CHAPTER 1

Running Maple

Computer Systems

What Computer System Are You Using?

Maple software runs on almost every major computer system including mainframes and desktop systems. Maple can also be set up to run across a network and even between systems. Almost all implementations of Maple are graphically-based systems where you can both type on a keyboard and use a mouse to navigate a window (e.g., desktop systems running MS Windows, the MacOS or a Linux-based system.) The Maple commands we discuss in this guide will work on all of these implementations.

> **Note:** Text-based implementations of Maple are also available. These are often found on mainframe systems, and are usually of interest only to people who are doing remote logins, or to advanced users who want the performance advantage that comes from using a text-based system.

Starting the Software

You should follow the instructions that came with the Maple software to install it on your computer system. Once you've completed the installation, you're ready to explore Maple.

Starting Maple obviously depends on the system you are using.

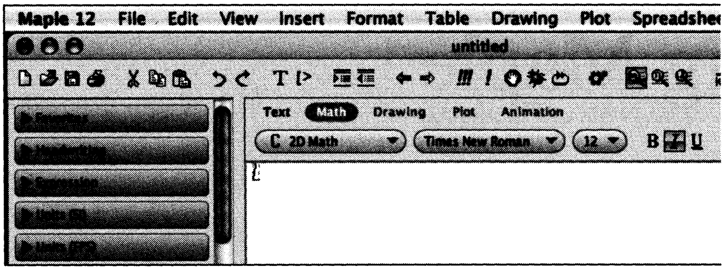
- On most PCs and Macintosh computers, you will typically find the icon of the Maple application in a window. Click (or double-click) on the icon.
- On Linux, or other command-line operating systems, you will usually enter the command **`xmaple`** or **`maple`** (or a local equivalent, depending on how the software has been installed and how your system is configured).

> **Note:** If you run Maple over a network, you may need to check with your system manager for the starting procedure.

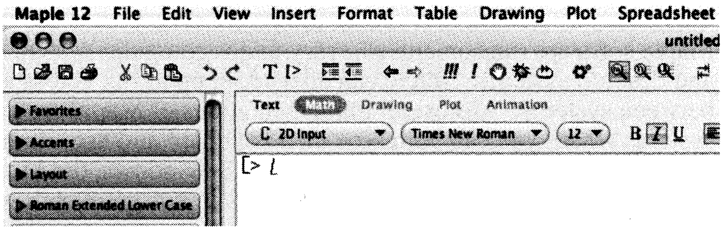
If this is the first time you start Maple, you will be presented with the choice to open a document or a worksheet. In the document mode, you can mix Maple's computation with your text to create an interactive document. In the worksheet mode, Maple's input has to be entered line by line in groups. (Please see the graphics of these two modes on the next page.)

Document is the default mode, but we find that beginners are more comfortable with the worksheet mode. Nevertheless, the Maple input we discussed in this book will work on both modes. If you want to switch from one mode to the other, check the Q & A section at the end of this chapter.

When Maple is launched, a new window will be opened with the cursor flashing, awaiting your input.



Document Mode

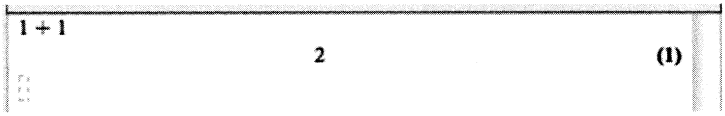


Worksheet Mode

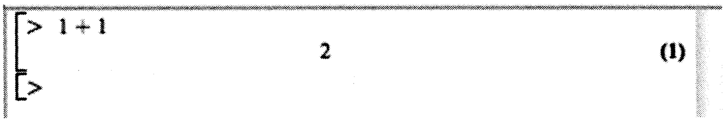
Notice that in the worksheet mode, Maple gave you the prompt symbol ">" automatically. You don't enter it yourself.

Input and Output

Maple is interactive software. For almost every entry you make, Maple will provide a direct response. Once you launch Maple and see the flashing cursor, you can type the three characters [1 + 1] and then press the evaluation key (either the [*enter*] or [*return*] key, depending on your computer system). Maple will give you the response "2" on a new line, and your window should look something like this:

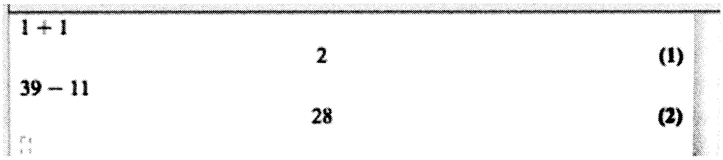


Document Mode

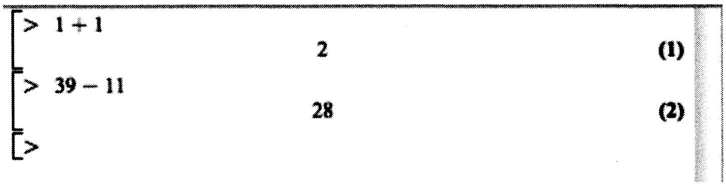


Worksheet Mode

Now you enter the characters [39 - 11] and then press the evaluation key again ([*enter*] or [*return*]) and you'll see the result of 28:



Document Mode



Worksheet Mode

Throughout the rest of this manual, we will not show you windows. Instead, we'll show all of our input, with the response in plain text. Thus, our sequence above would be listed as:

Keystrokes: **1+1** [*enter*]

1 + 1

2

Keystrokes: **39-11** [*enter*]

39 − 11

28

Quit

When you've had enough and want to exit Maple, you can simply choosing the **Quit Maple** item in the **Maple** menu on the top of the screen. On a text based system, simply execute

```
quit
```

2-D Math Input

By default, Maple opens in 2-D Math input mode in a document. This allows you to enter the input in the same way as you would write it mathematically. As a result, you can use Maple with a minimum of syntax and mix mathematical notation with text for explanation. For example, to compute $\frac{2}{3}+1$, as you begin to type

2/

the input on the screen will automatically turn to the fraction form as shown below with a flashing cursor at the denominator waiting for your input.

$$\frac{\mathbf{2}}{}$$

You can now enter **3** in the denominator. Press [→] (the right arrow key) on your keyboard and the cursor returns to the central position. Now type **+1**. You will see

$$\frac{\mathbf{2}}{\mathbf{3}}+\mathbf{1}$$

Press the evaluation key [*enter*] and Maple responds with

$$\frac{5}{3}$$

As another example, to enter $x^2 + 3x$ in 2-D Math mode, you will type **x^2** then press [→], and finish the typing with **+3*x** to see the input:

$$x^2 + 3 \cdot x$$

(When you type ^, the cursor will move to the superscript position. Press [→] to make the cursor move out of the superscript position.)

A Quick Tour

For the rest of this chapter, we'll show you some of Maple's capabilities. We present these examples only to whet your appetite. You can follow along at your computer by typing (exactly!) what we show below. In later chapters, we'll give you a more complete explanation of how to use these commands and how to use palettes to graphically produce the commands.

Note: When you input the following commands in Maple, make sure that:

- You use upper- and lower-case characters exactly as we do. Maple is very "case sensitive." If you use the wrong capitalization, Maple won't understand what you mean.
- Use exactly the type of brackets we show. There are three types of brackets: **[** *square brackets* **]**, **(** *parentheses* **)** and **{** *curly braces* **}**. Each has its own meaning in Maple. If you use the wrong one, Maple will be confused.
- You can continue input from one line to another in Maple by pressing either [*shift - return*] or [*return*] (depending on your system). For our examples, we recommend that you break lines exactly as we show in the text.
- You must always press the evaluation key [*enter*] to see the output.
- Your output might appear slightly different from ours in some of the following examples. We explain why in Q & A at the end of this chapter.

Calculator

Maple does all the work of a hand-held electronic calculator. You can enter numerical expressions and Maple will do the arithmetic:

Keystrokes: **235.567*441.235/623.45** [*enter*]

$$\frac{235.567 \cdot 441.235}{623.45}$$

$$166.7181092$$

Keystrokes: **sin(0.3)** [*enter*]

$$\sin(0.3)$$

$$0.2955202067$$

But Maple can go much further. Try this factorial computation!

Keystrokes: **`289!`** [*enter*]

289!

```
2079866075306145164348895732262527092227125189083652864966524223 17\
405760295930638776430109826354519132675660433931363055910963871453\
772379754931444766652739192303201763588723618347593740385428725846\
122572271049818916876323493243976023302916666394540247449307010665\
731331903556896427962603583291932028351318888786128689538489086713\
005449989591695585446014805881310771610743578696897019623882572957\
311572603371048376255338230572538584588079078669943174850854858995\
580594914244562856410918607028520410684463210865874698240000000000\
000000000000000000000000000000000000000000000000000000000000
```

Your calculator cannot do this!

Solving Equations

Maple can solve complicated equations and even systems of equations in many variables. For example, the equations $2x+5y=37$ and $x-3y=21$ have a simultaneous solution:

Keystrokes: **`solve({2*x+5*y=37,x-3*y=21},{x,y})`** [*enter*]

$$solve(\{2\cdot x+5\cdot y=37,\ x-3\cdot y=21\},\ \{x,y\})$$

$$\left\{x=\frac{216}{11},\ y=\frac{-5}{11}\right\}$$

Maple can also find solutions to equations numerically. For example, the equation $x=\cos(x)$ has a solution very close to $x=0.75$. We can find it with:

Keystrokes: **`fsolve(x=cos(x),x)`** [*enter*]

$$fsolve(x=\cos(x),\ x)$$

$$0.7390851332$$

You will learn more about solving equations in Chapter 6.

Algebra

Maple is very good at algebra. It can work with polynomials:

Keystrokes: **`expand((x-2)^2`** [→] **`*(x+5)^3`** [→] **`)`** [*enter*]

$$expand((x-2)^2\cdot(x+5)^3)$$

$$x^5+11x^4+19x^3-115x^2-200x+500$$

> **Note:** Use [→] after the **2** to move out of the superscript. The [→] will also move you out of the denominator of a fraction or the radical of a square root.

Keystrokes: **factor(x^5** [→] **+11*x^4** [→] **+19*x^3** [→] **-115*x^2** [→] **-200*x+500)** [enter]

$$factor(x^5 + 11 \cdot x^4 + 19 \cdot x^3 - 115 \cdot x^2 - 200 \cdot x + 500)$$

$$(x-2)^2(x+5)^3$$

Are you impressed? Maple also knows standard trigonometric identities such as $\sin^2(x) + \cos^2(x) = 1$:

Keystrokes: **simplify(sin(x)^2** [→] **+cos(x)^2** [→] **)** [enter]

$$simplify\left(\sin(x)^2 + \cos(x)^2\right)$$

$$1$$

> Note that $\sin(x)^2$ means $(\sin(x))^2$ and is different from $\sin(x^2)$.

That one was easy, but you probably forgot that $\sec^2(x) - \tan^2(x) = 1$:

$$simplify\left(\sec(x)^2 - \tan(x)^2\right)$$

$$1$$

You will learn more about doing algebra in Maple in Chapter 5.

Calculus

Maple even knows a lot about calculus! We can find the derivative of the function $f(x) = x/(1+x^2)$.

Keystrokes: **diff(x/(1+x^2** [→] **)** [→] **,x)** [enter]

$$diff\left(\frac{x}{(1+x^2)}, x\right)$$

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

A complicated integral such as $\int \frac{1}{1+x^3}\, dx$ is handled rather easily.

Keystrokes: **int(1/(1+x^3** [→] **)** [→] **,x)** [enter]

$$int\left(\frac{1}{1+x^3}, x\right)$$

$$-\frac{1}{6}\ln(x^2 - x + 1) + \frac{1}{3}\sqrt{3}\arctan\left(\frac{1}{3}(2x-1)\sqrt{3}\right) + \frac{1}{3}\ln(1+x)$$

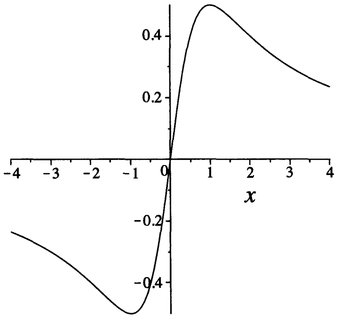
In Chapter 2, we will show you how to use palettes for differentiation and integration. Chapters 12 through 15 demonstrate many of the Calculus capabilities of Maple. The Appendix A also contains information on Maple commands useful for learning Calculus.

Graphing in the Plane

Maple does everything a standard graphing calculator does and does it better. For example, to see the graph of the function $f(x) = x/(1+x^2)$ over the interval $-4 \le x \le 4$, you can use:

Keystroke: **`plot(x/1+x^2`** [→] [→] **`,x=-4..4)`** [*enter*]

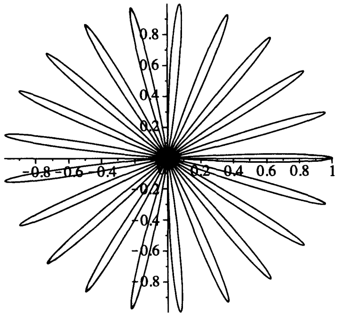
$$plot\left(\frac{x}{1+x^2}, x = -4..4\right)$$



We can see a "daisy" with:

Keystroke: **`plot([cos(21*t)*cos(t),cos(21*t)*sin(t),t=0..2*Pi])`** [*enter*]

$$plot([\cos(21 \cdot t) \cdot \cos(t),\ \cos(21 \cdot t) \cdot \sin(t),\ t = 0..2 \cdot \mathrm{Pi}])$$



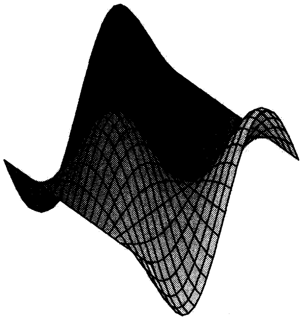
Chapters 9, 10, and 11 contain the details of the two-dimensional plotting capabilities of Maple.

Plotting in Space

Maple does a wonderful job with three-dimensional graphics. Let us show you two examples.

Keystrokes: **`plot3d(sin(x)*cos(y),x=0..2*Pi, y=0..2*Pi)`** [*enter*]

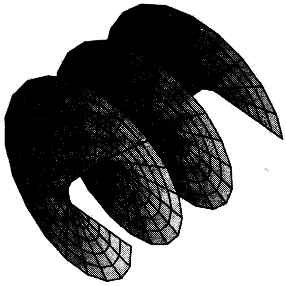
$$plot3d(\sin(x) \cdot \cos(y),\ x = 0..2 \cdot \mathrm{Pi},\ y = 0..2 \cdot \mathrm{Pi})$$



Keystrokes: **plot3d([t,r*cos(t),r*sin(t)],r=0..1,t=0..6*Pi,** *shift - return*

grid=[8,60]) *enter*

$$plot3d([t,\ r\cdot\cos(t),\ r\cdot\sin(t)],\ r=0..1,\ t=0..6\cdot\mathrm{Pi},\ grid=[8,60])$$



Chapters 16 and 20 contain most of the information you need to create three-dimensional pictures.

Changing the 3-D View

If you click and drag the mouse button with the cursor/arrow at any point inside a 3-D picture, the picture will rotate according to how you move the mouse. This allows you to see the 3-D picture from any viewpoint.

Chapter 20 contains more information on how you can utilize Maple's interface to more easily control your view of 3-D graphics.

Programming and Simulation

Maple has its own programming language. You can use it to write code just as you would in Java, C or BASIC. Here's a simple routine to simulate the flipping of a coin several times and return the number of heads observed:

Keystrokes:

```
coinflips := proc( howmany ) [shift - return]
 local heads; [shift - return]
 heads := 0; [shift - return]
 from 1 to howmany do [shift - return]
 if RandomTools[Generate](choose([head,tail])) [shift - return]
   =head then [shift - return]
     heads := heads + 1; [shift - return]
```

```
end if; [shift - return]
end do; [shift - return]
printf("%a heads seen in %a flips", [shift - return]
     heads, howmany ); [shift - return]
end proc; [enter]
```

You can use this routine:

Keystrokes: `coinflips( 100 );` [enter]

```
55 heads seen in 100 flips.
```

Keystrokes: `coinflips( 1000 );` [enter]

```
491 heads seen in 1000 flips.
```

Chapter 27 introduces the basic features of the Maple programming language.

More Examples

The "More Examples" sections of this guide present examples involving more mathematics. Students of mathematics, science, and engineering may find these of interest.

Here's one such example. Not too many people know about the Bessel functions. But if you're learning physics, you might want to know what Maple has available for you in Bessel functions.

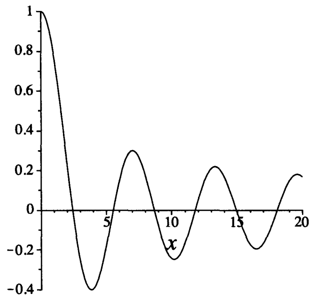
Special Functions

The Bessel function of order 0, $J_0(x)$, is a solution to the differential equation:

$$x^2y'' + xy' + x^2y = 0 .$$

You can see its graph with:

Keystrokes: `plot(BesselJ(0,x),x=0..20)` [enter]



The smallest, positive zero of $J_0(x)$ is at approximately $x = 2.40483$:

Keystrokes: `fsolve(BesselJ(0,x)=0,x=1)` [enter]

2.404825558

Useful Tips

If you want to try the examples in this book, you don't have to enter the Maple input one by one by hand. We've made available copies of the Maple documents and worksheets used to create the inputs you see in this Guide. After downloading them, you can copy and paste into your own Maple document and begin experimenting. Please visit the authors' websites mentioned in the Preface.

Troubleshooting Q & A

Most chapters of this guide end with a Troubleshooting Q & A section, where we answer some common questions we think you will have. But there's not too much you can ask yet, except:

Question... Maple is giving me unexpected evaluations when I use trigonometric or other named functions in 2-D Math. What am I doing wrong?

Answer... A common mistake when entering 2-D Math input is to carelessly leave a space between the function name and the left parenthesis. In 2-D Math input a space is implied multiplication, but it may be hard to see. For example:

Keystrokes: **`sin(1.2)`** *enter*

sin(1.2)

0.9320390860

But if you type a space, then the answer will be very wrong.

Keystrokes: **`sin`** *space* **`(1.2)`** *enter*

sin (1.2)

1.2 sin

Question... Suppose I make a mistake in the input, or I do not finish the entry, but I hit *enter* anyway. How will Maple respond?

Answer... Most likely you will get an error message from Maple. Don't worry. You can simply move the cursor back to your input, correct the mistake or continue the typing to finish the input.

Question... The output I get when testing some of the examples in this text does not look exactly like what's shown here. What's going on?

Answer... All the outputs we shown in the text are from the document mode of Maple 12. If you use a different version of Maple, or use the worksheet mode instead, you may get an output in a different format or arrangement. However despite this inconsistency, the actual computation and the final result should not be affected.

Question... If I set up the document mode as the default, how can I start Maple with worksheet mode instead, or vice versa?

Answer... No matter whether you are in a document mode or worksheet mode, you can always open a new document or worksheet by choosing **File** → **New** → **Worksheet Mode** or **Document Mode** from the menu bar.

You can also reset the default style of user interface that Maple starts by default. To do this, click on the **Tools** menu and select the **Options ...** entry from the list. A popup window will appear. Now, click on the **Interface** tab and look for the drop-down list labeled by "**Default format for new worksheets:**". Select **worksheet** or **document**, then click **Apply Globally**.

Question... What do I do now?

Answer... Turn the page and start learning about Maple!

CHAPTER 2

Maple Input Modes

2-D Input Mode

2-D Math Input By default, when you open Maple document or worksheet, it is ready for 2-D input. As we have seen in the examples in the first chapter, 2-D input allows us to enter and automatically format a mathematical expression in the same way as we would write it by hand. Here is a short summary of some of the special keys used for 2-D input:

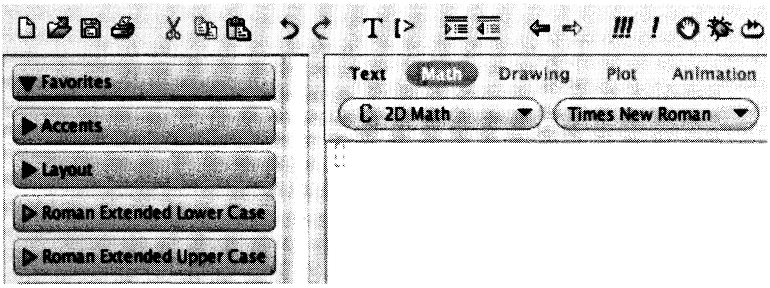
Key(s)	*What it does*	*Example*
^	Exponent	*Keystrokes*: **`3^2`** The input appears as: 3^2
/	fraction	*Keystrokes*: **`3/1+x`** The input appears as: $\frac{3}{1+x}$
*	Multiplication	*Keystrokes*: **`x*y`** The input appears as: $x \cdot y$
[→] (Right arrow key)	Navigating expressions	*Keystrokes*: **`3^2`** [→] **`x`** The input appears as: $3^2 x$

> **Note**: Although one can also use the [space] (space bar) in place of * for multiplication, we do not recommend it because Maple doesn't automatically insert a multiplication symbol. It is difficult for a novice to pick up the difference between the inputs *x y* (which means *x* times *y*) and *xy* (which means the name *xy*).

