A Maple Tutorial

Contents

4

Calculus   **13**

5

Differential Equations   **17**

5

Adding text to a Maple session   **23**

6

Printing   **25**

7

Saving your work   **27**

8

Getting Help   **29**

1

Getting Started

Maple is a powerful mathematical calculator, often called a computer algebra system (CAS) or symbolic mathematics program, because unlike most calculators it can do algebra and calculus symbolically. For example, it can expand the product $(x+1)^{30}$ into the normal representation of this polynomial, and it can factor the result back into the original form. It can compute the derivative or antiderivative of $3x/\sqrt{x^2+2x+47}$. It can also serve as a numerical calculator, but again unlike most calculators it can do integer and rational number arithmetic exactly, no matter how many digits are involved. If it is undesirable or impossible to express a result exactly, Maple can do numerical approximation. Finally, Maple can do some quite sophisticated graphing in both 2 and 3 dimensions.

To learn how to use some of Maple's features, you should work through this tutorial step by step. What you should type will be shown in the left hand column in `typewriter style`; comments and explanations will appear at the right. You should do everything listed here, but you should feel free to experiment too. If you leave something out or do something extra it may change the results of some later computations. (If you want to experiment along the way, try using different names for functions and variables than the ones in the examples.)

To start Maple, log in on one of the workstations in the math lab. Click on the Maple button on the bar at the bottom of the screen. When the Maple window comes up, you are ready to give commands to Maple.

For the most part, the commands you give will be typed in the main Maple window. Some commands can be given by making selections from menus, or by clicking on buttons

on the top bar. For example, to leave Maple, you may move the mouse pointer to the button titled 'File' and click on the left mouse button. When you do, a list of further commands appears. If you click on the last command, 'Exit', Maple will quit and the window will disappear.

The instructions given here are for Maple version 9. To make Maple 10 act like Maple 9, choose **Options** from the **Tools** menu; choose the **Display** tab; change **Input display** to **Maple Notation**; click on **Apply Globally**. You should only have to do this the first time you use Maple.

2

Algebra

2.1 NUMBERS

```
2+3;
123456789* 987654321;
```
You must type the semicolon at the end of the line to signal the end of your expression.

```
2+3
34*98
-45;
```
Type this as is, without the semicolon on the first line. Maple will warn you about the missing semicolon, but will usually do the calculation anyway. Sometimes you may want to start a new line as part of the same expression; you can do this by holding the shift key while you press return. Do this after the "98".

```
2/5 + 1/3;
23/45-167/345*2/3;
```
All rational arithmetic is done exactly.

```
2^5;
2**5;
3^27;
ifactor(%);
```

Maple knows standard mathematical notation, including these two symbols for exponentiation. The percent sign (%) denotes the previous result (it's called the 'ditto' operator). The command `ifactor` does integer factoring; the command `factor` factors polynomials.

```
12345/98765;
evalf(%);
```

You can force Maple to approximate values as 'floating point numbers' (i.e., using decimal notation). Some of Maple's commands do this automatically, as we will see later.

```
I^2;
evalf(Pi);
evalf(exp(1));
```

The number $i = \sqrt{-1}$ is denoted by I; the case is significant—a lower case 'i' is different. Pi denotes π; again case is significant. It is very easy to mistakenly type 'pi' instead of 'Pi'. When you do, things will almost certainly not work properly, but you will probably get no error messages—you just have to watch out for this. There is no built-in symbol for $e \approx 2.71828$; use the function `exp(x)` whenever you need e^x.

2.2   Variables and Expressions

Maple can manipulate expressions involving variables as well as numbers. Variables can be assigned values, either actual numeric values or entire expressions.

```
(x+5)^2;
f:=%;
x:=5;
f;
x^2;
```

Maple simply repeats the first line since x has no value. The second line saves the expression $(x + 5)^2$ with name f; the ":=" symbol acts like an equal sign in normal mathematics or in the programming language C. The third line assigns x the value 5.

```
x:='x';
x;
f;
```

This clears the value of x—this is not the same as setting it to zero! Note that Maple remembers the 'real' definition of f—it is still an expression involving x, not the fixed value 100.

```
2x;
2*x;
```

You must always type the multiplication symbol—Maple will not accept the first line even though it is standard notation. If you make a simple typing mistake like this, you can fix it without retyping the command. In this case, Maple should correctly guess where the problem is and place the cursor between the 2 and the x; simply type the * and press return. In any case, you can always move the mouse pointer to the position of the mistake and click the left button, then make your corrections. You can adjust your position in the line using the arrow keys to the right of the main keyboard. When you press the return key, the command will be re-evaluated.

```
expand(f);
factor(%);
factor(x^2+4*x+5);
```

If a polynomial doesn't factor, Maple leaves it alone. (As a general rule, if Maple cannot perform a calculation it will simply restate the command.)

