, or *zero vector*).
4. There is an element $-v \in V$ such that $v + (-v) = 0$ (Additive inverse).

Anybody who has studied abstract algebra should recognize that the above four axioms show that V must be an Abelian group. Continuing on:

5. $a(u + v) = au + av$ (Distributive property).
6. $(a + b)u = au + bu$ (Distributive property).
7. $(ab)u = a(bu)$ (Associativity of scalar multiplication).
8. $1v = v$ where $1 \in \mathbb{F}$ is the multiplicitave identity.

Scalar multiplication and vector addition interact in a very similar fashion to the way the more familiar addition and multiplication of real or complex numbers interact.

We should already be familiar with some vector spaces, even if we have never thought of them in such terms. The points of the Cartesian plane can be thought of as vectors, and indeed we have used *Maple* to manipulate or produce them earlier. To be specific, the set V in this case is the set $\mathbb{R}^2$ or $\mathbb{R} \times \mathbb{R}$ which consists of ordered pairs of real numbers ($\mathbb{R}^2 := \{(x, y) \,|\, x \in \mathbb{R} \text{ and } y \in \mathbb{R}\}$) and the field is the real numbers $\mathbb{R}$. It is straightforward to check that these two sets satisfy all the axioms above.

In a similar fashion we can see that the points in three-dimensional real space, or $\mathbb{R}^3$ also form a vector space over the real numbers. In fact, any n-dimensional real space

$\mathbb{R}^n := \{(x_1, \ldots, x_n) \mid x_i \in \mathbb{R}\}$ forms a vector space over the real numbers, with vector addition and scalar multiplication working as it does for $\mathbb{R}^2$ and $\mathbb{R}^3$.

We can extend this idea a little further by replacing the real numbers with complex numbers and end up with $\mathbb{C}^n := \{(z_1, \ldots, z_n) \mid z_i \in \mathbb{C}\}$ with vector addition and scalar multiplication working in precisely the same way as with the real case (excepting, of course, that we perform complex multiplication). Let's have a quick look at that in *Maple*.

> $u, v := \langle 1 + 2 \cdot I, 2 + 3 \cdot I, 3 + 4 \cdot I \rangle, \langle 5 + 6 \cdot I, 6 + 7 \cdot I, 7 + 8 \cdot I \rangle$

$$u, v := \begin{bmatrix} 1 + 2\,\mathrm{I} \\ 2 + 3\,\mathrm{I} \\ 3 + 4\,\mathrm{I} \end{bmatrix}, \begin{bmatrix} 5 + 6\,\mathrm{I} \\ 6 + 7\,\mathrm{I} \\ 7 + 8\,\mathrm{I} \end{bmatrix}$$

> $u + v; (9 + 10 \cdot I) \cdot u$

$$\begin{bmatrix} 6 + 8\,\mathrm{I} \\ 8 + 10\,\mathrm{I} \\ 10 + 12\,\mathrm{I} \end{bmatrix}$$

$$\begin{bmatrix} -11 + 28\,\mathrm{I} \\ -12 + 47\,\mathrm{I} \\ -13 + 66\,\mathrm{I} \end{bmatrix}$$

We may even construct vector spaces of polynomials of fixed degree, or matrices of a fixed size. It is left as an exercise for the reader to verify the vector space axioms for both of these cases We denote the space of $m \times n$ matrices as $M_{m,n}(\mathbb{F})$, and the space of degree n polynomaials as $P_n(\mathbb{F})$ (both over the field $\mathbb{F}$).

We now demonstrate a useful correlation between the vector space $P_n(\mathbb{F})$ and the vector space $\mathbb{F}^{n+1}$. Let $p_0 + p_1 x + \cdots + p_n x^n \in P_n(\mathbb{F})$ (so $p_i \in \mathbb{F}$). We take the vector $(p_0, p_1, \ldots, p_n) \in \mathbb{F}^{n+1}$, and consider it as being equivalent.We can freely change between these forms; given the vector we can easily construct the equivalent polynomial, and given the polynomial we can easily construct the vector. For this example we use $n = 4$, however this idea works for any n.

> $P := p_0 + p_1 \cdot x + p_2 \cdot x^2 + p_3 \cdot x^3 + p_4 \cdot x^4$
> $Q := p_0 + q_1 \cdot x + q_2 \cdot x^2 + q_3 \cdot x^3 + q_4 \cdot x^4$

$$P := p_0 + p_1\,x + p_2\,x^2 + p_3\,x^3 + p_4\,x^4$$
$$Q := q_0 + q_1\,x + q_2\,x^2 + q_3\,x^3 + q_4\,x^4$$

> $u :=' u' :$
> $u_p, u_q := Vector(5, i \rightarrow p_{i-1}), Vector(5, i \rightarrow q_{i-1})$

$$u_p, u_q := \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix}, \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}$$

We demonstrate that vector addition and polynomial addition produce equivalent results, as do vector and polynomial scaling.

> $expand(k \cdot P)$

$$k\,p_0 + k\,p_1\,x + k\,p_2\,x^2 + k\,p_3\,x^3 + k\,p_4\,x^4$$

> $k \cdot u_p$

$$\begin{bmatrix} k\,p_0 \\ k\,p_1 \\ k\,p_2 \\ k\,p_3 \\ k\,p_4 \end{bmatrix}$$

> $P + Q$

$$p_0 + p_1 \cdot x + p + 2 \cdot x^2 + p_3 \cdot x^3 + p_4 \cdot x^4 + q_0 + q_1 \cdot x + q + 2 \cdot x^2 + q_3 \cdot x^3 + q_4 \cdot x^4$$

> $collect(\%, x)$

$$(p_4 + q_4)\,x^4 + (q_3 + p_3)\,x^3 + (q_2 + p_2)\,x^2 + (q_1 + p_1)\,x + p_0 + q_0$$

> $u_p + u_q$

$$\begin{bmatrix} p_0 + q_0 \\ q_1 + p_1 \\ q_2 + p_2 \\ q_3 + p_3 \\ p_4 + q_4 \end{bmatrix}$$

As previously mentioned, this works for any value of n (which should be clear, given what we know about vectors and polynomials). We have, in effect, shown that polynomials can be thought of as $n+1$ vectors, and that they can be interchangeably used as vector spaces.

Incidentally, converting between forms in *Maple* is quite straightforward. If we multiply the column vector $(1, x, x^2, x^3, x^4)$ on the left of our previous vectors we will retrieve our polynomial. To go the other way, we use the **CoefficientVector** from the PolynomialTools package, which returns a vector of the coefficients of a polynomial (as the name suggests).

> $\langle 1|x|x^2|x^3|x^4 \rangle \cdot (u_p + u_q)$

$$p_0 + q_0 + (q_1 + p_1)\,x + (q_2 + p_2)\,x^2 + (q_3 + p_3)\,x^3 + (p_4 + q_4)\,x^4$$

> $PolynomialTools[CoefficientVector](P, x), PolynomialTools[CoefficientVector](Q, x)$

$$\begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix}, \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}$$

This is all a special case of a result we look at a little later; that any finite-dimensional vector space with dimension n can be thought of as $\mathbb{F}^n$. This is of key use to us with *Maple*.

3.2.2 Linear Combinations

Let V be a vector space, and let $v_1, v_2, \ldots, v_k \in V$. A *linear combination* of these vectors is a new vector of the form $v = a_1 v_1 + a_2 v_2 + \cdots + a_k v_k$ where the a_i are scalars (i.e., $a_i \in \mathbb{F}$). That $v \in V$ follows directly from the axioms of a vector space.

We may wish to consider the set of all possible linear combinations of a set of vectors, which we call the *span* of those vectors, for example, span$(\{v_1, v_2, \ldots, v_k\})$ which is equal to $\{a_2 v_2 + \cdots + a_k v_k : a_i \in \mathbb{F}\}$.

Let $S \subset V$ be a nonempty and finite set of vectors. A natural question arises as to how we may tell whether a vector v is a linear combination of the vectors in S. Equivalently this same question may be phrased as whether a vector is in the span of S. For example, is the vector $(7, 8, 9)$ a linear combination of the vectors $(1, 2, 3)$ and $(4, 5, 6)$?

For simplicity, we consider only the case of the vector spaces $\mathbb{F}^n$. Think of what a linear combination would mean in this case. If v is a linear combination of vectors in S then

$$a_1 \begin{bmatrix} v_{1_1} \\ \vdots \\ v_{1_n} \end{bmatrix} + a_2 \begin{bmatrix} v_{2_1} \\ \vdots \\ v_{2_n} \end{bmatrix} + \cdots + a_k \begin{bmatrix} v_{k_1} \\ \vdots \\ v_{k_n} \end{bmatrix} = v = \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix}$$

which is equivalent to

$$\begin{bmatrix} v_{1_1} & v_{2_1} & \cdots & v_{k_1} \\ \vdots & \vdots & \ddots & \vdots \\ v_{1_n} & v_{2_n} & \cdots & v_{k_n} \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix}$$

which, in turn, is a system of linear equations, just as we dealt with in Section 3.1.2. If the linear system has at least one solution, then the vector v is a linear combination of the vectors in S. Indeed if there are many solutions, then there are many ways to represent v as such a linear combination, but this does not detract in any way from the fact that it is a linear combination. If, however, there is no solution to the system, then v cannot be written as a linear combination of the vectors in S.

This will be true for any set of vectors $v_1, \ldots, v_k$, as long as our vector space is $\mathbb{F}^n$. We know how to solve such systems, so we can now easily answer our example question from above.

> $LinearSolve(\langle\langle 1, 2, 3\rangle | \langle 4, 5, 6\rangle\rangle, \langle 7, 8, 9\rangle)$

$$\begin{bmatrix} -1 \\ 2 \end{bmatrix}$$

It would seem that there is a linear combination here, with coefficients -1 and 2. We can, of course, check this easily.

> $-1 \cdot \langle 1, 2, 3 \rangle + 2 \cdot \langle 4, 5, 6 \rangle$

$$\begin{bmatrix} 7 \\ 8 \\ 9 \end{bmatrix}$$

This gives us a new way to look at simultaneous linear equations as well. Recall from Section 3.1.2 our first (and simplest) example $x + y = 2$ and $2x + y = 3$ which had solution $x = 1, y = 1$. We constructed from this the linear system

$$\begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

which we may now, equivalently, consider as finding $(2, 3)$ as a linear combination of $(1, 2)$ and $(1, 1)$. The solution $1, 1$ clearly works for this interpretation.

> $\langle\langle 1, 2 \rangle | \langle 1, 1 \rangle\rangle \cdot \langle 1, 1 \rangle$

$$\begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

> $1 \cdot \langle 1, 2 \rangle + 1 \cdot \langle 1, 1 \rangle$

$$\begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

We may use this same linear system technique with polynomials, as well, because we established a correspondence between polynomials and n-tuples (meaning vectors in $\mathbb{F}^n$). Let us then ascertain whether $-4x^4 + x^3 + 5x^2 + 3x - 13$ is a linear combination of $x^4 + 3x^3 + 7x^2 - 6$, $2x^4 + 4x^2 - 5x + 2$ and $5x^3 - 3x^2 + 12x - 5$. To do this, we need to consider them as the vectors $(-13, 3, 5, 1, -4)$, $(-6, 0, 7, 3, 1)$, $(2, -5, 4, 0, 2)$, and $(-5, 12, -3, 5, 0)$.

> $p_1 := x^4 + 3 \cdot x^3 + 7 \cdot x^2 - 6;$
> $p_2 := 2 \cdot x^4 + 4 \cdot x^2 - 5 \cdot x + 2;$
> $p_3 := 5 \cdot x^3 - 3 \cdot x^2 + 12 \cdot x - 5;$
> $q := -4 \cdot x^4 + x^3 + 5 \cdot x^2 + 3 \cdot x - 13$

$$p_1 := x^4 + 3x^3 + 7x^2 - 6$$
$$p_2 := 2x^4 + 4x^2 - 5x + 2$$
$$p_3 := 5x^3 - 3x^2 + 12x - 5$$
$$q := -4x^4 + x^3 + 5x^2 + 3x - 13$$

> $u_1, u_2, u_3 := seq(PolynomialTools[CoefficientVector](p_i, x), i = 1..3);$

$$u_1, u_2, u_3 := \begin{bmatrix} -6 \\ 0 \\ 7 \\ 3 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ -5 \\ 4 \\ 0 \\ 2 \end{bmatrix}, \begin{bmatrix} -5 \\ 12 \\ -3 \\ 5 \end{bmatrix}$$

```
> v := PolynomialTools[CoefficientVector](q, x)
```

$$v := \begin{bmatrix} -13 \\ 3 \\ 5 \\ 1 \\ -4 \end{bmatrix}$$

Observe that the vector u_3 is incorrect, as *Maple* saw only a degree-3 polynomial, and not the degree-4 polynomial we intended. This is easily fixed by appending a 0 to the bottom of u_3 using the angle bracket notation.

```
> u_3 := ⟨u_3, 0⟩
```

$$u_3 := \begin{bmatrix} -5 \\ 12 \\ -3 \\ 5 \\ 0 \end{bmatrix}$$

Now we may solve the problem

```
> LinearSolve(⟨u_1|u_2|u_3⟩, v)
```

$$\begin{bmatrix} 2 \\ -3 \\ -1 \end{bmatrix}$$

```
> 2 · p_1 − 3 · p_2 − p_3 = q
```
$$-4\,x^4 + x^3 + 5\,x^2 + 3\,x - 13 = -4\,x^4 + x^3 + 5\,x^2 + 3\,x - 13$$

and there we have it.

3.2.3 Linear Independence

Along with the question of whether a vector may be expressed as a linear combination of other vectors comes a related but different question. If we have a nonempty and finite set $S \subset V$, can we express one of them (any one) as a linear combination of the others? If we can, then we say that the set S is *linearly dependent*. Conversely, if we cannot express any of the vectors in S as a linear combination of the others, then we say the set is *linearly independent*.

This is different from the question of whether an arbitrary vector $v \in V$ is in the span of S. The difference here is that previously v could have been any vector at all, whereas now we are asking whether vectors in the subset S are linear combinations of other vectors in the same subset which is a more specific question. Another way to think about this idea is whether we can remove vectors from S and still have the same span.

Another, equivalent, formulation of linear (in)dependence is to say that a finite set $\{v_1, \ldots, v_n\}$ of vectors is linearly independent if and only if the only solution to $a_1 v_1 + a_2 v_2 + \cdots + a_n v_n = 0$ is the trivial solution $a_1 = a_2 = \cdots = a_n = 0$. To see that this is

an equivalent notion, suppose that $v_1 \neq 0$ can be expressed as a linear combination of the other vectors,

$$v_1 = a_2 v_2 + \cdots + a_n v_n \implies 0 = a_2 v_2 + \cdots + a_n v_n - v_1$$

showing that the zero vector can be obtained as a nontrivial linear combination of the vectors. If v_2 or any other v_i can be expressed as a linear combination of the other vectors, then we may similarly construct the zero-vector as a non-trivial linear combination. Conversely, now suppose that

$$a_1 v_1 + a_2 v_2 + \cdots + a_n v_n = 0 \ \text{ where } \ a_1 \neq 0$$

Then

$$v_1 = \left(-\frac{a_2}{a_1} \right) v_2 + \cdots + \left(-\frac{a_n}{a_1} \right) v_n$$

showing that v_1 can be written as a linear combination of the other vectors. If v_2 or any other v_i is nonzero, then we can similarly show that it can be written as a linear combination of the other vectors.

We have now shown that the two notions of linear independence (and, thus, linear dependence) are equivalent.

This, in turn, allows us to link these linear independence notions to systems of linear equations and matrix invertibility. If we wish to see if a set, S, of vectors is linearly independent, we need to see if the zero vector can be expressed as a linear combination of the vectors in S. This, as we have seen, can be accomplished by solving a linear system.

For example, we have a look at the vectors $(-1, 2, 3, 4)$, $(1, -2, 3, 4)$, $(1, 2, -3, 4)$, and $(1, 2, 3, -4)$, and see if they are a linearly independent set. We do this by solving the linear system

$$\begin{bmatrix} -1 & 1 & 1 & 1 \\ 2 & -2 & 2 & 2 \\ 3 & 3 & -3 & 3 \\ 4 & 4 & 4 & -4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

> $v := \langle -1, 2, 3, 4 \rangle, \langle 1, -2, 3, 4 \rangle, \langle 1, 2, -3, 4 \rangle, \langle 1, 2, 3, -4 \rangle$

$$v := \begin{bmatrix} -1 \\ 2 \\ 3 \\ 4 \end{bmatrix}, \begin{bmatrix} 1 \\ -2 \\ 3 \\ 4 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \\ -3 \\ 4 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \\ 3 \\ -4 \end{bmatrix}$$

> $A := \langle v_1 | v_2 | v_3 | v_4 \rangle$

$$A := \begin{bmatrix} -1 & 1 & 1 & 1 \\ 2 & -2 & 2 & 2 \\ 3 & 3 & -3 & 3 \\ 4 & 4 & 4 & -4 \end{bmatrix}$$

> $LinearSolve(A, \langle 0, 0, 0, 0 \rangle)$

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

The only solution is the trivial solution, therefore the vectors are linearly independent. If we add a new vector into the mix, $(1, 2, 3, 4)$, we can ask the same question.

> $v := v, \langle 1, 2, 3, 4 \rangle$

$$v := \begin{bmatrix} -1 \\ 2 \\ 3 \\ 4 \end{bmatrix}, \begin{bmatrix} 1 \\ -2 \\ 3 \\ 4 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \\ -3 \\ 4 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \\ 3 \\ -4 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$$

> $A := \langle v_1 | v_2 | v_3 | v_4 | v_5 \rangle$

$$A := \begin{bmatrix} -1 & 1 & 1 & 1 & 1 \\ 2 & -2 & 2 & 2 & 2 \\ 3 & 3 & -3 & 3 & 3 \\ 4 & 4 & 4 & -4 & 4 \end{bmatrix}$$

> $LinearSolve(A, \langle 0, 0, 0, 0 \rangle)$

$$\begin{bmatrix} \_t10_4 \\ \_t10_4 \\ \_t10_4 \\ \_t10_4 \\ -2 \_t10_4 \end{bmatrix}$$

Now we have a linearly independent set. In fact, we can read straight from the solution to the linear system that $2v_5 = v_1 + v_2 + v_3 + v_4$.

> $t \cdot v_1 + t \cdot v_2 + t \cdot v_3 + t \cdot v_4$

$$\begin{bmatrix} 2t \\ 4t \\ 6t \\ 8t \end{bmatrix}$$

Looking at the question of linear dependence as a linear system problem, in the case of $\mathbb{F}^n$, if we have more than n vectors, then we have a linear system with an infinite number of solutions. This linear system would correspond to a set of simultaneous equations with more unknowns as equations. As such, any set of more than n vectors from the vector space $\mathbb{F}^n$ must be linearly dependent.

Furthermore, if we have exactly n vectors then we have a square matrix, and the solution to $Ax = 0$ being the trivial solution is equivalent to the matrix being invertible, by the theorem in Section 3.1.3. In fact, we may now add a new equivalence to this list; that the column vectors are linearly independent.

We now look at this same phenomenon with polynomials and their correspondence to n-tuples which we have been using. We start by using the same polynomials from Section 3.2.2, as we already know the answer to the linear system. Recall that $2p_1 - 3p_2 - p_3 = q$ or, in other words $2p_1 + 3p_2 - p_3 - q = 0$ and so the set $\{p_1, p_2, p_3, q\}$ is linearly dependent.