■ **Example**. To enter $\frac{x^2}{1+x}+5$ with 2-D Math input, we type:

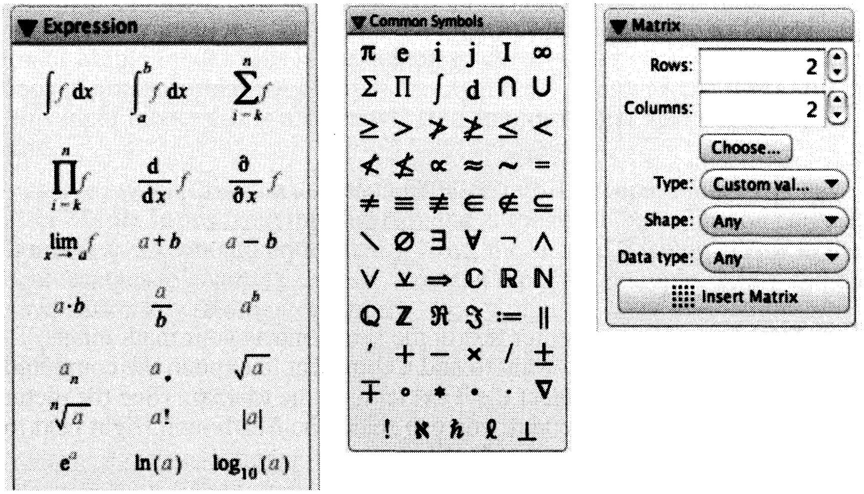
Keystrokes: **`x^2`** [→] **`/1+x`** [→] **`+5`**

Palettes Input In addition to directly entering an expression using only keystrokes, you can use pre-configured palettes to format and organize the terms of your input. By default, more than 20 palette docks are displayed in the left pane of the Maple document.



(If you don't see the palettes, you can load them from the main menu by selecting **View → Palettes → Show All Palettes.**)

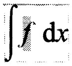
The most frequently used palettes are the **Expression**, **Matrix**, and **Common Symbols**. When you click on a palette dock, all of the templates, symbols, and other items inside that particular palette will be displayed. You can insert a template into your session by pointing the cursor at the icon and clicking the left mouse button.



By using templates from the **Expression** Palette, we can easily ask Maple to integrate, differentiate, find limits and summations, and so on. We will quickly illustrate how they work with the following example.

■ **Example.** To evaluate $\int \frac{1}{2+\pi x^2}\,dx$ using palettes:

- Open the **Expression** palette by clicking on its button. Click on the indefinite integral template ($\boxed{\int f\,\mathrm{d}x}$) in the **Expression** palette. A new expression is created in your document. Notice that *f* is highlighted in the expression.



- Click on the fraction template ($\boxed{\frac{a}{b}}$)in the **Expression** palette. Now you see:

$$\int \frac{f}{b}\,dx$$

- Type **1**, then press the *tab key* to move to the denominator. (Every time you press the *tab key*, you move from one box in the expression to another box.) Then type **2 +** , and then click on the multiplication template ($\boxed{a \cdot b}$). Now you see the expression:

$$\int \frac{1}{2+a\cdot b}\,dx$$

- Open the **Common Symbols** palette. Click on the $\boxed{\pi}$ symbol in that palette. Then press the *tab key*.

$$\int \frac{1}{2+\pi\cdot b}\,dx$$

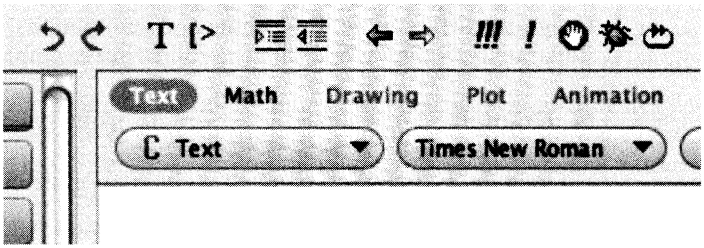
- Click on the power template ($\boxed{a^b}$) in the **Expression** palette. Type **x**, press the *tab* key, then type **2**, and press the *tab* key again. Now you see the completed expression more like what you would see in a textbook. Hit $\boxed{enter}$ to see Maple's output.

$$\int \frac{1}{2+\pi\cdot x^2}\,dx$$

$$\frac{1}{2}\,\frac{\sqrt{2}\arctan\left(\frac{1}{2}\sqrt{\pi}\;x\;\sqrt{2}\right)}{\sqrt{\pi}}$$

Adding Text to Math Input

You can enter text in the same line as your math input. This is particularly helpful when you want to add a comment to explain the computation. In order to do so, you have to select the **Text** icon on the toolbar. (See the picture below.) After you finish typing the text, you can select the **Math** icon (right next to the **Text** icon) on the tool bar to continue the Math input.



■ **Example**. We want to comment that the area is found by computing the integral $\int_0^2 x\,dx$. The procedure is:

- Select the **Text** icon on the toolbar. Type **`The area can be computed with`**.

- Select the **Math** icon on the toolbar. From the **Expression** palette, select the definite integral template ($\int_a^b f\,dx$).
- Notice that the placeholder for *a* is highlighted. Type **0**, then press the *tab* key, type **2**, press the *tab* key again, and then type **x**.
- Press the evaluation key (**Return/Enter**). We have:

The area can be computed with $\int_0^2 x\,dx$

$$2$$

1-D Input Mode

1-D Math Mode

You can also enter input in 1-D Math mode, which Maple refers to as Maple Notation or Maple Input. This is the default mode for older versions of Maple, where no automatic formatting of input was available. Many Maple users still prefer to use this input mode; this is particularly true amongst users who write Maple programs.

If you want to switch from the default 2-D mode into 1-D input mode, you have to first select **Maple Input** from the **Insert** menu at the menu bar. (Or you can use **command-M** as a keyboard equivalent.) Now, the slanted flashing cursor in the document changes to a vertical cursor. You can type, for example, **2/3;** and then hit the evaluation key (**Return** or **Enter**). You will get

```
2/3;
```

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

Notice that with 1-D Math input,

- input appears in red and in a fixed-width font;
- Maple does not automatically format your input with a numerator and denominator as you type; and
- all 1-D input must end with a semicolon.

Comparison of 1-D and 2-D Input Modes

You can intermix the use of 1-D and 2-D math modes throughout a Maple document. Each mode has its advantages. Nevertheless, some users will be strong advocates of using one mode over the other. So it helps to compare.

Comparing Keystrokes

■ **Example**. We want to enter the expression $\frac{1}{1+x^2}+5$ into Maple.

- Using the 2-D Math input

Keystrokes: **`1/1+x^2`** [→] [→] **`+5`**

(The first [→] brings the cursor out of the exponent position, and the second [→] brings us out of the denominator.)

- Using the 1-D Math input

Keystrokes: **`1/(1+x^2) + 5;`**

Notice that in the 1-D Math input, we have to use parentheses to group the terms in the denominator together. If you enter instead

Keystrokes: **`1/1+x^2 + 5;`**

Maple will misinterpret your input as $\frac{1}{1}+x^2+5$, and return $6+x^2$.

Converting Between Input Modes

You can convert between 1-D and 2-D Math by highlighting the input to be converted and choosing **Convert To** from the **Format** menu, then selecting between the **1-D Math input** option or the **2-D Math input** option.

For example, you can convert 2-D Math inputs,

$$\frac{2}{3}+1$$

and

$$x^2+3\cdot x$$

into 1-D Math inputs

```
2/3+1;

x^2+3*x;
```

respectively.

Which Mode To Choose?

Now, let us give you a simple, direct comparison between the 2-D Math input and 1-D input.

	2-D Input	1-D Input
Advantages	• Easier for beginners because it matches how mathematics is typically presented in textbooks. • Input errors are easier to identify because you see the expression as you type.	• Looks like calculator syntax. • Easier for users to replicate because every keystroke is recorded.

Disadvantages	• Subtle differences in the output can lead to confusion. For example, the 2-D inputs "*a b*" and "*ab*" have very different interpretations. • The expressions "sin (Pi/2)" and "sin(Pi/2)" also illustrate another error that is difficult to detect. • Some mistakes cannot be undone without completely restarting the expression.	• Matching parentheses is essential. Maple's built-in parenthesis matching is helpful. • All groupings must be done with parentheses; **[***square brackets***]**, **{***curly braces***}**, and **(***parentheses***)** have different uses in Maple. (See Chapter 6.) • When entering a complicated math expression, users may sometimes fail to group some of the terms correctly or leave off a matching parenthesis and hence get an error message.
Differences in keystrokes	• [→] is used to move the cursor within the expression. • [space] means (implied) multiplication, but we discourage its use.	• Use [→] to move cursor to the right. • [space] has no special meaning. • Must end an input with : or ; • Parentheses are required to correctly group terms.

Which Input Method Will This Book Represent?

Each input method has its own advantage and disadvantage. In the first few chapters of this book, we will show you both the 2-D and 1-D inputs, but later on we use only the traditional 1-D Maple input. Throughout this manual, when you see, for example,

Keystrokes: **3^2** [→] **/5**

$$\frac{3^2}{5}$$

The first line shows the keystrokes used to enter the expression. This is followed by the actual 2-D Math input display. On the other hand, if you see

```
3^2/5;
```

that is the input for 1-D Math.

Readers should feel free to use either method, but be ready to convert to 1-D if Maple gives an unexpected result. In the other direction, converting to 2-D is often useful to check whether the Maple command is asking for what the user hopes to compute.

Useful Tips

In 2-D input, do not use spaces for implicit multiplication. Instead, use * explicitly.

💡💡 When working with palettes, the *tab* key moves through the input areas of the current template.

💡 Make a habit of using **:** or **;** to end a Maple input in both 2-D and 1-D input. This will avoid unnecessary warnings and error messages during input.

Troubleshooting Q & A

Question... How come [→] (the right-arrow-key) does not work in moving the cursor from one field to the other in 2-D input?

Answer... In regular 2-D Math input, you use [→] to move the cursor about the various terms in the expression. To move the cursor about the terms of a palette, you use the *tab key*. In both modes, you can use the mouse to move the cursor directly to the location within the expression where you want to insert (or change) the input.

Question... When I convert an expression to 1-D, Maple adds a semicolon to my expression. Why?

Answer... 1-D Math was designed for text based systems and set up to have no ambiguities. A semicolon indicates both the end of an expression and the instruction that Maple should evaluate the expression and report back on its value.

CHAPTER 3

Calculator Features

Simple Arithmetic

Basic Arithmetic Operations

In this chapter, we will show you how Maple can work as a calculator. We begin with the basic arithmetic operations used in Maple:

Command	*What It Does*
`+, -`	Add, Subtract
`*, /`	Multiply, Divide
`^`	Raise to a power (exponentiation)

Here is an example to evaluate $\frac{23}{5} - \frac{3}{5} + 5(2^3)$ with either 2-D or 1-D input:

Keystrokes: **23/5** [→] **-3/5** [→] **+5*2^3** [→]

44

> **Note:** [→] is the right-arrow-key on the keyboard, we can use it to get out of the denominator and the exponent as discussed in Chapter 2.

```
23/5-3/5+5*2^3;
```

44

We can also use parentheses to group terms together. For example, the expression $(3+4)\left(\frac{4-8}{5}\right)$ can be entered in 2-D or 1-D Math mode with:

Keystrokes: **(3+4)*((4-8)/5** [→] **)**

```
(3+4)*((4-8)/5);
```

$$-\frac{28}{5}$$

> **Note:** Use **(** *parentheses* **)** to group terms in expressions. Do not use **[** *square brackets* **]** or **{** *curly braces* **}** -- they mean something different. (See Chapter 6.)

Precedence

In 1-D Math input, Maple follows the laws of precedence of multiplication over addition and so on, just as you do by hand. For example,

```
9/6*22+5;
```

38

actually computes $(\frac{9}{6} \times 22) + 5$. A common mistake is to think of the input as either $\frac{9}{6\times 22} + 5$ or $\frac{9}{6} \times (22 + 5)$. In 2-D Math input, Maple follows the visual grouping. This means you see exactly the expression that Maple actually computes.

Comments

You can add a comment to any expression by starting the comment with the pound sign **#**. For example in 2-D Math mode,

Keystrokes: **27*3 # This multiplies 27 and 3.**

$27 \cdot 3$ ***# This multiplies 27 and 3.***

81

Or in 1-D mode, we enter this as the following (and don't forget to add the semicolon).

```
27*3; # This multiplies 27 and 3.
```

81

Maple will ignore the phrase **This multiplies 27 and 3**. It is for your own reference.

We'll use comment lines throughout this guide to write short reminders about what's being emphasized in certain examples.

Previous Results and % Syntax

As a convenience, Maple lets you use the percentage sign, %, to stand for "the last result obtained." In this way, you can avoid retyping output when you want to work with it in your next input.

For example,

Keystrokes: **3200*12**

$3200 \cdot 12$

38400

Keystrokes: **% - 6500 # Here % refers to 38400**

$\% - 6500$ ***# Here % refers to 38400.***

31900

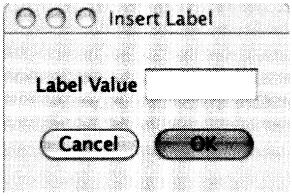
In 1-D Math input mode, we can enter:

```
(1000 + %)^2;            # Here % refers to 31900.
```

1082410000

You can use two percentage signs %% for the "second-last result obtained," and three percentage signs %%% for the third-last. However, this convention does not work with more than 3 percentage signs.

You can also refer to specific output by its equation label. (The equation label is the number in parentheses at the right-hand edge of each output.) To do that you can type **CTRL+L** (or **Command+ L** for Mac). An **Insert Label** dialog is displayed (see the picture below). Enter the label number and click OK.



Output Styles

Numeric Output

Maple gives you an exact (symbolic) value for almost every numeric expression:

`(3+9)*(4-8)/1247*67;` # for 1-D mode, or

$$\frac{(3+9)\cdot(4-8)}{1247}\cdot 67 \quad \textit{\# for 2-D mode}$$

$$-\frac{3216}{1247}$$

You can force Maple to give you an answer that looks like the decimal answer you'd get on a calculator by using **`evalf`** (for floating point evaluation) with parentheses around an expression:

`evalf((3+9)*(4-8)/1247*67);` #for 1-D mode, or

$$evalf\left(\frac{(3+9)\cdot(4-8)}{1247}\cdot 67\right) \quad \textit{\#for 2-D mode}$$

$$-2.578989575$$

By default Maple shows floating-point answers with 10 digits (including those to the left of the decimal point). You can see more digits in the answer – say 40 – with:

`evalf((3+9)*(4-8)/1247*67, 40);` # for 1-D mode, or

$$evalf\left(\frac{(3+9)\cdot(4-8)}{1247}\cdot 67, 40\right) \quad \textit{\# for 2-D mode}$$

$$-2.578989574979951884522854851643945469126$$

Results like these are called **approximate numeric values** in Maple.

Scientific Notation

Maple uses a modified standard scientific notation to display results when the numbers either get very large or very small:

`evalf(1234567890);` # for both 1-D and 2-D modes

$$1.234567890\ 10^9$$

```
0.0000003492836;              # for both 1-D and 2-D modes
```

$$3.492836\ 10^{-7}$$

The spaces in the output above represent (implied) multiplication.

Built-in Constants and Functions

Built-in Constants

The mathematical constants used most often are already built into Maple. You can either use the **Common Symbols** Palette (see Chapter 2), or type them in directly. Be careful using upper-case and lower-case characters when you use these constants.

Constant	*Value*	*Explanation*	Maple
π	3.1415926...	Ratio of a circle's circumference to its diameter	**`Pi`**
e	2.71828...	Natural exponential	**`exp(1)`**
i	$i=\sqrt{-1}$	Imaginary unit, a solution to x^2+1=0	**`I`**
∞	∞	(Positive) infinity	**`infinity`**

For example, to see the value of π to 45 significant digits, we use:

```
evalf(Pi, 45);               # for both 1-D mode and 2-D mode
```

$$3.14159265358979323846264338327950288419716940$$

To compute the numerical value of $\pi^4 - 5e$, we type:

```
evalf(Pi^4 -5*exp(1));       # for 1-D mode, or
```

$\boldsymbol{evalf\left(\pi^4 - 5\cdot \exp(1)\right)}$ **# for 2-D mode**

$$83.81768194$$

Built-in Functions

Maple has many built-in functions. Here are the functions that you will probably use the most. (Some of these can be found in the **Expression** Palette.)

Function(s)	*Sample(s)*	Maple *Name(s)*
Natural logarithm	$\ln(x)$	**`ln(x)`**
Logarithm to base *a*	$\log_a x$	**`log[a](x)`**
Exponential	e^x	**`exp(x)`**
Absolute value	$\lvert x\rvert$	**`abs(x)`**
Square root	$\sqrt{x}$	**`sqrt(x)`**

Trigonometric	$\sin(x)$, $\cos(x)$, ...	**sin(x), cos(x), tan(x), cot(x), sec(x), csc(x)**
Inverse trigonometric	$\sin^{-1}(x)$, $\cos^{-1}(x)$, ...	**arcsin(x), arccos(x), arctan(x), arccot(x)**, etc.
Hyperbolic	$\sinh(x)$, $\cosh(x)$, ...	**sinh(x), cosh(x), tanh(x), coth(x), sech(x), csch(x)**
Inverse hyperbolic	$\sinh^{-1}(x)$, $\cosh^{-1}(x)$, ...	**arcsinh(x), arccosh(x), arctanh(x), arccoth(x)**, etc.

For example,

```
sin(Pi);
```

$$0$$

```
evalf(sin(180));        # 180 radians, NOT 180 degrees.
```

$$-.8011526357$$

```
arctan(1);
```

$$\frac{1}{4}\pi$$

```
exp(ln(exp(1)));
```

e

Notes:

(1) Maple uses radian measure for all angles. To convert from degrees to radians, multiply by **Pi/180**.

(2) Maple refers to e^1 as **e** in the output. Unfortunately, you cannot use **e** (bold face letter e) when you enter input. You have to write **exp(1)** for input.

(3) A common mistake for the beginner is to enter **e^x** for the exponential function e^x, it has to be **exp(x)** instead.

Error Messages

If you make a mistake in your input, Maple *displays an error message*. In 2-D Math input, Maple will also put a red box around the part of the output where the mistake was detected. In 1-D Math input, Maple places the cursor on the position where the error was detected. Note that the actual error can be elsewhere in the expression.

Common Mistakes

The most common mistakes for beginners usually involve mismatching (or omitting) parentheses and forgetting to write the multiplication symbol, as you see in the following examples:

```
3·(4 − 5))+6 # too many right parentheses
Error, unable to match delimiters
                                        3·(4 − 5))+6 # too many right parentheses

sin 3  # it should be sin(3)
Error, missing operation
                                        sin 3 # it should be sin(3)

5 + 2^x 3  # it should be 5 + 2^x·3
Error, missing operation
                                        5 + 2^x 3 # it should be 5 + 2^x·3
```

More Examples

Approximate Numbers and Exactness

Working with **approximate numeric values** is just like working with values on a hand-held calculator. These numbers sometimes lose precision as values get rounded off in arithmetic.

■ **Example**. Maple makes a big distinction between the exact number π and a numerical approximation for it. For example, **`sin(517*Pi)`** is an exact quantity with an exact answer:

`sin(517*Pi);` #for 1-D mode, or

sin(517 · Pi) ***#for 2-D mode***

$$0$$

If you use a numerical approximation for 517π, you don't get an exact zero:

`sin(evalf(517*Pi));` #for 1-D mode, or

sin(*evalf*(517 · Pi)) ***#for 2-D mode***

$$-9.407689571\ 10^{-8}$$

This is very close to zero (it is, after all, –0.00000009407689571) and it's probably acceptable for the work you will be doing. But it's not exact.

Useful Tips

Don't forget to type the asterisk (`*`) when multiplying terms. This is the most common mistake made by beginners. For example, you might incorrectly enter **`cos(0)23`** instead of **`cos(0)*23`** and get an error message. However, *in 2-D Math Input mode,* if a number comes before a function, this will be interpreted as multiplication. But, if the same input is entered using 1-D Math Input, Maple considers this to be an error.

For example, in 2-D Math Input mode, **`23cos(0)`** is interpret as 23*cos(0), but in 1-D mode **`23cos(0);`** produce an error message that states "`missing operator or `;``." Because of this confusion, it is better to use * explicitly for multiplication.

Never try to multiply terms together in Maple using only `(` *parentheses* `)`. For example, in mathematical writing you can write (cos(Pi))(4) to denote $\cos(\pi) \times 4$. However, Maple has a different interpretation.

(cos(Pi))(4)

$$-1$$

Use parentheses in expressions to clarify what you mean. This helps avoid mistakes. For example, you might think that **`3^2*x`** means 3^{2x}, but it doesn't! It is actually $(3^2)x$ because the square is done before the multiplication of **`2`** and **`x`**. To get 3^{2x}, you should write **`3^(2*x)`**.

Many of the Maple built-in functions (e.g., **`sin`**, **`cos`**, **`ln`**) are understood to be defined for *complex* arguments. As a result, you may sometimes get an unexpected result that involves a complex number.

Use the ditto operator (`%`) sparingly. We recommend using the `%` symbol only for short sequences of one or two calculations that you don't expect to repeat later. In general, it is better to use equation labels when you need to refer to a previous result,

Avoid using **`exp(1)`** together with ^ to describe an exponential function. For example, the expression e^{2x} is better written as **`exp(2*x)`**, rather than **`exp(1)^(2*x)`**. (The latter expression has to be simplified before Maple recognizes it as being the same as **`exp(2*x)`**.)

Troubleshooting Q & A

Question... When I tried to evaluate a built-in function, Maple gave an error message, "`Error, unexpected number.`" What should I check?

Answer... Make sure that you included parentheses when using Maple functions. For example, a common mistake for beginners is to type:

```
sin 2;
Error, unexpected number
```

The correct input should be **`sin(2);`**.

Question... When I entered **`sin(2)`**, Maple just gave me the same thing back again. It didn't evaluate it. Why?

```
sin(2);
```

sin(2)

Answer... Maple always gives you an exact answer. When you write sin(2) in mathematics, you don't try to simplify it. Neither does Maple. However, if you want a decimal approximation for sin(2), use **evalf**.

```
evalf(sin(2));
```

0.9092974268

Question... I cannot get a numerical value of π from Maple using **evalf**. Why?

Answer... It is a common mistake to enter π as **pi** instead of **Pi**. Maple will consider **pi** as the Greek letter "π" rather than the geometrical constant "π". Therefore, **pi** does not have any numerical value.

```
pi, evalf(pi), Pi, evalf(Pi);
```

π, π, π, 3.141592654

Question... When I tried to evaluate a built-in function, Maple just returned the input unevaluated. I then used **evalf**, but still Maple did not give me a numerical value. What should I check?