WARNING: You can go back anywhere in your Maple session and change a command. When you do this you change *only* the result of that command—you do not change any subsequent commands that depend on the result. If you want to re-execute subsequent commands, you must move the cursor to each command in turn and press the return key. Do not overuse this editing ability—it can make the chain of events very difficult to follow. It often is better to copy the command, paste it in at the end of the session, modify it there, and execute it as a new command. To copy and paste a command, do this: Position the mouse pointer at the beginning of the command you want to copy, press the left mouse button and hold it down, move the mouse cursor to the end of the command, and release the mouse button. (If the command is on a single line, you may also select the whole command by positioning the cursor in the line, and clicking the left mouse button three

times quickly—this is a "triple click".) The command should now be highlighted. Position the mouse pointer where you want to paste and click the middle button.

2.3 EVALUATION AND SUBSTITUTION

It usually is not a good idea to assign values to variables—they are more useful as variables. If you want Maple to evaluate an expression you can give the variables temporary values.

```
f:=A*(x+3)^3;
subs(A=3,f);
subs({x=5,A=2},f);
x; A; f;
```

Note here that you use a plain '=' inside the substitution command. Two or more substitutions are done by enclosing them in curly braces. The values of f, A and x *don't* change.

```
subs(x=z^2,f);
f:=subs(x=z^3,f);
f;
```

You can substitute whole expressions for variables, not just values. To force the value of f to change, just reassign it—but be careful. It usually is better to give the expression a new name in case you need the original definition of f later.

```
f:=x^2+3*x+4;
f(1);
g:=x -> x^2+3*x+4;
g(1);
h:=unapply(f,x);
h(1);
h(x);
h(x^3);
```

Maple makes a crucial distinction between an expression like $x^2 + 3x + 4$ and the function which assigns to each value of x the value $x^2 + 3x + 4$. In this example, f is an expression, g and h are functions. The same distinction exists in mathematics, but usually it is possible to ignore it, and indeed in many instances Maple will accept either a function or an expression. To employ the normal mathematical notation $f(1)$, however, f must be a function. This example shows two ways to define a function—directly, or using an existing expression. The last two examples show that you can evaluate a function on expressions, not just numbers.

```
map(g,[-3,-2,-1,0,1,2,3]);
map(g,{-3,-2,-1,0,1,2,3});
```

Using the 'map' operation and a function (not an expression) you can evaluate a function on a list of values. The list can be enclosed in square brackets or in set braces—but notice the difference! The set braces produce a set as a result, meaning that repeated values are shown only once, and the order does not necessarily correspond to the order of the input values. In general, the square bracket form is more useful.

2.4 SOLVING EQUATIONS

Maple will solve a limited class of equations exactly and a broader class approximately.

```
solve(x^2-x-10=10,x);
solve(x^2-x-20,x);
solutions:=solve(g(x),x);
solve(x^5+4*x^3-3*x^2+10,x);
solve(tan(x)=x+1,x);
allvalues(%);
solve(tan(x)=x,x);
allvalues(%);
solve(sin(x)=cos(x)-1,x);
```

The *solve* command will give exact answers if possible, including complex answers. If Maple can't solve an equation it may print nothing, or it may display the equation in a standard form, preceded by 'RootOf'. You can ask Maple to attempt to evaluate roots displayed as a RootOf using `allvalues`. Maple will attempt to find all solutions for an equation or a RootOf, but often will not succeed.