> $p_1 := x^4 + 3 \cdot x^3 + 7 \cdot x^2 - 6;$
> $p_2 := 2 \cdot x^4 + 4 \cdot x^2 - 5 \cdot x + 2;$
> $p_3 := 5 \cdot x^3 - 3 \cdot x^2 + 12 \cdot x - 5;$
> $q := -4 \cdot x^4 + x^3 + 5 \cdot x^2 + 3 \cdot x - 13$

$$p_1 := x^4 + 3x^3 + 7x^2 - 6$$
$$p_2 := 2x^4 + 4x^2 - 5x + 2$$
$$p_3 := 5x^3 - 3x^2 + 12x - 5$$
$$q := -4x^4 + x^3 + 5x^2 + 3x - 13$$

> $2 \cdot p_1 - 3 \cdot p_2 - p_3 - q$

$$0$$

Let's try some larger polynomials. We use $90x^5 + 80x^4 + 19x^3 + 88x^2 - 82x - 70$, $41x^5 + 91x^4 + 29x^3 + 70x^2 - 32x - 1$ and $52x^5 - 13x^4 + 82x^3 + 72x^2 + 42x + 18$.

> $p := 90x^5 + 80x^4 + 19x^3 + 88x^2 - 82x - 70$, $41x^5 + 91x^4 + 29x^3 + 70x^2 - 32x - 1$, $52x^5 - 13x^4 + 82x^3 + 72x^2 + 42x + 18 :$

> $p_1; p_2; p_3$

$$90x^5 + 80x^4 + 19x^3 + 88x^2 - 82x - 70$$
$$41x^5 + 91x^4 + 29x^3 + 70x^2 - 32x - 1$$
$$52x^5 - 13x^4 + 82x^3 + 72x^2 + 42x + 18$$

> $u := seq(PolynomialTools[CoefficientVector](p_i, x), i = 1..3)$

$$u := \begin{bmatrix} -70 \\ -82 \\ 88 \\ 19 \\ 80 \\ 90 \end{bmatrix}, \begin{bmatrix} -1 \\ -32 \\ 70 \\ 29 \\ 91 \\ 41 \end{bmatrix}, \begin{bmatrix} 18 \\ 42 \\ 72 \\ 82 \\ -13 \\ 52 \end{bmatrix}$$

> $LinearSolve(\langle u_1 | u_2 | u_3 \rangle, \langle 0, 0, 0, 0, 0, 0 \rangle)$

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

We can see that our polynomials are linearly independent. We now add two more polynomials to this list, $-19x^5 - 68x^4 - 89x^3 + 66x + 77$ and $-80x^5 - 19x^4 - 62x^3 + 81x^2 + 22x + 50$.

> $p := p, \ -19\,x^5 - 68\,x^4 - 89\,x^3 + 66\,x + 77, \ -80\,x^5 - 19\,x^4 - 62\,x^3 + 81\,x^2$
  $+\, 22\,x + 50 :$

> $u := u, seq(PolynomialTools[CoefficientVector](p_i, x), i = 4..5)$

$$u := \begin{bmatrix} -70 \\ -82 \\ 88 \\ 19 \\ 80 \\ 90 \end{bmatrix}, \begin{bmatrix} -1 \\ -32 \\ 70 \\ 29 \\ 91 \\ 41 \end{bmatrix}, \begin{bmatrix} 18 \\ 42 \\ 72 \\ 82 \\ -13 \\ 52 \end{bmatrix}, \begin{bmatrix} 77 \\ 66 \\ 0 \\ -89 \\ -68 \\ -19 \end{bmatrix}, \begin{bmatrix} 50 \\ 22 \\ 81 \\ -62 \\ -19 \\ -80 \end{bmatrix}$$

> $LinearSolve(\langle u_1|u_2|u_3|u_4|u_5\rangle, \langle 0,0,0,0,0,0\rangle)$

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Our collection of vectors is still linearly independent. We add one more polynomial
$162\,x^5 - 73\,x^4 + 45\,x^3 + 9\,x^2 + 36\,x - 24$

> $p := p, \ 162\,x^5 - 73\,x^4 + 45\,x^3 + 9\,x^2 + 36\,x - 24 :$

> $u := u, PolynomialTools[CoefficientVector](p_6, x)$

$$\begin{bmatrix} -70 \\ -82 \\ 88 \\ 19 \\ 80 \\ 90 \end{bmatrix}, \begin{bmatrix} -1 \\ -32 \\ 70 \\ 29 \\ 91 \\ 41 \end{bmatrix}, \begin{bmatrix} 18 \\ 42 \\ 72 \\ 82 \\ -13 \\ 52 \end{bmatrix}, \begin{bmatrix} 77 \\ 66 \\ 0 \\ -89 \\ -68 \\ -19 \end{bmatrix}, \begin{bmatrix} 50 \\ 22 \\ 81 \\ -62 \\ -19 \\ -80 \end{bmatrix}, \begin{bmatrix} -24 \\ 36 \\ 9 \\ 45 \\ -73 \\ 162 \end{bmatrix}$$

> $A := \langle u_1|u_2|u_3|u_4|u_5|u_6\rangle$

$$A := \begin{bmatrix} -70 & -1 & 18 & 77 & 50 & -24 \\ -82 & -32 & 42 & 66 & 22 & 36 \\ 88 & 70 & 72 & 0 & 81 & 9 \\ 19 & 29 & 82 & -89 & -62 & 45 \\ 80 & 91 & -13 & -68 & -19 & -73 \\ 90 & 41 & 52 & -19 & -80 & 162 \end{bmatrix}$$

> $Determinant(A)$

$$0$$

We have a zero determinant, which tells us that the matrix is not invertible, which in turn tells us that the column vectors are linearly dependent. This, in turn, tells us that we have a linearly dependent set of polynomials. We go ahead and solve the linear system.

> $LinearSolve(A, \langle 0, 0, 0, 0, 0, 0 \rangle)$

$$\begin{bmatrix} -\_t12_5 \\ \_t12_5 \\ -\_t12_5 \\ -\_t12_5 \\ \_t12_5 \\ \_t12_5 \end{bmatrix}$$

We may read directly from this solution that $-t \cdot p_1 + t \cdot p_2 - t \cdot p_3 - t \cdot p_4 + t \cdot p_5 = -t \cdot p_6$.

> $-t \cdot p_1 + t \cdot p_2 - t \cdot p_3 - t \cdot p_4 + t \cdot p_5$

$-t \left(90\,x^5 + 80\,x^4 + 19\,x^3 + 88\,x^2 - 82\,x - 70\right) + t \left(41\,x^5 + 91\,x^4 + 29\,x^3 + 70\,x^2 - 32\,x - 1\right) - t \left(52\,x^5 - 13\,x^4 + 82\,x^3 + 72\,x^2 + 42\,x + 18\right) - t \left(-19\,x^5 - 68\,x^4 - 89\,x^3 + 66\,x + 77\right) + t \left(-80\,x^5 - 19\,x^4 - 62\,x^3 + 81\,x^2 + 22\,x + 50\right)$

> $simplify(\%)$

$$24\,t - 162\,tx^5 + 73\,tx^4 - 45\,tx^3 - 9\,tx^2 - 36\,tx$$

> $sort(\%)$

$$-162\,tx^5 + 73\,tx^4 - 45\,tx^3 - 9\,tx^2 - 36\,tx + 24\,t$$

3.2.4 Basis and Dimension

Having discussed linear combinations and linear independence, we now come to the notion of the *basis* for a vector space. If we have a subset $S \subset V$ of vectors, then it is possible that the span of S (the collection of all linear combinations of the vectors) may indeed be the entire vector space V. If, in addition, S is linearly independent, then we say that S is a *basis* of V.

Because a basis, B say, must be linearly independent, then no vector in B can be made from a linear combination of the other vectors, and so if we remove a basis vector, the span of the resultant set will no longer be all of V. Furthermore, if we add any vector, $v \in V$ to B, then the new set will no longer be linearly independent (because v will be able to be written as a linear combination of the other basis vectors). As such we can think of a basis as the smallest set of vectors that span a given vector space.

We should be familiar with some standard bases. For example, $\mathbb{F}^3$ has basis $(1, 0, 0)$, $(0, 1, 0)$, and $(0, 0, 1)$. In fact $\mathbb{F}^n$ has a standard basis of n vectors

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \ldots, \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

An interesting result, which we do not prove, is that any basis for a given vector space will always have the same number of elements. This gives rise to the notion of *dimension* of a vector space. The dimension of a vector space V is the number of elements in any basis for the space.

If we look to polynomial spaces again, we may see that the space of all degree-n polynomials has a basis consisting of the $n + 1$ elements $\{1, x, x^2, \ldots, x^n\}$ or, alternatively, $\{x^k : 0 \le k \le n \text{ and } k \in \mathbb{Z}\}$. In fact, what we have been doing with our correspondence between polynomials and n-tuples has been to establish a correspondence between their basis vectors. More correctly, we have established a correspondence between degree-n polynomials and $(n + 1)$-tuples.

In fact, this notion extends to any finite-dimensional vector space, and allows us to treat any n-dimensional vector space as being $\mathbb{F}^n$. This allows us to use *Maple* vectors to perform calculations with any finite-dimensional vector space, just as we have previously been doing with polynomial spaces. This result is nontrivial, and should be proved in any good linear algebra text. We do not prove it here.

The standard bases are not the only basis for any given space; in fact there are many. In fact, if V is an n-dimensional vector space, then any linearly independent subset of V with exactly n elements will be a basis for V. We explore this a little now.

We choose $(1, 2)$ and $(-1, 3)$ as a linearly independent set of vectors in $\mathbb{R}^2$. That these vectors are linearly independent should be clear, but we shall verify this with *Maple* regardless.

> $v = \langle 1, 2 \rangle, \langle -1, 3 \rangle$

$$v := \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \begin{bmatrix} -1 \\ 3 \end{bmatrix}$$

> $Determinant(\langle v_1 | v_2 \rangle)$

$$5$$

Our claim is that any vector (a, b) in $\mathbb{R}^2$ may be expressed as a linear combination of our basis vectors. If we ask *Maple* this directly, we do not get an entirely satisfactory answer.

> $c_1 \cdot v_1 + c_2 \cdot v_2$

$$\begin{bmatrix} c_1 - c_2 \\ 2\,c_1 + 3\,c_2 \end{bmatrix}$$

However, we are asking a question about when (or, in this case, if) a vector is a linear combination of other vectors. We know that this problem can be solved with a linear system. Attacking the problem in this way gives a better answer.

> $LinearSolve(\langle v_1 | v_2 \rangle)$

$$\begin{bmatrix} \dfrac{3}{5}\,a + \dfrac{1}{5}\,b \\ -\dfrac{2}{5}\,a + \dfrac{1}{5}\,b \end{bmatrix}$$

> $\left(\dfrac{3a}{5} + \dfrac{b}{5} \right) \cdot v_1 + \left(-\dfrac{2a}{5} + \dfrac{b}{5} \right) \cdot v_2$

$$\begin{bmatrix} a \\ b \end{bmatrix}$$

In fact, we may use this technique to verify the claim that any n linearly independent vectors of an n-dimensional vector space will form a basis for the space. We do this by treating the problem as the equivalent question of whether an arbitrary vector may be constructed as a linear combination of the basis vectors, which in turn may be thought of as solving the vector equation $Ax = b$. In this case, A is the square matrix constructed with the columns being the column vectors of the (alleged) basis vectors. If the basis vectors are linearly independent, then the matrix is invertible, and so the solution $Ax = b$ has a unique solution for every b.

Let's look at this with some polynomials now.

> $p := x^3,\ x^3 + 1,\ x^3 + x + 1,\ x^3 + x^2 + x + 1$
$$p := x^3,\ x^3 + 1,\ x^3 + x + 1,\ x^3 + x^2 + x + 1$$

> **for** i **from** 1 **to** 4 **do** p_i **od**
$$x^3$$
$$x^3 + 1$$
$$x^3 + x + 1$$
$$x^3 + x^2 + x + 1$$

We have defined 4 linearly independent degree-3 polynomials. We convert these into equivalent 4-tuples, and verify the linear independence.

> $u := seq(PolynomialTools[CoefficientVector](p_i, x), i = 1..4)$
$$u := \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

> $Determinant(\langle u_1 | u_2 | u_3 | u_4 \rangle)$
$$-1$$

Now all that remains to be seen is whether a general polynomial $dx^3 + cx^2 + bx + a$ can be constructed as a linear combination of the polynomials u_i. We answer this question in $\mathbb{F}^4$ using our equivalencies, and verify the answer in $P_3(\mathbb{F})$.

> $LinearSolve(\langle u_1 | u_2 | u_3 | u_4 \rangle, \langle a, b, c, d \rangle)$
$$\begin{bmatrix} -a + d \\ -b + a \\ -c + b \\ c \end{bmatrix}$$

> $(-a + d) \cdot p_1 + (-b + a) \cdot p_2 + (-c + b) \cdot p_3 + c \cdot p_4$
$$(-a + d)x^3 + (-b + a)(x^3 + 1) + (-c + b)(x^3 + x + 1) + c(x^3 + x^2 + x + 1)$$

> $simplify(\%)$
$$x^3 d + a + bx + cx^2$$

> $sort(\%)$
$$dx^3 + cx^2 + bx + a$$

And so, not only can we see that the polynomials form a basis for $P_3(\mathbb{F})$, but we even have a formula for the coefficients the basis vectors will have for any given polynomial.

3.3 Linear Transformations

3.3.1 Introduction to Linear Transformations

We now look at functions between vector spaces. Such functions are sometimes called *transformations*. We say a function (or transformation) is *linear* if it preserves linear combinations. That is if we have vector spaces U, V, and a function $f : V \to V$, we say that f is linear if

$$f(c_1 v_1 + c_1 v_1) = c_1 f(v_1) + c_2 f(v_2) \text{ for each } c_1, c_2 \in \mathbb{F} \text{ and } v_1, v_2 \in V$$

or, in other words if we put a linear combination of vectors into the function, what we get out is the same linear combination, but of the images of our original vectors.

Let's look at two simple examples of functions from $\mathbb{R}^2 \to \mathbb{R}^2$.

$$f : \mathbb{R}^2 \to \mathbb{R}^2 \qquad\qquad g : \mathbb{R}^2 \to \mathbb{R}^2$$
$$\begin{bmatrix} a \\ b \end{bmatrix} \mapsto \begin{bmatrix} 2a + b \\ b - a \end{bmatrix} \qquad\qquad \begin{bmatrix} a \\ b \end{bmatrix} \mapsto \begin{bmatrix} a^2 \\ b \end{bmatrix}$$

> $f := v \to \langle 2 \cdot v_1 + v_2, v_2 - v_1 \rangle$
$$f := v \to \langle 2\, v_1 + v_2, v_2 - v_1 \rangle$$

> $g := v \to \langle v_1^2, v_2 \rangle$
$$g := v \to \langle v_1^2, v_2 \rangle$$

Having defined our functions, we may now ascertain whether they are linear.

> $f\left(c_1 \cdot \langle u_1, u_2 \rangle + c_2 \cdot \langle v_1, v_2 \rangle\right) = c_1 \cdot f(\langle u_1, u_2 \rangle) + c_2 \cdot f(\langle v_1, v_2 \rangle)$
$$\begin{bmatrix} 2\,c_1 u_1 + 2\,c_2 v_1 + c_1 u_2 + c_2 v_2 \\ c_1 u_2 + c_2 v_2 - c_1 u_1 - c_2 v_1 \end{bmatrix} = \begin{bmatrix} c_1\,(2\,u_1 + u_2) + c_2\,(2\,v_1 + v_2) \\ c_1\,(u_2 - u_1) + c_2\,(v_2 - v_1) \end{bmatrix}$$

> $simplify(\%)$
$$\begin{bmatrix} 2\,c_1 u_1 + 2\,c_2 v_1 + c_1 u_2 + c_2 v_2 \\ c_1 u_2 + c_2 v_2 - c_1 u_1 - c_2 v_1 \end{bmatrix} = \begin{bmatrix} 2\,c_1 u_1 + 2\,c_2 v_1 + c_1 u_2 + c_2 v_2 \\ c_1 u_2 + c_2 v_2 - c_1 u_1 - c_2 v_1 \end{bmatrix}$$

We see that f is linear.

> $g\left(c_1 \cdot \langle u_1, u_2 \rangle + c_2 \cdot \langle v_1, v_2 \rangle\right) = c_1 \cdot g(\langle u_1, u_2 \rangle) + c_2 \cdot g(\langle v_1, v_2 \rangle)$
$$\begin{bmatrix} (c_1 u_1 + c_2 v_1)^2 \\ c_1 u_2 + c_2 v_2 \end{bmatrix} = \begin{bmatrix} c_1 u_1^2 + c_2 v_1^2 \\ c_1 u_2 + c_2 v_2 \end{bmatrix}$$

We see that because, in general, $c_1 u_1^2 + c_2 v_1^2 \neq (c_1 u_1 + c_2 v_1)^2$ it is the case that g is not linear.

3.3.2 Linear Transformations as Matrices

We know that every vector in a given vector space may be written as a linear combination of basis vectors. We also know that a linear transformation preserves linear combination. It follows then that once we know how a linear transformation modifies the basis vectors of a vector space, we know how it will modify any vector in the space.

We now extend our equivalence between vector spaces and n-tuples a little. If $v_1, \ldots, v_n$ is a basis for an n-dimensional vector space, then we may write an arbitrary vector $u = c_1 v_1 + \cdots + c_n v_n$ simply as the n-tuple $(c_1, \ldots, c_n)$. This is precisely what we have been doing previously with polynomials and matrices. However, we may use this idea to write the same vector in $\mathbb{F}^n$ in many different ways.

Let us recall an example from Section 3.2.4, where we showed that the vectors $(1, 2)$ and $(-1, 3)$ formed a basis for $\mathbb{R}^2$. Of course, these vectors are already written in terms of the standard basis $(1, 0)$ and $(0, 1)$ in that $(1, 2) = 1 \cdot (1, 0) + 2 \cdot (0, 1)$ and $(-1, 3) = -1 \cdot (1, 0) + 3 \cdot (0, 1)$. We already see our principle in action. However, things can become confusing, as we are not altogether accustomed to thinking of (a, b) as a shorthand notation for these linear combinations, and worse still we are about to be comparing different bases. So we now give these bases names. Let S (for "standard") be the basis $\{(1, 0), (0, 1)\}$ and let B (for, simply, "basis") be our alternate basis $\{(1, 2), (-1, 3)\}$.

At this stage, it is quite important that we be aware of just which basis we are referring to at any one time. We do this by denoting what basis a vector is in respect to by using a subscript. That is, for some basis $A = \{a_1, \ldots, a_n\}$, the vector $(u_1, \ldots, u_n)_A$ denotes the vector that is equal to $u_1 a_1 + \cdots + u_n a_m$. If we do not specify a basis in this way, and it is not abundantly clear from the context which we mean, then we mean the standard basis.

Returning to our example, let us take the vector $v = (5, 15)$. To be more clear, we mean $(5, 15)_S$. We know from our example in Section 3.2.4 that we can write this vector as a linear combination of our basis B with the coefficients $c_1 = \frac{3}{5} \cdot 5 + \frac{1}{5} \cdot 15 = 6$ and $c_2 = -\frac{2}{5} \cdot 5 + \frac{1}{5} \cdot 15 = 1$ so that $(5, 15) = 6 \cdot (1, 2) + 1 \cdot (-1, 3)$. All of this is with respect to the usual basis, remember, and so should feel familiar. However, because of this, we could equally well write the vector with respect to the alternate basis B and say that $v = (6, 1)_B$. There is no question that we refer to the same actual vector within our vector space. The reader who is familiar with representation of natural numbers in different bases should see a similarity with this idea.