Answer... Check your spelling. Most likely, you've misspelled the name of a built-in function or constant. For example, you may have used **cosine(Pi)** or **cos(pi)** instead of **cos(Pi)**.

Question... I got the error message "`Error, missing operator or `;``" when I entered a complicated arithmetic expression. What should I check?

Answer... There are many possible errors that will generate this error message. You should:

- Check the spelling of all of the commands and variables.
- Check if all the **(** *parentheses* **)** are in the right places. In particular, make sure you correctly matched right and left parentheses.
- Check that you typed * for multiplication. For example:

  ```
  sin(5)3 + 7;
  Error, missing operator or `;`
  ```

  Instead, use.

  ```
  sin(5)*3 + 7;
  ```

CHAPTER 4

Variables and Functions

Variables

Immediate Assignment

With a calculator, you can store a value into memory and recall it later. With a more advanced calculator, you can store several, different values under names such as A, B, C, and so on, and then use one or more of them in later calculations.

You can do even better in Maple. You can assign a name to any Maple expression or value and then recall it whenever you want. You do this using a colon followed by an equal sign, **:=**, that is the symbol for **immediate assignment**. (Note that there is no space between the colon and equal sign.) For example:

```
a := 3.4;                    # We assign a to have the value 3.4.
```

3.4 # Output in the document mode.

$a := 3.4$ # Output in the worksheet mode.

> Please note that from now on, in order to avoid repetition, we will only show you the 1-D input, in cases where the same keystrokes can be applied to the 2-D input. Also, we will only show you the output in the document mode.

Once you've made an assignment, you can recall its value or use it in an expression:

```
a;
```

3.4

```
a+2;
```

5.4

```
a^2;                         #You will see a^2 in the case of 2-D mode.
```

11.56

How Expressions Are Evaluated

You may want to know how Maple keeps track of all the symbols and variables that you have defined.

Say we assign the name **myLunch** to the sum of **apple** and 3 times **banana**.

```
myLunch := apple + 3*banana;
```

$apple + 3\ banana$

Maple spits back the definition because **apple** and **banana** have no associated values. Now suppose we give the value 2 to **apple** and then reevaluate **myLunch**:

```
apple := 2;                  # Now apple has the value 2.
```

```
myLunch;
```

$2 + 3\, banana$

(When Maple reevaluates **myLunch,** it substitutes the value 2 for **apple**.)

If we now define the value of **banana** to be 3 and reevaluate **myLunch**:

```
banana := 3;          # Now banana has the value 3.
```

3

```
myLunch;
```

11

When Maple reevaluates **myLunch**, it replaces **apple** and **banana** by their respective values, and simplifies the resulting expression to obtain 11. Bon appétit!!

Redefining and Clearing Symbols

Once a name has been assigned a value or an expression, Maple retains the association until you redefine it, explicitly clear it, or end your session.

```
a := 3.4;             # We assign a to be 3.4.
```

3.4

```
a := 5;               # We reassign a to be 5.
```

5

```
a + 2;                # Maple uses the most recently assigned value of a.
```

7

The easiest way to ask Maple to forget about an assignment is to use the **unassign** command.

```
unassign('a');        # Note that we use single quotes around a.
a;
```

a

You can also **unassign** assignments for several names at the same time, using one single statement:

```
unassign('myLunch','apple','banana');
```

Rules for Names

Names you use can be made up of letters and numbers, subject to the following two rules:

- You can't use a name that begins with a number. For example, **2app** is not an acceptable name because in 1-D input, Maple will think that you forgot to type an operator between "**2**" and "**app**," and so will give you an error message. In 2-D input mode, Maple will think that you mean "**2**" times "**app**."
- You can't choose names that conflict with Maple's own names. For example, you can't name one of your own variables **sin**.

All of the following are examples of legitimate names that you could use:

```
a, m, p1, A, area, Perimeter, Batman, classsOf2012
```

> **Note:** Maple distinguishes upper case and lower case characters. For example, the names **Batman**, **batman**, and **batMan** are different.

Evaluation Command

You can substitute values into an expression without defining the variables explicitly. The substitution command, **eval**, is used in the form:

eval(*expression, set of substitutions using =* **)**

For example, to substitute $x = 2$ and $y = 5$ into the expression $x^2 - 2xy$:

```
eval(x^2-2*x*y, {x=2, y=5 });                    #1-D input, or
```

$eval(x^2 - 2 \cdot x \cdot y, \{x = 2, y = 5\})$ ***# 2-D input***

$$-16$$

One advantage of using the **eval** command is that the evaluation of a variable is temporary and is not remembered by Maple.

```
x;            # The value of x is unchanged by the previous substitution.
```

$$x$$

Finding the Larger Root of a Quadratic Equation

■ **Example**. We want to find the larger root of each of the following quadratic equations: $2x^2 + 5x - 6 = 0$ and $2x^2 + x - 3 = 0$.

The roots of a quadratic equation $ax^2 + bx + c = 0$, with $a \neq 0$, are found using the quadratic formula $(-b \pm \sqrt{b^2 - 4ac})/(2a)$. The larger root, when $a > 0$, is thus:

```
unassign('a','b','c','largerRoot');
largerRoot := (-b+sqrt(b^2-4*a*c))/(2*a);   # 1-D input, or
```

$$largerRoot := \frac{\left(-b + \text{sqrt}\left(b^2 - 4 \cdot a \cdot c\right)\right)}{(2 \cdot a)} \qquad \text{\# 2-D input}$$

$$\frac{1}{2}\frac{-b + \sqrt{b^2 - 4ac}}{a}$$

Here are the larger roots of each of the two equations:

```
eval(largerRoot, {a = 2, b = 5, c = -6});
```

$$-\frac{5}{4} + \frac{1}{4}\sqrt{73}$$

```
eval(largerRoot, {a = 2, b = 1, c = -3});
```

$$-\frac{1}{4} + \frac{1}{4}\sqrt{25}$$

The above answer can actually be simplified to 1. (Maple does not notice that $\sqrt{25}$ = 5.) You can tell Maple to do so by using the **simplify** command, which will be discussed in detail in the next chapter.

```
simplify(%);
```

$$1$$

Functions

Defining Functions

Maple has many built-in functions such as **sqrt**, **sin**, and **tan**. You can define your own functions as well.

To define a function $f(x)$ in Maple, you use:

```
f := x -> formula ;
```

The "arrow symbol" **->** is formed by entering the minus sign and greater than sign together, with no space(s) in between.

For example, you can define the function $f(x) = x^2 + 5x$ in Maple with:

```
f := x -> x^2+5*x;
```
#1-D input, or

$f := x \to x^2 + 5 \cdot x$ **#2-D input**

$$x \to x^2 + 5x$$

Now, we can do some evaluations:

```
f(7.1);
```

$$85.91$$

```
f(a);
```

$$a^2 + 5a$$

```
f(x+1);
```

$$(x+1)^2 + 5x + 5$$

```
f(f(y));
```

$$(y^2 + 5y)^2 + 5y^2 + 25y$$

Functions with More Than One Variable

Functions may have more than one variable. A simple example is the computation of the average speed of an automobile.

If an automobile travels *m* miles in the span of *t* minutes, then its average speed in miles per hour is given by the expression $\frac{m}{(t/60)} = \frac{60m}{t}$. We then have a speed function "$f(m, t) = \frac{60m}{t}$":

```
speed := (m,t) -> 60*m/t ;
```
#1-D input, or

$speed := (m, t) \to \frac{60 \cdot m}{t}$ **#2-D input**

$$(m,t) \to \frac{60m}{t}$$

If a distance of 45 miles is traveled in 30 minutes, the average speed will be 90 m.p.h.:

```
speed(45,30);
```

$$90$$

More Examples

Functions of Split Definition

Functions sometimes cannot be defined using a single formula. For example, the famous Heaviside function is defined by:

$$H(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{if } x \le 0 \end{cases}$$

To define this function in Maple, we use the **piecewise** command as follows:

```
h:= x -> piecewise(x > 0, 1, x <= 0, 0);     #1-D input, or
```

h* := *x* → *piecewise*(*x* > 0, 1, *x* ≤ 0, 0)** ***#2-D input

$$x \to \text{piecewise}(x > 0, 1, x \le 0, 0)$$

(The characters "**<=**" in the command mean ≤, less than or equal to.)

To use **piecewise** to define a function having two branches, write:

piecewise(*condition1* **,** *result1* **,** *condition2* **,** *result2* **);**

This means that if *condition1* is satisfied, then *result1* will be used; otherwise, Maple will use *result2* if *condition2* is satisfied. The following table shows some operators you will use to check conditions.

Operator	*Meaning*
=	Equal to
>	Greater than
>=	Greater than or equal to
and	And

Operator	*Meaning*
<>	Not equal to
<	Less than
<=	Less than or equal to
or	Or

The syntax for the **piecewise** command can also be extended. For example, the following function *f* has three branches and can be defined as you see below:

$$f(x) = \begin{cases} 1-x, & \text{if } 1 < x < 3 \\ x^2, & \text{if } 0 \le x \le 1 \\ x+2, & \text{if } x < 0 \text{ or } x \ge 3 \end{cases}$$

```
f := x -> piecewise( 1 < x and x < 3, 1-x,
   0 <= x and x <= 1, x^2, x < 0 or x >= 3, x+2);
```

To see this definition more clearly, we recommend that you write the conditions one on each line (using *shift – return* on some systems) and line them up carefully:

```
f := x -> piecewise( 1 < x and x < 3,    1-x,
                     0 <= x and x <= 1,  x^2,
                     x < 0 or x >= 3,    x+2);
```

Useful Tips

Never assign values to any of the names **`x`**, **`y`**, **`z`**, or **`t`**. Otherwise, Maple will confuse them with the variables **`x`**, **`y`**, **`z`**, or **`t`** that you typically use when defining functions.

You should always use **`unassign`** before defining functions. It will help you to avoid potential conflicts between variable names and function definitions.

Most of Maple's built-in names use only lower case letters. Several names introduced in later versions begin with capital letters. One way to easily distinguish your names from Maple's is to use names that begin with a lowercase letter and include one or more uppercase letters (e.g., **`myFunction`**, **`newVar`**).

Even if you start a new worksheet or document in a *single* Maple session, all the variables or functions that you defined in earlier worksheets will still be retained until you quit Maple.

You can define **`e := exp(1);`**. This will help to simplify many of your inputs. For example, $e^{0.3}$ can now be entered as **`e^0.3`**, instead of **`exp(1)^0.3`**. However, it is still preferred to use **`exp(0.3)`**.

Troubleshooting Q & A

Question... When I defined a new variable or function, I got the error message "`attempting to assign to ... which is protected.`" What did I do wrong?

Answer... If you try to rename a built-in function or built-in constant, Maple will give you this error message. You cannot choose names that conflict with Maple's own names.

Question... I tried to define a function, but I couldn't get it to work. What should I check?

Answer... Three things usually bother function definitions.

- First, check for the proper syntax. Make sure you use a colon-equal definition. Also, don't leave any space between the – and > keys in forming the –> sign. (Common mistakes include using **`=`** instead of **`:=`** and using **`f(x):=`** instead of **`f := x ->`**.)

- Second, check that the formula of the function is entered correctly. Some common mistakes are to use expressions such as **cos x**, **e^x**, and so on.
- Third, your function or variable name may conflict with something else you used earlier in your Maple session. For example, you may have once defined:

  ```
  x := 3;
  ```

  Later you define:

  ```
  f := x -> x^2;
  f(x);
                  9
  ```

  The result is not the x^2 you expected, but $3^2 = 9$ since **x** has the value 3. You can check if this is the case by evaluating **x**. Make sure you explicitly clear the variable(s) before you define your function:

  ```
  unassign('x');
  f := x -> x^2;
  ```

Question... What is the difference between **=** (equal sign) and **:=** (colon-equal)?

Answer... The symbol **=** means equality, while **:=** means assignment. Equality is a test that gives a true or false answer. Assignment is an action that either gives a name to a value or defines a function.

Question... When I tried to evaluate a function, I got the error message "`... uses a 2nd argument, ... , which is missing.`" What went wrong?

Answer... This message means that you did not supply enough variables to evaluate the function.

A common mistake is to define a function of one variable, say,

```
f := x -> x^2;
```

Later, you use the same name **f** to define a function of two variables, say,

```
f := (x, y) -> x + 2*y;
```

The original definition of $f(x) = x^2$ has now been erased. If you type **f(x)**, you will get an error message complaining about not having enough variables (because Maple is now expecting two variables for the function).

Question... When I used **unassign**, I got the error message "`... cannot unassign.`" What went wrong?

Answer... Make sure that you enclose the variable name that you want to clear in single quotes. For example:

```
a := 3;
unassign(a);          # This will give you an error message.
Error, (in unassign) cannot unassign `3'
unassign('a');        # This is correct.
```

CHAPTER 5

Computer Algebra

Working with Polynomials and Powers

The expand and factor Commands

In this chapter you will learn how to use Maple to perform common algebraic operations. Let's start with polynomials.

The **expand** command does exactly what its name says it does:

```
expand((x-2)*(x-3)*(x+1)^2);        # for 1-D input, or
```

$expand\left((x-2)\cdot(x-3)\cdot(x+1)^2\right)$ ***# for 2-D input***

$$x^4 - 3x^3 - 3x^2 + 7x + 6$$

The **factor** command is basically the inverse of the **expand** command:

```
factor(x^4-3*x^3-3*x^2+7*x+6);        # for 1-D input, or
```

$factor\left(x^4 - 3\cdot x^3 - 3\cdot x^2 + 7\cdot x + 6\right)$ ***# for 2-D input***

$$(x-2)\,(x-3)\,(x+1)^2$$

Here are some more examples:

Example	*Comment*
`factor(x^2-3);` $x^2 - 3$	Although $x^2 - 3 = (x+\sqrt{3})(x-\sqrt{3})$, **factor** will not give radicals in its answer.
`factor(x^2-3,sqrt(3));` $-(x+\sqrt{3})\,(-x+\sqrt{3})$	We can give Maple a list of radicals it is allowed to use in factorization.
`expand((x-1.54)*(3.2*x-2.9));` $3.2x^2 - 7.828x + 4.466$ `factor(%);` $3.2\,(x - .90625000)\,(x - 1.540000)$	**factor** does a nice job even when you use numerical coefficients.
`factor((-2+6*I)+(-5-3*I)*x + (1-I)*x^2);` $(1-I)\,(-x+1+2I)\,(-x+2I)$	**factor** works even when the coefficients are complex.
`factor(x^2+1);` $x^2 + 1$	**factor** won't use complex numbers unless at least one of the coefficients is a complex number.

factor(x^2+1, I); $-(x+I)(-x+I)$	Or you give **factor** explicit permission to use **I** in its answer.
expand((x-y+z)^3); $x^3-3x^2y+3x^2z+3xy^2-6xyz+3xz^2-y^3+3y^2z-3yz^2+z^3$ **factor(%)** $(x-y+z)^3$	The **expand** and **factor** commands can also be used for polynomials with more than one variable.

The simplify Command

The **simplify** command tries to produce an expression that Maple thinks is the simplest form. For example:

simplify(4^(1/2)+5); # 1-D input, or,

$$simplify\left(4^{\left(\frac{1}{2}\right)}+5\right)$$ ***# 2-D input***

$$7$$

simplify((x+2)^2-4);

$$x^2+4x$$

Please note that the **simplify** command will not factor a polynomial into its "simplest" form. That is the job for **factor**.

factor((x+2)^2-4);

$$x(x+4)$$

The symbolic Option

By default, Maple is very careful in simplifying. It understands that **sqrt(x^2)** is not always the same as **x** for all complex numbers. Instead, **sqrt(x^2)** simplifies to **x** times the complex sign of **x**.

simplify(sqrt(x^2));

$\text{csgn}(x)\,x$ # csgn stands for complex sign.

Sometimes we want such expressions of powers simplified anyway. In those cases we use the **symbolic** option of the **simplify** command.

simplify(sqrt(x^2), symbolic);

$$x$$

The **symbolic** option of the **simplify** command handles $\sqrt{x^2}$ by treating it formally as $(x^2)^{1/2}$ and rewriting it to be $x^{2(1/2)}=x^1$ (without regard for the sign of x). Similarly:

simplify((x^6)^(1/3));

$$\left(x^6\right)^{1/3}$$

simplify((x^6)^(1/3), symbolic);

$$x^2$$

Working with Rational Functions

The simplify and convert Commands

A rational function is an expression of the form $\frac{\textit{a polynomial}}{\textit{another polynomial}}$. The following are three common algebraic operations involving rational functions.

- Combining terms over a common denominator can be done with the **simplify** command. For example, to combine $\frac{2}{3x+1}+\frac{5x}{x+2}$:

  ```
  simplify( 2/(3*x+1) + (5*x)/(x+2));
  ```

  $$\frac{7x+4+15x^2}{(3x+1)\,(x+2)}$$

- Splitting up rational functions into partial fractions can be done with the **convert** command if you specify the **parfrac** option. For example, to split up $\frac{11x^2-17x}{(x-1)^2(2x+1)}$:

  ```
  convert((11*x^2-17*x)/((x-1)^2*(2*x+1)),parfrac,x);
  ```

  $$\frac{5}{2x+1}+\frac{3}{x-1}-\frac{2}{(x-1)^2}$$

- The **convert** command also does long division when you use the **parfrac** option. For example, to find $(x^5-2x^2+6x+1)\div(x^2+x+1)$:

  ```
  convert((x^5-2*x^2+6*x+1)/(x^2+x+1),parfrac,x);
  ```

  $$x^3-x^2-1+\frac{2+7x}{x^2+x+1}$$

Working with Trigonometric and Hyperbolic Functions

Basic Trigonometric and Hyperbolic Identities

The **expand** command can work on expressions that involve trigonometric and hyperbolic functions.

```
expand(sin(2*x));
```

$$2\sin(x)\cos(x)$$

```
expand(cosh(2*x));
```

$$2\cosh(x)^2-1$$

The **simplify** command can also be used for trigonometric and hyperbolic functions, but the results may not come out the way you think they should. For example, **simplify** recognizes that $\sin^2 x + \cos^2 x = 1$, but does not recognize the identity $\cos^2 x - \sin^2 x = \cos(2x)$.

```
simplify(sin(x)^2+cos(x)^2);
```

$$1$$

```
simplify(cos(x)^2-sin(x)^2);
```

$$2\cos(x)^2 - 1$$

(Notice that Maple always "simplifies" trigonometric functions by using the cosine function instead of the sine function.)

The combine Command

The **combine** command is more effective in working with trigonometric and hyperbolic functions. For example,

```
combine(cos(x)^2-sin(x)^2);
```

$$\cos(2x)$$

```
combine(2*sin(x)*cos(x));
```

$$\sin(2x)$$

```
combine(cosh(x)^2+sinh(x)^2);
```

$$\cosh(2x)$$

```
combine(sin(2*x)*cos(3*x));
```

$$\frac{1}{2}\sin(5x) - \frac{1}{2}\sin(x)$$

Useful Tips

You have to be careful when you use the **symbolic** option in the **simplify** command. Most of the time you use it, you will be assuming that all the quantities you're working with are inside certain subsets of the real numbers.

The **combine** command can also be used very effectively to simplify exponential, logarithmic, and power functions.

Troubleshooting Q & A

Question... When I used **simplify**, **expand**, or **factor** on a polynomial, I got a number. What went wrong?

Answer... First, it is possible that after expansion or simplification, all the terms in your polynomial canceled out and left you with a constant.

If this is not the case, check whether you assigned a value to the variable at some earlier time. For example, you may have assigned

```
x := 5;
```

earlier, then after awhile you typed:

```
expand((x-3)^2*(4*x+5));
```

$$100$$

You asked Maple to expand (5-3)^2*(4*5+5) = 100 which is a number! To correct this, type:

```
unassign('x');
```

and re-execute the **expand** command.

Question... When I used **simplify** or **expand** on a simple polynomial, I got the wrong answer. What went wrong?

Answer... Check to see that you remembered to put an asterisk (*****) between terms being multiplied together. Parenthesized expressions don't get multiplied when they're written next to each other. Consider:

```
expand((x+1)(x+2)^2);
```

$$x(x+2)^2 + 2x(x+2) + 1$$

Maple interprets the expression **(x+1)(x+2)^2** as the function $(x+1)^2$ evaluated at $(x+2)$. This is almost certainly not what you wanted to do.

Question... I tried **simplify**, **expand**, **factor, combine,** and several other commands on an expression, and I can't get the type of expression I was expecting. What should I do?

Answer... There are more advanced techniques for controlling the way Maple simplifies an expression. But the practical answer may be that the software just can't get you to where you want to be using algebra alone. You may need to look for a different approach.

For example, Maple can't directly simplify $\tanh^{-1}\left((e^x - e^{-x})/(e^x + e^{-x})\right) = x$. But by graphing $\tanh^{-1}\left((e^x - e^{-x})/(e^x + e^{-x})\right)$, you see that it looks like $y = x$. (Chapter 9 does graphing.) You can also compute that $\tanh^{-1}\left((e^x - e^{-x})/(e^x + e^{-x})\right)$ has derivative 1, so that's almost enough to establish the identity. (Chapter 12 will show you how to do derivatives.)

CHAPTER 6

Working with Equations

Equations and Their Solutions

The solve Command for a Single Equation

Maple's **solve** command will solve an equation for an "unknown" variable. You use it in the form:

```
solve( an equation , unknown variable );
```

For example,

```
solve( 2*x+5 = 9, x );          #1-D input, or,
```

$solve(2 \cdot x + 5 = 9, x)$ ***#2-D input***

$$2$$

Notice that equations in Maple are written using the equal sign "**=**".

You can check that $x = 2$ is the correct solution to the above equation, by evaluating the equation with $x = 2$:

```
eval(2*x+5=9, x=2 );
```

$$9 = 9$$

You can even have Maple check that this substitution gives the correct answer by evaluating the result as a Boolean expression.

```
evalb( eval( 2*x+5 = 9, x=2) );
```

$$true$$

This means that after the substitution $x = 2$, it is true that the left-hand side of the equation equals the right-hand side.