Do not rely on the solve command alone when looking for solutions to equations—at the very least, you should graph your functions so that you know where roots appear to be. Note that if you give an expression instead of an equation Maple assumes that you mean '= 0'.

```
map(g,[solutions]);
simplify(%);
```

In some cases, Maple will not automatically simplify as much as it can, so you need to nudge it a bit. In this case, of course, we knew that the values should be zero. If the result had not simplified to zero, careful scrutiny of the problem would be called for.

```
f:=x^3+3*x^2+x+5:
solve(f,x);
fsolve(f,x);
fsolve(f,x,complex);
```

You can end a line with a colon instead of a semicolon—it tells Maple not to display the result. This can be particularly handy when the result is very long, or simply when you either know what it is or do not need to know (as when you will be using the result to compute a further result). The *fsolve* operation tells Maple to approximate the roots. In this example, Maple can compute the roots exactly, but it may be more informative to see approximations. Note that Maple only computes real roots unless the 'complex' option is present. If the function is not a polynomial, fsolve will find at most one solution, but for polynomials it will find all solutions.

```
g:=x -> exp(x)+x^2-10*x;
fsolve(g(x),x);
fsolve(g(x),x,-2..2);
g(%);
fsolve(g(x),x,2..4);
g(%);
```

You can specify a range in fsolve—you would normally do this when you know there are multiple roots and you want to approximate a particular one. In this case, a graph of the function clearly indicates a root near 0 and a root near 3. Because g is a function, and fsolve expects an expression, it is necessary to use $g(x)$ in fsolve. (I made g a function so that it would be easy to check the answers.)

```
solve({3*x+5*y=6,4*x-6*y=-8},
      {x,y});
h1:=3*x-6*y-5;
h2:=-4*x*y+7;
solve({h1,h2},{x,y});
evalf(%);
fsolve({h1,h2},{x,y});
fsolve({h1,h2},{x,y},{x=-2..0,y=-2..0});
```

Maple can solve equations simultaneously, with either solve or fsolve. As with single equations, fsolve may not find all solutions, but you can specify ranges for x and y to force fsolve to find particular solutions.

3

Plotting

Maple can do two and three dimensional graphing. There are many ways to adjust the plots that Maple produces; we'll cover just the basics here.

```
plot(f,x=-10..10);
plot(g(x),x=-5..5);
plot(g,-10..10);
plot(g,-10..10,y=-10..10);
```

Maple will accept either an expression or a function to plot, but notice that the form of the range is a bit different in the two cases. For an expression, use 'x=' before the range, but for a function leave it out. You can also specify the range for y explicitly. This is often useful if the function gets very large on the interval.

You can change the appearance of the plot in a variety of ways. Click in the plot and you should see a border appear around it; then you can explore the items on the plot menu. If you prefer, you can have plots come up in separate windows; use the **Options** item on the **Tools** menu in the main Maple window, then open the Display tab and change the plot display.

```
plot([1+2*sin(t),t,t=-Pi..Pi],
    coords=polar);
```

You can do plots in polar coordinates. Inside the square brackets, the first expression is the radius, the second is the angle, and the third is the range of the variable. In this case, the angle and the variable are the same, but in general the angle could be a more complicated function of the variable.

```
plot({x^2,x^3},x=-5..5);
plot({
   [1+2*sin(t),t,t=-Pi..Pi],
   [cos(4*t),t,t=-Pi..Pi]},
   coords=polar);
```

You can put more than one curve in the same plot by enclosing a list of expressions in curly braces.

```
r:=sqrt(x^2+y^2);
f:=sin(r)/r;
plot3d(f,x=-10..10,
        y=-10..10);
```

Try changing the appearance of a 3D plot by using the **Style**, **Color**, **Axes** and **Projection** menus. If you position the cursor in the drawing area and press *and hold* the left mouse button, you can move the mouse to change the orientation of the plot—try it.

```
with(plots):
polarplot(1+sin(t));
polarplot(sin(t/3));
polarplot(sin(t/3),
    t=0..3*Pi);
```

When you start Maple, only the most basic plotting commands are available. By typing "`with(plots)`" you tell Maple to load a number of special purpose plotting functions. One of the most useful is the polarplot routine shown here. By default, Maple uses the range $[-\pi, \pi]$; the last command shows how to specify a different range.