So now, when we see a vector written as an n-tuple we should think of this as the list of coefficients of some basis, even if that basis is simply the standard basis. In particular the vector $(c, 0, \ldots, 0)$ represents the vector which is the first basis vector scaled by a constant c, $(0, c, 0, \ldots, 0)$ as the second basis vector scaled by a constant c, and so on.

Observe, now, the effect matrix multiplication has on these vectors. We think of a matrix as being a collection of column vectors.

```
>  M := ⟨⟨u₁, u₂⟩|⟨v₁, v₂⟩⟩
```
$$M := \begin{bmatrix} u_1 & v_1 \\ u_2 & v_2 \end{bmatrix}$$

Observe that due to the nature of matrix multiplication, multiplying M on the right by the vector $(c, 0)$ will yield the column vector (cu_1, cu_2). Similarly multiplying M on the right by the vector $(0, d)$ will yield the column vector (dv_1, dv_2).

> $M.\langle c, 0 \rangle$

$$\begin{bmatrix} u_1\, c \\ u_2\, c \end{bmatrix}$$

> $M.\langle 0, d \rangle$

$$\begin{bmatrix} v_1\, d \\ v_2\, d \end{bmatrix}$$

It should be clear that this will hold for any $n \times m$ matrix and n-tuple. Furthermore multiplying M on the right by the vector (c, d) will yield the vector $(cu_1 + dv_1, cu_2 + dv_2) = c(u_1, u_2) + d(v_1, v_2)$.

> $M.\langle c, d \rangle = c \cdot \langle u_1, u_2 \rangle + d \cdot \langle v_1, v_2 \rangle$

$$\begin{bmatrix} u_1 c + v_1 d \\ u_2 c + v_2 d \end{bmatrix} = \begin{bmatrix} u_1 c + v_1 d \\ u_2 c + v_2 d \end{bmatrix}$$

Again, because of the way matrix multiplication is performed, this idea extends to any $n \times m$ matrix and n-tuple.

The point to all this is to demonstrate that matrix multiplication coincides with a linear combination of the column vectors of the matrix. It can be fairly easily checked that matrix multiplication itself is a linear transformation. What is less immediately obvious (but is strongly suggested by the above observations) is that every linear transformation between finite-dimensional vector spaces can be represented by matrix multiplication, for a suitable matrix.

We put this information together now. We know that a vector is just a shorthand way of writing a linear combination of basis vectors. We also know that linear transformations preserve linear combinations. Finally we know multiplying a matrix on the left by a vector creates a linear combination of the row vectors of that matrix. So, if we calculate the image of the basis vectors under the transformation, and then construct the matrix whose column vectors are those image vectors (in the correct order), then multiplying that matrix on the right by any vector (written with respect to the basis) will be exactly the same as applying the linear transformation to the vector directly.

For example, using our linear function f from Section 3.3.1,

> $M := \langle f(\langle 1, 0 \rangle) | f(\langle 0, 1 \rangle) \rangle$

$$\begin{bmatrix} 2 & 1 \\ -1 & 1 \end{bmatrix}$$

> $f(\langle a, b \rangle) = M.\langle a, b \rangle$

$$\begin{bmatrix} 2\,a + b \\ b - a \end{bmatrix} = \begin{bmatrix} 2\,a + b \\ b - a \end{bmatrix}$$

We use this concept now to rotate some plots in $\mathbb{R}^2$. Specifically, to rotate them around the origin (as rotation around an arbitrary point is not, in general, a linear transformation). We need only know what happens to the standard basis vectors $(1, 0), (0, 1)$ under the rotation. It is simpler to use polar co-ordinates, however, be warned that polar co-ordinates (r, θ) most emphatically are not a linear combination of basis vectors. They refer to a distance from the origin and an angle from the x-axis. To avoid confusion, we use the polar form of complex numbers $e^{i\theta}$ when referring to points in polar form.

As complex numbers, our basis vectors are simply $1 = e^{i0}$ and $e^{i\frac{\pi}{2}}$. Rotating anti-clockwise by an angle of θ we have that

$$
\begin{array}{rcccl}
(1,0) = & 1 & \mapsto & e^{i\theta} & = (\cos(\theta), \sin(\theta)) \\
(0,1) = & e^{i\frac{\pi}{2}} & \mapsto & e^{i(\frac{\pi}{2}+\theta)} & = \left(\cos\left(\frac{\pi}{2}+\theta\right), \sin\left(\frac{\pi}{2}+\theta\right)\right)
\end{array}
$$

and we can now construct our matrix, which we construct as a function of θ so that we may reuse it for different angles.

> $R := \text{theta} \rightarrow \left\langle \langle \cos(\text{theta}), \sin(\text{theta}) \rangle | \left\langle \cos\left(\frac{\text{Pi}}{2} + \text{theta}\right), \sin\left(\frac{\text{Pi}}{2} + \text{theta}\right) \right\rangle \right\rangle$

$$
R := \theta \rightarrow \left\langle \langle \cos(\theta), \sin(\theta) \rangle | \left\langle \cos\left(\frac{\text{Pi}}{2} + \theta\right), \sin\left(\frac{\text{Pi}}{2} + \theta\right) \right\rangle \right\rangle
$$

> $R(\text{theta})$

$$
\begin{bmatrix}
\cos(\theta) & -\sin(\theta) \\
\sin(\theta) & \cos(\theta)
\end{bmatrix}
$$

The observant student will notice that *Maple* has simplified our second column vector for us. These verifications can easily be verified $(\cos(\pi/2 + \theta) = -\sin(\theta)$ and $\sin(\pi/2 + \theta) = \cos theta)$. This is the standard form for a rotation matrix in $\mathbb{R}^2$.

Let us first rotate a parabola through an angle of $\frac{1}{3}\pi$. In order for this linear function to be applied, the matrix must be multiplied with a vector, so we need to use the vector equation (or parameterized equation) for the parabola. That is, $(x, y) = (t, t^2)$.

> $R\left(\frac{\text{Pi}}{3}\right) . \langle t, t^2 \rangle$

$$
\begin{bmatrix}
\frac{1}{2} t - \frac{1}{2} \sqrt{3}\, t^2 \\
\frac{1}{2} \sqrt{3}\, t + \frac{1}{2} t^2
\end{bmatrix}
$$

This gives us our parameterization of the rotated parabola, but we need to turn it into a list with a range for t so that the **plot** function will know what to do with it.

> $convert(\%, list)$

$$
\left[\frac{1}{2} t - \frac{1}{2} \sqrt{3} t^2, \frac{1}{2} \sqrt{3} t + \frac{1}{2} t^2 \right]
$$

> $P := [op(\%), t = -2..2]$

$$
\left[\frac{1}{2} t - \frac{1}{2} \sqrt{3} t^2, \frac{1}{2} \sqrt{3} t + \frac{1}{2} t^2, t = -2\ldots 2 \right]
$$

And now we can plot it.

> $plot(P, scaling = constrained)$
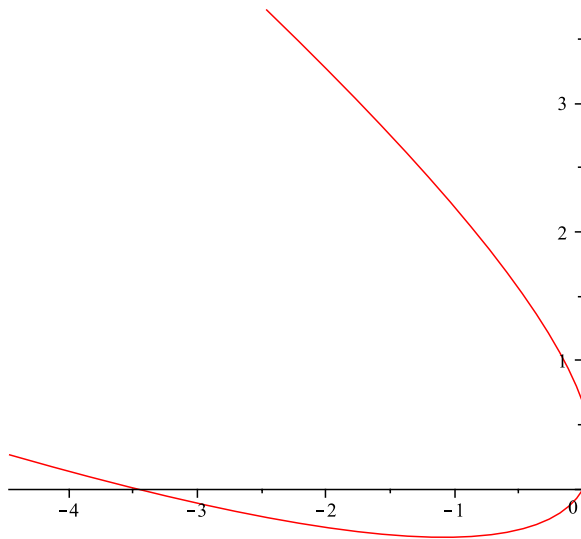
And now we rotate a sine curve clockwise by $\frac{1}{4}\pi$.

> $plot\left(\left[op\left(convert\left(R\left(-\dfrac{\text{Pi}}{4}\right).\langle t, \sin(t)\rangle, list\right), t = -2\cdot\text{Pi}..2\cdot\text{Pi}\right)\right],\right.$
> $\left.scaling = constrained\right)$
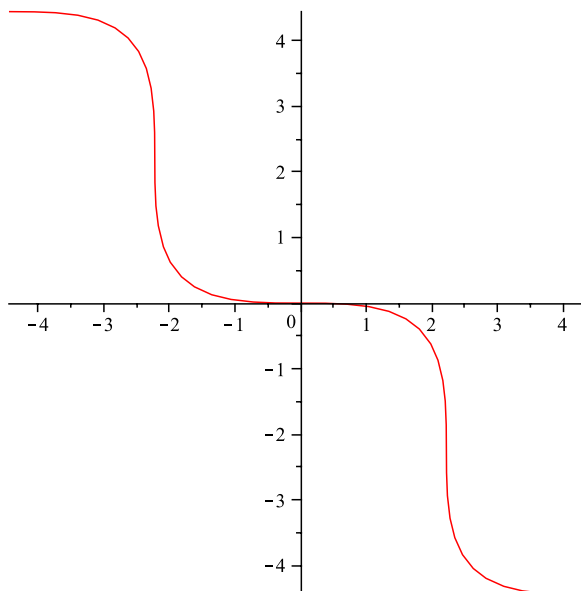
Finally we make an observation. Because every linear transformation can be represented as matrix multiplication, it should then be clear that a linear transformation may only produce linear combinations of the components of a vector. In other words, given an arbitrary vector $(v_1, \ldots, v_n)$ we should never see $v_i^k, v_i v_j$ or anything similar as components of the resultant vector. We can only see linear combinations of the v_i,

as a result of matrix multiplication. Looking back at our examples from Section 3.3.1 it should be quite clear now that $g : (a, b) \mapsto (a^2, b)$ could not possibly be a linear mapping, whereas $f : (a, b) \mapsto (2a + b, b - a)$ clearly is.

3.3.3 Eigenvectors and Eigenvalues

If we have a linear transformation, $T : V \to V$ say, operating on a vector space V, then it might be the case that there is a vector $v \in V$ that doesn't get moved at all by T, and is only scaled. That is, $T(v) = \lambda v$. We call such a vector an *eigenvector*, and λ the *eigenvalue* corresponding to the eigenvector. Because 0 always remains unchanged by a linear transformation, we do not consider it to be an eigenvector. Note that if $\lambda = 1$ then v is not only an eigenvector, but it is a fixed point of the mapping (as it remains unchanged by the transformation).

The question now is how do we find these eigenvectors? First, instead of thinking of $T(v)$ as the image of v under the transformation T, we consider T to be a matrix (which we can do in light of the previous section). So we are looking for solutions to the vector equation $Tv = \lambda v$, which we may equivalently write $Tv = \lambda I v$, where I is the identity matrix. Then

$$Tv = \lambda I v \;\Rightarrow\; Tv - \lambda I v = 0 \;\Rightarrow\; (T - \lambda I)v = 0$$

and we now have a vector equation corresponding to a linear system of the sort we have dealt with earlier in Sections 3.1.2 and 3.2.

We know that 0 is not an eigenvector, but will be a solution to the linear system, so we need the linear system to have a nontrivial solution. We know from our theorem that this will not happen if the determinant of the matrix is nonzero, so we are looking for values of λ where the determinant of $T - \lambda I$ is zero.

Let us have a look at our linear transformation f from Section 3.3.1.

> $T := \langle f(\langle 1, 0 \rangle) | f(\langle 0, 1 \rangle) \rangle$

$$\begin{bmatrix} 2 & 1 \\ -1 & 1 \end{bmatrix}$$

> $T - \mathrm{lambda} \cdot IdentityMatrix(2)$

$$\begin{bmatrix} 2 - \lambda & 1 \\ -1 & 1 - \lambda \end{bmatrix}$$

The resultant matrix $T - \lambda I$ is simply the matrix T with *lambda* subtracted from each entry on the diagonal. This always happens no matter the size of the matrix. When we calculate the determinant of this (or any other) matrix, we end up with a polynomial in λ.

> $Determinant(\%)$

$$3 - 3\lambda + \lambda^2$$

This is called the *characteristic polynomial* of T. *Maple* can calculate this directly using the **CharacteristicPolynomial** function from the LinearAlgebra package. To use this function we also need to tell it the name we want to use for the variable of the polynomial.

> $CharacteristicPolynomial(T, \text{lambda})$

$$3 - 3\lambda + \lambda^2$$

We do not yet have solutions for v, but we have solutions for λ. We know the only possible eigenvalues are the values of λ that solve $\chi(\lambda) = 0$ where χ is the characteristic polynomial. Once we know the values for λ (of which there can only be finitely many, for they are the roots of a polynomial) then we may substitute them into our linear system and solve for v.

> $solve(\% = 0)$

$$\left[\frac{3}{2} + \frac{1}{2}I\sqrt{3}, \frac{3}{2} - \frac{1}{2}I\sqrt{3}\right]$$

We can see straight away that our linear transformation f has no eigenvectors, because the possible eigenvalues are all complex, and there is no possible way our integer-valued matrix multiplied by any real-valued vector could produce a complex scaling of that vector.

Let us look at another example—one that actually has eigenvectors this time—and in three dimensions.

> $T := \langle\langle -1, -6, 4\rangle | \langle 2, 6, -2\rangle | \langle 0, 0, 1\rangle\rangle$

$$\begin{bmatrix} -1 & 2 & 0 \\ -6 & 6 & 0 \\ 4 & -2 & 1 \end{bmatrix}$$

> $CharacteristicPolynomial(T, \text{lambda})$

$$-6 + \lambda^3 - 6\lambda^2 + 11\lambda$$

> $factor(\%, \text{lambda})$

$$(\lambda - 1)(\lambda - 2)(\lambda - 3)$$

We can see straight away that $\lambda \in 1, 2, 3$ are solutions to the characteristic polynomial, and so are eigenvalues of the matrix T. This is great, but we still need to know what the corresponding eigenvectors are. All we know at the moment is that, for our linear map T, there is a vector which stays the same (after T is applied), there is another vector which becomes twice as large, and yet another vector which becomes three times as large. But we have no idea which vectors they may be.[1]

In order to find the vectors, we return to our vector equation $(T - \lambda I)v = 0$. However, we now know the only values of λ that could possibly satisfy this equation, and if we substitute these values into the above equation (one by one) we have three separate equations of the form $Mv = 0$ where $M = (T - \lambda I)$. These are, of course, linear systems that we should be quite familiar and adept with by now. So we simply solve these linear systems.

---

[1] The astute reader who is familiar with matrix operations should be able to make a good guess, from the matrix of T alone, as to which vector remains unchanged.

```
> for lambda in [1, 2, 3] do
    'lambda' = lambda, LinearSolve(T − lambda · IdentityMatrix(3), ⟨0, 0, 0⟩)
  od;
  lambda :=' lambda' :
```

$$\lambda = 1, \begin{bmatrix} 0 \\ 0 \\ \_t5_3 \end{bmatrix}$$

$$\lambda = 2, \begin{bmatrix} \_t6_3 \\ \dfrac{3}{2} \_t6_3 \\ \_t6_3 \end{bmatrix}$$

$$\lambda = 3, \begin{bmatrix} \_t7_1 \\ 2 \_t7_1 \\ 0 \end{bmatrix}$$

And we have our answer, although we need to think about it for a second. Maple has given us a whole family of vectors for each eigenvalue. This shouldn't surprise us, for if we have an eigenvector v, then $Tv = \lambda v$ for some λ. But because T is linear, then $T(kv) = kTv = k\lambda v$ for any $k \in \mathbb{F}$ in our field of scalars. So any scale of v is also an eigenvector corresponding to the eigenvector λ. What we usually do here is to just pick the simplest looking such vector to represent all of the possible vectors. In this spirit we declare that $(0, 0, 1), (1, \frac{3}{2}, 1)$ and $(1, 2, 0)$ are the eigenvectors corresponding to the eigenvectors $1, 2$, and 3, respectively.

To demonstrate this, we check this answer in *Maple*. We simplify the parameter in the vectors by calling it t.

```
> T . ⟨0, 0, t⟩
```

$$\begin{bmatrix} 0 \\ 0 \\ t \end{bmatrix}$$

```
> T . ⟨t, 3/2 · t, t⟩
```

$$\begin{bmatrix} 2t \\ 3t \\ 2t \end{bmatrix}$$

```
> T . ⟨t, 2 · t, 0⟩
```

$$\begin{bmatrix} 3t \\ 6t \\ 0 \end{bmatrix}$$

We explore one more example. In the previous example we had a 3×3 matrix, and we ended up with three distinct eigenvalues, and three distinct single-dimensional families of eigenvectors, each one corresponding to a particular eigenvalue. This does not always

happen in general.[2] Furthermore, there is no requirement that eigenvalues be nonzero (unlike eigen*vectors*). The example we now look at demonstrates both of these points.

> $T := \langle\langle 2, 4, 3, 1\rangle | \langle -5, -19, -15, -29\rangle | \langle 6, 24, 19, 38\rangle | \langle 0, 0, 0, 2\rangle\rangle$

$$\begin{bmatrix} 2 & -5 & 6 & 0 \\ 4 & -19 & 24 & 0 \\ 33 & -15 & 19 & 0 \\ 17 & -29 & 38 & 2 \end{bmatrix}$$

> $CharacteristicPolynomial(T, \text{lambda})$

$$\lambda^4 - 4\lambda^3 + 5\lambda^2 - 2\lambda$$

> $factor(\%, \text{lambda})$

$$\lambda\,(\lambda - 2)\,(\lambda - 1)^2$$

This time we have $0, 1$, and 2 as the only eigenvalues, however, notice that in this case 1 is a repeated root of the characteristic polynomial, and so we also consider it to be a repeated eigenvalue of the linear operator (or, equivalently, matrix) T. We now find the eigenvectors.

> **for** lambda **in** $[0, 1, 2]$ **do**
> $'\text{lambda}' = \text{lambda}, LinearSolve(T - \text{lambda} \cdot IdentityMatrix(4), \langle 0, 0, 0, 0\rangle)$
>   **od**;
> $\text{lambda} :=' \text{lambda}' :$

$$\lambda = 0, \begin{bmatrix} 2\_t5_4 \\ 8\_t5_4 \\ 6\_t5_4 \\ \_t5_4 \end{bmatrix}$$

$$\lambda = 1, \begin{bmatrix} 5\_t6_2 - 6\_t6_3 \\ \_t6_2 \\ \_t6_3 \\ 24\_t6_2 - 32\_t6_3 \end{bmatrix}$$

$$\lambda = 2, \begin{bmatrix} 0 \\ 0 \\ 0 \\ \_t7_4 \end{bmatrix}$$

Now, we have $(2, 8, 6, 1)$ as "the" eigenvector corresponding to eigenvalue 0 as well as $(0, 0, 0, 1)$ as "the" eigenvector corresponding to eigenvalue 2. Of course, any scale multiple of these vectors is also a corresponding eigenvector. In particular, this means that any scale of the vector $(2, 8, 6, 1)$ will be turned into the zero vector by our transformation T. Eigenvalue 1 is more interesting, but we verify the easy ones first.

---

[2] Indeed, we have already seen an example with no eigenvalues or eigenvectors.

> $T \cdot \langle 2t, 8t, 6t, t \rangle$

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

> $T \cdot \langle 0, 0, 0, t \rangle$

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 2t \end{bmatrix}$$

Now, to our repeated eigenvalue, 1: we need to recognize that the vector given to us by *Maple* has two parameters, and so is a two-dimensional family of vectors. The parameterization may be written as $t \cdot (5, 1, 0, 24) + s \cdot (-6, 0, 1, -32)$. We could think of $(5, 0, 1, 24)$ and $(-6, 0, 1, -32)$ as being two separate eigenvectors of T, both corresponding to the eigenvalue 1 (and, of course, any scale of each vector) as follows.