Here are a few more examples involving **solve**:

Equation	*To Solve It in Maple*	*Comment*
Solve $x^2 - 3x + 1 = 0$ for x	`solve( x^2-3*x+1 = 0, x );` $\frac{3}{2}+\frac{1}{2}\sqrt{5},\ \frac{3}{2}-\frac{1}{2}\sqrt{5}$ `evalf(%);` 2.618033988, 0.381966012	Equations can have more than one solution. We can see a numerical answer with the **evalf** command.
Solve $y^2 - ay = 2a$ for y.	`solve( y^2 - a*y = 2*a, y );` $\frac{1}{2}a+\frac{1}{2}\sqrt{a^2+8a},\ \frac{1}{2}a-\frac{1}{2}\sqrt{a^2+8a}$	If the equation involves other variables, Maple will treat them as parameters.

Solve $x^3+x^2=-3x$ for x	`solve( x^3+x^2 = -3*x, x );` $0, -\frac{1}{2}+\frac{1}{2}I\sqrt{11}, -\frac{1}{2}-\frac{1}{2}I\sqrt{11}$ `evalf( %, 5 );` $0., -0.50000+1.6583I, -0.50000-1.6583I$	Here two of the solutions are complex. I stands for $\sqrt{-1}$. Numerical approximations up to 5 digits are obtained in this way.
Solve $x+\sin x=\cos x$ for x	`solve( x+sin(x) = cos(x), x );` `Warning, solutions may have been lost`	No solution is returned when Maple cannot solve an equation.

Note: The **`solve`** command works very well for equations involving polynomials. However, it often has much less success with trigonometric, exponential, logarithmic, or hyperbolic functions.

The solve Command for a System of Equations

The **`solve`** command can also be used to solve a system of equations. For two equations in two unknown variables, you use this form of the command:

`solve({` *`equation1`*, *`equation2`* **`}, {`** *`variable1`*, *`variable2`* **`});`**

■ **Example.** To solve the equations $3x+8y=5$ and $5x+2y=7$ in the variables x and y, use:

```
solve( {3*x+8*y = 5, 5*x+2*y = 7}, {x,y} );
```

$$\left\{x=\frac{23}{17},\ y=\frac{2}{17}\right\}$$

■ **Example.** To solve the equations $3xy-y^2=-4$ and $2x+y=3$:

```
solve( {3*x*y - y^2 = -4, 2*x + y = 3}, {x,y} );
```

$$\left\{x=-\frac{1}{2}RootOf(5\_Z^2-9\_Z-8, label=\_L1)+\frac{3}{2},\ y=RootOf(5\_Z^2-9\_Z-8, label=\_L1)\right\}$$

Maple gives the answer in terms of the roots of the polynomial $5z^2-9z-8$. (The expression $label=\_L1$ indicates that the values of x and y are built from the same root of the polynomial in the answer.) Since the polynomial is quadratic, we expect two sets of solutions. To get a list of solutions, we use the **`allvalues`** command:

```
allvalues( % );
```

$$\left\{x=\frac{21}{20}-\frac{1}{20}\sqrt{241},\ y=\frac{9}{10}+\frac{1}{10}\sqrt{241}\right\},\left\{x=\frac{21}{20}+\frac{1}{20}\sqrt{241},\ y=\frac{9}{10}-\frac{1}{10}\sqrt{241}\right\}$$

■ **Example.** If you try

```
solve({x+y = 0, x+y = 1},{x,y});
```

No solutions are returned because there are no solutions to this system of equations.

■ **Example.** We can also solve systems of equations involving more than two variables. For example:

```
allvalues( solve( {x+2*y-z=1, x-y+z^2=2, y+x=2*z},
                  {x,y,z} ) );
```

$$\left\{x=6\sqrt{2}-7,\ y=-2\sqrt{2}+3,\ z=-2+2\sqrt{2}\right\},$$
$$\left\{x=-7-6\sqrt{2},\ y=3+2\sqrt{2},\ z=-2-2\sqrt{2}\right\}$$

Numerical Solutions for Equations

Numerical Answers from solve

The **solve** command can also use efficient numerical techniques to approximate roots of polynomial and a few other simple functions. In order to do this, at least one of the coefficients in the equation has to be a floating point number. For example:

```
solve( x^5-x^3 = 1, x );
```

$$RootOf(\_Z^5-\_Z^3-1, index=1),\quad RootOf(\_Z^5-\_Z^3-1, index=2),$$
$$RootOf(\_Z^5-\_Z^3-1, index=3),\quad RootOf(\_Z^5-\_Z^3-1, index=4),$$
$$RootOf(\_Z^5-\_Z^3-1, index=5)$$

Now, if we replace "1" with "1.0" in the equation (i.e., replacing the exact value 1 with its numerical equivalent 1.0), then Maple will report numerical approximations for the answers:

```
solve( x^5-x^3 = 1.0, x );   # use 1.0 instead of 1
```

$$1.236505703,\ 0.3407948662+0.7854231030\,I,\ -0.9590477179+0.4283659562\,I,$$
$$-0.9590477179-0.4283659562\,I,\ 0.3407948662-0.7854231030\,I$$

This trick will also work for systems of equations that involve polynomials or simple functions.

```
solve( {3*x*y - y^2 = 4, 2*x^2 + y = 9.0}, {x,y} );
                                          # use 9.0 instead of 9
```

$$\{x=1.614445936, y=3.787128638\}, \{x=2.031216860, y=.7483161321\},$$
$$\{x=-2.199468588, y=-0.6753241393\}, \{x=-2.946194209, y=-8.360120631\}$$

If you used the same **solve** command without a decimal point in the equation:

```
solve( {3*x*y - y^2 = 4, 2*x^2 + y = 9}, {x,y} );
```

the solution would involve the roots of a quartic equation. Using **allvalues** to obtain explicit representations of the solutions produces several pages of output!

Numerical Solutions Using fsolve

To find numerical solutions for equations in general, use the **fsolve** command. It has the same syntax as **solve**. For a single equation, use:

fsolve(*an equation* **,** *variable to solve for* **);**

and, for a system of equations, use:

fsolve({ *equation1*, *equation2*}, { *variable1*, *variable2* });

For example:

```
fsolve( x^5-x^3-1=0, x );
```

1.236505703

In general the **fsolve** command gives a single answer. However, if the equation involves only polynomials of one variable, you can indicate the number of roots to look for with the **maxsols** option. Specify the **complex** option if you want complex values returned as well.

```
fsolve(x^5-x^3-1=0,x, complex, maxsols=5);
```

−0.9590477179 − 0.4283659563 I, −0.9590477179 + 0.4283659563 I,
0.3407948662 − 0.7854231030 I, 0.3407948662 + 0.7854231030 I, 1.236505703

You can usually use **fsolve** to find solutions of a complicated equation whenever **solve** fails.

```
solve( exp(x^2) -50*x^2+3*x = 0, x );
   Warning, solutions may have been lost
solve( exp(x^2) -50*x^2+3*x = 0.0, x );
   Warning, solutions may have been lost
fsolve( exp(x^2) -50*x^2+3*x = 0, x );
```

−0.115494211

The two **solve** commands return no output other than a warning that solutions may be lost; the **fsolve** command returns an approximate numerical solution.

Guiding fsolve with a Range

Using fsolve to Find a Solution in a Specific Interval

Sometimes we are interested in a root other than the one that **fsolve** returns. We can guide **fsolve** by giving it a range in which to look for the solution. The form of the command becomes:

fsolve(*equation* , *variable* , *range for the solution* **);**

■ **Example.** We are interested in using **fsolve** to find solutions to the trigonometric equation $\tan(x) = x$. If we know that there is a solution in each of the intervals $-\pi/2 \le x \le \pi/2$, $\pi/2 \le x \le 3\pi/2$, and $3\pi/2 \le x \le 5\pi/2$, we can specify appropriate ranges. We then find three solutions.

```
fsolve( tan(x) -x = 0, x,  -Pi/2..  Pi/2);
fsolve( tan(x) -x = 0, x,   Pi/2..3*Pi/2);
fsolve( tan(x) -x = 0, x, 3*Pi/2..5*Pi/2);
```

0
4.493409458
7.725251837

fsolve with a Range in a System of Equations

To solve a system of two equations for the two unknowns x and y, restricted to the intervals $x_0 \le x \le x_1$ and $y_0 \le y \le y_1$, you use **fsolve** with three arguments:

```
fsolve( { equation1, equation2 }, {x,y},
                  { x = x0..x1, y = y0..y1 });
```

■ **Example.** The system $y^2 - x^3 = 5$ and $y = x - 3\cos x + 4$ has a solution inside the intervals $-2 < x < -1$ and $1 < y < 2$. We can pinpoint it with:

```
fsolve( { y^2-x^3=5, y=x-3*cos(x)+4 }, {x,y},
        { x=-2..-1, y=1..2 } );
```

$$\{x = -1.231437723,\ y = 1.769915245\}$$

More Examples

Extracting Solutions from the Results of solve

Sometimes you will want to work with the answers you get from **solve** and **fsolve** without having to retype them. This example will show you the two steps that will enable you to do so.

Consider the equation $x^2 - 3x + 1 = 0$. It has two solutions:

```
solve(x^2-3*x+1 = 0, x);
```

$$\frac{3}{2} + \frac{1}{2}\sqrt{5},\ \frac{3}{2} - \frac{1}{2}\sqrt{5}$$

• Step 1. Make the output of the **solve** command the contents of a list and give the list a name. (We discuss lists in the next chapter.)

```
ans := [ solve(x^2-3*x+1 = 0, x) ];
```

$$\left[\frac{3}{2} + \frac{1}{2}\sqrt{5},\ \frac{3}{2} - \frac{1}{2}\sqrt{5}\right]$$

• Step 2. Identify each of the answers with **ans[1]** and **ans[2]**.

```
ans[1];
```

$$\frac{3}{2} + \frac{1}{2}\sqrt{5}$$

```
ans[2];
```

$$\frac{3}{2} - \frac{1}{2}\sqrt{5}$$

Let's check that **ans[2]** is really a solution to $x^2 - 3x + 1 = 0$:

```
simplify( eval(x^2 - 3*x + 1, x=ans[2]) );
```

$$0$$

How about:

```
(x - ans[1])*(x - ans[2]);
```

$$\left(x-\frac{3}{2}-\frac{1}{2}\sqrt{5}\right)\left(x-\frac{3}{2}+\frac{1}{2}\sqrt{5}\right)$$

```
expand( % );
```

$$x^2-3x+1$$

Useful Tips

fsolve works much more quickly using *numeric methods* than **solve** does using *algebraic methods*.

fsolve can deal with all kinds of equations, but it only gives a single answer in most cases. (It can be given a range in which to look for a solution, but that removes the automation.) **solve** will try to find up to 100 solutions, but in many cases, cannot solve the equations symbolically.

We think that the best approach is first to use **solve**, including a floating point number in one of the equations to find a numerical solution. Then, work with the output!

Maple can return all solutions to some transcendental equations when the environment variable **_EnvAllSolutions** is set to true. For example:

```
solve(sin(x) = 1/2);    # The answer is in the first quadrant.
```

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

```
_EnvAllSolutions := true:
solve(sin(x) = 1/2);    # The general solution.
```

$$\frac{1}{6}\pi+\frac{2}{3}\pi\_B1\sim+2\pi\_Z1\sim$$

(In the output, $\_B1\sim$ is a binary constant (0 or 1) and $\_Z1\sim$ is an integer constant.)

The following table can help you remember the difference between the various types of brackets. But beware – they are not interchangeable!

Syntax Element	*Purpose*	*Example*
(*parentheses*)	(i) Grouping terms in a computation (ii) Arguments of commands	`(x^2+3)*(x–1)` `sin(x^2)`
[*square brackets*]	List (i.e. an ordered list)	`[x, y, z]`
{ *curly braces* }	Set (i.e., an unordered list)	`{x–3 = y, 5*x+y = 2}`
List [*square brackets*]	Specify a position in a list	`soln[1]`

Troubleshooting Q & A

Question... When I used **solve** or **fsolve**, I got an error message. What should I check first?

Answer... You should first check if you made a mistake in the input of the equation(s).

- Check that you entered the formula correctly. Common mistakes include misspelling names of the Maple built-in functions, and forgetting to type the multiplication symbol "*".
- Did you remember to include the variable(s) to solve for in the input?
- If you are using more than one equation, make sure all the curly braces and commas are located in the right places.

Question... When I used **solve** or **fsolve**, I got the error message "`a constant is invalid as a variable.`" What should I check?

Answer... Check that the variable or variables you are trying to solve for have no assigned values. For example, you may have earlier assigned:

```
x := 3;
```

Later, you try:

```
solve( x^2-1=0, x);
Error, (in solve) a constant is invalid as a variable, 3
```

You should have cleared the variable(s) before using **solve**.

```
unassign('x');
```

Question... I used **solve** and Maple gave me the output "*RootOf*". What does this mean?

Answer... For example, if you try:

```
sol := [solve(sqrt(x) = (x-3)^2, x)];
```

$$[\mathit{RootOf}(-\_Z+\_Z^4-6\_Z^2+9,\ \mathit{index}=1)^2,\ \mathit{RootOf}(-\_Z+\_Z^4-6\_Z^2+9,\ \mathit{index}=2)^2]$$

Maple tells you that the solutions to the equation $\sqrt{x}=(x-3)^2$ are the squares of the 1^{st} and 2^{nd} roots of the polynomial $-z+z^4-6z^2+9$. This is not helpful. However, you can see the numerical value of these answers with:

```
evalf({sol[1], sol[2]});
```

$$\{1.835964860, 4.452626878\}$$

If the equation involves polynomial only, you can also use the **allvalues** command to convert the answer to a friendlier form:

```
allvalues(%);
```

Question... I used **`solve`** but could not understand Maple's output. It had the symbols `%1`, `%2`, and so on. What does this mean?

Answer... If the result from **`solve`** is too lengthy, Maple will use the symbols `%1`, `%2`, etc., to represent expansions of subexpressions that occur several times.

Question... Maple did not give any output from **`solve`** or **`fsolve`**. Why not?

Answer... This usually means one of three things.

- Maple does not know how to solve your equation(s) with the **`solve`** command.
- If you used **`fsolve`** with a specified interval, Maple was unable to locate a root in that interval. Make sure that the interval(s) you gave contains a root.
- There is no solution to your equation or system of equations. A single equation might reduce to an absurdity (e.g., $x = x + 1$ reduces to $0 = 1$). A system of equations may be inconsistent (e.g., $x + y = 1$ and $x + y = 2$).

Question... When I used **`fsolve`**, I got an error message that says "`...should use exactly all the indeterminates.`" or "`...is in the equation, and is not solved for.`" What happened?

Answer... *Maple* cannot use a numerical method to approximate the solution, because your equation involves a constant that is not defined. For example, if **a** is not defined, *Maple* can do:

```
solve(x^2 = a, x);
```

But you will get an error message with:

```
fsolve(x^2 = a, x);
```

Question... **`fsolve`** failed to give me a solution to an equation in an interval where I know there is a solution. What happened?

Answer... **`fsolve`** uses a version of Newton's method to search for an answer. The procedure will not always find an answer. It is particularly vulnerable to functions whose graphs have narrow spikes on otherwise well-behaved regions.

For example, away from the origin, $\tan(x) = x^3$ has this feature, and it is hard for **`fsolve`** to locate a root without a very precise range.

Question... After solving an equation, I get real and complex roots. How can I only select the real roots?

Answer... To remove any term from *your list* that contains "**`I`**", use the following **`remove`** command.

`remove(has,` *your list*, **`I);`**

For example,

```
soln := [solve(x^7-5*x+4=0.0, x)];
```

[1., 0.8851280726, 0.5229014897 + 1.183774264 I, -0.7607643824 + 1.155768213 I, -1.409402287, -0.7607643824 - 1.155768213 I, 0.5229014897 - 1.183774264 I]

```
realSoln := remove(has, soln, I);
```

[1., 0.8851280726, -1.409402287]

Only the real solutions are selected.

Similarly, if you want to select those roots that have non-zero imaginary part, you can use:

```
imaginarySoln := select(has, soln, I);
```

You will learn more about **select** and **remove** in Chapter 24.

CHAPTER 7

Sets, Lists, and Sequences

Lists and Sets

What Is a List?

A **list** in Maple is an expression in which elements are separated by commas and enclosed in **[** *square brackets* **]**. For example, each of the following is a list.

```
[2, 5, 7, 10, -3, -25];        # Each element is a number.
```

[2, 5, 7, 10, -3, -25]

```
[3, [1,2], 6, `good`, `bad`, -2, `ugly`];
```

#The elements consist of numbers, a list of numbers and names.

[3, [1,2], 6, *good*, *bad*, -2, *ugly*]

What Is a Set?

A **set** in Maple is an expression in which elements are separated by commas and enclosed in **{** *curly braces* **}**. For example, each of the following is a set.

```
{2, 5, 7, 10, -3, -25};        # Each element is a number.
```

{-25, -3, 10, 7, 5, 2}

```
{3, [1,2], 6, `good`, `bad`, -2, `ugly`};
```

#The elements consist of numbers, a list of numbers and names.

{–2, 3, 6, *bad*, *good*, *ugly*, [1, 2]}

The Difference between Sets and Lists

A **list** is an ordered set. We can have repetition in the elements. On the other hand, ordering and redundancy of elements in a **set** does not matter.

```
{5,4,3,2,1};    # The arrangement of the elements in a set doesn't matter.
```

{1, 2, 3, 4, 5}

Also, duplicate elements are automatically removed from a set, but not from a list:

```
{1,2,2,3,3,3,4,4,4,4,5,5,5,5,5};
```

{1, 2, 3, 4, 5}

```
[1,2,2,3,3,3,4,4,4,4,5,5,5,5,5];
```

[1, 2, 2, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 5, 5]

Lists and sets are important structures in Maple. Many of Maple's inputs and outputs are expressed using lists or sets. For example, Maple gives us two sets for the output when we solve this system of equations:

```
solve({x+y=2, x^2+y = 2}, {x,y});
```

$$\{x = 1, y = 1\},\ \{x = 0, y = 2\}$$

We can then collect the solutions to form a list:

```
ans :=  [solve({x+y=2, x^2+y = 2},{x,y})];
```

$$[\{x = 1, y = 1\}, \{y = 2, x = 0\}]$$

Working with Elements of a List

Maple lets you work with the elements of a list directly. You can access each element using the **[** *square brackets* **]** notation.

For example, consider the list,

```
myList := ["red", 2, 3, 3.153, [1,3,5], sin(z), Pi,
        "history", Fred];
```

["red", 2, 3, 3.153, [1, 3, 5], sin(z), π, "history", *Fred*]

The following table shows some examples and how you can work with the elements of this list:

Maple Command	*Explanation*
myList[4]; 3.153	Returns the 4th element of the list.
myList[-3]; π	Returns the entry that is 3rd from the end of the list.
myList[3..6]; [3, 3.153, [1,3,5], sin(z)]	Returns a list of the 3rd through 6th elements of the **myList**.
op(myList[3..6]); 3, 3.153, [1,3,5], sin(z) or you can use **myList[3..6][];** 3, 3.153, [1,3,5], sin(z)	The **op** command will let us see the elements without a list. The empty **[** *square brackets* **]** at the end will give the same result.
myList[5]; [1,3,5] **myList[5][2];** 3 **myList[5,2];** #same as above 3	If an element of a list is a list itself, we can retrieve its entries in the same manner, using **[***square bracket***]** syntax.
myList[2] := 88; 88 **myList;** ["red", 88, 3, 3.153, [1, 3, 5], sin(z), π, "history", *Fred*]	This syntax lets you directly change the 2nd element of **myList**. You can see that 2nd element of this list has been replaced.

Sequences

The seq Command

The **seq** command can be used to generate a sequence of elements that can be defined by a mathematical formula. For example, to make a sequence of the form n^2, for each integer $1 \le n \le 16$, we will type:

```
seq(n^2, n=1..16);
```

1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225, 256

In general, the **seq** command is used in the form:

seq(*expression in a variable n*, *n* **=** n_0**..** n_1**);**

Here are a few examples:

Maple Command	*Remark*
`seq( n^2, n=-2..8 );` 4, 1, 0, 1, 4, 9, 16, 25, 36, 49, 64	The variable *n* runs from –2 to 8.
`seq( [cos(n),n/(n+1)], n = 1..3 );` $\left[\cos(1), \frac{1}{2}\right], \left[\cos(2), \frac{2}{3}\right], \left[\cos(3), \frac{3}{4}\right]$	The expression in the **seq** command can be a list itself. In this case, we form a table of coordinate pairs.
`seq( x^n, n=0..7 );` $1, x, x^2, x^3, x^4, x^5, x^6, x^7$	Each element can be a symbolic expression.

By combining the **seq** command with **[** *square brackets* **]** or **{** *curly braces* **}**, we can create a list or set. For example,

```
[ seq( n, n=1..22 ) ];
```

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22]

```
{ seq( cos(n), n =1..5 ) };
```

{cos(1), cos(2), cos(3), cos(4), cos(5)}

More Examples

An Experiment in Factoring

Let us look at the factorization of $x^n + 1$. With the help of the **seq** command, we can see the factorization for several values of n very quickly, say for $n = 2, 3, \ldots,$ 10.

```
seq(factor(x^n+1), n=2..10);
```

$$x^2+1,\ (x+1)(x^2-x+1),\ x^4+1,\ (x+1)(x^4-x^3+x^2-x+1),$$
$$(x^2+1)(x^4-x^2+1),\ (x+1)(1-x+x^2-x^3+x^4-x^5+x^6),\ x^8+1,$$
$$(x+1)(x^2-x+1)(x^6-x^3+1),\ (x^2+1)(x^8-x^6+x^4-x^2+1)$$

Do you notice that all the coefficients in all the factors above are either 1 or –1? Try repeating the command with **n = 2..100**. You will see the same thing!

You may conclude that every coefficient in every factor of $x^n + 1$ will be ±1, no matter what n is. But before you celebrate your latest discovery, check the factorization of $x^{105} + 1$. Surprise!