4

Calculus

4.1 LIMITS

```
f:=x^2+4*x+30:
limit(f,x=3);
subs(x=3,f);
g:=sin(x)/x;
limit(g,x=0);
limit(g,x=infinity);
g:=x/abs(x);
limit(g,x=0);
limit(g,x=0,left);
limit(g,x=0,right);
```

The *limit* command `limit(f,x=a)` computes the limit $\lim_{x \to a} f(x)$. Maple can handle many non-trivial limit problems and limits at infinity.

4.2 DIFFERENTIATION

```
f:=x->x^3+4*x^2+3*x+4:
fp:=diff(f(x),x);
critpts:=solve(fp,x);
map(f,[critpts]);
evalf(%);
evalf(critpts);
plot({f(x),fp},x=-5..5,
              y=-5..10);
```

The *diff* command does differentiation. Here I have found the derivative with respect to x, then the x coordinates of the critical points of f, then the value of f at these points, then floating point representations for the x and y coordinates. Because Maple can do calculations exactly, you should do approximations at the end; it would not be a good idea to approximate *critpts* before substitution into f (though in this simple case the results are almost identical). Note that if you click in the plot, the coordinates of the point under the cursor appear at the upper left of the Maple window. Youe can use this to approximate critical points and double check the computed values.

4.3 IMPLICIT DIFFERENTIATION

```
eqn:=x^2+y^2=1;
xp:=implicitdiff(eqn,x,y);
yp:=implicitdiff(eqn,y,x);
subs({x=1/2,y=sqrt(3/4)},yp);
```

Notice that the order of the second two arguments to the `implicitdiff` function is important—xp is dx/dy and yp is dy/dx. The last line computes y' at a specific point, $(1/2, \sqrt{3}/2)$.

4.4 INTEGRATION

```
f:=(2*z-3)*sqrt(z-3);
g:=int(f,z);
diff(g,z);
factor(%);
```

The *int* command shown here will compute an antiderivative. The computer is not infallible, and you may not notice a typing mistake, so it's not a bad idea to use Maple to check your work. Here, I've taken the derivative to check the antiderivative.

```
int(f,z=5..10);
simplify(%);
evalf(%);
subs(z=10,g)-subs(z=5,g);
```

To do definite integration give the limits after the variable. Of course, you can also do it yourself using the antiderivative. Notice that the answer can look quite different before simplification.

```
f:=1/sqrt(x):
int(f,x=0..1);
f:=1/x^2:
int(f,x=0..1);
int(f,x=1..infinity);
f:=exp(-x^2):
int(f,x=-infinity..infinity);
int(sin(x),x=0..infinity);
```

Maple can do many improper integrals exactly. If the integral does not converge, Maple may answer with '∞' if appropriate, or it may give an error message.

```
f:=tan(x^2):
int(f,x);
int(f,x=0..1);
evalf(%);
Int(f,x=0..1);
evalf(%);
```

Maple can't find antiderivatives of some functions. In such cases you can use *evalf* to approximate the value of a definite integral. If you don't want Maple to even try to find an antiderivative, use the capital I form of the integration operator.

5

Differential Equations

5.1 INTRODUCTION

A *differential equation* is an equation in which one or more derivatives of an unknown function appear. The function itself may also occur. The *order* of a differential equation is the order of the highest-order derivative of the function appearing in the differential equation. The challenge is to find the unknown function, or at least to find out some interesting information about it. Here are two differential equations, one is first-order, and the other is a second-order differential equation:

$$\frac{dy}{dx} - 3y = x + \sin x$$

$$y'' - 2y' + y = 0$$

A *solution* of a differential equation is a function which satisfies the equation. Maple will give you these solutions to the differential equations above:

$$y(x) = -\tfrac{1}{3} - \tfrac{1}{9} - \tfrac{1}{10}\cos(x) - \tfrac{3}{10}\sin(x) + e^{(3x)}\_C1$$

$$y(x) = \_C1\ e^x + \_C2\ e^x\ x$$

in which $\_C1$ and $\_C2$ represent constants which could have any value. (We would likely write them as C_1 and C_2. Maple sometimes writes the constant factor after the variable. In the solution of the first differential equation above, the last term is what we would write as $C_1 e^{3x}$.)

You should test these "solutions" to see that each does work in its differential equation.