> $T \cdot \langle 5t, 0, t, 24t \rangle$

$$\begin{bmatrix} 5t \\ t \\ 0 \\ 24t \end{bmatrix}$$

> $T \cdot \langle -6t, 0, t, -32t \rangle$

$$\begin{bmatrix} -6t \\ 0 \\ t \\ -32t \end{bmatrix}$$

However, this is really only half the story. What *Maple* has given us is a two-dimensional family of vectors, any of which are left unchanged by our transformation T

> $\langle 5t, 0, t, 24t \rangle + \langle -6s, 0, s, -32s \rangle \, ; T \cdot \%$

$$\begin{bmatrix} 5t - 6s \\ t \\ s \\ 24t - 32s \end{bmatrix}$$

$$\begin{bmatrix} 5t - 6s \\ t \\ s \\ 24t - 32s \end{bmatrix}$$

Observe, firstly, that each family of eigenvectors is, in fact, a vector subspace of the space which the matrix acts on; verification is left as an exercise for the reader. These spaces are called *eigenspaces*.

Observe, secondly, that in this, and the previous, example the dimensions of the eigenspaces all add up to the dimension of the whole space. In the most recent case we had eigenspaces of degree 2, 1, and 1 which add together to give us 4 which was the dimension of the vector space upon which the matrix acted. This is not always the case, but we explore the cases in which it happens in the next section.

3.3.4 Diagonalization

We motivate this section with an observation regarding a previous example. Let us look at our 3×3 matrix from the previous section,

$$\begin{bmatrix} -1 & 2 & 0 \\ -6 & 6 & 0 \\ 4 & -2 & 1 \end{bmatrix}$$

and its eigenvectors, $(0, 0, 1), (1, 3/2, 1)$ and $(1, 2, 0)$ which correspond to eigenvalues $1, 2$ and 3 respectively. In order to avoid messy fractions, we use the vector $(2, 3, 2)$ instead of $(1, 3/2, 1)$ as they are from the same eigenspace.

> $T := \langle\langle -1, -6, 4\rangle | \langle 2, 6, -2\rangle | \langle 0, 0, 1\rangle\rangle$

$$\begin{bmatrix} -1 & 2 & 0 \\ -6 & 6 & 0 \\ 4 & -2 & 1 \end{bmatrix}$$

> $e_1, e_2, e_3 := \langle 0, 0, 1\rangle, \langle 2, 3, 2\rangle, \langle 1, 2, 0\rangle$

$$e_1, e_2, e_3 := \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 3 \\ 2 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix}$$

The observation we make is simple. These eigenvectors are linearly independent, and because there are three of them, they form an alternate basis for $\mathbb{R}^3$. This is an observation that we can easily check in *Maple* thanks to our work back in Section 3.2.

> $LinearSolve(\langle e_1 | e_2 | e_3\rangle, \langle 0, 0, 0\rangle)$

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Now, inasmuch as our vectors e_1, e_2, and e_3 form a basis (which we call B), then we can write any vector $v \in \mathbb{R}^3$ as a linear combination of these new basis vectors. We use *Maple* to find an explicit formula for this new combination, by solving the appropriate linear system (just as we did in Section 3.2.4).

> $LinearSolve(\langle e_1|e_2|e_3\rangle, \langle x, y, z\rangle)$

$$\begin{bmatrix} -4\,x + 2\,y + z \\ 2\,x - y \\ -3\,x + 2\,y \end{bmatrix}$$

> $ChgBasis := v \rightarrow LinearSolve(\langle e_1|e_2|e_3\rangle, v)$

$$f := v \rightarrow LinearAlgebra\text{:-}LinearSolve(\langle e_1|e_2|e_3\rangle, v)$$

Let's see what the vector $(1, 1, 1)$ becomes in terms of the basis B.

> $ChgBasis(\langle 1, 1, 1\rangle)$

$$\begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix}$$

It is quite elementary to verify that $(-1, 1, -1)_B = -1 \cdot (0, 0, 1) + 1 \cdot (2, 3, 2) - 1 \cdot (1, 2, 0) = (1, 1, 1)$.

The question that may occur now, is what will our linear operator T do to vectors written in terms of our new basis? Clearly T has not changed, and neither have the vectors themselves, just how we write them. However, because we are writing the vectors slightly differently now, our old matrix for T will probably not be appropriate anymore, as it was expecting to be multiplied by a vector that contained the coefficients of the standard basis, and not our new basis.

If we think about this for a bit, we can apply what we already know about linear transformations to obtain an answer. We know that, because T is linear, then $T(\lambda_1 v_1 + \lambda_2 v_2) = \lambda_1 T(v_1) + \lambda_2 T(v_2)$. We apply this to an arbitrary vector v written in terms of our basis B; that is, $v = \lambda_1 e_1 + \lambda_2 e_2 + \lambda_3 e_3$. When we do this we get that

$$\begin{aligned} T(v) &= T(\lambda_1 e_1 + \lambda_2 e_2 + \lambda_3 e_3) \\ &= \lambda_1 T(e_1) + \lambda_2 T(e_2) + \lambda_3 T(e_3) \\ &= \lambda_1 \cdot 1 \cdot e_1 + \lambda_2 \cdot 2 \cdot e_2 + \lambda_3 \cdot 3 \cdot e_3 \end{aligned}$$

because, in this case e_1, e_2, and e_3 are eigenvectors corresponding to eigenvalues 1, 2, and 3, respectively. To put this more succinctly

$$T\left((\lambda_1, \lambda_2, \lambda_3)_B\right) = (\lambda_1, 2\lambda_2, 3\lambda_3)_B$$

which can be achieved by multiplication (on the left) by the matrix

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

We demonstrate this for an arbitrary three-dimensional vector, $v = (x, y, z)$. First, we show what the transformation does to the vector directly. We call this new vector w.

```
> w := T.⟨x, y, z⟩
```

$$w := \begin{bmatrix} -x + 2\,y \\ -6\,x + 6\,y \\ 4\,x - 2\,y + z \end{bmatrix}$$

Recall from earlier that v when written as a vector with respect to the basis B is

$$v = (-4x + 2y + z, 2x - y, -3x + 2y)_B$$

Now, from our recent calculations, we should expect $T(v)$ to be equal to

$$T(v) = (1\cdot(-4x+2y+z), 2\cdot(2x-y), 3\cdot(-3x+2y))_B = (-4x+2y+z, 4x-2y, -9x+6y)_B$$

which is, as it happens, exactly what happens.

```
> ChgBasis(w)
```

$$\begin{bmatrix} -4\,x + 2\,y + z \\ 4\,x - 2\,y \\ -9\,x + 6\,y \end{bmatrix}$$

Let us look a little more closely at the change of basis. One should notice that it is a linear transformation, as it is the solution of a linear system of equations. As such we should be able to represent it as matrix multiplication. We construct the matrix as we did in Section 3.3.2, by seeing what the change of basis does to the standard basis vectors. We call the matrix B.

```
> B := ⟨ChgBasis(⟨1, 0, 0⟩)| ChgBasis(⟨0, 1, 0⟩)| ChgBasis(⟨0, 0, 1⟩)⟩
```

$$B := \begin{bmatrix} -4 & 2 & 1 \\ 2 & -1 & 0 \\ -3 & 2 & 0 \end{bmatrix}$$

The form of this matrix should not be surprising when we compare it to the elements of the form of an arbitrary vector (v, above). We see a column of 4, 2, and -3 which are precisely the coefficients of x, another column of 2, -1, and 2 which are the coefficients of y, and 1, 0, and 0 which are the coefficients of z.

With this information now we may perform the steps above with simple matrix multiplication, using this matrix B, and the diagonal matrix above, which we name Di. We would prefer to call this diagonal matrix simply D, but recall that *Maple* uses **D** for differentiation, and it is a reserved name. We use *Maple*'s **DiagonalMatrix** function from the LinearAlgebra package for the purpose.

```
> Di := DiagonalMatrix([1, 2, 3])
```

$$Di := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

```
> Di.B.⟨x, y, z⟩
```

$$\begin{bmatrix} -4\,x + 2\,y + z \\ 4\,x - 2\,y \\ -9\,x + 6\,y \end{bmatrix}$$

This vector, which we have calculated a number of ways now, is still with respect to our basis B (not to be confused with our matrix B). It should follow that if we can change our basis from the standard basis to the basis B, then we should be able to freely change back. That is, the change of basis should be an invertible function. This is reinforced when we remember that the coefficients of the standard basis are unique to each unique vector, as are the coefficients of the vectors in our basis B. What we have here is an isomorphism, which really ought to be invertible. We should expect, then, that the matrix B has an inverse, and that multiplying by this inverse should undo the basis change, as clearly $B^{-1}B = I$.

> $B^{-1}.\%$

$$\begin{bmatrix} -x + 2y \\ -6x + 6y \\ 4x - 2y + z \end{bmatrix}$$

Now, it should occur to us that we have never yet tried to see what our linear function T actually does to an arbitrary vector. This is an egregious oversight, which we correct immediately.

> $T.\langle x, y, z \rangle$

$$\begin{bmatrix} -x + 2y \\ -6x + 6y \\ 4x - 2y + z \end{bmatrix}$$

So now we have two matrix representations for our linear operator T. One is just the matrix we started with, and the other is the product of three matrices, one of which is diagonal $B^{-1} Di B$. In fact, we should expect that $T = B^{-1} Di B$.

> $T = B^{-1}.Di.B$

$$\begin{bmatrix} -1 & 2 & 0 \\ -6 & 6 & 0 \\ 4 & -2 & 1 \end{bmatrix} = \begin{bmatrix} -1 & 2 & 0 \\ -6 & 6 & 0 \\ 4 & -2 & 1 \end{bmatrix}$$

So now, one might ask, just what is B^{-1}?

> B^{-1}

$$\begin{bmatrix} 0 & 2 & 1 \\ 0 & 3 & 2 \\ 1 & 2 & 0 \end{bmatrix}$$

It is precisely the matrix consisting of our eigenvectors, in the exact order we used them back when we solved the linear system for the *ChgBasis* function we wrote.

> $\langle e_1 | e_2 | e_3 \rangle$

$$\begin{bmatrix} 0 & 2 & 1 \\ 0 & 3 & 2 \\ 1 & 2 & 0 \end{bmatrix}$$

This is an example of a *diagonalizable* matrix. That is, an $n \times n$ matrix, M say, which may be written as a product of matrices $P D P^{-1}$ where D is a diagonal matrix. Alternately, we might say that $P^{-1} M P$ is a diagonal matrix. If we rename B^{-1} to be P in the previous example, then we see that this is satisfied.

Let us look now at the next example from the previous section (Section 3.3.3).

> $T := \langle\langle 2, 4, 3, 1\rangle | \langle -5, -19, -15, -29\rangle | \langle 6, 24, 19, 38\rangle | \langle 0, 0, 0, 2\rangle\rangle$

$$\begin{bmatrix} 2 & -5 & 6 & 0 \\ 4 & -19 & 24 & 0 \\ 33 & -15 & 19 & 0 \\ 17 & -29 & 38 & 2 \end{bmatrix}$$

We should remember that this example had eigenvalues $0, 1, 2$ and that the eigenspace associated with eigenvalue 1 was two-dimensional. Specifically, we had eigenvector $(2, 8, 6, 1)$ corresponding to eigenvalue 0, eigenvectors $(5, 1, 0, 24)$ and $(-6, 0, 1, -32)$ formed a basis for the eigenspace corresponding to eigenvalue 1, and eigenvector $(0, 0, 0, 1)$ corresponded to eigenvalue 2.

It is prudent to talk about the multiplicity of an eigenvalue as well as the dimension of the eigenspace. The multiplicity of the eigenvalue, is simply its multiplicity in as a root of the characteristic polynomial. In this example the characteristic polynomial is $x(x - 2)(x - 1)^2$, and so $x = 0$ and $x = 2$ are single roots, whereas $x = 1$ is a double root. Hence the eigenvalues 0 and 2 had multiplicity 1, and eigenvalue 1 had multiplicity 2. Similarly the dimensions of the eigenspaces are the same as the multiplicity of the corresponding eigenvalue.

We ask *Maple* to calculate the eigenvectors directly.

> $E := Eigenvectors(T)$

$$E := \begin{bmatrix} 0 \\ 1 \\ 1 \\ 2 \end{bmatrix}, \begin{bmatrix} 2 & \dfrac{5}{24} & \dfrac{2}{3} & 0 \\ 8 & \dfrac{1}{24} & \dfrac{4}{3} & 0 \\ 6 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

We see here that *Maple* has calculated a vector of our eigenvectors, as well as a matrix consisting of eigenvectors as its columns. Well, almost. We see $(2, 8, 6, 1)$ as well as $(0, 0, 0, 1)$ as the first and last column, as well as $(5/24, 1/24, 0, 1)$ which is clearly a scale of $(5, 1, 0, 24)$. However, we have a mysterious column $(2/3, 4/3, 1, 0)$.

If we remember that a general form of an eigenvector corresponding to eigenvalue 1 as $t \cdot (5, 1, 0, 24) + s \cdot (-6, 0, 1, -32)$, then setting $t = 4/3$ and $s = 1$ we get that $5t - 6s = 2/3$ and that $24t - 32t = 0$ giving us that our mysterious vector $(2/3, 4/3, 1, 0)$ is just another eigenvector corresponding to eigenvalue 1. It should also be clear that this vector and $(5/24, 1/24, 0, 1)$ are linearly independent, and so form a basis of the eigenspace corresponding to eigenvalue 1.

We can now produce our diagonal matrix, and see if we can diagonalize the original matrix.

> $Di, P := DiagonalMatrix(E_1), E_2$

$$Di, P := \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}, \begin{bmatrix} 2 & \dfrac{5}{24} & \dfrac{2}{3} & 0 \\ 8 & \dfrac{1}{24} & \dfrac{4}{3} & 0 \\ 6 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

We could, at this point, simply calculate $P \; Di \; P^{-1}$ and see if the matrix we get is equal to T, however, it is probably prudent and illustrative to check that the eigenvectors we have are all linearly independent (and thus form a basis for $\mathbb{R}^4$). Fortunately, we already have the eigenvectors handily arranged into the matrix named P, so establishing the linear independence is precisely the same as verifying that the vector equation $Px = 0$ has only the trivial solution.

> $LinearSolve(P, \langle 0, 0, 0, 0 \rangle)$

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Thus we see that these vectors form the required basis, and so we can be quite sure that diagonalization will work. We see that indeed it does.

> $P.Di.P^{-1}$

$$\begin{bmatrix} 2 & -5 & 6 & 0 \\ 4 & -19 & 24 & 0 \\ 33 & -15 & 19 & 0 \\ 17 & -29 & 38 & 2 \end{bmatrix}$$

In both of the above examples, our matrix has had exactly as many linearly independent eigenvectors in number as the dimension of the vector space upon which it acts. That is, we had a 3×3 matrix with 3 linearly independent eigenvectors, and also a 4×4 matrix with 4 linearly independent eigenvectors. The process of diagonalization that we have used in both of these examples is one of using these eigenvectors as a basis for the underlying space. With this basis we find that we may apply the function simply by multiplying the coefficients of the basis vectors by fixed values (and hence the diagonal matrix), and then change back to the regular basis.

It should be clear then that as long as we have n linearly independent eigenvectors for an $n \times n$ matrix, M say, we can always follow this procedure, and we will thus always have a diagonalizable matrix. It turns out that this must always happen and that, in fact, an equivalent definition of an $n \times n$ matrix being diagonalizable is that it has exactly n linearly independent eigenvectors. To see that this is equivalent, we must see that a diagonalizable matrix M will always have exactly n linearly independent eigenvectors. We do not prove this here, but the proof is quite elementary and can be found in any good linear algebra text.

Why is diagonalization desirable? Well, diagonalization has applications in the solving of differential equations, as well as recurrence relations, and more besides. However, a more down to earth reason is that it can make taking powers of the matrix much simpler.

Without diagonalization, to compute T^n (which is just $TT \cdots T$ where there are n multiplications) we would have to compute n matrix multiplications. In the case where n is large, such a computation might be prohibitively time consuming, even for a computer.

In the case that T is diagonalizable, then we can write $T = PDP^{-1}$, and so

$$T^n = TT \cdots T = PDP^{-1}PDP^{-1} \cdots PDP^{-1} = PD^nP^{-1}$$

where each P^{-1} is canceled out by multiplication by P in the middle of the expression, leaving only a single P on the left and P^{-1} on the right, and n Ds all multiplied together in the middle forming D^n.

Now, the power of a diagonal matrix is very simple to take. One simply raises each entry on the diagonal to the power n. The reader is encouraged to experiment with some simple 2×2 or 3×3 examples of diagonal matrices to see why. For this it is actually more illustrative to perform the calculations by hand in order to see the pattern of multiplication.

Clearly this is a potentially very great decrease in time taken to find large powers of a matrix. Imagine trying to find T^{100} the long way, compared to just three matrix multiplications, plus some work to find eigenvectors and eigenvalues. This is not just a speedup of hand calculations, either. Matrix multiplication on computers can be similarly sped up this way, and it is quite likely that *Maple* itself uses such techniques (and likely more sophisticated ones as well).

We look at one more example of a diagonalizable matrix. This time, we do so in complex space. We take this example from the previous section as well.

```
> T := ⟨⟨2, -1⟩|⟨1, 1⟩
```

$$\begin{bmatrix} 2 & 1 \\ -1 & 1 \end{bmatrix}$$

We should remember, from our earlier computation, that this example had no real eigenvalues, but it did have complex eigenvalues. If we think about this matrix as a linear transformation from $\mathbb{C}^2 \to \mathbb{C}^2$, then the roots of the characteristic equation, and thus the eigenvalues are $\frac{1}{2}(3 + i\sqrt{3})$ and $\frac{1}{2}(3 - i\sqrt{3})$.

```
> Eigenvectors(T);
  Di, P := DiagonalMatrix(%₁), %₂
```

$$\begin{bmatrix} 3/2 + 1/2\,i\sqrt{3} \\ 3/2 - 1/2\,i\sqrt{3} \end{bmatrix}, \begin{bmatrix} \left(-1/2 + 1/2\,i\sqrt{3}\right)^{-1} & \left(-1/2 - 1/2\,i\sqrt{3}\right)^{-1} \\ 1 & 1 \end{bmatrix}$$

$$Di, P := \begin{bmatrix} 3/2 + 1/2\,i\sqrt{3} & 0 \\ 0 & 3/2 - 1/2\,i\sqrt{3} \end{bmatrix}, \begin{bmatrix} \left(-1/2 + 1/2\,i\sqrt{3}\right)^{-1} & \left(-1/2 - 1/2\,i\sqrt{3}\right)^{-1} \\ 1 & 1 \end{bmatrix}$$

We see that the vectors are linearly independent by finding the determinant of the matrix P, remembering that a nonzero determinant is equivalent to the expression $Tx = 0$ having only the trivial solution, and thus the column vectors being linearly independent.

```
> Determinant(T)
```

$$\frac{4\,i\sqrt{3}}{\left(-1 + i\sqrt{3}\right)\left(1 + i\sqrt{3}\right)}$$

We have exactly two linearly independent, complex eigenvectors, which must therefore form a basis for $\mathbb{C}^2$. We can diagonalize the matrix.

```
> simplify(P. Di .P⁻¹)
```

$$\begin{bmatrix} 2 & 1 \\ -1 & 1 \end{bmatrix}$$

And there we have it. Note the we need to simplify the expression. The result of the computation $P. Di .P^{-1}$ on its own is complicated, and messy, and too troublesome to print.