Drawing a Nonagon

■ **Example**. A regular nonagon is a regular polygon of nine sides (i.e., all nine sides have the same length and all nine angles are equal). It can be formed by joining together the points with coordinates $(\cos(2\pi n/9), \sin(2\pi n/9))$, where n = 1, 2, . . ., 9.

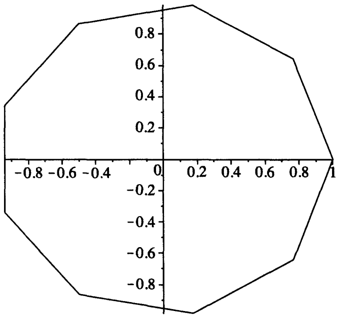
We can create a list of these points using the **seq** command:

```
coordlist:=
    [seq([cos(n*2*Pi/9),sin(n*2*Pi/9)],n=0..9)];
```

$$[[1, 0], [\cos(\tfrac{2}{9}\pi), \sin(\tfrac{2}{9}\pi)], [\cos(\tfrac{4}{9}\pi), \sin(\tfrac{4}{9}\pi)],[-\tfrac{1}{2},\tfrac{1}{2}\sqrt{3}],$$
$$[-\cos(\tfrac{1}{9}\pi),\sin(\tfrac{1}{9}\pi)], [-\cos(\tfrac{1}{9}\pi),-\sin(\tfrac{1}{9}\pi)], [-\tfrac{1}{2},-\tfrac{1}{2}\sqrt{3}],$$
$$[\cos(\tfrac{4}{9}\pi),-\sin(\tfrac{4}{9}\pi)], [\cos(\tfrac{2}{9}\pi),-\sin(\tfrac{2}{9}\pi)], [1,0]]$$

If we join the points one by one, we will create the nonagon. This can be done using the **plot** command. (**plot** will be discussed in details in Chapters 9 and 10.)

```
plot( coordlist );
```



Troubleshooting Q & A

Question... When I used the **seq** command, Maple gave me no output. What does that mean?

Answer... This means that the sequence you asked for is empty, because the index range you specified does not make sense. Make sure that $n_0 \le n_1$ in the command

seq(*expression***, n=**n_0**..**n_1**);**

CHAPTER 8

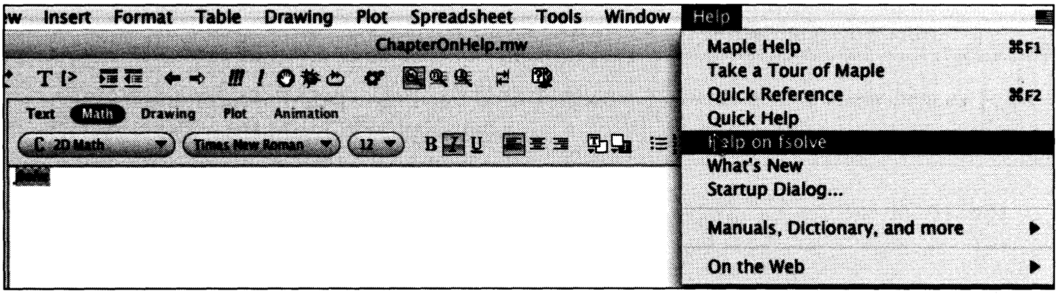
Getting Help and Loading Packages

Getting Help

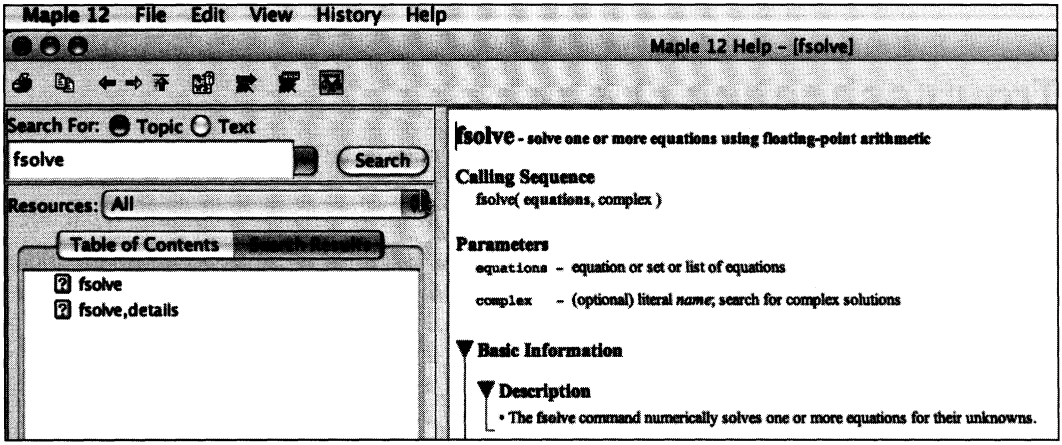
Help for Specific Maple Commands

You can always get some help on how to use a *Maple* command by typing the command, highlighting it, going to the **Help** menu in the menu bar, then selecting "**Help on** *command*" - the fifth item in the menu. A help window will open, giving you details of the command, including options and examples.

For example, if you forget how to use the **`fsolve`** command, you can type **`fsolve`**, highlight this string, then select "**Help on fsolve**" from the **Help** menu.

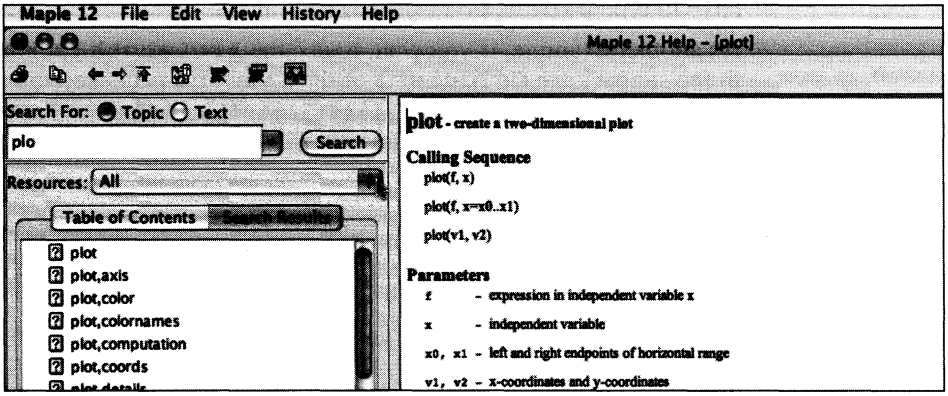


The help window that comes up gives help for the command and a list of other related commands.



Looking Up a Command

Sometimes, you won't remember the exact name of a command that you want use. Say, you remember that it started with **`plo`**, but you just can't recall its full name. You can type **`plo`**, highlight it, and select "**Help on plo**" from the **Help** menu. The help browser will open with Maple's best guess as to the command you're looking for. It will also list many other options for you on the left side in case it guessed wrong.



Packages

Loading a Package

Maple has a built-in vocabulary of several hundred commands. But still these are not enough for everyday usage. Additional commands are available in the Maple "packages." A few of the more common packages are: **`Algebraic`**, **`Calculus`**, **`Linear Algebra`**, **`plots`**, **`Statistics`**, **`Student`**, etc..

Before using a command defined in a package, you have to load the package using the **`with`** command. For example, the **`animate`** command is defined in the **`plots`** package. We will load the package with:

```
with(plots);
```

[animate, animate3d, animatecurve, arrow, changecoords, complexplot, complexplot3d, conformal, conformal3d, contourplot, contourplot3d, coordplot, coordplot3d, densityplot, display, dualaxisplot, fieldplot, fieldplot3d, gradplot, gradplot3d, graphplot3d, implicitplot, implicitplot3d, inequal, interactive, interactiveparams, intersectplot, listcontplot, listcontplot3d, listdensityplot, listplot, listplot3d, loglogplot, logplot, matrixplot, multiple, odeplot, pareto, plotcompare, pointplot, pointplot3d, polarplot, polygonplot, polygonplot3d, polyhedra_supported, polyhedraplot, rootlocus, semilogplot, setcolors, setoptions, setoptions3d, spacecurve, sparsematrixplot, surfdata, textplot, textplot3d, tubeplot]

The output shows all of the commands defined in the **`plots`** package that have been loaded. (You can hide this list by using a colon to terminate the previous command line.)

Now we can use the **`animate`** command:

```
animate(sin(2*x*t), x=-2..2, t=-2..2);
```

Now, click on the picture, then click the play button near the top of the window and the show starts! (Chapter 26 discusses animation in detail.)

Using a Command Without Loading the Package

You can also load and use a single command from a package without loading the whole package. For example, if you only want to use the **polygonplot** command defined in the **plots** package, you can type:

plots[polygonplot](*input values for the command* **);**

Maple will then know where to find the definition of that command. However, since you have not loaded the **polygonplot** command, the next time you use it, you have to type **plots[polygonplot]** again.

As another example, if you want to use the **ArcLength** command which is defined in the subpackage **Calculus1** of the **Student** package, you type:

Student[Calculus1][ArcLength](*input values for the command***);**

Useful Tips

The help pages contain examples showing how the commands can be used. Pick the one or two sample commands that are closest to what you are trying to do, execute them, and see how they work.

Another way of invoking help is to use the **help** or **?** commands:

help(*command* **);** #or you can use

?*command***;**

For example, to get help on the **fsolve** command, you can type:

?fsolve *enter*

When you load a package, finish the command with a colon instead of a semicolon,

with(*package* **):**

Then Maple will not display the whole list of the commands in that package, which can be very lengthy.

You can also load a package from the menu bar. In the **Tools** menu, select "**Load Package**", then select the package you want to load. The list shows the most commonly-used packages, and the last item "**List All Packages...**" lets you see the complete list of packages available.

To unload a package, use the **unwith** command.

unwith(*package* **):**

CHAPTER 9

Making 2-D Pictures

Drawing the Graph of a Function

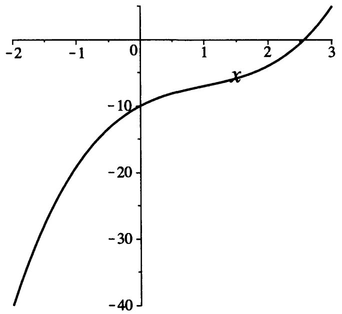
The plot Command

A common operation in mathematics is to plot the graph of a function, such as a polynomial, over a given interval. Maple does this with its **plot** command. To plot a function of x over an interval $a \le x \le b$, you type:

```
plot( function, x = a..b );
```

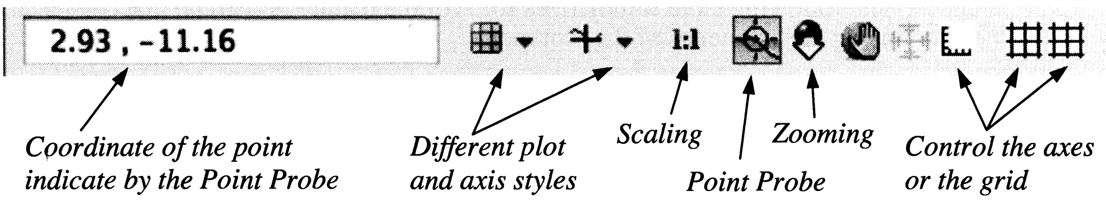
For example, the following plots the graph of $y = x^3 - 3x^2 + 5x - 10$ for x in the interval $-2 \le x \le 3$.

```
plot(x^3-3*x^2+5*x-10, x=-2..3);
```

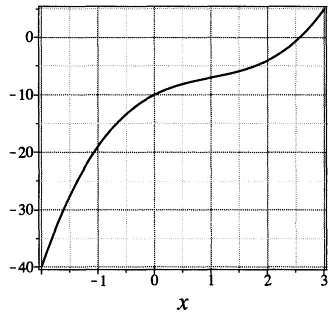


Plot Options Tool Bar

When you click once on a picture from the **plot** command, the **Plot options** tool bar will show up near the top of the worksheet. By clicking various buttons in the options bar, you can alter the appearance of the picture with different plot styles, axes styles, and scales.

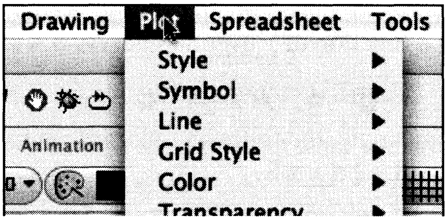


For example, the following graph shows the result of plotting the same graph as above, then choosing the boxed icon for the axes and adding a grid of lines.

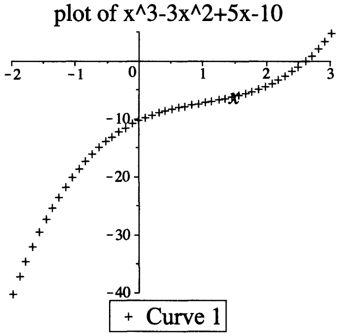


Plot Options Menus

When you click on a picture, the sub-menus of the **Plot** menu in the menu bar at the top of the Maple window become active, as you see in the picture below. This lets you control even more options for the plot.



You can then ask Maple to redraw the graphic by choosing various options from these menus. For example, we can repeat the `plot` command above, then add a title, and legend, change the style to points, and make the symbol a cross to obtain the graph below.



The following table summarizes some of the feature of a graph that can be controlled using the submenus of the Plot menu.

Menu	*Comment*
Style	Choose if graphs are drawn as curves or collections of points
Symbol / Line	Choose styles for lines and points in a picture.
Legend	Add and edit legends to a picture. Most useful when the plot has more than one function.
Title	Add and edit a title and caption to the picture

Axes	Choose a style for the axes. The **`properties`** option opens a dialog box that lets you set the viewing window for the plot.
Projection	Choose either **`constrained`** or **`unconstrained`** scaling. Serves same purpose as the 1:1 button in context menu bar.
Manipulator	Controls the action produced by a mouse click or drag in the plot.
Export	Lets you export the picture in various formats, such as postscript, GIF, JPG, and so on.

Note: You should use **`constrained`** scaling in the **`Projection`** menu or the 1:1 button on the **Plot options** tool bar, whenever you want to see angles or circles properly. For example, a circle can look like an ellipse unless you specify **`constrained`** scaling.

More Options with the plot Command

Plotting Multiple Functions

The **`plot`** command allows you to graph several functions or expressions simultaneously, all on the same set of axes, over a common interval $a \le x \le b$. You use the following format.

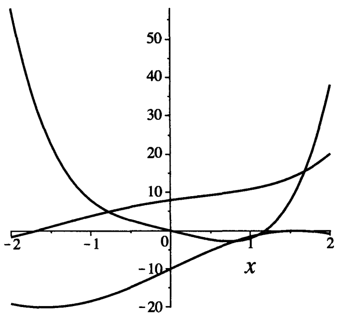
`plot([` *function1*, *function2*, *etc.* **`], x =`** *a*..*b* **`);`**

For example, to see the graphs:

$$y = 3x^4 - 5x\,,\ y = 10\sin(x) - 10\,,\ \text{and}\ y = 5\cos(x) + 3e^x$$

on the interval $-2 \le x \le 2$, you type:

```
plot([3*x^4-5*x, 10*sin(x)-10, 5*cos(x) + 3*exp(x)],
      x = -2..2);
```



Notice the use of [*square brackets*] to group the functions together. This indicates the order in which the graphs are going to be added to the plot. A drawback to this command is that all the functions are plotted on the same domain (in this case **`x =`**

-2..2). Later in this chapter, we will see another method to combine different pictures.

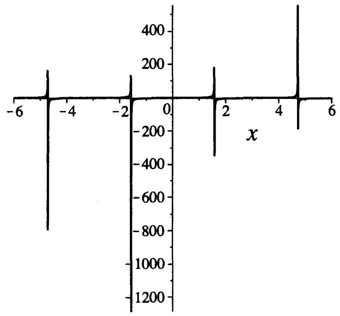
When drawing multiple functions, you may also want to use a legend to identify the graphs. You do this by choosing the **Show Legend** option in the **Legend** menu under the **Plot** menu. You can also use the **Edit Legend** menu to specify a legend.

Restricting the *y*-axis

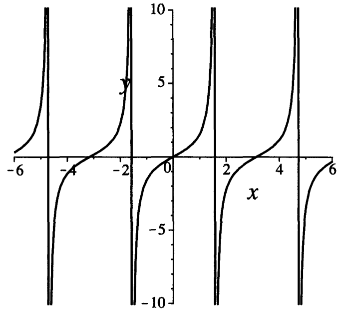
If a graph contains a vertical asymptote or has a very large range of *y*-values, some of its interesting features may not be visible. That's because Maple tries to show you the whole graph. The solution is to tell Maple what portion of the viewing window you want to see.

For example, the graph of the tangent function has vertical asymptotes. Consider the difference between Maple's default picture and one where we restrict the graph to show only *y*-values with $-10 \le y \le 10$.

```
plot( tan(x), x=-6..6 );
```



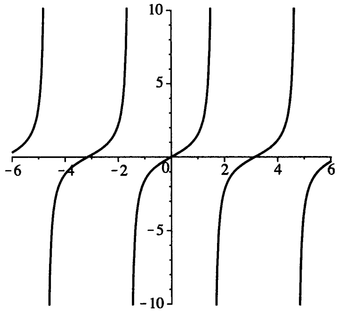

```
plot( tan(x), x=-6..6, y=-10..10 );
```



Working with Asymptotes and Discontinuities

Maple draws asymptotes in the picture above because the curve is constructed by connecting points, left to right, within the specified domain. You can instruct Maple to not connect points across a discontinuity by adding the optional argument **discont=true** in the **plot** command.

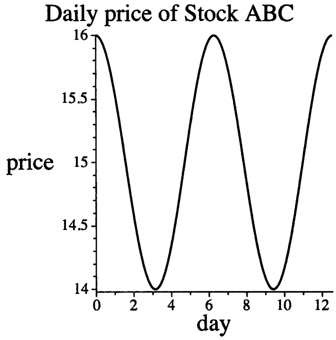
```
plot( tan(x), x=-6..6 , y=-10..10, discont=true);
```



Labeling Pictures

You can add a title to a plot and change the labels on the axes by using the **title** and **labels** options in the **plot** command.

```
plot( 15+cos(x), x=0..4*Pi ,
   labels = [ "day", "price" ],
   title = "Daily price of Stock ABC" );
```



More Advanced Drawings

Colors

You can change the color of a curve by specifying the **color** option inside the **plot** command. You have 155 pre-defined colors to choose from. Some of the colors Maple knows are:

"Aquamarine", "Black", "Blue", "Navy", "Coral", "Cyan", "Brown", "Gold", "Green", "Grey", "Magenta", "Maroon", "Orange", "Pink", "Plum", "Red", "Tan", "Turquoise", "Violet", "Wheat", "White", and **"Yellow"**.

Say, let us draw a picture in orange:

```
plot(x^3+2*x, x=-3..2, color = "Orange");
```

Sorry, we cannot show you the picture here, because this text is printed in black and white!

Alternatively, we could use the **COLOR** function with specified **RGB** values to create a color. This is done using a **COLOR** structure in which the amounts of red, blue, and green light in the final color are specified. For example, **COLOR(RGB,1,0,0)** is red, while **COLOR(RGB,1,0,1)** is purple.

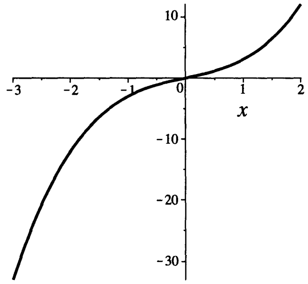
We can redraw the same picture in orange using **RGB** values of 1, 0.647 and 0.

```
plot(x^3+2*x, x=-3..2, color = COLOR(RGB,1,0.647,0));
```

Thickness

One way to distinguish curves in a black-and-white graph is to control the thickness of a curve. This is done by setting the **thickness** option to have the value 1 for a thin pen (default), 2 for a medium pen, 3 for a thick pen, 4 for a thicker pen, etc.. The larger the value, the thicker will be the graph.

```
plot(x^3+2*x, x=-3..2, thickness = 3);
```



Combining Plots with the display Command

You can combine different plots with different styles into a single picture by using the **display** command, which is loaded as part of the **plots** package.

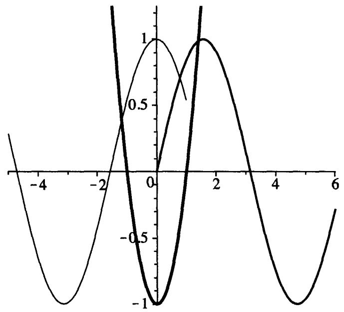
First, name each of the plots. Then, use the **display** command to combine these named graphics into a single output.

```
graph1 := plot( cos(x), x=-5..1, color=green):
graph2 := plot( sin(x), x=0..6,
                        thickness=2, color=yellow):
graph3 := plot( x^2-1, x=-1.5..1.5,
                        thickness=3, color=red):
```

We ended each command above with a colon instead of a semicolon. The colon serves the same purpose as a semicolon, except that the colon indicates that Maple should perform the desired calculation internally without displaying the result.

In this case, if we had used a semicolon, we could have still stored the plots, but then Maple would have displayed a harmless output "*PLOT*(...)."

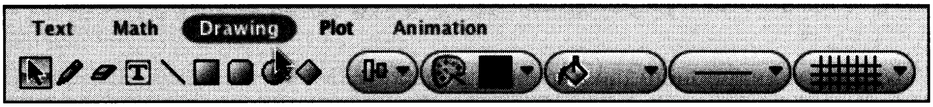
```
with(plots):     # We have to load the plots package to use display.
display( [ graph1, graph2, graph3 ] );
```



Annotating a Plot with Text and Drawings

The Drawing Tool Bar

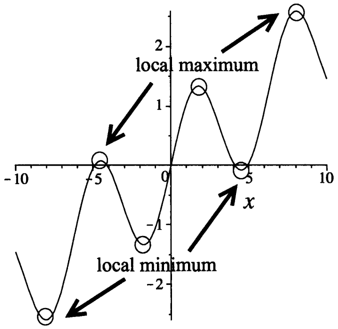
Maple lets you annotate a plot by adding text and drawings. When you click once on a plot, the **Plot** options tool bar will show up near the top of the worksheet. Click on the **Drawing** button (see the picture below), the Drawing tool bar will show up with the standard drawing tools. You can use these tools to work directly on the plot.