The principal Maple tool for solving differential equations is `dsolve`. Here is how you would ask Maple to solve the differential equations above:

```
dsolve(diff(y(x),x)-3*y(x)=x+sin(x),y(x));
dsolve(diff(y(x),x$2)-2*diff(y(x),x)+y(x)=0,y(x));
```

The unknown function in these examples is called `y(x)`. This tells Maple that you are thinking of `x` as the independent variable, and `y` as dependent. You could use almost any combination of letters, though, as long as you were consistent.

In writing the differential equation, you use `diff(y(x),x)` for y', i.e., for dy/dx. For the second derivative, use `diff(y(x),x$2)`. If you were to need the third derivative, you would use `diff(y(x),x$3)`.

5.2   THE GENERAL FORM OF DSOLVE

```
dsolve(deqs,vars,opt)
```

`deqs` is a differential equation or a set of differential equations to be solved.

`vars` is the variable or set of variables to be solved for. This was simply `y(x)` above.

`opt` is one or more optional instructions we'll mention later.

5.3   SOLVING A FIRST ORDER DIFFERENTIAL EQUATION

```
dsolve(diff(f(t),t)=f(t),f(t));
```

Maple will tell you that the solution of $f'(t) = f(t)$ is $f(t) = e^t\ \_C1$, i.e., $C_1 e^t$.

```
difeq := diff(f(t),t)=f(t);
dsolve(difeq,f(t));
```

In using `dsolve`, you can give a name to a differential equation. These two lines accomplish the same as the one line above.

```
dsolve({diff(f(t),t)=f(t),f(0)=5},f(t));
```
This finds the solution that satisfies the *initial condition* $f(0) = 5$. When you have additional information that allows the values of the constants (_C1, etc.) to be evaluated, equations giving this information are included with the differential equation, all enclosed in braces { }. This becomes the set `deqs` sent to `dsolve`.

There are various ways of sending the same information to Maple. Here are some variations, using the differential equation and initial condition above:

```
difeq := diff(f(t),t)=f(t);
dsolve({difeq,f(0)=5},f(t));
```
Assign a name to the differential equation.

```
difeq := diff(f(t),t)=f(t);
init := f(0)=5;
dsolve({difeq,init},f(t));
```
Assign names to the differential equation and to the initial condition, and then form the set with these two as members.

```
difeqs:={diff(f(t),t)=f(t),f(0)=5};
dsolve(difeqs,f(t));
```
Put the differential equation and the initial condition together in a set, and then send that set to `dsolve`.

```
dsolve(difeqs,f(t));
expr := rhs(%);
g := unapply(expr,t);
```
`dsolve` returns an equation. `rhs` extracts the right side of the equation, i.e., the expression giving the formula for the solution. Then **unapply** turns this into a *function* that you can use. This example gives the name **g** to the function.

The advantage of giving names to the differential equation and/or the equation specifying the initial condition is that these tend to stand out, and you can easily change the initial condition, say, and then invoke `dsolve` again.

5.4   Solving a Second Order Differential Equation

```
deq := diff(y(x),x$2)+y(x)=0;
dsolve(deq, y(x));
```
The differential equation to be solved is $y'' + y = 0$. `dsolve` will give you the solution $y(x) = \_C1\ \cos(x) + \_C2\ \sin(x)$.

```
deq := diff(y(x),x$2)+y(x)=0;
init := y(0)=1, D(y)(0)=3;
dsolve({deq,init},y(x));
```
For a second order differential equation, you can specify two initial conditions. Typically you give the value of the function and of its first derivative at some point. **Notice that you use D, not diff, to indicate derivative in the initial conditions!**

```
de := diff(y(t),t) = a*y(t) + b*t;
dsolve(de,y(t));
```
Maple can handle symbolic constants.

$$y(t) = \frac{-bta - b + e^{(at)}\_C1a^2}{a^2}$$

```
a := 4: b := -2: dsolve(de,y(t));
```
Integer factors are fine:

$$y(t) = \frac{1}{2}t + \frac{1}{8} + e^{(4t)}\_C1$$

```
a := 4.1: b := -2: dsolve(de,y(t));
```
But `dsolve` does not like floating point (decimal) factors. If your differential equation involves these, you can ask for a numerical solution. (See the next section.) You can, however, use rational (fractional) constants.