Finally we look at an example of a matrix that is not diagonal. We have not proven that an $n \times n$ matrix must have n linearly independent eigenvectors for it to be diagonalizable, although we have referred the reader to where such a proof may be found. We take this as read, however, and show a matrix with too few eigenvectors.

> $T := \langle \langle 6, 4 \rangle | \langle -9, -6 \rangle$

$$\begin{bmatrix} 6 & -9 \\ 4 & -6 \end{bmatrix}$$

> $Eigenvectors(T);$

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \dfrac{3}{2} & 0 \\ 1 & 0 \end{bmatrix}$$

Interpreting the output, we have a repeated eigenvalue of 0, and two eigenvectors, $(3/2, 1)$ and $(0, 0)$. However, we know the zero vector cannot be an eigenvector. Furthermore, taking linear combinations of these vectors yields only a one-dimensional eigenspace. Let's look at this a little closer to see what's going on.

> $CharacteristicPolynomial(T, \text{lambda})$

$$\lambda^2$$

Well that's an easy one to solve. Clearly $\lambda = 0$ is the only eigenvalue with multiplicity of 2. We will now manually calculate the eigenvectors. Recall that we need to solve the linear system $(T - \lambda I)x = 0$, which in this case collapses to $Tx = 0$.

> $LinearSolve(T, \langle 0, 0 \rangle)$

$$\begin{bmatrix} \dfrac{3}{2} \_t0_2 \\ \_t0_2 \end{bmatrix}$$

And here we see only a single-parameter vector, and hence only a one dimensional eigenspace. Because the multiplicity of the eigenvalue is greater than the dimension of the eigenspace we say that the eigenspace is *deficient*. This is sufficient to render our matrix T as not being invertible.

3.4 Exercises

1. a. Create the following vectors and matrices using the angle bracket $\langle\rangle$ notation.

    i. $\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$
    ii. (x, y, z, w)
    iii. $\begin{bmatrix} 1 & -2 & 7 \\ 8 & 4 & -5 \\ 7 & 9 & 2 \end{bmatrix}$

    Create the matrix twice, once as a row of column vectors and then as a column of row vectors.

    b. Create the following row vectors using the **Vector** function.

       i. $(\pi, \pi, \pi, \pi, \pi)$
       ii. $(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$

    Create the following as column vectors using the **Vector** function
    iii. $u = (u_1, \ldots, u_8)$ where $u_i = i^i$.
    iv. $v = (v_1, \ldots, v_{10})$ where v_i is the ith Fibonacci number.
    Create the following matrices using the **Matrix** function

    v. $\begin{bmatrix} e^2 & e^2 & e^2 & e^2 \\ e^2 & e^2 & e^2 & e^2 \\ e^2 & e^2 & e^2 & e^2 \\ e^2 & e^2 & e^2 & e^2 \end{bmatrix}$
    vi. $\begin{bmatrix} m_{1,1} & m_{1,2} & \cdots & m_{1,10} \\ m_{2,1} & m_{2,2} & \cdots & m_{2,10} \\ \vdots & \vdots & \ddots & \vdots \\ m_{5,1} & m_{5,2} & \cdots & m_{5,10} \end{bmatrix}$

    c. Create functions to produce the following general matrices.
       i. An $n \times n$ matrix $A = [a_{i,j}]$ where $a_{i,j} = \binom{n}{i} + \binom{n}{j}$ and $\binom{k}{m}$ are the binomial coefficients.
       ii. An $n \times m$ matrix $B = [b_{i,j}]$ where $b_{i,j} = i^3 + j^2$.
       iii. An $n \times m$ matrix $C = [c_{i,j}]$ where $c_{i,j} = f(i, j)$ for an arbitrary 2-variable function f.

2. a. Calculate the following matrix products

    i. $\begin{bmatrix} 1 & 5 & 9 \\ 2 & 6 & 10 \\ 3 & 7 & 11 \\ 4 & 8 & 12 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \end{bmatrix}$
    ii. $\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$

    b. Recall that the dot product between two vectors allows the calculation of the angle between the vectors with the formula

    $$u.v = |u| \cdot |v| \cdot \cos\theta$$

    Use this formula to find the angle between the following vectors. Also, find a function in the LinearAlgebra package for calculating the angle between vectors, and use it to verify your calculations.

    i. $(1, 0, 1)$ and $(1, 1, 0)$ iii. $(1, 2, 3, 4, 5, 6)$ and $(6, 5, 4, 3, 2, 1)$

    ii. $(2, 2, 2, 2)$ and $(3, 0, 3, 3)$ iv. $(1, 2, 3)$ and $(4, 5, 6)$

    Note: For a vector v, the value $|v|$ may be calculated in *Maple* with the **VectorNorm** command (from the LinearAlgebra package, of course), using the command *VectorNorm*$(v, 2)$. The ", 2" is necessary and tells *Maple* that we are calculating the 2-norm.

  c. Recall that the vector cross-product is an operation that can only be performed on three-dimensional vectors, and that it calculates a new vector perpendicular to the two vectors used in its calculation.

    Find the *Maple* command to perform cross-products, and use it to calculate the cross-product of the following vectors. Verify that the cross-product is, indeed, perpendicular to the two vectors.

    i. $(1, 0, 0)$ and $(0, 0, 1)$ ii. $(1, -2, 3)$ and $(3, 2, -1)$

3.   a. Plot the following systems of equations, and attempt to identify from the plot whether the system is solvable. Solve the system using linear algebraic techniques. Plot the solution space if there is more than one solution.

    i. $2x - 3y = -2$ iii. $2x + z = 1$

        $2x + y = 1$ $-3x + z = 3$

        $3x + 2y = 1$ $2y + z = 4$

                                       $-2y + z = 2$

    ii. $-x + y = 16$ iv. $x + y + z = 3$

        $2x + y = -17$ $-x + y + 3z = 3$

        $6x + 2y = -56$ $4x + y - 2z = 3$

  b. Solve the following linear systems and verify the solution. Express the solutions as vector equations.

    i. $4x + 3y + 2z + w = 1$ ii. $2x + y + z + w = 1$

       $x + 2y + 3z + 4w = 2$ $x + 2y + z + w = 2$

                                    $x + y + 2z + w = 3$

                                    $x + y + z + 2w = 4$

4. Let A_n be the $n \times n$ matrix where

$$a_{i,j} = \begin{cases} 2 & \text{if } i = j \\ 1 & \text{otherwise} \end{cases}$$

  and let $b = (b_1, \ldots, b_n)$ be the n-vector where $b_i = i$.

  Solve the linear system $A_n\, x = b$ for all values of n up to 10. Form a hypothesis about the solution for general n, and test that hypothesis for $n = 100$, and any other values of n that you choose.

5. Use row reduction to attempt to find inverses for the following matrices. If they are invertible, then give the inverse matrix, otherwise explain why they are not invertible.

a.
$$\begin{bmatrix} -90 & 30 & -4 \\ -54 & 57 & 17 \\ -27 & -69 & -40 \end{bmatrix}$$

b.
$$\begin{bmatrix} 40 & 71 & 40 & 72 \\ -51 & -44 & 83 & -26 \\ -21 & 79 & 41 & -39 \\ -5 & 17 & -76 & -58 \end{bmatrix}$$

c.
$$\begin{bmatrix} 22 & -3 & -56 & -18 & -72 \\ 29 & 7 & 80 & -94 & -55 \\ -84 & -62 & -42 & -26 & 89 \\ 64 & -84 & -12 & 23 & 59 \\ 7 & -65 & -30 & 27 & 60 \end{bmatrix}$$

6. Which of the following matrices will always give a unique solution to the vector equation $Ax = b$, which are invertible, and which will become the identity matrix in reduced row echelon form?

a.
$$\begin{bmatrix} 15 & 33 & 0 & -97 & 47 & 48 \\ -70 & 0 & 0 & -84 & 66 & 65 \\ 0 & -19 & -65 & 13 & 0 & 0 \\ 0 & -61 & 65 & 0 & -55 & -78 \\ -38 & 0 & 0 & 64 & -42 & 0 \\ -53 & 0 & -41 & 0 & -9 & 0 \end{bmatrix}$$

c.
$$\begin{bmatrix} 76 & 66 & 98 & -37 & 82 & -153 \\ 93 & -18 & -89 & -10 & -88 & 76 \\ 56 & -18 & -50 & 87 & -85 & -26 \\ -58 & 79 & 92 & 46 & 34 & -35 \\ 68 & 10 & -89 & -16 & 1 & 46 \\ 40 & -72 & 32 & 78 & -26 & -196 \end{bmatrix}$$

b.
$$\begin{bmatrix} -62 & 32 & 0 & 61 & 31 & -124 \\ 0 & -94 & 0 & 93 & -1 & 0 \\ -74 & 0 & 51 & 0 & -23 & -46 \\ -86 & -25 & 39 & 0 & -72 & -94 \\ -36 & -20 & 0 & 0 & -56 & -72 \\ 0 & -69 & 0 & -26 & -95 & 0 \end{bmatrix}$$

d.
$$\begin{bmatrix} 25 & -16 & -38 & 57 & -32 & 99 \\ 94 & -9 & -18 & 27 & -74 & 29 \\ 12 & -50 & 87 & -93 & -4 & 44 \\ -2 & -22 & 33 & -76 & 27 & 92 \\ 50 & 45 & -98 & -72 & 8 & -31 \\ 10 & -81 & -77 & -2 & 69 & 67 \end{bmatrix}$$

7. For the matrices in Exercise 6 that were invertible, find a sequence of row operations that will produce the matrix when performed on the identity matrix. (In other words, find the expression of the matrix as a product of elemenary matrices).

8. a. For the following, state whether the first vector is a linear combination of the other vectors.

   i. $(1, 2), (-1, 4), (2, -3)$ iii. $(-1, 2, 3), (1, 2, -3), (1, 10, -3)$
   ii. $(1, 2, 3), (-1, 2, 3), (1, -2, 3)$ iv. $(-1, 2, 3, -4), (3, -2, -1, 5), (7, -2, 3, 7),$
                                                   $(1, 2, 5, -3)$

   b. For the following, state whether the first polynomial is a linear combination of the other polynomials.
      i. $71 x^4 - 136 x^3 + 142 x^2 + 264 x + 265$

      $$-31 x^4 - 54 x^3 + 88 x + 31$$
      $$-82 x^4 - 13 x^3 - 71 x^2 - 86$$

ii. $67\,x^5 - 31\,x^4 + 92\,x^3 + 44\,x^2 + 29\,x + 99$

$\qquad 69\,x^5 + 8\,x^4 + 27\,x^3 - 4\,x^2 - 74\,x - 32$

$\qquad -2\,x^5 - 72\,x^4 - 76\,x^3 - 93\,x^2 + 27\,x + 57$

$\qquad -77\,x^5 - 98\,x^4 + 33\,x^3 + 87\,x^2 - 18\,x - 38$

9. Which of the sets of vectors and polynomials from Exercise 8 are linearly indepen-
   dent, and which are linearly dependent?

10. The matrix space $M_2(\mathbb{R})$ has a standard basis:

$$
\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix},
\begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix},
\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix},
\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}
$$

And so has dimension 4.

   a. Extend this notion to find a standard basis for $M_n(\mathbb{R})$. What is the dimension
      of $M_n(\mathbb{R})$? Use this to establish a correspondence between matrices in $M_n(\mathbb{R})$
      and $\mathbb{R}^m$ for suitable m.
   b. For the following sets of matrices calculate whether the first is a linear combi-
      nation of the others. Which of these sets are linearly independent, and which
      are linearly dependent?

   i. $\begin{bmatrix} -50 & 45 \\ -22 & -81 \end{bmatrix},
        \begin{bmatrix} 50 & -16 \\ 10 & -9 \end{bmatrix},
        \begin{bmatrix} 25 & 12 \\ 94 & -2 \end{bmatrix},
        \begin{bmatrix} 31 & -80 \\ -50 & 43 \end{bmatrix}$

   ii. $\begin{bmatrix} -36 & 1 \\ -99 & 53 \end{bmatrix},
        \begin{bmatrix} -61 & 77 \\ -48 & 9 \end{bmatrix},
        \begin{bmatrix} 24 & 86 \\ 65 & 20 \end{bmatrix},
        \begin{bmatrix} -25 & 76 \\ 51 & -44 \end{bmatrix}$

   iii. $\begin{bmatrix} -67 & 16 & 60 \\ 22 & 9 & -95 \\ 14 & 99 & -20 \end{bmatrix},
        \begin{bmatrix} 82 & 18 & -62 \\ 72 & -59 & -33 \\ 42 & 12 & -68 \end{bmatrix},
        \begin{bmatrix} -70 & 29 & -1 \\ 41 & 70 & 52 \\ 91 & -32 & -13 \end{bmatrix},
        \begin{bmatrix} -14 & 21 & 19 \\ 60 & 90 & 88 \\ -35 & 80 & -82 \end{bmatrix}$

11. A vector in $\mathbb{R}^3$ may be rotated around any of the three axes. Rotation around the
    z-axis is equivalent to rotating in the xy plane. Similarly rotating around the y-axis
    is equivalent to rotating in the xz plane and rotating around the x-axis is equivalent
    to rotating in the yz plane. We call these rotations R_{xy}, R_{xz} and R_{zy}, respectively.

   a. Convince yourself that each of these rotations is a linear transformation on $\mathbb{R}^3$.
   b. Construct rotation matrices for R_{xy}, R_{xz}, and R_{zy} for a rotation by an arbitrary
      angle θ.
   c. Show that any one of these rotations can be realized as a composite transfor-
      mation using only the other two and their inverses.
   d. How would you rotate a vector around an arbitrary line in $\mathbb{R}^3$?
   e. Plot a cube with a face pointing in the direction of the vector $(1, 1, 1)$.

12. Find the eigenvectors and eigenvalues of the following matrices.

   a. $\begin{bmatrix}
      -300 & 296 & -36 & 24 & 0 \\
      -309 & 305 & -36 & 24 & 0 \\
      -365 & 356 & -40 & 27 & 0 \\
      -699 & 680 & -92 & 61 & 0 \\
      328 & -320 & 44 & -24 & 4
      \end{bmatrix}$
   b. $\begin{bmatrix}
      1277 & -336 & -1668 & 1572 & 288 \\
      744 & -187 & -972 & 900 & 168 \\
      1634 & -432 & -2138 & 2021 & 370 \\
      548 & -144 & -718 & 679 & 124 \\
      1722 & -456 & -2259 & 2133 & 395
      \end{bmatrix}$

13. The *square root* of a (square) matrix, M say, is a matrix A such that $A \cdot A = M$.

a. Find the square root of the following matrices.

i.
$$\begin{bmatrix} 16 & 0 & 0 & 0 \\ 0 & 49 & 0 & 0 \\ 0 & 0 & 64 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

ii.
$$\begin{bmatrix} 36 & 0 & 0 & 0 & 0 & 0 \\ 0 & 16 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 49 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 81 \end{bmatrix}$$

What is the square root of an arbitrary diagonal matrix? (Justify your answer)

b. Find the square root of the following diagonalizable matrices from Sections 3.3.3 and 3.3.4.

i.
$$\begin{bmatrix} -1 & 2 & 0 \\ -6 & 6 & 0 \\ 4 & -2 & 1 \end{bmatrix}$$

ii.
$$\begin{bmatrix} 2 & -5 & 6 & 0 \\ 4 & -19 & 24 & 0 \\ 33 & -15 & 19 & 0 \\ 17 & -29 & 38 & 2 \end{bmatrix}$$

Hint: Use the fact that these matrices are diagonalizable, and the properties of powers of diagonalizable matrices.

14. Diagonalize the following matrices.

a.
$$\begin{bmatrix} 9 & -45 & 9 & -9 \\ 0 & 0 & 0 & 0 \\ 2 & -10 & 2 & -2 \\ 3 & -15 & 3 & -3 \end{bmatrix}$$

b.
$$\begin{bmatrix} 4 & 10 & -16 & 44 \\ -3 & 15 & -9 & 24 \\ -8 & 16 & 2 & -4 \\ -2 & 4 & -1 & 5 \end{bmatrix}$$

c.
$$\begin{bmatrix} 18 & -8 & 0 & -16 \\ 18 & -7 & 0 & -18 \\ 0 & 0 & 2 & 0 \\ 8 & -4 & 0 & -6 \end{bmatrix}$$

d.
$$\begin{bmatrix} -300 & 296 & -36 & 24 & 0 \\ -309 & 305 & -36 & 24 & 0 \\ -365 & 356 & -40 & 27 & 0 \\ -699 & 680 & -92 & 61 & 0 \\ 328 & -320 & 44 & -24 & 4 \end{bmatrix}$$

3.5 Further Explorations

1. A *positive* matrix $A = (a_{i,j})$ is a matrix over $\mathbb{R}$ where $a_{i,j} > 0$ for every i and j. In other words it is a real matrix with all positive entries. The Peron–Frobenius theorem states that such a matrix has a unique, largest, real eigenvalue, and a corresponding eigenvalue with all positive entries.
More technically stated:

Theorem 3 (Peron–Frobenius). *Let $A = (a_{i,j})$ be an $n \times n$ positive matrix. Then the following hold.*

    a. *There is a unique eigenvalue $r \in \mathbb{R}$ with the property that for every other eigenvalue λ it is the case that $|\lambda| < r$.*

    b. *The eigenvalue r is a simple root of the characteristic polynomial, and so is a degree 1 eigenvalue.*

    c. *There is an eigenvector $v = (v_1, \ldots, v_n)$ corresponding to the eigenvalue r has the property that $v_i > 0$ for every $1 \leq i \leq n$.*

    d. *The eigenvector v (above) is the only eigenvector with nonnegative entries.*

The eigenvalue r is sometimes called the Peron root *or the* Peron–Frobenius eigenvalue.

Be aware that the eigenvalues λ in 1a could potentially be complex, in which case the absolute value is the complex modulus. Similarly, as a consequence of 1d any other eigenvector of A (i.e., an eigenvector corresponding to a different eigenvalue) must have either a negative entry, or a complex one.

2. Recall recurrence relations from Section 1.3.3. The solution to a first order recurrence relation is quite straightforward. We may use a similarly straightforward approach to systems of recurrence relations (often called *difference equations*). Suppose we have n recurrence relations $a_1(k), \ldots, a_n(k)$ which are interlinked in some way. That is,

$$a_1(k) = \lambda_{1_1} a_1(k-1) + \lambda_{1_2} a_2(k-1) + \cdots + \lambda_{1_n} a_n(k-1)$$
$$a_2(k) = \lambda_{2_1} a_1(k-1) + \lambda_{2_2} a_2(k-1) + \cdots + \lambda_{2_n} a_n(k-1)$$
$$\vdots$$
$$a_n(k) = \lambda_{n_1} a_1(k-1) + \lambda_{n_2} a_2(k-1) + \cdots + \lambda_{n_n} a_n(k-1)$$

We may decompose this into something very reminiscent of a linear system. Let

$$A := \begin{bmatrix} \lambda_{1_1} & \lambda_{1_2} & \ldots & \lambda_{1_n} \\ \lambda_{2_1} & \lambda_{2_2} & \ldots & \lambda_{2_n} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{n_1} & \lambda_{n_2} & \ldots & \lambda_{n_n} \end{bmatrix} \text{ and } a(k) := \begin{bmatrix} a_1(k) \\ a_2(k) \\ \vdots \\ a_n(k) \end{bmatrix}$$

then

$$a(k) = A \cdot a(k-1)$$

and by the same argument we used in Section 1.3.3 we can see that

$$a(k) = A^k \cdot a(0)$$

How would you ascertain the long-term behavior of such a system?

In the special case that the elements of every row of A add to 1, we have a representation of a time-homogeneous Markov chain. This is not actually a difference equation, however. In this case we are representing some system with n states that between which it may transition. Each state is represented by a row and a column. The element $a_{i,j}$ (being the element in row i and column j) is the probability that the system will move to the state represented by column j if it is currently in the state represented by row i.