For example, we can start with the `plot` command:

```
plot(sin(x)+.2*x, x = -10 .. 10);
```

The **Drawing** tool can be used to indicate the local maximum and minimum points of the graph. This gives the annotated graph.

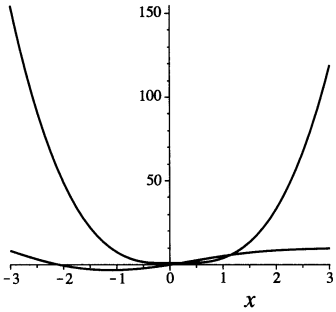


More Examples

Zooming In

■ **Example.** Consider $f(x) = \sqrt{1+10x^4-20x^5+25x^6}$ and $g(x) = x^2 + 5\sin(x)$:

```
f := x -> sqrt( 1 + 10*x^4 - 20*x^5 + 25*x^6);
g := x -> x^2 + 5*sin(x);
plot([f(x),g(x)], x=-3..3);
```



To get a better estimate of the intersection near 1, we can zoom in on the graph by successively shrinking down the x-interval in the **plot** command.

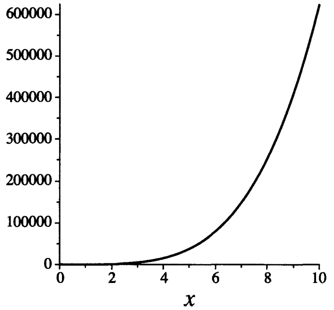
```
plot([f(x),g(x)], x=1 .. 1.5);
plot([f(x),g(x)], x=1.13 .. 1.15);
plot([f(x),g(x)], x=1.135 .. 1.145);
```

"Optical Illusion"

■ **Example.** Suppose we plot the function $f(x) = 64x^4 - 16x^3 + x^2$:

```
f := x -> 64*x^4-16*x^3+x^2;
```

```
plot(f(x), x=0..10);
```



You can easily get the impression that the graph is always increasing. However, notice that the vertical range is very large, between 0 and 600,000, so the picture is not sharp enough to show any small dips. In particular, the graph looks like a straight line for x between 0 and 2; this certainly is not the case.

```
plot(f(x), x=0..2);      # Still looks OK.
plot(f(x), x=0..1);      # Still looks OK.
plot(f(x), x=0..0.5);    # Still looks OK.
plot(f(x), x=0..0.2);    # Surprise!!
```

Useful Tips

If you're trying to arrange several graphs with different plot options on the same graph, it's always easier to draw the graphs separately and then use the **display** command to put them together.

Maple uses a variety of colors to draw 2-D graphs. These may look pleasing on screen, but may not look very good when printed on a gray scale printer. We suggest that unless you have a color printer available, you add the option **color = black** to your plots.

If you have a **plot** that involves complicated functions, always try it first with one function at a time to make sure that each **plot** comes out OK by itself.

You will find other methods of making 2-D pictures in Chapters 10 and 11. Also, there are many other options in **plot** that we have not discussed. You can find them using **?plot[options]** and experiment with them.

Troubleshooting Q & A

Question... I got an empty picture from **plot** with an error message about "...`empty plot.`" What went wrong?

Answer... This usually indicates that Maple cannot evaluate your input function numerically. Check whether you made a typo in the input. Some common mistakes are:

- You typed the name of a built-in function incorrectly.
- You used the wrong variable.
- You specified an interval in which the input function is not defined.

Check whether your function really gives numbers! Do you get numbers when you enter **f(–1.)**, **f(0.)**, **f(1.)** and so on?

Question... When I combined various pictures using the **display** command, I got the reply *display(PLOT(......* but no picture. Where was my mistake?

Answer... Make sure that you load the **plots** package before using the **display** command. Type:

```
with(plots);
```

and re-enter your **display** command.

Question... When I combined various plots using the **display** command, I got an error message. Where was my mistake?

Answer... Check if you typed the names of the plots correctly in your **display** command. If there is no misspelling there, then you should **display** each picture one at a time in order to find out which one is causing the problem. Then recheck their definitions. (A common mistake is to use = instead of := in defining the plots.)

CHAPTER 10

Plotting Parametric Curves Line Segments, and Points

Parametric Curves

Plotting Parametric Curves

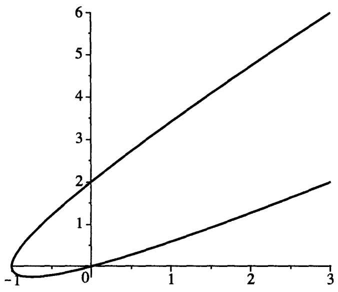
In the previous chapter you saw how to use **plot** to draw curves that are graphs of functions. But not all curves are the graphs of functions.

A two-dimensional (2-D) parametric curve is written in the form $(x(t),\ y(t))$. The **plot** command can be used to draw the curve $(x(t),\ y(t))$, defined on an interval $a \le t \le b$. The command has the form:

```
plot( [ x(t), y(t), t = a..b ] );
```

For example, to see the curve $(t^2-1,\ t+t^2)$ for $-2 \le t \le 2$, we use:

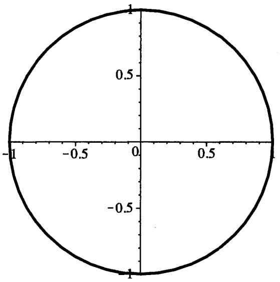
```
plot( [t^2-1, t+t^2, t=-2..2] );
```



Plot Options

The options that we discussed in the previous chapter (e.g., **color, scaling** and **thickness**) also work for this particular format of **plot** command. For example, to draw the unit circle $(\cos t,\ \sin t)$ for $0 \le t \le 2\pi$ with a blue color, and the two axes measured in the same scale, you use:

```
plot( [cos(t),sin(t), t=0..2*Pi], thickness = 3,
      scaling = constrained, color = blue);
```



Plotting Line Segments and Polygonal Paths

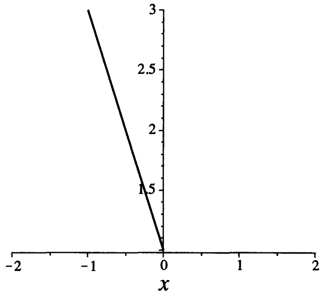
Line Segments

The **plot** command can also be used to plot a line segment joining the points (x_0, y_0) and (x_1, y_1). The syntax is:

```
plot( [[x0, y0], [x1, y1]], x = a..b );
```

For example, to see the line segment joining (0, 1) to (–1, 3):

```
plot( [[0,1],[-1,3]], x=-2..2 );
```



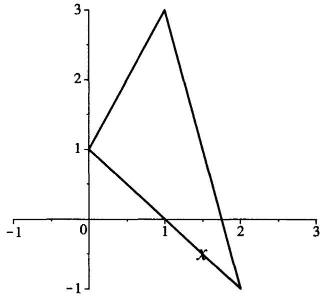
Polygonal Paths

A similar construction is used to plot a polygonal path connecting *n* points.

```
plot([[x0, y0], [x1, y1], ..... ,[xn, yn]], x = a..b);
```

For example, to plot the triangle with vertices (0, 1), (1, 3), and (2, –1) you type:

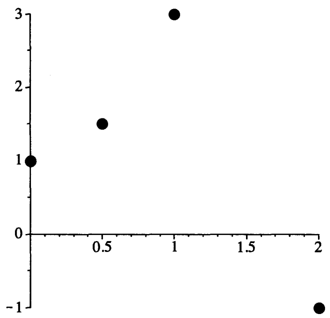
```
plot([[0,1],[1,3],[2,-1],[0,1]], x=-1..3);
```



Plotting Individual Points

To have Maple plot only points without drawing the line segments that connect them, include the optional argument **`style=point`**. The shape and size of each point is controlled by the **`symbol`** and **`symbolsize`** options, respectively. This is demonstrated in the following example.

```
plot([[0,1],[1,3],[2,-1], [0.5,1.5]], style = point,
     symbol = solidcircle, symbolsize = 18);
```



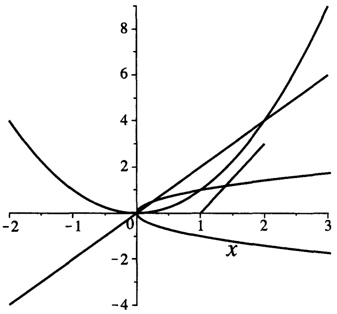
Plotting Multiple Curves in One Plot

Multiple Curves at Once

You can plot several parametric curves, graphs, and lines all in one picture, using a single **`plot`** command. To do this, you put all the expressions of the curves inside **[** *square brackets* **]**.

■ **Example**. To draw the curves (t^2, t) for $-2 \le t \le 2$, and (t, t^2) for $-1 \le t \le 1$, the line segment from (1, 0) to (2, 3), and the graphs $y = 2x$, $y = x^2$ for $-2 \le x \le 3$, we can use:

```
plot( [ [t^2,t, t=-2..2], [t,t^2, t=-1..1],
        [[1,0],[2,3]], 2*x, x^2 ], x=-2..3 );
```

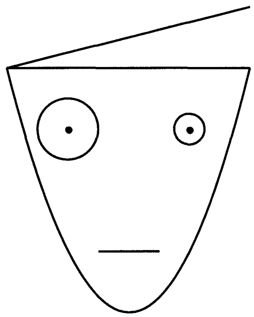


Multiple Pictures Together

You can also combine several 2-D pictures that you've already created with the **`display`** command. (The **`display`** command is defined inside the **`plots`** package. This approach is used when the pictures you want to combine do not have the same viewing window or do not all use the same options.

■ **Example**. To create the "open skull" picture that you see below, we will combine several pictures. The face is made from a parabola and two straight lines; the eyes from two circles; and the mouth from a straight line.

```
face := plot([x^2, 4, 4.5+x/4],x=-2..2, color = red):
eyes := plot([ [1+cos(t)/4, 3+sin(t)/4, t=0..2*Pi],
               [-1+cos(t)/2, 3+sin(t)/2, t=0..2*Pi]],
               color = brown):
pupil := plot([[-1,3],[1,3]], style=point,
               color=green, symbol=solidcircle,
               symbolsize=18 ):
mouth := plot([[-0.5,1],[0.5,1]], color = blue):
with(plots):
display([face,eyes,pupil,mouth],
        scaling=constrained, axes=none );
```



Useful Tips

💡💡💡 You may be overwhelmed by all the different plotting formats that we have discussed. The following table can help you to remember them:

Expression of the Form	*What It Plots*
`plot(` [$f_1(x)$, $f_2(x)$], `x =` $a..b$);	Graphs of $f_1(x)$, $f_2(x)$ for $0 \le x \le b$
`plot(` [$x(t)$, $y(t)$, `t =` $a..b$]);	Parametric curve $(x(t), y(t))$, $a \le t \le b$
`plot(` [[a, b], [c, d]]);	Line segment from (a, b) to (c, d)
`plot(`[[a, b], [c, d]], `style=point);`	Points (a, b) and (c, d)

💡💡 If you need to draw a complicated picture that consists of several graphs or curves, always draw each picture individually and combine them with the **`display`** command. You may be able to use a single **`plot`** command to draw the entire picture, but if you get an error message, it can be difficult to locate the mistakes.

Troubleshooting Q & A

Question... I tried to draw one parametric curve with **plot**, but instead Maple gave me a picture of two curves. What went wrong?

Answer... When entering a parametric curve, it is common to forget to include the interval *inside* the [*square brackets*]. For example, instead of typing **plot([t, t^2, t = –2..2]);** you may make the mistake of typing:

```
plot([t, t^2], t=-2..2);
```

Then Maple will draw the *graphs* $y = t$ and $y = t^2$!! This is not what you want.

Question... I tried to draw a parametric curve with **plot** but got an error message "`Error, (in plot) unexpected options: ....`" What should I check?

Answer... This error message suggests that you should check the syntax of the **plot** command. Make sure that you used **[** *square brackets* **]** and **(** *parentheses* **)** correctly . For example, a common mistake is to input the expression **(t, t^2)** or **(t, t^2, t = –2..2)** instead of **[t, t^2, t = –2..2]** for the parametric curve (t, t^2), $-2 \le t \le 2$.

Question... I tried to draw a parametric curve with **plot** but got an error message "`Error, (in plot) expecting a real constant as range ....`" What should I check?

Answer... This error message suggests that you made a mistake in specifying the range for the parameter. For example, check to see if you used a variable to define the range. Also, a common mistake is to use **pi** instead of **Pi**.

Question... I tried to graph a parametric curve with **plot** but got an error message. What went wrong?

Answer... This usually indicates that Maple cannot evaluate your input function numerically. Check whether you made a typo in the input. Common mistakes include:

- You typed the wrong name of a built-in function.
- You used the wrong variable.
- You specified an interval in which the input function is not defined.

Question... I tried to use one of the commands from the **plots** package but Maple just echoes back the command. Why won't Maple execute the command?

Answer... The most common cause of this problem is not loading the **plots** package. Enter and execute the command **with(plots):** and re-execute your command.

CHAPTER 11

Polar and Implicit Plots

Plotting in Polar Coordinates

The polarplot Command

Curves expressed in polar coordinates can be graphed with the **polarplot** command. To access this command it is necessary to load the **plots** package. (See the discussion of packages in Chapter 8.)

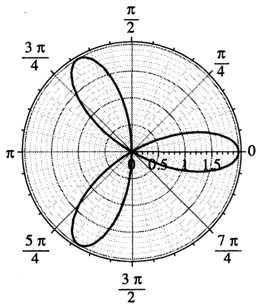
```
with(plots):
```

If the curve is given by $r = f(\theta)$, for $\theta_1 \le \theta \le \theta_2$, we can draw it with the following:

```
polarplot(f(θ), theta = θ1 .. θ2);
```

Notice that this form is very similar to that of the **plot** command you already know. For example, to plot the three-leaf rose, use:

```
with(plots):
polarplot( 2*cos(3*theta) , theta = 0..2*Pi );
```

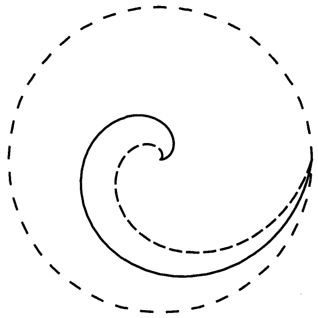


Plotting Multiple Curves with Style

Just as with the **plot** command, several curves can be plotted at once with the **polarplot** command by enclosing them in **[** *square brackets* **]**, separated by commas. Also, you can supply many of the options that work with the **plot** command.

■ **Example**. A command to graph the spirals $r = \frac{\theta}{2\pi}$, $r = \left(\frac{\theta}{2\pi}\right)^2$, and the circle $r = 1$, all in one plot, is:

```
polarplot( [ theta/(2*Pi), (theta/(2*Pi))^2, 1 ],
           theta=0..2*Pi, axes=none,
           linestyle=[1,3,6], color=black);
```



Notice how the **linestyle** option is used to make the curves easily identifiable in a black-and-white plot by dashing lines. The **axes=none** instructs Maple to omit the polar axes.

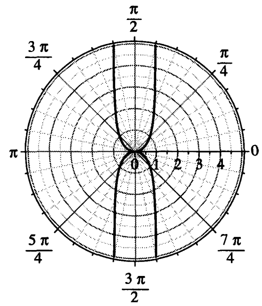
Cartesian Plotting Tricks Revisited

When plotting in polar coordinates, we face many of the same issues we found in Cartesian coordinates in order to get a nice picture. For example, the graph of $r = \tan(\theta)$ has discontinuities and produces very large values of r. Thus the command:

```
polarplot(tan(theta), theta=0..2*Pi);
```

will not show much detail, especially near the origin. We will get a better picture by limiting the values of r and θ using the **coordinateview** option. Also we use **discont=true** to take care of the spurious lines from the discontinuities.

```
polarplot( tan(theta), theta=0..2*Pi,
           coordinateview=[0..5, 0..2*Pi],
           discont=true);
```



Plotting Graphs of Equations

The implicitplot Command

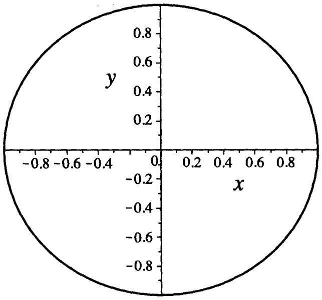
When a curve is given by an equation in variables x and y, you can draw the curve with the **implicitplot** command (defined in the **plots** package).

You use the command in the form:

```
with(plots):
implicitplot( an equation in x and y , x = a..b, y = c..d);
```

For example, to see the unit circle $x^2+y^2=1$ for $-1 \le x \le 1$, $-1 \le y \le 1$, use:

```
with(plots):
implicitplot( x^2 + y^2 = 1, x=-1..1, y=-1..1);
```



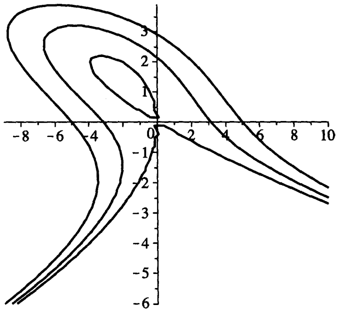
(Note that the circle is not round because the x and y axes are not of the same scale. You can correct this by adding the option **scaling = constrained**.)

Multiple Curves and Styles

You can also use [*square brackets*] or the **display** command to plot multiple implicit curves. The format is the same as you've used already with the **plot** and **polarplot** commands. Many of the options you use with the **plot** command work with **implicitplot** as well.

■ **Example**. We want to see the curves $x^2+3xy+y^3=25$, $x^2+3xy+y^3=10$, and $x^2+3xy+y^3=0$ in the intervals $-10 \le x \le 10$ and $-6 \le y \le 4$. Let's draw each curve with a different color:

```
f := (x,y) -> x^2+3*x*y+y^3;
pict1 := implicitplot( f(x,y) = 25, x=-10..10,
            y=-6..4, color = red):
pict2 := implicitplot( f(x,y) = 10, x=-10..10,
            y=-6..4, color = yellow):
pict3 := implicitplot( f(x,y) = 0, x=-10..10,
            y=-6..4, color = green):
display( [pict1,pict2,pict3] );
```



This picture looks like the Chinese character for "Wind." We can get the same plot with a single call to **implicitplot** as follows:

```
implicitplot( [ f(x, y)=25, f(x, y)=10, f(x, y)=0 ],
              x = -10..10, y = -6..4,
              color = [red, yellow, green]);
```

Useful Tips

You may be overwhelmed by all the different plotting commands that we have discussed. Don't worry, because **plot** is the most commonly used command, while **polarplot** and **implicitplot** are for special situations:

To Draw	*Use*
Graph of a function $f(x)$	**plot**
Parametric curve $(x(t), y(t))$	**plot**
Curve in polar coordinates	**polarplot**
Curve given by an equation	**implicitplot**

Beginning with Maple 12, the **polarplot** command adds a polar grid and uses constrained scaling by default. The grid can be removed either by modifying the plot with the toolbar after execution or by using the **axes=none** or **gridlines =false** option in the command. You can change the axes to Cartesian with the **axiscoordinates = cartesian** option.

The **display** command lets you combine the results from different kinds of 2D-plot commands (such as **plot**, **polarplot**, **implicitplot**) on a single graph.

A parametric curve of the form $x = r(t)\cos(\theta(t)),\ \ y = r(t)\sin(\theta(t))$, can be plotted using polar coordinates with the **polarplot** command or the **plot** command (using the **coords=polar** option) in the following formats.

```
polarplot([ r(t), θ(t) ,t = a..b]);
plot([ r(t), θ(t) ,t = a..b], coords=polar);
```

The **algcurves** package can be used for graphing polynomial equations implicitly. The **plot_real_curve** command in this package usually draws a more accurate curve. For example:

```
implicitplot(x^2=y^3, x=-3..3, y=-3..3,
             view=[-3..3,-3..3]);
```

and

```
with( algcurves ):
plot_real_curve( x^2-y^3, x, y, view=[-3..3,-3..3] );
```

draw the same curve, but the **plot_real_curve** command works better near the critical point.

Troubleshooting Q & A

Question... When I tried to use the **polarplot** or **implicitplot** command, Maple gave the command back without doing anything. What happened?

Answer... Usually, there are two reasons why this might happen:

- You probably forgot to load the **plots** package before using these commands. Enter the following, and then try your commend again.

```
with(plots):
```

- Check to see that you correctly spelled the command names **polarplot** or **implicitplot**.

Question... I got a warning message "`... unable to evaluate the function`" and then Maple showed a picture of a horizontal line with labeled polar coordinate axes. Is that picture correct?

Answer... No, that picture is not correct. Check whether you made a typo in the input. Common mistakes are:

- You mistyped the name of a built-in function/variable.
- Your input function contains a variable other than **theta** (a common mistake is to include **r**, the radius variable, in the input).
- You did not match up parameters correctly (e.g., you used **t** in the function but used **theta** when you specified the interval).

Question... I got an error message from **implicitplot**. What should I look for?

Answer... There are two major problem areas in using **implicitplot**.

- Make sure that the equation you entered is really an equation with an equal sign "=". Check that the equation is entered correctly. Did you remember to type * for multiplication?
- Make sure that your equation has two variables that do not have values. Executing **unassign('x','y');** before using **x** and **y** in your equation for **implicitplot** is highly recommended.

Question... The plot produced by **implicitplot** is very coarse, or it missed some parts of the level curve. What can I do to improve the quality of the plot?

Answer... The **implicitplot** command works by searching a mesh of points in the viewing window. By default this grid is 26×26. The size of this grid can be controlled by the **grid** option in **implicitplot**. For example,

```
f := (x,y) -> x^2+y^2:
implicitplot( [f(x,y)=1, f(x,y)=0.73, f(x,y)=0.31],
              x=-3..3, y=-3..3, grid=[90,90] );
```

CHAPTER 16

Making Graphs in Space

Graphing Functions of Two Variables

The plot3d Command

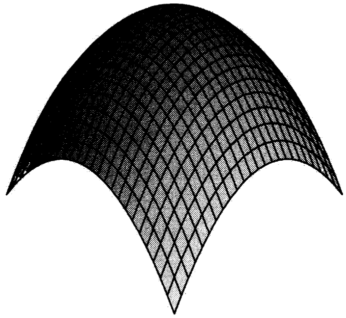
Plotting the graphs of functions of two variables with Maple parallels the graphing of functions of one variable. The main change is using **plot3d** instead of **plot**.

In the **plot3d** command, you input an expression in terms of the independent variables x and y, and specify bounds for the x- and y-variables as $x_0 \le x \le x_1$ and $y_0 \le y \le y_1$. The **plot3d** command then has the form:

plot3d(*an expression of x and y*, **x =** x_0**..**x_1**, y =** y_0**..**y_1 **);**

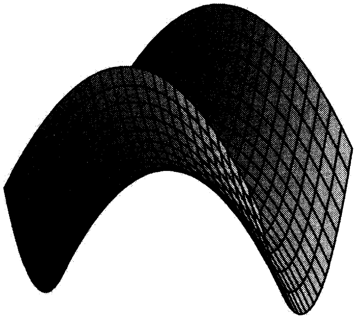
For example, the surface whose height is $z = 4 - x^2 - y^2$ above the xy-plane, over the rectangle $-2 \le x \le 2$ and $-2 \le y \le 2$, is seen with:

```
plot3d( 4-x^2-y^2, x = -2..2, y = -2..2);
```



■ **Example.** The graph of the function $f(x,y) = x^2 - y^2$ looks like a saddle.

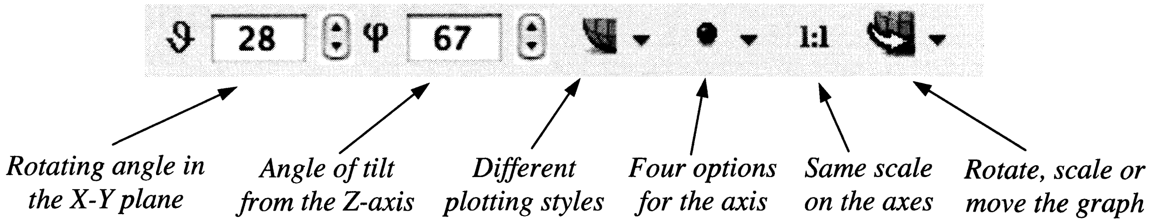
```
plot3d( x^2 - y^2, x = -3..3 , y = -3..3 );
```



Changing the 3-D View and options from the tool bar and menu bar

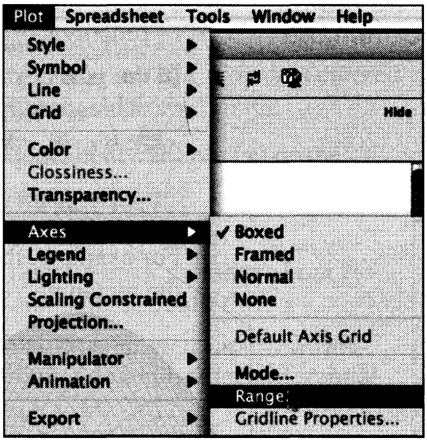
Move the cursor to any point in Maple's graphical output, then click and hold the (left) mouse button. As you drag the mouse with the button held down, the picture rotates. This allows you to see the picture from any viewpoint.

When you click on the picture the context bar changes, showing the viewing angles and a series of drop down menus for viewing options. You can use these controls to choose the type of display you want.



Clicking on a 3-D graph also changes the menu bar at the top of the page, activating the settings in the **Plot** menu. The menus allow you to change the settings for several options, including **Style**, **Symbol**, **Line**, **Color**, **Axes**, **Legend**, **Projection**, and **Lighting**

For example, if you create a picture using **`plot3d`**, you may want to use the **Ranges** option under the **Axes** menu to adjust the intervals for the *x, y* and *z* axes. (See the picture on the right.)

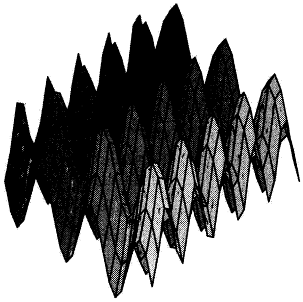


Specifying Options

The grid Option

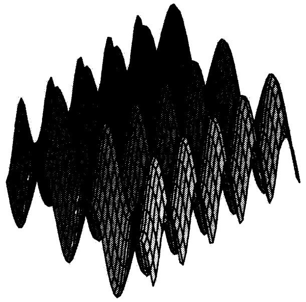
The **`grid`** option is used to instruct **`plot3d`** to sample more points when constructing a plot. Consider:

```
plot3d( sin(3*y)+cos(5*x), x = -Pi ..Pi, y=-Pi..Pi );
```



The picture does not look very good. But if we add the option **`grid = [50,50]`**, we'll see the graph shown with values sampled from a 50×50 grid, instead of the default 25 × 25 grid. This gives a smoother picture, but takes about four times longer to compute and draw.

```
plot3d( sin(3*y)+cos(5*x), x = -Pi ..Pi, y=-Pi..Pi,
        grid=[50,50] );
```



The view Option

As with graphs of functions of one variable, we sometimes need to control the range of the dependent variable in 3-D graphs. With the **plot3d** command we use the **view** option. For example, if we try the graph

```
plot3d(1/(x^2+y^2), x=-2..2, y=-2..2);
```

The result is not very informative because the values of this function close to the origin are very large and distort the scale of the z-axis. If you try,

```
plot3d(1/(x^2+y^2), x=-2..2, y=-2..2, z=-5..5); #error
```

you will get an error message. However, you can use the **view** option as follow

```
plot3d(1/(x^2+y^2), x=-2..2, y=-2..2, view =-5..5);
```

This truncated graph enables you to see more of the detail in the surface.

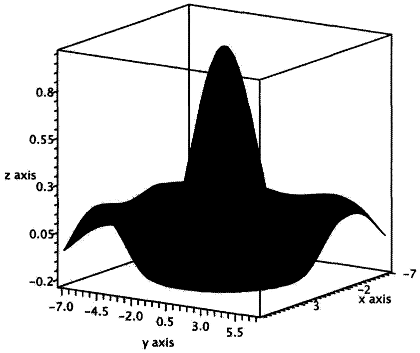
Other Useful Options

The following table summarizes some of the more common options that you can use to add more "character" to a picture with **plot3d**.

Option	*What It Does*
`axes = normal`	Shows the three axes in a normal way.
`axes = boxed`	Surrounds the picture with a box.
`labels = ["x","y","z"]`	Provides names to label each of the three axes.
`scaling = constrained`	Scales the graphic so that units in each direction have the same length.
`color =` *the color you want*	Draw the graphic with the specified color.

■ **Example.** The sombrero has equation $f(x,y) = \sin(\sqrt{x^2+y^2})/\sqrt{x^2+y^2}$:

```
S := sin(sqrt(x^2+y^2))/sqrt(x^2+y^2):
plot3d( S, x=-7..7, y=-7..7, grid=[50,50], color=red,
    axes=boxed, labels=["x axis","y axis","z axis"]);
```



Surfaces in Cylindrical and Spherical Coordinates

Surfaces are sometimes described using the cylindrical or spherical coordinate system. Maple can draw such surfaces easily by setting the **coords** option to **cylindrical** or **spherical**, respectively.

The coords = cylindrical Option

Points in the cylindrical coordinate system are described by quantities r, θ, and z, where

- r is the horizontal radial distance of the point from the z-axis;
- θ is the horizontal angle measured from the positive x-axis; and
- z is the vertical distance from the xy-plane.

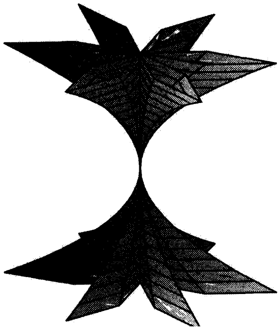
To draw the surface $r = f(\theta, z)$ for $\theta_0 \leq \theta \leq \theta_1$ and $z_0 \leq z \leq z_1$, you enter:

```
plot3d( f(θ,z), theta = θ0..θ1, z = z0..z1
        coords = cylindrical);
```

Note: When using **coords=cylindrical**, you must enter the interval for **theta** first, then the interval for **z**. You do not have to use these names, but Maple assumes this ordering when it creates the plot.

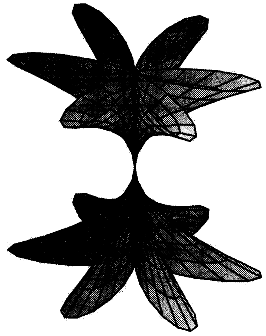
For example, to see the surface $r = z^2 \cos^2(3\theta)$, for $0 \leq \theta \leq 2\pi$ and $-2 \leq z \leq 2$:

```
plot3d( z^2*(cos(3*theta))^2, theta=0..2*Pi,
        z=-2..2, coords = cylindrical );
```



Notice that this surface is very "choppy" as the horizontal angle θ varies, yet the surface is quite smooth along the vertical z-direction. This suggests that we should increase the sampling of points in the θ variable (from 25 to, say, 80) but decrease the sampling in z (from 25 to, say, 15). We can do this with:

```
plot3d( z^2*(cos(3*theta))^2, theta=0..2*Pi,
        z=-2..2, coords=cylindrical, grid=[80,15]);
```



The coords = spherical Option

Points in the spherical coordinate system are described by quantities ρ, θ, and ϕ, where

- ρ is the radial distance in space of the point from the origin;
- θ is the horizontal angle measured from the positive x-axis; and
- ϕ is the vertical angle measured from the positive z-axis.

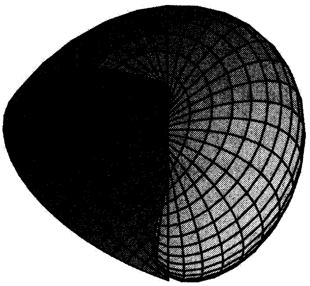
To draw the surface $\rho = f(\theta,\phi)$, $\theta_0 \le \theta \le \theta_1$, and $\phi_0 \le \phi \le \phi_1$, you will enter:

```
plot3d( f(θ,φ), theta = θ0..θ1, phi = φ0..φ1,
        coords = spherical );
```

> **Note:** When using **coords=spherical**, you have to enter the interval for **theta** first, then the interval for **phi**.

For example, to see the surface $\rho = \sqrt{\theta}(3+\cos\phi)$, $0 \le \theta \le 3\pi/2$, and $0 \le \phi \le \pi$:

```
plot3d( sqrt(theta)*(3+cos(phi)), theta =0..3*Pi/2,
        phi=0..Pi, coords=spherical);
```



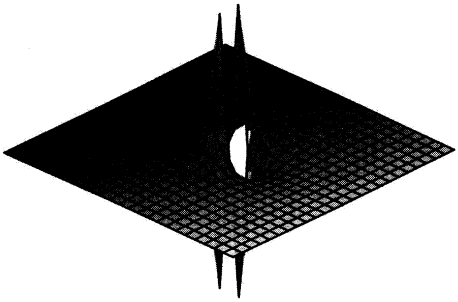
More Examples

Choosing the Right Coordinate Systems

In some cases, a surface given in rectangular coordinates will look better if you draw it using cylindrical or spherical coordinates.

■ **Example.** The surface $z = \dfrac{x^2 - y^2}{(x^2 + y^2)^2}$ can be plotted with:

```
plot3d( (x^2-y^2)/(x^2+y^2)^2, x=-3..3, y=-3..3 );
```

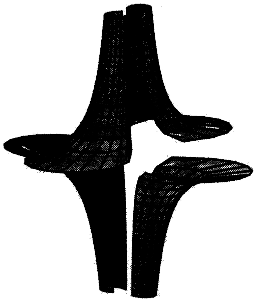


The picture is choppy especially near the origin. However, if we use cylindrical coordinates, the equation of the surface becomes

$$z = \frac{x^2 - y^2}{(x^2 + y^2)^2} = \frac{(r\cos\theta)^2 - (r\sin\theta)^2}{((r\cos\theta)^2 + (r\sin\theta)^2)^2} = \frac{r^2(\cos^2\theta - \sin^2\theta)}{r^4} = \frac{\cos 2\theta}{r^2},$$

which means $r^2 = \cos(2\theta)/z$, or equivalently $r = \sqrt{\cos(2\theta)/z}$. (The surface needs to be expressed in the form $r = f(\theta, z)$ in order to use **coords=cylindrical**.)

```
S := (theta,z) -> sqrt(cos(2*theta)/z):
top := plot3d( S(theta,z), theta=0..2*Pi,
       z=0.1..2, grid=[50,15], coords=cylindrical):
bottom := plot3d( S(theta,z), theta=0..2*Pi,
       z=-2..-0.1, grid=[50,15], coords=cylindrical):
plots[display]( [top,bottom], orientation=[36,77]);
```



This picture is nicer! Since this surface is most easily described with z as a function of r and θ, the graph is even nicer if we plot it as a parametric surface in cylindrical coordinates. We will do that in chapter 20.

Useful Tips

The **plot3d** command can be used to plot with non-rectangular domains. For example, to graph the part of the surface $z = x^2 + y$ above the region in the xy-plane bounded by $y = x^2$ and $y = 2 - x^2$ use:

```
plot3d( x^2+y, y=x^2..2-x^2, x=-1..1, axes=normal );
```

This approach can be used in other coordinate systems as well. (When viewed from above, **orientation = [0, 0]** , you see the projection of this surface onto the xy-plane.)

For 2-D graphs having multiple curves, Maple automatically uses a different color for each curve. For 3-D graphs, color variations represent depth. This means that if multiple surfaces are graphed together, they will have the same coloring scheme.

To designate separate colors for different surfaces in the same picture, you can either plot each one separately using a different **color** option and then use the **display** command to put them together, or you can use square brackets **[ ]** to make list of expressions to plot and make a corresponding list of colors. For example:

```
pict1 := plot3d(10-x^2, x=-3..3, y=-3..3, color=red):
pict2 := plot3d(y^2, x=-3..3, y=-3..3, color=green):
plots[display]( [pict1, pict2] );
```

or:

```
plot3d( [10-x^2, y^2], x=-3..3, y=-3..3,
        color=[red, green]);
```

If you want to create a 3-D graph of an expression that Maple has produced as output from a previous command, simply position the cursor over this expression and click the output with the right mouse button (or *control –* click for Mac users). A **context-sensitive menu** pops up. You can then select **Plots** → **3D-Plot** → **x,y**. Maple will automatically show you the picture. If you want to add more surfaces to this plot, simply select the expressions (expressed in x and y) and drag the input into the previous picture.

The interactive **Plot Builder** is useful for learning the syntax of new plotting options. You launch **Plot Builder** by selecting **Tools** → **Assistants** → **Plot Builder** from the menu bar. Enter the expressions to be plotted and click OK. Select the type of plot to be created, then click **Options**. Specify the settings that you want, clicking **Preview** to check the current status of the plot before clicking **Plot** to see the final plot. (Clicking **Command** returns the actual Maple plot command used to create the plot without actually showing the plot.)

You can use **plot3d** to draw parametric surfaces in space. We will discuss this in detail in Chapter 20.

Troubleshooting Q & A

Question... I tried to draw a 3-D picture but got an error message telling me about a "`Plotting error.`" What happened?

Answer... This means that Maple has trouble using the function you specified to generate points for the picture. Check these items:

- Did you mistype the input?
- Did you use the same variables in the function as you used in specifying the intervals?
- Is the function well-defined in the given intervals?

Another common mistake is to use the **`plot`** command to draw a 3-D picture; make sure you use the **`plot3d`** command.

Question... I drew the graph of a function using **`plot3d`**, but the picture looks different from the one shown in my textbook. Why is that?

Answer... Two plots may sometimes look different because they are drawn in different viewing windows, with different scalings, or from different perspectives. Adjust the viewing window with the **`view`**, the **`scaling`**, and **`orientation`** options respectively.

Question... Can I use **`display`** to combine the graphics created using context-sensitive menus?

Answer... You cannot use **`display`** to combine pictures constructed with context-sensitive menus, including drag-and-drop.

The pictures created from context-sensitive menus are based on the **`smartplot3d`** command. Although the outputs from **`smartplot3d`** and **`plot3d`** appear to be similar, their internal structures are quite different. The reason is that the pictures from **`smartplot3d`** allow several manipulations that are not offered in the standard plots.

For example, you can control-drag an expression into a **`smartplot3d`** picture to draw that expression. As a result, we cannot combine these pictures using **`display`** as we can with standard plots.

Question... The picture I got using **`coords = cylindrical`** or **`coords = spherical`** was completely wrong. What should I check?

Answer... Check these three areas:

- Make sure you typed the input function and the intervals of the two parameters correctly.
- With **`coords = cylindrical`** you have to enter the θ-interval **`theta`** **`=`** $\theta_0 .. \theta_1$ first, followed by the z-interval **`z =`** $z_0 .. z_1$. If you enter these in the wrong order, Maple will reverse the sense of the variables.
- With **`coords = spherical`** you have to enter the θ-interval **`theta =`** $\theta_0 .. \theta_1$ first, then the ϕ-interval **`phi =`** $\phi_0 .. \phi_1$. If you enter these in the wrong order, Maple will draw an incorrect picture.

CHAPTER 19

Matrices and Vectors

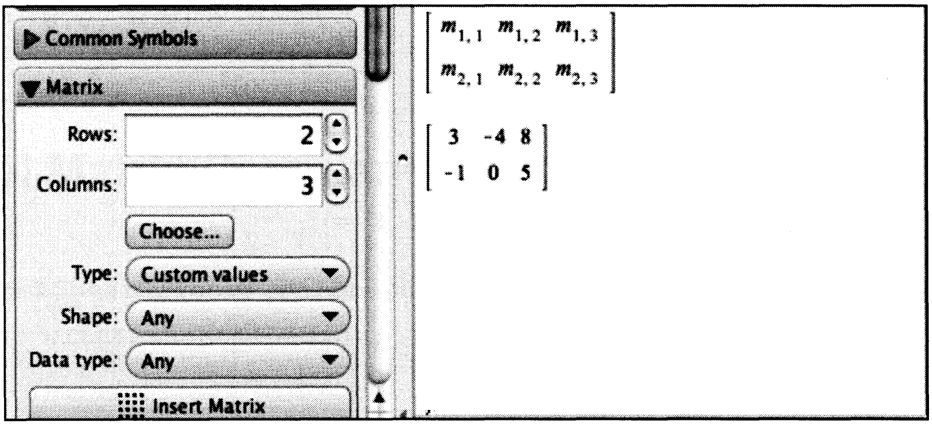
Matrices

Defining Matrices

The easiest way to define a matrix in Maple is by using the **Matrix** palette. For example, to enter the matrix $\begin{pmatrix} 3 & -4 & 8 \\ -1 & 0 & 5 \end{pmatrix}$, you select the **Matrix** palette, set the number of rows to 2, set the number of columns to 3, press the **Insert Matrix** button, then type:

Keystrokes: **3** *tab* **-4** *tab* **8** *tab* **-1** *tab* **0** *tab* **5**

The picture below shows the palette and the result of using it, both before and after typing the entries.



You can also use the **`Matrix`** command to define a matrix in the form:

`Matrix([ [` *row*$_1$ *entries* **`], [`** *row*$_2$ *entries* **`],`** *etc.* **`]);`**

For example:

```
Matrix([ [3,-4,8], [-1,0,5] ]);
```

$$\begin{bmatrix} 3 & -4 & 8 \\ -1 & 0 & 5 \end{bmatrix}$$

> **Note:** The **`Matrix`** command needs both the **`[`** *square brackets* **`]`** around the lists and **`(`** *parentheses* **`)`** at the outer level of the command. A shortcut construction using **<** *angle brackets* **>** is discussed a little later in this chapter, see also **?<>** .

Basic Operations Involving Matrices

You can do matrix addition and subtraction, scalar multiplication and scalar addition (adding a scalar to the main diagonal), multiplication of a matrix by a matrix or vector, and exponentiation of a square matrix by using the standard operators **+**, **–**, *****, **.** (dot), and **^**.

Consider the matrices $A=\begin{bmatrix}-1 & 0\\ 1 & 2\end{bmatrix}$, $B=\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}$ and $C=\begin{bmatrix}3 & -4 & 8\\ -1 & 0 & 5\end{bmatrix}$.

```
A := Matrix([[-1,0],[1,2]]):
B := Matrix([[1, 2],[3,4]]):
C := Matrix([[3, -4, 8],[-1,0,5]]):
```

Operation	*Example*	*Maple Command*
Addition (**+**)	$A+B=\begin{bmatrix}-1 & 0\\ 1 & 2\end{bmatrix}+\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}$	**A + B;** $\begin{bmatrix}0 & 2\\ 4 & 6\end{bmatrix}$
Subtraction (**-**)	$A-B=\begin{bmatrix}-1 & 0\\ 1 & 2\end{bmatrix}-\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}$	**A - B;** $\begin{bmatrix}-2 & -2\\ -2 & -2\end{bmatrix}$
Scalar multiplication (*****)	$5C=5\begin{bmatrix}3 & -4 & 8\\ -1 & 0 & 5\end{bmatrix}$	**5*C;** $\begin{bmatrix}15 & -20 & 40\\ -5 & 0 & 25\end{bmatrix}$
Matrix multiplication (**.**)	$AC=\begin{bmatrix}-1 & 0\\ 1 & 2\end{bmatrix}\begin{bmatrix}3 & -4 & 8\\ -1 & 0 & 5\end{bmatrix}$	**A.C;** $\begin{bmatrix}-3 & 4 & -8\\ 1 & -4 & 18\end{bmatrix}$
Matrix Power (**^**)	$A^5=\begin{bmatrix}-1 & 0\\ 1 & 2\end{bmatrix}^5$	**A^5;** $\begin{bmatrix}-1 & 0\\ 11 & 32\end{bmatrix}$
Inverse of a Matrix**)** (**^(-1)**)	$A^{-1}=\begin{bmatrix}-1 & 0\\ 1 & 2\end{bmatrix}^{-1}$	**A^(-1);** $\begin{bmatrix}-1 & 0\\ \frac{1}{2} & \frac{1}{2}\end{bmatrix}$
Transpose of a Matrix**)** (**^%T)**	$A^T=\begin{bmatrix}-1 & 0\\ 1 & 2\end{bmatrix}^T$	**A^%T;** $\begin{bmatrix}-1 & 1\\ 0 & 2\end{bmatrix}$

Vectors

Column and Row Vectors

We use the same **Matrix** palette for creating vectors. If we reduce the number of rows or columns to one, the **Insert Matrix** button becomes an **Insert Vector** button. We can then create a row or column vector.