The similarity to difference equations is in taking higher powers of the matrix A in order to obtain a solution. If we have an n-vector, v_0 say, whose elements sum to 1, we can consider it to be a probability distribution of the states. That is, we consider the element v_i as being the probability that the state is in the state represented by row i, then the vector $v_1 = A \cdot v$ is the probability distribution of the system after a single transition, and the vector $v_k = A^k \cdot v$ is the probability distribution of the system after k transitions.

How would you ascertain the long term behavior of such a system?

3. Recall differential equations from Section 2.2.4. We may have interrelated differential equations in a similar manner to our difference equations above. Such equations are sometimes called *coupled* differential equations. Suppose we have the following system of differential equations.

$$y_1'(t) = \lambda_{1_1} y_1(t) + \lambda_{1_2} y_2(t) + \cdots + \lambda_{1_n} y_n(t)$$
$$y_2'(t) = \lambda_{2_1} y_1(t) + \lambda_{2_2} y_2(t) + \cdots + \lambda_{2_n} y_n(t)$$
$$\vdots$$
$$y_n'(t) = \lambda_{n_1} y_1(t) + \lambda_{n_2} y_2(t) + \cdots + \lambda_{n_n} y_n(t)$$

We construct a linear system. Let

$$A := \begin{bmatrix} \lambda_{1_1} & \lambda_{1_2} & \cdots & \lambda_{1_n} \\ \lambda_{2_1} & \lambda_{2_2} & \cdots & \lambda_{2_n} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{n_1} & \lambda_{n_2} & \cdots & \lambda_{n_n} \end{bmatrix}, \quad f(t) := \begin{bmatrix} f_1(t) \\ f_2(t) \\ \vdots \\ f_n(t) \end{bmatrix} \text{ and } f'(t) := \begin{bmatrix} f_1'(t) \\ f_2'(t) \\ \vdots \\ f_n'(t) \end{bmatrix}$$

Then our system of differential equations can be written as

$$f'(t) = A \cdot f(t)$$

If $\mu_1, \ldots, \mu_n$ are distinct eigenvalues of A with corresponding eigenvectors $v_1, \ldots, v_n$ then

$$f(t) = \sum_{i=1}^{n} c_i e^{\mu_i t} v_i$$

is the general form of the solution to where c_i are arbitrary constants. You can easily show that $e^{\mu t} v$ is a solution, and it's an easy step from there to see that a linear combination of solutions must also be a solution.

For second-order differential equations, $ay'' + by' + cy = 0$, we introduce a new function x such that $x = y'$ and we now have the following system of equations

$$y' = x$$
$$x'(t) = -\frac{b}{a}x - \frac{c}{a}y$$

which we can now evaluate using the matrix method above. Doing so will verify the characteristic polynomial method.

Extend this method to deal with higher-degree differential equations, and systems of coupled higher-degree differential equations. How might you cope with inhomogeneous cases?

Chapter 4
Visualization and Geometry: A Postscript

We conclude with a brief chapter on visualization and geometry. It should be noted here that although *Maple* is capable of handling geometric problems, it is inherently static. There is a growing variety of "interactive geometry" packages such as *Cinderella* and *GeoGebra* that allow a much more dynamic exploration to take place.

4.1 Useful Visualization Tools

Maple contains some useful tools for visualization that we discuss briefly here.

4.1.1 Text and Labeling

Many of the plots we produce may be enhanced with the addition of text to the plot to explain or label. Up until now we have only labeled the axes of a plot. We show here some more complicated labeling.

First, the **plot** and **plot3d** commands have some built-in labeling options (see the *Maple* help files regarding plotting options for more details). One simple change from our regular plots is to change which values are marked on the axes. A common use for this would be for plotting the sine function and showing the multiples of π along the horizontal axis.

The markings on any axis are called *tickmarks*, and are modified using the **tickmarks** argument to the **plot** command. The **tickmarks** argument takes a list that specifies the options for the horizontal axis and the vertical axis in that order. In our case we want the tickmarks on the horizontal axis to be shown at regular intervals of $\frac{1}{2}\pi$ in order to have the critical points of the sine curve suitably marked. We do not need, nor wish, for the vertical axis to be marked any differently from usual, and so we specify the **default** marking for it.

> $plot\left(\sin(\text{theta}), \text{theta} = 0..2 \cdot \text{Pi}, tickmarks = \left[spacing\left(\dfrac{\text{Pi}}{2}\right), default\right]\right)$
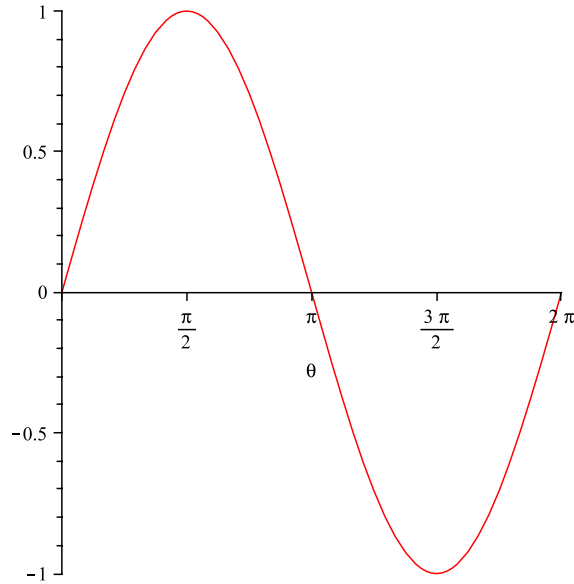
There are other handy labeling options as shown below. For example, we may label the axes, display a **legend** to multiple plots, or put a **caption** on the plot.

> $plot\left([\sin, \cos], 0..2 \cdot \text{Pi}, color = [red, blue], legend = [\sin, \cos],\right.$
>
> $labels = ["\text{Angle (radians)}", ""], tickmarks = \left[spacing\left(\dfrac{\text{Pi}}{2}\right), default\right],$
>
> $\left.caption = "\text{A comparison of sine and cosine curves}"\right)$
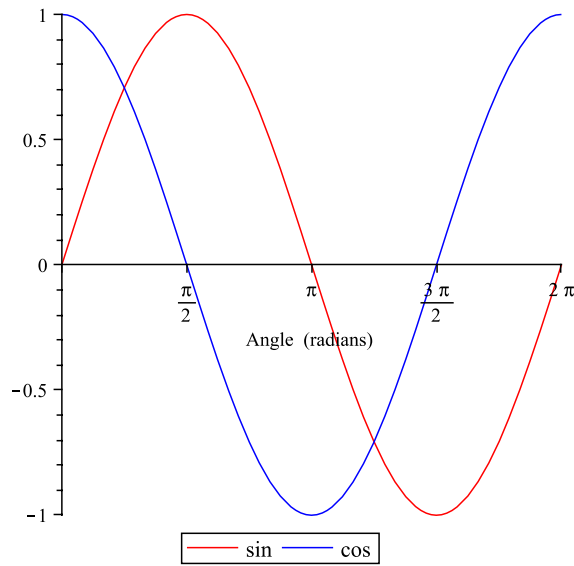
A comparison of sine and cosine curves

Regarding the labeling of the axes; we have previously had no trouble with the labeling of axes in our plots, and have been merrily labeling the axes with variable names for as long as we've been plotting functions. However, the **labels** option of the plot command allows us to label using arbitrary text that may even include spaces. This is something we would not have been able to do with our previous labeling methods (the reader is encouraged to verify this).

We may, if we wish, place arbitrary text anywhere on a plot using the **textplot** function from the plots package. For example, let's take the cubic $x^3 + x^2 - 5x - 5$, which we call p. Through simple analysis (which we leave as an exercise to the reader) we discover that p has zeroes at $x = 1$ and $x = \pm\sqrt{5}$. We also discover that p has turning points at $(1, -8)$ and $(-5/3, 40/27)$ which are a local minimum and maximum, respectively. We intend to plot this cubic, and label these interesting points.

First we should have a look at the curve. Notice that we need to put the caption in parentheses so that the entire mathematical expression becomes the caption, and *Maple* doesn't get confused with the two equal signs.

```
> with(plots):
  p := x → x³ + x² − 5x − 5
```
$$p := x \to x^3 + x^2 - 5x - 5$$

```
> pl := plot(p(x), −3.25..2.75, caption = (y = p(x)), labels = [x, y])
```
$$pl := PLOT(\ldots)$$

```
> pl
```


$$y = x^3 + x^2 - 5\,x - 5$$
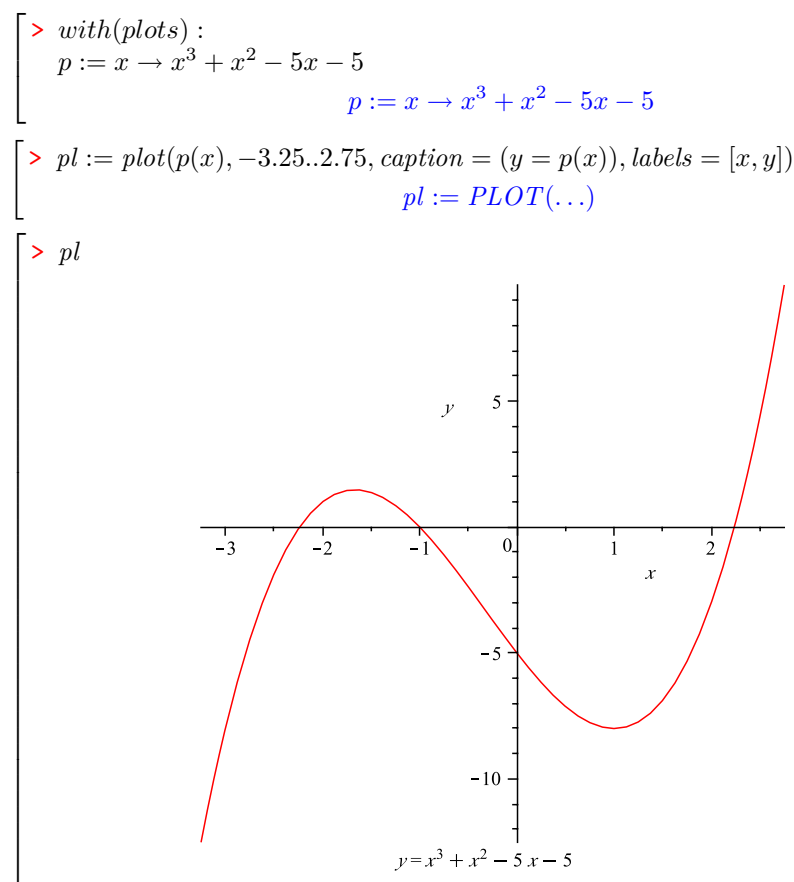
We start with the critical points. It should be noted here that the **textplot** function produces a plot, just as do any of our other plotting functions. If we want to put the text near the plot of our curve, we need to use the **display** function. It is for this reason that we assigned our first plot to the variable pl, above.

We want the label for the local maximum to sit immediately above the point in question, and the label for the local minimum to sit immediately below the point. We also place points on the plot in the appropriate places. Finally, we incorporate both

text and mathematics together for our labels using the **typeset** option, which may be used anywhere we have a label (including in our previous examples).

> $cpts := plot\left(\left[\left[-\dfrac{5}{3}, \dfrac{40}{27}\right], [1, -8]\right], style = point, color = black,\right.$
>
> $\left.symbol = solidcircle, symbolsize = 14\right);$
>
> $mx := textplot\left(\left[-\dfrac{5}{3}, \dfrac{40}{27}, typeset\left("Local Max: ", \left[-\dfrac{5}{3}, \dfrac{40}{27}\right]\right)\right], align = above\right);$
>
> $mn := textplot([1, -8, typeset("Local Min: ", [1, -8])], align = below)$

$$cpts := PLOT(\ldots)$$
$$mx := PLOT(\ldots)$$
$$mn := PLOT(\ldots)$$

> $critpoints := cpts, mx, mn$

$$critpoints := PLOT(\ldots), PLOT(\ldots), PLOT(\ldots)$$

> $display([pl, critpoints])$
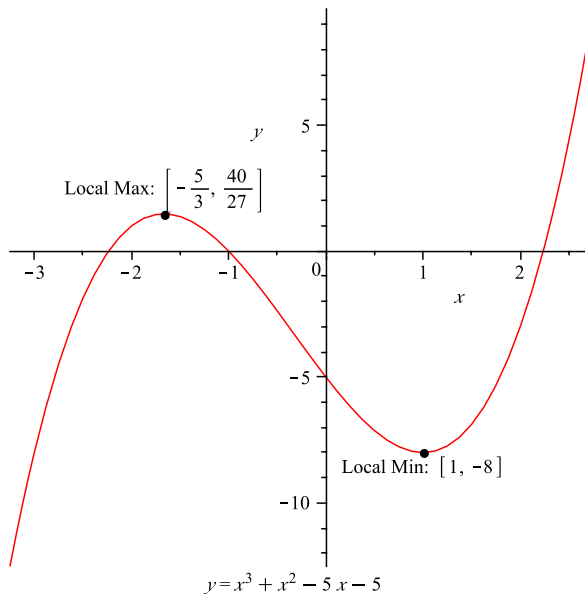
The syntax of the **textplot** function is quite straightforward. The function takes a list consisting of x-coordinate, y-coordinate, and text to be displayed, in that order. The function plots the text in an invisible box, and places the center of that box upon the point given by the x- and y-coordinates. We may modify the placement of the box using the *align* option. In the case above, we place the text above the local maximum, and below the local minimum.

We have also mixed text and mathematics together to form the label. The **typeset** option does this for us in a quite simple way. This option behaves much as a function, and concatenates its inputs. In the above example we provided only two inputs, the first one being text and the second one being mathematics. We may have as many parameters as we wish. The **typeset** option may be used anywhere that labels are being produced.

We now mark and label the zeroes of the function in a similar manner. We make the markings and labels for the zeroes blue so that they are more easily separable from the critical points.

> $zpts := plot([[-\operatorname{sqrt} 5, 0], [-1, 0], [\operatorname{sqrt}(5), 0]], style = point, color = blue,$
    $symbol = solidcircle, symbolsize = 14);$
  $z1 := textplot([-\operatorname{sqrt}(5), 0, -\operatorname{sqrt}(5)], align = [above, left]);$
  $z2 := textplot([\operatorname{sqrt}(5), 0, \operatorname{sqrt}(5)], align = [above, left]);$
  $z3 := textplot([-1, 0, -1], align = [above, right])$

$$zpts := PLOT(\ldots)$$
$$z1 := PLOT(\ldots)$$
$$z2 := PLOT(\ldots)$$
$$z3 := PLOT(\ldots)$$

> $zeroes := zpts, z1, z2, z3$

$$zeroes := PLOT(\ldots), PLOT(\ldots), PLOT(\ldots), PLOT(\ldots)$$

> $display([pl, zeroes])$

$$y = x^3 + x^2 - 5x - 5$$
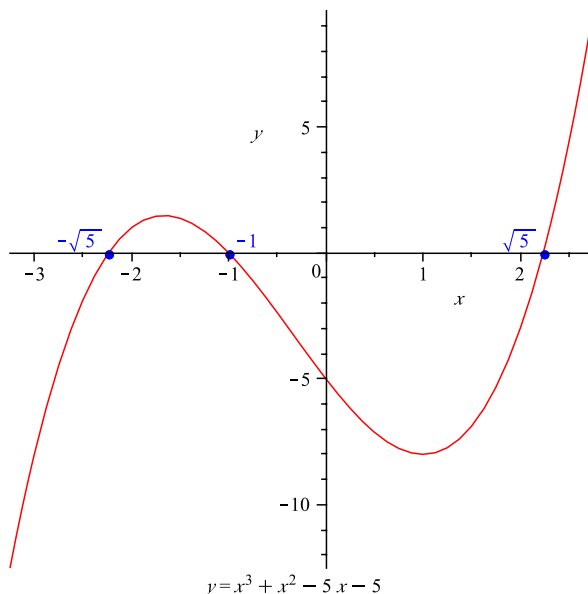
The labels for the zeroes are much simpler, and do not need to be typeset. We do, however, need to be more careful about the placement. Observe that previously, when we placed the text above (or below) a point, it was still centered horizontally. If we were to do that with our zeroes, labeling the curve would cut through the text. Instead we need to place the text above and to the right, or left of the point. The positioning information is put in a list in this case, however, the order is not important ($[above, left]$ is the same as $[left, above]$).

We may now put everything together, and see the final result.

> $display([pl, critpoints, zeroes])$

$$y = x^3 + x^2 - 5x - 5$$
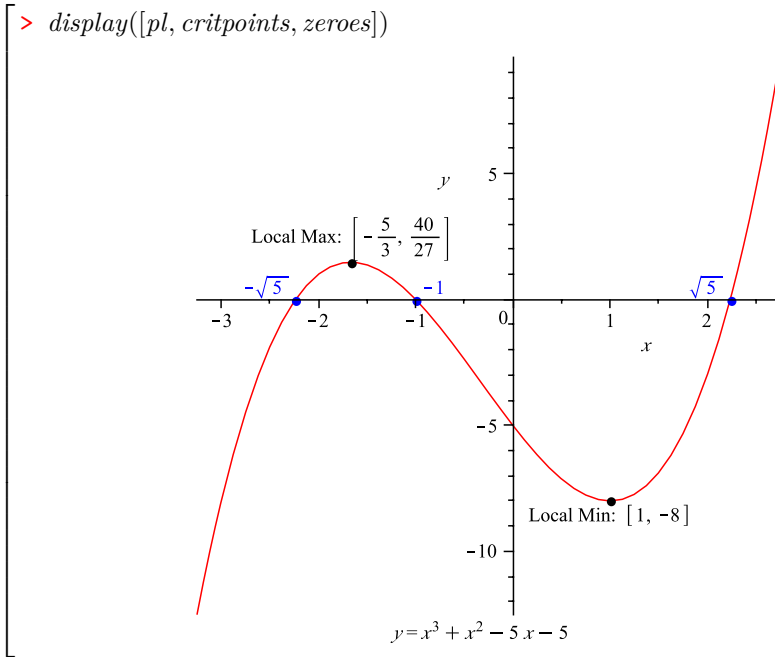
4.1.2 Polygons, Polyhedra, and so on

In addition to plotting functions (implicit or otherwise), there may be other things we wish to visualize. *Maple* provides a package named plottools that allows for some more unusual display functions.

We may visualize basic polygons and polyhedra. The **polygon** function takes a list of points which it then uses to create a polygon.

> $plots[display](polygon([[0,0],[3,1],[2,1]], color = blue))$
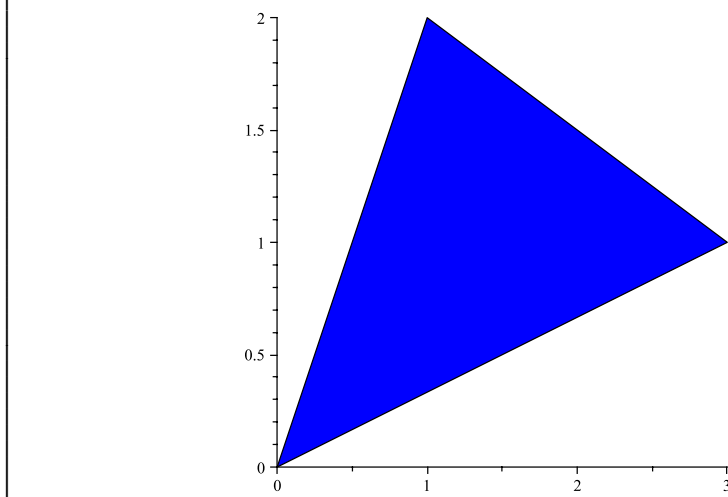
With some judicious use of the **seq** command we can create a regular 11-gon.

> $vertices := \left[seq\left(\left[\cos\left(\dfrac{2 \cdot \text{Pi} \cdot k}{11} \right), \sin\left(\dfrac{2 \cdot \text{Pi} \cdot k}{11} \right) \right], k = 1..11 \right) \right] :$

> $plots[display](polygon(vertices, color = blue))$
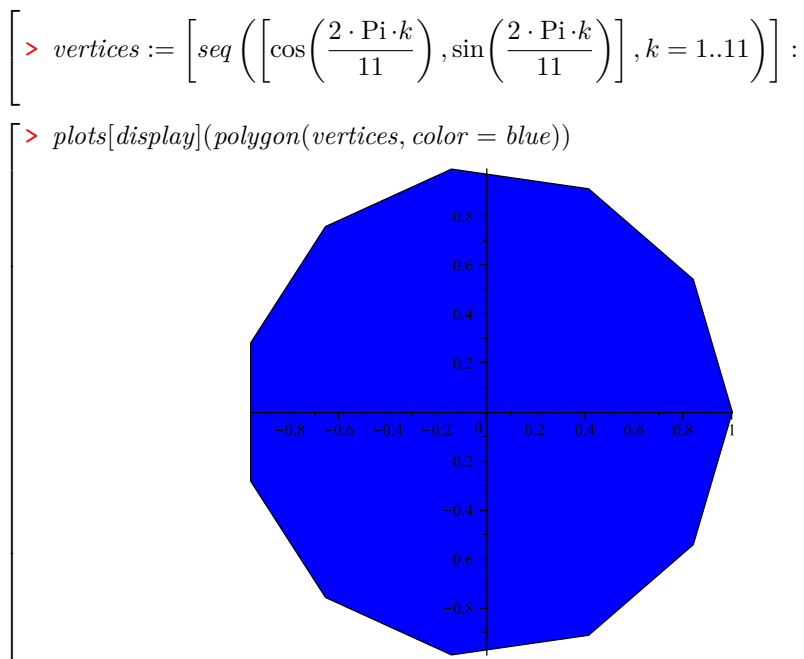
Note that the order of the vertices is important. The **polygon** functions draws the boundary of the polygon by drawing lines between the vertices in the order they appear (1st to 2nd then 2nd to 3rd and so on until the nth and then finally from the nth to the 1st). If the boundary lines intersect, then the interior changes appropriately.

> $plots[display](polygon([0, 0], [1, 0], [0, 1], [1, 1]), color = blue))$
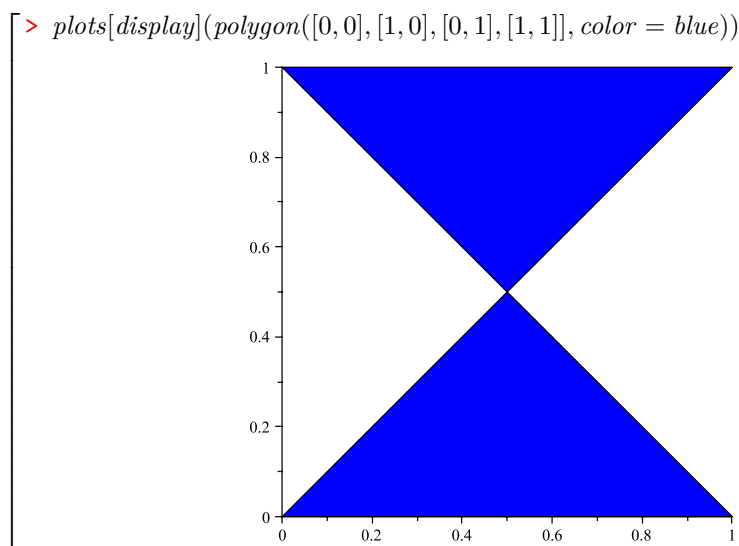
Note here that there is a very similar function in the plots package named **polygonplot**, as well as a three-dimensional variant **polygonplot3d**.

Getting back to our plottools, however, we have some functions to visualize the regular polyhedra. These functions take a center and an optional scale. The interested reader can consult the *Maple* help files for more information.

> $plots[display](icosahedron([0,0,0]))$
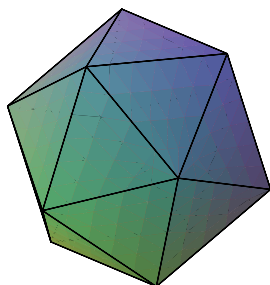
The plottools package also provides some shortcuts to some more common plots, for instance, **line** and **hyperbola**. Below we use the latter as a shortcut to plot the hyperbola

$$\frac{(x-2)^2}{1^2} - \frac{(y-2)^2}{1^2} = 1$$

> $plots[display](hyperbola([2,2],1,1,-3..3))$
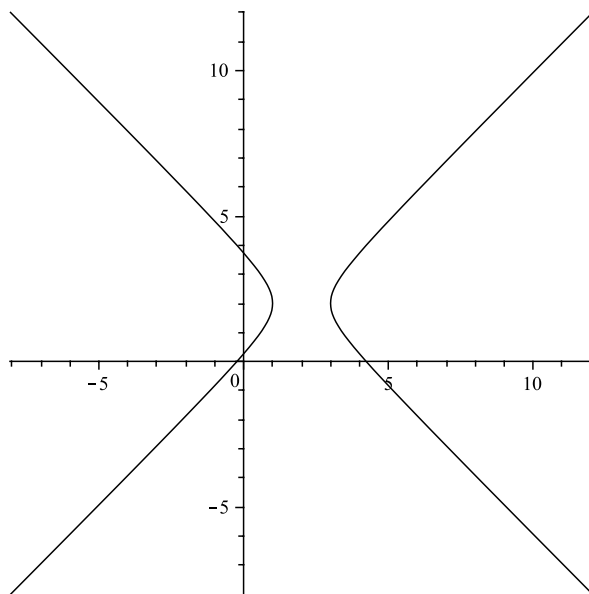
The interval $-3..3$ is not a simple interval of the x-axis, as should be clear from the plot. Instead it is used to form a pair of parameterized intervals, in a way best explored in the help files.

Finally the plottools package provides a small suite of functions for manipulating other plots. We may **project**, **rotate**, **scale**, **transform**, and several others. Each of these functions takes a $PLOT$ object of some sort, and performs computations on the points of the plot.

For instance, recall that to rotate a plot of the sin function in Section 3.3.2 we first created a vector with parameterized coordinates for the sin function which we then transformed using matrix multiplication and a rotation matrix. We also had to convert the vector back to a list that we could then use in the **plot** command. We now perform this same task (i.e., rotating a sin curve) using the **rotate** function from the plottools package.

> $S := plot(\sin(\text{theta}), \text{theta} = -2 \cdot \text{Pi} .. 2 \cdot \text{Pi})$
$$S := PLOT(\dots)$$

> $rotate\left(S, \dfrac{-\text{Pi}}{4}\right)$
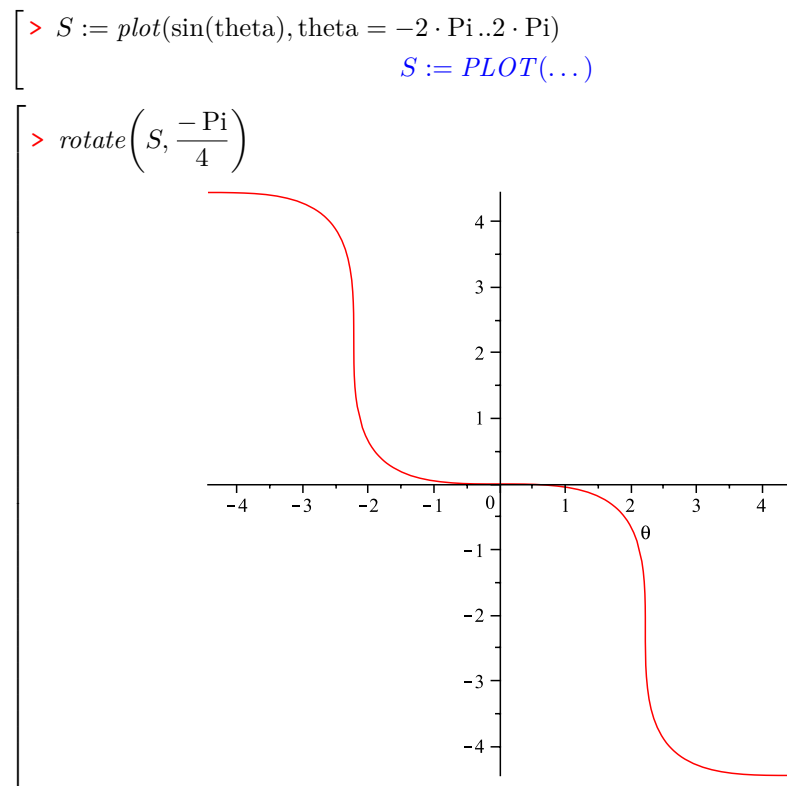
This is simpler than the method employed in Section 3.3.2, however, it should be noted that these functions are operating *numerically* upon the points calculated by the original **plot** command. Contrast this with our earlier approach which was working symbolically upon a parameterization of the curve. Remembering that the **plot** command samples points (quite intelligently) and then fills in the space between these points, it is possible that for highly irregular plots, that the sampling for the original plot may not be appropriate for the rotated plot (and similarly for the other functions that modify *PLOT* structures). As always, one should not be complacent, and should be aware of what one is doing, and attempting to do when using a CAS.

The functions provided by the plottools package can be a little esoteric. We have endeavored here to give some idea as to the "flavor" of the package, but more important, we have pointed out its existence to the reader. The interested reader, by this stage of the book, should be capable of independently using the help files to gain further knowledge of the contents of this package.

We close this section now with one final example of a function that manipulates a *PLOT* structure, and an esoteric one at that. This final example is chosen for no better reason than it looks "kinda cool" to the author. The function is the **stellate** function, which takes a three-dimensional[1] *POLYGONS* construct, and turns each polygon in the construction into a pyramid, thus creating something that looks a lot like a star (in most cases). We apply this to the icosahedron, because we have already seen it un-stellated earlier.

---

[1] Be careful, it is not entirely clear from the help page for this function that it requires three-dimensional polyhedra.

> $plots[display](stellate(icosahedron()))$
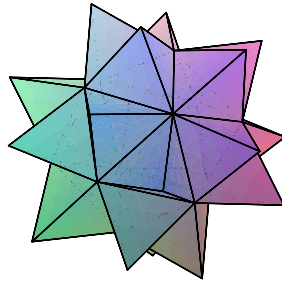
4.2 Geometry and Geometric Constructions

Maple, unsurprisingly, comes with a geometry package which is named, even more unsurprisingly, geometry. We show some geometric constructions using this package. All constructions given in this section are performed in $\mathbb{R}^2$. Again we remind the reader that these constructions are much better explored using dynamic geometry software.

The geometry package operates somewhat differently from the *Maple* we have become familiar with thus far. These differences are dealt with and explained in the course of our explorations in this section. However, it is expected that by now readers should be more than competent enough with *Maple* and its associated help file system to be able to make sense of this package (and others) on their own. As such we do not explain the functions in as much detail as we might have done previously.

4.2.1 Constructing a Circle Given Three Points

Given three points, we may find a unique circle that passes through all three of those points. This fact is related to the fact that the perpendicular bisector of any two chords on a circle will always intersect at the center of that circle.

Let us start with three points. In this case we take the points

$$(1,5), (4,2) \text{ and } \left(1 + \frac{9\sqrt{10}}{10}, 2 + \frac{3\sqrt{10}}{10}\right)$$

These are points that have been chosen because they lie on a familiar circle. We start with these points in *Maple*.

> $with(geometry):$

> $point(p1, 1, 5), point(p2, 2, 4), point\left(p3, 1 + \frac{9 \cdot \text{sqrt}(10)}{10}, 2 + \frac{3 \cdot \text{sqrt}(10)}{10}\right)$
$$p1, p2, p3$$
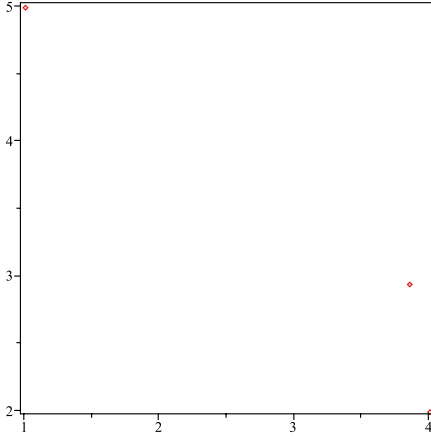
Here we see our first change from familiar *Maple*. The above *Maple* code creates three points named $p1$, $p2$, and $p3$. We did not, however, use the assignment operator as we would have normally. Instead the first parameter of the **point** function is the name of the point. All geometric elements in the geometry package are created in this way. It is worth noting here that this function does not like subscripts in the variable names.

We may obtain information about our various geometric objects using the **detail** function. Fortunately this works pretty much as we would expect.

> $detail(p1)$

name of the object	$p1$
form of the object	$point2d$
coordinates of the point	$[1, 5]$

And we may draw our geometric objects using the **draw** function. Note that for these geometry objects we *must* use **draw**, because **plot** will not work. Fortunately some of the options we may use are similar.

> $draw([p1, p2, p3], scaling = constrained)$

The way in which we construct the circle is to suppose that our three points lie on the surface of some circle. If this were the case, then we could create up to three chords of that circle by drawing line segments between the three points. We create these line segments using the **segment** command.

> $segment(s1, p1, p2), segment(s2, p2, p3)$

$$s1, s2$$

> $draw([p1, p2, p3, s1, s2], scaling = constrained)$
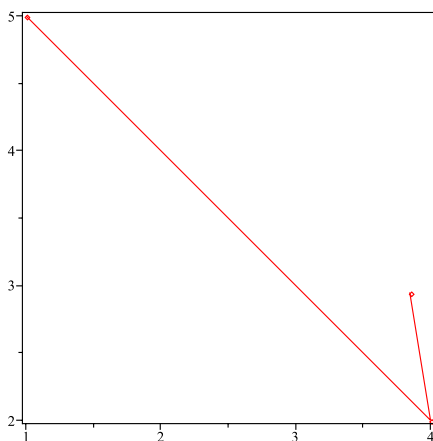
In order to find the center of our hypothetical circle, we need only two of these chords. For simplicity's sake, we chose the chords $p1\,p2$ and $p2\,p3$. Observe, however,

that whichever two chords we chose would always have had a point in common. This is
critical. We now construct the perpendicular bisector of each of these chords. This we
do with the **PerpenBisector** function.

> $PerpenBisector(pb1, p1, p2), PerpenBisector(pb2, p2, p3)$

$$pb1, pb2$$

Notice that the **PerpenBisector** function used the points, and not the segments.
We could have done completely without the segments, but because they help with the
visualization, we keep them. We should look at our construction now.

> $draw([p1, p2, p3, s1, s2, pb1, pb2], scaling = constrained)$
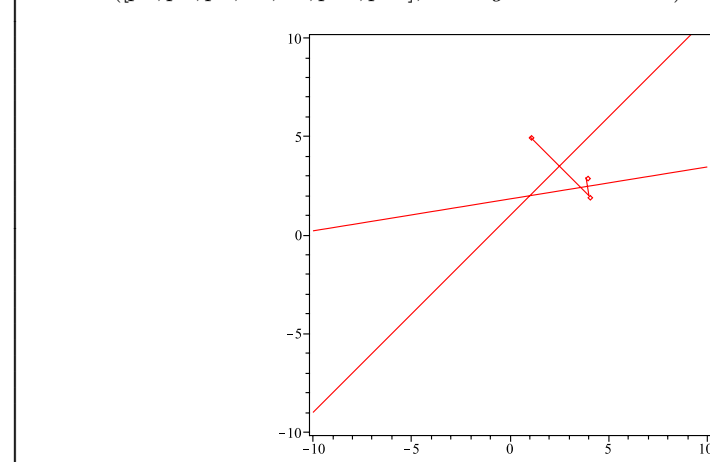
So now we have an intersection of two perpendicular bisectors drawn from chords
on a hypothetical circle. How do we know this circle even exists? One way would be
to try to draw the circle whose center is at the intersection of the two bisectors and
using one of the points as a radius. If we did this then maybe that circle would also
pass through the other three points in which case we would have our desired circle.
Doing this, however, would not guarantee that the same construction would work for a
different three points.

So instead, let's think about these bisector lines for a bit. If we take any point, p say,
on the bisector of, say, $p1\,p2$, then we know that point is equidistant from $p1$ and $p2$. If we
imagine the isosceles triangle $p\,p1\,p2$ we should be convinced of this fact. In particular,
then, the point of intersection between our two bisectors—let's be presumptions and
call it c—is equidistant to $p1$ and $p2$. By a similar argument, applied to the chord $p2\,p3$,
the intersection point c is also equidistant from the points $p2$ and $p3$. So it must be
the case that all three of our points are equidistant from c, which is the same as saying
that they lie on a circle whose center is c and whose radius is the distance between the
points and c.

We now know that not only is our hypothetical circle actually a real circle, but that
it makes no difference which three points we choose. This is because as we observed
above, there must always be a common point that our two chords share. The proof that
this circle is unique is left as an exercise for the reader.

Let's now go and draw our circle. We find the intersection of the two bisectors using
the **intersection** function and we create our circle using the **circle** function. We find
the distance between two points by using the **distance** function, which behaves like
functions we are more used to dealing with in *Maple*.

> $intersection(c, pb1, pb2)$

$$c$$

> $r := distance(c, p1)$

$$r := \sqrt{9}$$

> $circle(circ1, [c, r])$

$$circ1$$

> $draw([circ1, c, p1, p2, p3, s1, s2, pb1, pb2], scaling = constrained)$
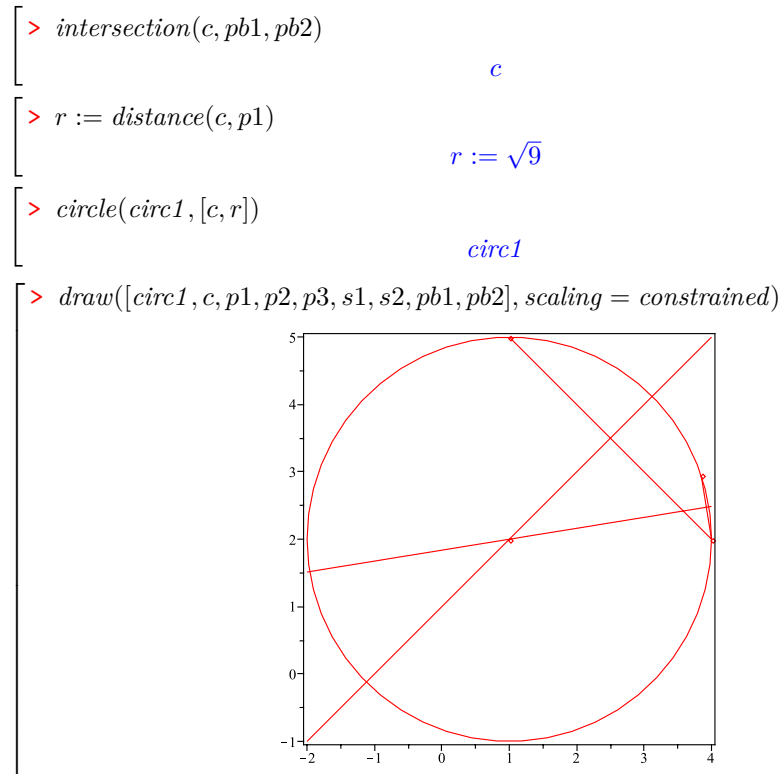
As is usually the case with these things, we could simply have given the initial three points to the **circle** function, and it would have produced the correct circle for us. Having done so would not have led us to the proof of correctness of the above technique, however. Nonetheless, it is useful to know, and readers are encouraged to investigate the other forms of the **circle** function and, indeed, the geometry package in its entirety on their own.