Shortcut Syntax

There is a shorthand syntax for defining matrices or vectors that uses angle brackets as delimiters. For example, a column vector can be entered as `<1, 2, 3>`, and a row vector as `< 1 | 2 | 3 >`. Note that commas (`,`) separate rows and vertical bars (|) separate columns. For example, a 2×3 matrix $\begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix}$ can be entered as:

```
< < 1, 2 > | < 3, 4 > | < 5, 6 > >;
```

or

```
< <1 | 3 | 5 >, <2 | 4 | 6 > > ;
```

The standard addition, subtraction and scalar multiplication of vectors are defined using the operators **+** , **–** and *****.

For example:

```
U := <1|2|3>:    # row vector
V := <4|3|-1>:   # row vector
W := <3,2>:      # column vector
U+V, 2*U, 2*U-3*V, 4*W;
```

$$[5\ 5\ 2],\ [2\ 4\ 6],\ [-10, -5, 9],\ \begin{bmatrix} 12 \\ 8 \end{bmatrix}$$

Dot Product

The dot operator "**.**" represents the dot product between two vectors of the same type (row or column). Using the vectors *U*, *V*, and *W*, that we defined earlier, we calculate

```
U.V, W.W;
```

$$7, 13$$

Some Useful Matrix Commands

LinearAlgebra Package

The **`LinearAlgebra`** package includes many commands for matrix computation:

```
with(LinearAlgebra):   #Load the package first
```

Operation	*Maple Command and Example*

`CrossProduct`
– Find the cross product between two column vectors or two row vectors of dimension three.
	`A := < 1,2,1 >:`
	`B := < 1,1,3 >:`
	`CrossProduct( A, B );`
	$\begin{bmatrix} 5 \\ -2 \\ -1 \end{bmatrix}$

`Determinant`
– Find the determinant of a square matrix.
	`B := < <2,3,-2> | <5,1,1> | <1,2,0> >:`
	`Determinant(B);`
	-19

Eigenvalues
– Find the eigenvalues of a square matrix.
	`C := < <2,-1,1> | <1,0,3> | <0,1,1> >;`
	`Eigenvalues(C);`
	$\begin{bmatrix} -1 \\ 2 \\ 2 \end{bmatrix}$

Eigenvectors – Find the eigenvectors of a square matrix.	`Eigenvectors(C, output = list);` $\left[\left[-1, 1, \left\{\begin{bmatrix} \frac{1}{4} \\ -\frac{3}{4} \\ 1 \end{bmatrix}\right\}\right], \left[2, 2, \left\{\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}\right\}\right]\right]$ This means that the eigenvalue –1 has multiplicity 1 with eigenvector [1/4, –3/4, 1] and eigenvalue 2 has multiplicity 2 with a single eigenvector [1, 0, 1]. (C does not have a full set of eigenvectors.)

The zip and map Commands

In addition to the standard matrix and vector operations, the **map** command lets you apply operations to each entry of a matrix, and the **zip** command applies operations to corresponding entries of matrices of the same size.

```
A := Matrix( [[-1,0], [1,2]] ):

B := Matrix( [[1,2], [3,4]] ):
map( x->sqrt(x), B );        #Take the square root of each element.
```

$$\begin{bmatrix} 1 & \sqrt{2} \\ \sqrt{3} & 2 \end{bmatrix}$$

```
zip( (x,y)->x*y, A, B ); #Multiply the corresponding entries.
```

$$\begin{bmatrix} -1 & 0 \\ 3 & 8 \end{bmatrix}$$

More Examples

Matrices and Systems of Linear Equations

A major use of linear algebra is to solve systems of linear equations. You convert the system of equations into an augmented matrix, apply row operations to reduce the matrix, then convert back to a system of equations. The conversion to a matrix is done with the **GenerateMatrix** command in the **LinearAlgebra** package.

```
eqns := [2*x +y-z= 1, x+3*z = 4, -5*x-3*y+z = 2]:
vars := [x, y, z]:

with(LinearAlgebra):
A1 := GenerateMatrix(eqns, vars, augmented=true);
```

$$\begin{bmatrix} 2 & 1 & -1 & 1 \\ 1 & 0 & 3 & 4 \\ -5 & -3 & 1 & 2 \end{bmatrix}$$

Once we have converted the system of equations to a matrix, we can find the reduced row echelon form by using Gauss-Jordan elimination.

```
A2 := ReducedRowEchelonForm(A1);
```

$$\begin{bmatrix} 1 & 0 & 0 & \frac{23}{5} \\ 0 & 1 & 0 & -\frac{42}{5} \\ 0 & 0 & 1 & -\frac{1}{5} \end{bmatrix}$$

We can then use the **GenerateEquations** command to convert the matrix back to a system of equations and see the answer.

```
GenerateEquations(A2, vars);
```

$$\left[x = \frac{23}{5}, y = -\frac{42}{5}, z = -\frac{1}{5}\right]$$

The LinearSolve Command

The **LinearSolve** command can be used to solve linear systems without going through the explicit reduction of the matrix to row echelon form. For example using the augmented matrix **A1** we defined above,

```
A := A1[..,1..3]:        #A is the first three columns of A1.
b := A1[..,4]:           #b is the fourth column of A1.
with(LinearAlgebra):
LinearSolve( A, b );
```

$$\begin{bmatrix} \frac{23}{5} \\ -\frac{42}{5} \\ -\frac{1}{5} \end{bmatrix}$$

Vector Space operations

Another standard task in linear algebra is to find a basis of a vector space. Maple has commands to find the basis of a vector space and the basis of the intersection of two vector spaces. Let us define the vectors,

```
V1 := <0, 1, 0> :  V2 := <-2, 1, 0>:
V3 := <2, 1, 0> :  V4 := <0, 0, 1>:
V5 := <0, 1, 3> :
```

We use the **Basis** command to find a basis of a vector space that is the span of a set or list of vectors.

```
with(LinearAlgebra):
Basis( [ V1, V2, V3 ] );
```

$$\left[\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} -2 \\ 1 \\ 0 \end{bmatrix}\right]$$

This gives a basis of the vector space that is the span of the vectors *V1*, *V2* and *V3*.

To find the basis of the intersection of several vector spaces, we use the **IntersectionBasis** command.

```
IntersectionBasis([ [V2, V3], [V4, V5] ] );
```

$$\left[\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}\right]$$

This means that the vector [0, 1, 0] is the basis of the subspace which is the intersection of the two vector spaces Span($V2$, $V3$) and Span($V4$, $V5$).

Note that the basis vector of the intersection need not be a member of the basis of either of the spaces whose intersection is being computed.

The Student [LinearAlgebra] package

The **Student[LinearAlgebra]** package is an alternative package that can be used for linear algebra operations. It includes many commands for visualization and can help students develop their geometric understanding of linear algebra.

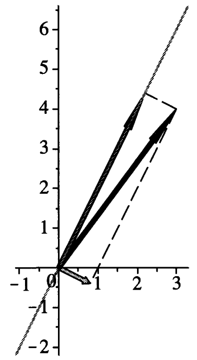
For example, the **ProjectionPlot** command shows the projection of one vector onto another; the **LinearTransformPlot** command shows the action of a linear transformation on the unit circle; the **EigenPlot** command shows what a linear map does to a collection of unit vectors; You can find out more about them from **?ProjectionPlot; ?LinearTransformPlot;** and **?EigenPlot;**.

When using these commands, setting the **infolevel** to **1** will give you more information. For example:

```
with(Student[LinearAlgebra]):
infolevel[Student[LinearAlgebra]] := 1:
ProjectionPlot(<3, 4>, <1, 2>);
```

```
Vector:     <3, 4>
Projection: <2.200, 4.400>
Orthogonal complement: <.8000, -.4000>
Norm of orthogonal complement: .8944
```

The Projection of a Vector
Onto a Line



Useful Tips

The **LinearAlgebra** package has many more useful commands, including:

- For extracting pieces of a matrix: **Row**, **Column**, **DeleteRow**, **DeleteColumn**, **Submatrix**, and **ArrayTools[Concatenate]**.
- For working with vectors: **Dimension**, **GramSchmidt**, **Normalize**, and **VectorAngle**.
- For constructing special matrices (and vectors): **BandMatrix**, **ConstantMatrix**, **DiagonalMatrix**, **IdentityMatrix**, **RandomMatrix**, **RandomVector**, **ScalarMatrix**, **ScalarVector**, **UnitVector**, **ZeroMatrix**, and **ZeroVector** (in **LinearAlgebra**).

Some of the commands in the **Student[LinearAlgebra]** package include:

- For step-by-step row reduction: **GaussJordanEliminationTutor**.
- For understanding the geometry of a linear system: **ColumnSpace**, **LinearSystemPlot**, **NullSpace**, **Rank**, and **RowSpace**.

You should be aware that some commands are included in both the **LinearAlgebra** and the **Student[LinearAlgebra]** packages. Examples include: **Eigenvalues**, **Eigenvectors**, **CharacteristicPolynomial**, **JordanForm**, **Determinant**, and **Trace**. The **Student[LinearAlgebra]** versions are generally identical to the ones found in **LinearAlgebra**.

Troubleshooting Q & A

Question... When I used one of the commands discussed in this chapter, Maple returned the command unevaluated. What should I check for?

Answer... Check that you load the corresponding package

```
with(LinearAlgebra):
with(Student[LinearAlgebra]):
```

Question... When I tried to create a matrix with the **Matrix** command, the result had extra zeroes that I did not enter. How did this happen?

Answer... Check that the rows or columns all have the same number of entries. The **Matrix** command pads short rows out with zeroes to give a rectangular array.

Question... When I tried to add, subtract, or multiply two matrices, I got an error message. What should I check for?

Answer... Check that the matrices in the command have compatible dimensions.

CHAPTER 20

Parametric Curves and Surfaces in Space

Parametric Curves in Space

The spacecurve Command

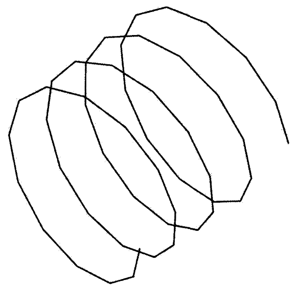
You use the **spacecurve** command, defined in the **plots** package, to create a 3-D plot of a parametric curve in space. To see the curve given as $(x(t), y(t), z(t))$, for $a \le t \le b$, type:

```
with(plots):
spacecurve( [ x(t), y(t), z(t), t = a..b ] );
```

This format is similar to the syntax of the **plot** command used for plane parametric curves. Here, however, the curve is defined with *three* parametric functions rather than *two*.

■ **Example.** The helix given parametrically by $(t, 3\cos(t), 3\sin(t))$, for $0 \le t \le 8\pi$, is drawn with:

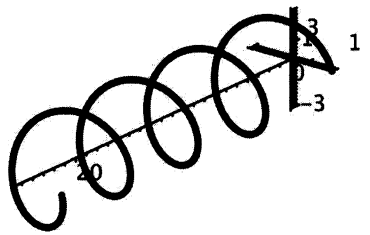
```
with(plots):
spacecurve( [t, 3*cos(t), 3*sin(t), t = 0..8*Pi] );
```



Options for spacecurve

Most options that you can use with **plot3d**, including **axes**, **labels**, **color**, and **scaling**, can also be used with **spacecurve**. The **numpoints** option provides some control of the resolution of the picture Maple produces. Here's a nicer picture of the same helix.

```
spacecurve( [t, 3*cos(t), 3*sin(t), t = 0..8*Pi],
            axes=normal,color=black, thickness=3,
            numpoints = 100, scaling=constrained);
```



Parametric Surfaces in Space

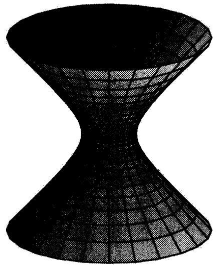
The plot3d Command

The **plot3d** command that we used to draw the graph of a two-variable function in Chapter 16 can also be used to draw a parametric surface in space. If a surface is defined parametrically by $((x(u,v), y(u,v), z(u,v))$ for $u_0 \le u \le u_1$ and $v_0 \le v \le v_1$, you enter:

```
plot3d( [ x(u,v) , y(u,v) , z(u,v) ],
         u = u0..u1, v = v0..v1 );
```

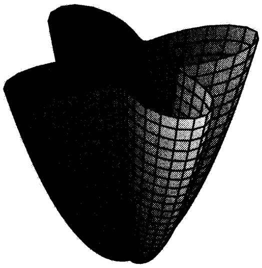
■ **Example.** To see a portion of the one-sheeted hyperboloid given parametrically by $(\cos(u)\cosh(v), \sin(u)\cosh(v), \sinh(v))$, for $0 \le u \le 2\pi$ and $-2 \le v \le 2$, you write:

```
H:=[cos(u)*cosh(v),sin(u)*cosh(v),sinh(v)]:
plot3d( H, u = 0..2*Pi, v = -2..2);
```



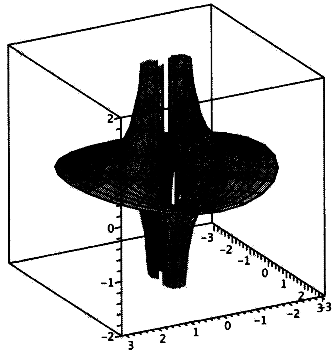
The surface $(v(2-\cos(4u))\cos(u), v(2-\cos(4u))\sin(u), v^2)$, for $0 \le u \le 2\pi$ and $0 \le v \le 2$, gives a very nice picture of a vase:

```
V:=[v*(2-cos(4*u))*cos(u),v*(2-cos(4*u))*sin(u),v^2]:
plot3d( V, u = 0..2*Pi, v = 0..2, grid = [60, 30] );
```



In Chapter 16 we looked at the surface $z = \dfrac{x^2 - y^2}{(x^2 + y^2)^2}$ noting that in cylindrical coordinates it simplified to $z = \dfrac{\cos(2\theta)}{r^2}$. In that chapter we solved for *r* as two functions of *z* and θ. The nicer alternative is to plot the surface parametrically, so that we can describe *z* as a function of *r* and θ, and describe the surface in parametric form as $(r\cos\theta, r\sin\theta, \dfrac{\cos(2\theta)}{r^2})$, for $0 \le \theta \le 2\pi$ and $r \ne 0$.

```
S := [ r*cos(theta), r*sin(theta),cos(2*theta)/r^2 ]:
plot3d( S, theta = 0..2*Pi, r = -3..3,
        view = -2..2, axes = boxed );
```



Plotting Multiple Curves and Surfaces

The **spacecurve** (or **plot3d**) command can sketch several curves (or surfaces) with a single command, just as you've already seen in the **plot**, **polarplot**, and **implicitplot** commands. Use one of these formats:

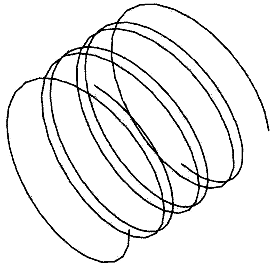
```
spacecurve( {[curve1, t=a1..b1],[curve2, t=a2..b2]} );

plot3d( [ [surface1 ], [surface2 ]],u = u0..u1, v = v0..v1);
```

> **Note:** Please notice that we need to use **{** *curly braces* **}** instead of **[** *square brackets* **]** to group the curves together in the **spacecurve** command.

Multiple Curves

■ **Example.** Consider the helixes $(t, -3\sin(t), 3\cos(t))$ with $\pi \le t \le 6\pi$ and $(t, 3\cos(t), 3\sin(t))$ with $0 \le t \le 8\pi$. They can be drawn together using:

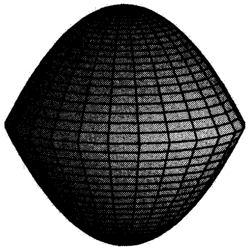
```
spacecurve( [[t,-3*sin(t), 3*cos(t),t=Pi..6*Pi],
[t,3*cos(t), 3*sin(t),t=0..8*Pi] ],
            numpoints=100, color=black );
```



Multiple Surfaces

■ **Example.** The paraboloid $(r\cos(t), r\sin(t), 2-r^2)$ opens down, while the paraboloid $(r\cos(t), r\sin(t), r^2)$ opens up. We can combine them to form a nice "beehive."

```
plot3d( [ [r*cos(t),r*sin(t),r^2],
          [r*cos(t),r*sin(t), 2-r^2] ],
          r = 0..1, t = 0..2*Pi);
```



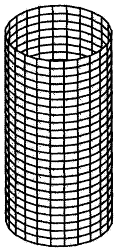
Shading, Coloring, and Transparency

Shading for Surfaces

The **plot3d** command colors the surface according to a default color and shading scheme. The goal is usually to use the color and shading (and lighting and more) to emphasize certain features of the surface.

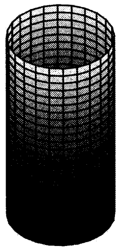
You can turn off Maple's default shading by setting **shading=none**. This results in a plain white surface. For example:

```
plot3d( [2*cos(t), 2*sin(t), z],
        t=0..2*Pi, z=-4..4, shading=none );
```



You can control the shading more directly by specifying, for example, **shading = zgrayscale** (colors lower points darker and higher points lighter) or **shading = zhue** (colors the lowest points blue and the highest points red). For example:

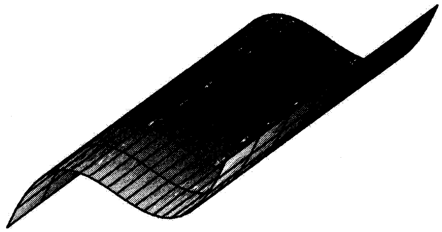
```
plot3d( [2*cos(t),2*sin(t),z],t = 0..2*Pi, z=-4..4,
        shading=zgrayscale, scaling=constrained );
```



Coloring Function

You can explicitly set the coloring of points on a surface by using the **color** option. If the color is a list of three numbers, this is interpreted as an **RGB** color. For example,

```
s := (u,v) -> [u^2, v, v^3];
plot3d( s(u,v), u=-2..2, v=-2..2,
        style=patch,
        color=[(u+v)/4, u^2/4, sin(v)^2]);
```



This means that each point $s(u,v)$ is painted with the color defined by **COLOR**(RGB, $\frac{u+v}{4}, \frac{u^2}{4}, \sin^2(v)$ **)**. For example, at the point $s(0,1)=[0,1,1]$ the color is **COLOR**(RGB, 1/4, 0, sin(1)^2**)**, a dark blue.

Transparency

When we plot several surfaces together we may want one surface to be partly transparent so that we can see another surface behind it. This can be done by setting the **transparency** option to be a number between 0 and 1. For example the following puts together a sphere and a cylinder, where we can see the cylinder through the sphere:

```
with(plots):
sphere := plot3d( 2, phi=0..Pi, theta=0..2*Pi,
                  coords = spherical,
                  color=red, transparency=.5 ):
cylinder := plot3d( 1, theta=0..2*Pi, z=-2..3,
                  coords=cylindrical,
                  color=green ):
display( [ sphere, cylinder ] );
```

(The figure is not shown here, since color transparency does not show up well in a static black and white image. Try it for yourself!!)

More Examples

Combining Graphics

You can combine curves, surfaces, and other three-dimensional images into a single graphic using the **display** command (just as we saw in Chapters 10 and 16).

■ **Example.** The upper hemisphere of the unit sphere $x^2+y^2+z^2=1$ is best plotted in spherical coordinates with $\rho=1$, for $0\le\theta\le 2\pi$ and $0\le\phi\le\pi/2$,

```
g1:=plot3d( 1, theta=0..2*Pi, phi=0..Pi/2,
            coords=spherical):
```

The point $P=(\frac{1}{2},\frac{1}{2},\frac{1}{\sqrt{2}})$ lies on this hemisphere. A normal (perpendicular) line to the hemisphere at P is given by $\vec{r}(t) = (\frac{1}{2}+t, \frac{1}{2}+t, \frac{1}{\sqrt{2}}+\sqrt{2}\,t)$. This command shows just a portion of the normal line:

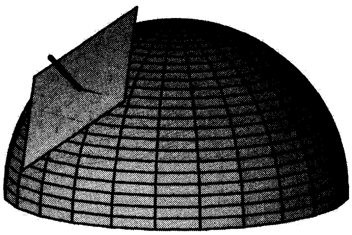
```
with(plots):
g2:=spacecurve( [1/2+t, 1/2+t, 1/sqrt(2)+sqrt(2)*t ],
                t=0..0.15, thickness=3, color=blue ):
```

Finally, the plane tangent to the hemisphere at P has equation $z=(2-x-y)/\sqrt{2}$. You can sketch a portion of it near the point P with:

```
g3:=plot3d( (2-x-y)/sqrt(2),
             x = 0.2..0.8, y = 0.2..0.8,
             grid=[2,2], color=cyan):
```

You can now see one of the nicest features of Maple, the ability to combine these graphics, despite the fact that each was drawn using a different type of command.

```
display( [g1,g2,g3], orientation=[97,78],
         scaling=constrained );
```



Notice how well the graphic convinces you that we have described both the tangent plane and the normal line correctly.

Troubleshooting Q & A

Question... When I tried to use the **spacecurve** or **display** command, Maple returned the input unevaluated. What went wrong?