> $circle(circ2, [p1, p2, p3])$

$$circ2$$

> $draw([circ2, p1, p2, p3])$
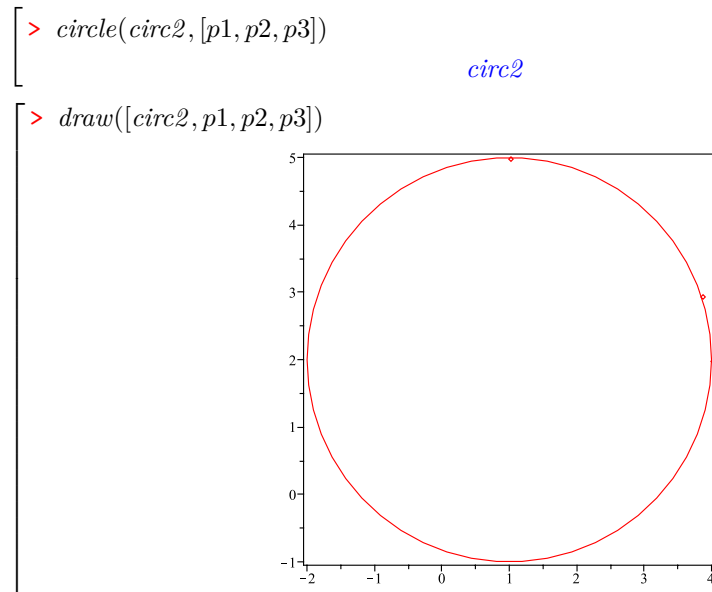
4.2.2 Constructing the Orthocenter of a Triangle

We perform one more construction. Inasmuch as we're now a little familiar with the working of the geometry package, we can proceed a little more quickly than usual.

   We start with a triangle, the points of which have been chosen arbitrarily by the author for no particular reason other than they fit fairly neatly within a square

> $point(p1, 1, 2), point(p1, -1, -2), point(p1, 3, -1)$

$$p1, p2, p3$$

> $segment(s1, p1, p2), segment(s2, p2, p3), segment(s3, p3, p1)$

$$s1, s2, s3$$

> $draw([s1, s2, s3])$
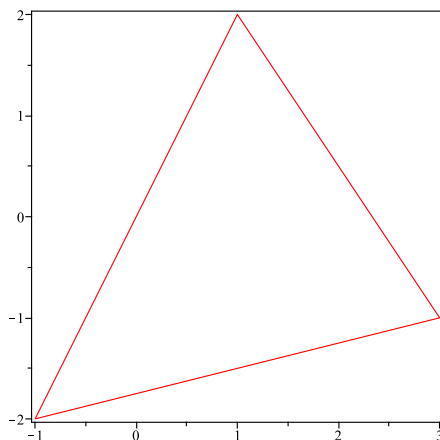
   To construct the orthocenter of the triangle, we draw the line perpendicular to each side and passing though that side's opposite corner. In this case, we take the lines perpendicular to $s1$ passing though $p3$, the line perpendicular to $s2$ passing though $p1$, and the line perpendicular to $s3$ passing through $p2$.

   These lines, as we will shortly see, intersect at a single point. In fact, performing this construction for any given triangle will always produce a single intersection point. We call this point the orthocenter. We give no proof that the lines always intersect at a single point, nor does our construction shed any light on why this might be. We produce but a single example.

   In order to create our intersection lines, we take a quick look at the **PerpendicularLine** function. Unfortunately, it turns out that this function takes a line and a point as its parameters, but so far we have only points and segments. Time to make some lines.

> $line(l1, [p1, p2]), line(l2, [p2, p3]), line(s3, [p3, p1])$

$$l1, l2, l3$$

   Our observations above are simply modified by replacing the segment with the appropriate line ($s1$ with $l1$, $s2$ with $l2$, and so on).

> $PerpendicularLine(pl1, p3, l1), PerpendicularLine(pl2, p1, l2),$
> $PerpendicularLine(pl1, p2, l3)$

$$pl1, pl1, pl1$$

> $draw([s1, s2, s3, pl1, pl2, pl3])$
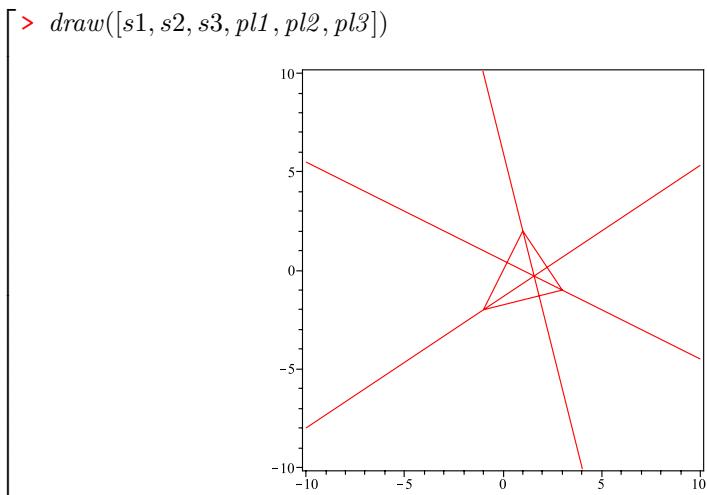
The three perpendiculars appear to meet at a point. In fact they do. In order to see this a little better we construct the segments of the perpendicular lines that lie within the triangle.

> $intersection(i1, l1, pl1), intersection(i2, l2, pl2), intersection(i3, l3, pl3)$

$$i1, i2, i3$$

> $segment(s4, i1, p3), segment(s5, i2, p1), segment(s6, i3, p2)$

$$s4, s5, s6$$

> $draw([s1, s2, s3, s4, s5, s6])$
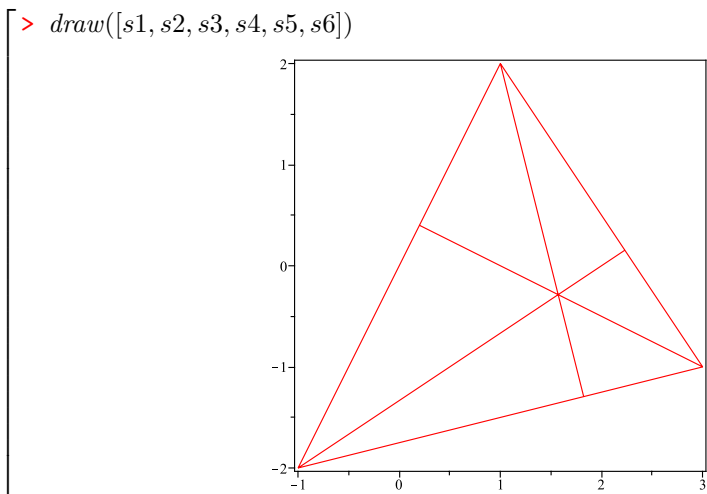
We now calculate the intersection of two of the perpendicular lines, and see that the point, which we call c, also lies on the third line.

> $intersection(c, pl1, pl2)$

$$c$$

> $IsOnLine(c, pl3)$

$$true$$

Appendix A
Sample Quizzes

A.1 Number Theory

Short Answer Section

The following questions are worth 1 point each. Answer the questions in your *Maple* file.

1. What is e^{13} evaluated to 23 significant figures?
2. Factorize $x^{12} + 3\,x^{10} - 23\,x^8 - 51\,x^6 + 94\,x^4 + 120\,x^2$.
3. Convert $\dfrac{e^{2x} - 1}{xe^{2x} + x}$ into an expression involving trig functions.
4. What is the partial fraction decomposition of the rational polynomial.

$$\frac{x^7 + 4\,x^6 + 25\,x^4 - 20\,x^3 + 53\,x^2 - 42\,x + 33}{x^8 - 2\,x^7 + 7\,x^6 - 12\,x^5 + 18\,x^4 - 24\,x^3 + 20\,x^2 - 16\,x + 8}$$

5. What are the first 20 terms of the sequence $\left\{ \dfrac{1}{k(k+1)} \right\}_{k=1}^{\infty}$?
6. Evaluate $\displaystyle\sum_{k=1}^{\infty} \frac{1}{k(k+1)}$.
7. Evaluate $\displaystyle\prod_{k=1}^{\infty} \left(1 - \frac{1}{2x^2} \right)$.
8. The following variable names are all names that *Maple* treats as Greek letters. Which of them are protected?

$$\text{delta, Delta, gamma, Gamma, GAMMA, pi, Pi, PI, zeta, Zeta, ZETA}$$

Long Answer Section

The following questions are worth 3 points each. Points are given for working. Answer the questions in your *Maple* file.

9. Let $a_n = 2n - 1$ and $s_n = n^2$ and define the sequences

$$A := \{a_n\}_{n=1}^{\infty} \quad S := \{s_n\}_{n=1}^{\infty}$$

It should be clear then that $A = \{1, 3, 5, \dots\}$ and $S = \{1, 4, 9, 16, \dots\}$.

a. Calculate the first 20 terms of the sequence

$$\{s_{n+1} - s_n\}_{n=1}^{\infty}$$

What is $s_{n+1} - s_n$?

b. Calculate the first 20 terms of the sequence

$$\left\{ \sum_{k=1}^{n} a_k \right\}_{n=1}^{\infty}$$

What is $\sum_{k=1}^{n} a_k$?

10. Recall that $\sum k^{-1}$ diverges. It may be shown that $\sum k^{-(1+\epsilon)}$ converges for any $\epsilon > 0$. For this question we let $\epsilon = \frac{1}{100}$

a. Evaluate the series $\sum_{k=1}^{\infty} 1/k^{\frac{101}{100}}$, and obtain a decimal approximation.

b. Calculate decimal approximations of the partial sums

$$\sum_{k=1}^{N} \frac{1}{k^{101/100}} \text{ for } N = 10, 100, 1000, 10000, 100000$$

and measure how much time each takes to calculate.

Notice that this series converges very slowly.

11. Let $\{f_n\}$ be the Fibonacci-like sequence defined by

$$f_n := f_{n-1} + f_{n-2} \quad f_1 = -2, f_2 = 3$$

a. Write a *Maple* function (using arrow notation or a procedure as you choose) to calculate the terms of this sequence.

b. What are the first 10 terms of this sequence?

c. What is the largest number in this sequence less than 1,000,000, and what is its index?

12. Let s be the first-order nonlinear recurrence relation defined by

$$s_n = n\, s_{n-1}^2$$

a. Let $s_0 = C$ and calculate the first 5 or so terms of the recurrence.

b. Solve the recurrence.

c. Verify the solution for at least 20 terms of the sequence, and in general if you can.

A.2 Calculus

Short Answer Section

The following questions are worth 1 point each. Answer the questions in your *Maple* file.

1. What is the limit of $\dfrac{x + \sin x}{\pi\, x}$ at $x \to \infty$?

2. Find $\displaystyle\lim_{x \to 0^+} \dfrac{-\cosh x}{x}$.

3. What is the derivative of $\dfrac{\cos x}{x}$?

4. What is the slope of the tangent to the curve $y = \dfrac{\cos x}{x}$ at $x = \frac{1}{3}\pi$?

5. Evaluate $\displaystyle\int_0^1 \log x\, dx$.

6. Find a function whose derivative is $\tanh x$.

7. Find the first partial derivatives of $z = x^2 - y^2$.

8. How many critical points does $z = x^2 - y^2$ have, and what kind of critical points are they?

Long Answer Section

The following questions are worth 3 points each. Points are given for working. Answer the questions in your *Maple* file.

9. A length of wire 10 meters long is cut in two. One of the pieces is bent into a square, the other into an equilateral triangle. Let x be the length of wire that is bent into the square (meaning that $10 - x$ is the length of wire bent into the triangle). Let A_s be the area of the square and let A_t be the area of the triangle.

   a. Define A to be the formula for the total area of the two shapes (i.e., $A :=$ $A_s + A_t$). Plot A.
   b. How much wire should be used for the square to maximize the total area?
   c. How much wire should be used for the square to minimize the total area?

10. The Airy functions $\mathrm{Ai}(z), \mathrm{Bi}(z)$ are the two independent solutions to the differential equation

$$y'' - z\,y = 0 \qquad\qquad\qquad (A.1)$$

   a. Solve the differential equation (A.1), and verify the solution.
   b. Plot the Airy functions together on the same axes. Make sure to show good detail of what the functions are doing.
   c. Find and plot a third solution, other than $y = \mathrm{Ai}(z)$ and $y = \mathrm{Bi}(z)$, to equation (A.1).

11. Use solids of revolution to verify the following volumes.

   a. The volume of a sphere with radius r ($4/3\pi r^3$)
   b. The volume of a cone with height h and radius r ($1/3\pi r^2 h$)

12. Consider the surface $z = \sin(x)\cos(y)$.

   a. Plot the surface z.
   b. Find a general formula, or formulae, for the critical points.
   c. Which critical points are maxima, which are minima, and which are saddle points?

A.3 Linear Algebra

The following questions are worth 1 point each. Answer the questions in your *Maple* file.

1. Calculate the vector $\pi \cdot (8, 1, 5, 1, 9) + e \cdot (1, 2, 5, 6, 6)$.

2. Calculate the matrix product $\begin{bmatrix} 3 & 1 & 6 \\ 0 & 0 & 7 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 0 \\ 5 & 4 \end{bmatrix}$.

3. Calculate the dot product of the vectors $(3, 3, 2, 3, 3, 3)$ and $(3, 2, 3, 3, 2, 1)$.

4. Find the angle between the vectors $(3, 3, 2, 3, 3, 3)$ and $(3, 2, 3, 3, 2, 1)$.

5. Find a vector perpendicular to the vectors $(5, 5, 3)$ and $(5, 5, 5)$.

6. Create the 9×10 matrix M whose entries $m_{i,j} = 17\,i\,j$.

7. Find the elementary matrix that will add k multiplied by row 7 to row 9 of a 10×10 matrix.

8. How many solutions are there to the vector equation $M \cdot x = 0$ where

$$
M := \begin{bmatrix}
9 & 4 & 7 & 8 & 5 \\
4 & 7 & 3 & 4 & 8 \\
7 & 7 & 0 & 5 & 7 \\
7 & 4 & 6 & 4 & 4 \\
7 & 6 & 2 & 2 & 6
\end{bmatrix}
$$

Long Answer Section

The following questions are worth 3 points each. Points are given for working. Answer the questions in your *Maple* file.

9. This question refers to the following simultaneous equations.

$$y + 3z = 2$$
$$8x + 2y + 2z = 3$$
$$3x + 3y + 5z = 2$$

   a. Solve the simultaneous equations. How many solutions are there?
   b. Plot the three surfaces in such a way that clearly shows the solution.

10. This question refers to the following three matrices

$$\begin{bmatrix} 1\,1\,7\,2 \\ 2\,2\,0\,9 \\ 7\,7\,2\,5 \\ 0\,0\,1\,5 \end{bmatrix}, \begin{bmatrix} 0\,4\,1\,4 \\ 0\,2\,1\,1 \\ 3\,3\,4\,4 \\ 6\,4\,9\,5 \end{bmatrix}, \begin{bmatrix} 0\,1\,1\,1 \\ 6\,1\,0\,0 \\ 2\,2\,6\,6 \\ 9\,4\,4\,4 \end{bmatrix}$$

   a. Which of the matrices may be expressed as a product of elementary matrices?
   b. For the matrices that may be expressed as a product of elementary matrices, find the sequence of elementary matrices whose product is that matrix. (Equivalently, you may find the sequence of row operations performed on the identity matrix.)

11. Let A be the matrix below, and let $p, q \in \mathbb{R}$.

$$A := \begin{bmatrix} p & q & 1-p-q \\ 1-p-q & p & q \\ q & 1-p-q & p \end{bmatrix}$$

   a. Create A in *Maple* as a function of p and q.
   b. By examining various numerical cases where $p > 0$, $q > 0$ and $1 - p - q > 0$, conjecture the behavior of the matrix A^n as $n \to \infty$.

12. This question refers to the following set of matrices

$$\begin{bmatrix} 7\,2\,9 \\ 1\,2\,6 \\ 2\,4\,8 \end{bmatrix}, \begin{bmatrix} 8\,4\,7 \\ 6\,7\,4 \\ 9\,5\,7 \end{bmatrix}, \begin{bmatrix} 0\,6\,0 \\ 2\,4\,0 \\ 5\,2\,5 \end{bmatrix}, \begin{bmatrix} 0\,7\,4 \\ 2\,7\,9 \\ 9\,3\,7 \end{bmatrix}$$
$$\begin{bmatrix} 0\,0\,9 \\ 3\,7\,1 \\ 0\,0\,3 \end{bmatrix}, \begin{bmatrix} 1\,0\,1 \\ 7\,3\,9 \\ 0\,0\,4 \end{bmatrix}, \begin{bmatrix} 7\,9\,1 \\ 1\,4\,5 \\ 1\,6\,3 \end{bmatrix}, \begin{bmatrix} 9\,7\,4 \\ 2\,1\,4 \\ 5\,9\,0 \end{bmatrix}, \begin{bmatrix} 5\,0\,8 \\ 7\,5\,4 \\ 9\,4\,8 \end{bmatrix}$$

   a. Do the matrices form a basis for $M_3(\mathbb{R})$? Justify your answer.
   b. Find the coefficients of a linear combination of these matrices for an arbitrary 3×3 matrix.

References

1. ANTON, H., AND RORRES, C. *Elementary Linear Algebra*, 7th ed. John Wiley and Sons Inc, Brisbane, 1994.
2. BAILEY, D., BORWEIN, J., CALKIN, N., GIRGENSOHN, R., LUKE, R., AND MOLL, V. *Experimental Mathematics in Action*, 1st ed. AK Peters, Wellesley, MA, 2007.
3. BORWEIN, J., AND BAILEY, D. *Mathematics by Experiment: Plausible Reasoning in the 21st Century*, 2nd ed. AK Peters, Wellesley, MA, 2008.
4. BORWEIN, J., BAILEY, D., AND GIRGENSOHN, R. *Experimentation in Mathematics: Computational Paths to Discovery*, 1st ed. AK Peters, Natick, MA, 2004.
5. BORWEIN, J., AND DEVLIN, K. *The Computer as Crucible: An Introduction to Experimental Mathematics.* AK Peters, Wellesley, MA, 2009.
6. GANDER, W., AND HREBÍCEK, J. *Solving Problems in Scientific Computing Using Maple and MATLAB*, 4th ed. Springer, New York, 2008.
7. GARVAN, F. *The Maple 5 Primer Rel 4.* Prentice Hall, Englewood Cliffs, NJ, 1997.
8. GARVAN, F. *The Maple Book.* Chapman and Hall/CRC, Boca Raton, FL, 2001.
9. HECK, A. *Introduction to Maple*, 3rd ed. Springer, New York, 2003.
10. KLIMEK, G., AND KLIMEK, M. *Discovering Curves and Surfaces with Maple.* Springer, New York, 1997.
11. ROVENSKI, V. Y. *Geometry of Curves and Surfaces with MAPLE.* Springer, New York, 2000.
12. STEWART, J. *Calculus*, 6th ed. Brooks/Cole, 2008.
13. TROTT, M. *The Mathematica Guidebooks*, 3rd ed. Springer, New York, 2004–2006.
14. WAGON, S. *Mathematica in Action*, 2nd ed. Springer, New York, 1999.

Index

211