5.5   NUMERICAL SOLUTION OF A DIFFERENTIAL EQUATION

```
de := diff(y(x),x)=(x-y(x))/
              (1+x^2+y(x)^2);
dsolve(de,y(x));
```

Maple cannot find an exact solution to

$$\frac{dy}{dx} = \frac{x - y}{1 + x^2 + y^2}.$$

```
init := y(0)=-1;
f1 := dsolve({de,init},y(x),numeric);
```

But if you give it initial conditions (one for a first order equation, two for a second order equation, and so on,) it will build a routine to approximate the value of a solution for any value of x.

```
f1(0);f1(2.5);
```

$$[x = 0, y(x) = -1]$$
$$[x = 2.5, y(x) = .3034671108279335]$$

Having obtained a numerical solution, you can use a special command to plot a graph of the solution. The graph is drawn from the values which are approximated by the numerical routine produced by **dsolve numeric**. You can examine the graph and estimate any values of the independent variable that might make the value of the function 0, for example.

```
plots[odeplot](f1,[x,y(x)]);
```

f1 is the name given in the command to solve numerically. **[x,y(x)]** tells Maple that you want a plot of **y(x)** against **x**. **x** and **y(x)** are the names used in the **dsolve numeric** command. **x** will be on the horizontal axis, and **y(x)** on the vertical.

```
plots[odeplot](f1,[x,y(x)],0..3);
```

You can also specify a range on the horizontal axis, as this does, or on both horizontal and vertical axes. As before, you could give this plot structure a name in order to view it later with a **display** command.

5.6   Solving a System of Differential Equations

If several quantities are interrelated, then an expression for the rate of change of each may be a function of two or more of the quantities. For example, two quantities $x(t)$ and $y(t)$ may be related by

$$dx/dt = ax(t) + by(t)$$
$$dy/dt = cx(t) + dy(t)$$

with a, b, c, d being constants. For this system, Maple will be able to find an exact (analytical) solution, but in many cases it's necessary to give initial conditions and ask for a numeric solution.

```
de1 := diff(x(t),t)=a*x(t)+b*y(t);
de2 := diff(y(t),t)=c*x(t)+d*y(t);
dsolve({de1,de2},{x(t),y(t)});
```

This will produce solutions for $x(t)$ and $y(t)$ in terms of a, b, c, d and two constants of integration _C1 and _C2.

```
a := -2: b := -1: c := 1: d := -2:
init := x(0) = 1, y(0) = 2;
sys := {de1,de2,init};
dsolve(sys,{x(t),y(t)});
```

If you give the constant factors values, and give initial conditions, then Maple can give you specific formulas for the solutions.

```
sol:=dsolve(sys,{x(t),y(t)},numeric);
plots[odeplot](sol,[t,x(t)],0..5);
plots[odeplot](sol,[t,y(t)],0..5);
```

This will produce numerical solutions. As before, you can use **ode-plot** to get graphs of these solutions.

5

Adding text to a Maple session

You can add plain text to a *Maple* session. This is especially useful if you are going to give a printed or electronic version of a session to someone else, since it lets you explain what is going on. It is also a good idea to add text to remind yourself at a later date what you were doing and why.

To add text above a command, position the cursor anywhere in the command. Then open the Insert menu, and then the Paragraph submenu, and choose Before Cursor. When you type you will get ordinary text that will not be interpreted as part of the command.

6

Printing

BEFORE YOU PRINT: You should clean up your session before you print, so that you do not waste paper and so that the printed copy is attractive and easy to read. If you want to print only part of the session, first save the session (see the section *Saving your work*) and then cut out the portions of the session that you do not want to print. You might want to save the trimmed version using a different name.

To print your worksheet, choose **Print** from the **File** menu, or click on the print button (the fourth little button from the left in the menu bar).

If you would like to print just a plot, you need to display it in a window of its own. You can do this by changing the plot display option, as described in the *Plotting* section. Then position the cursor in the plot command and press return. The plot should open in a new window, and you can use the Print menu in that window to print the plot.

7

Saving your work

You can save your Maple worksheet so that you can come back to it at a later time. Choose the **Save** or **Save As**... item from the **File** menu in Maple; a new window will pop up. (Use **Save As** if you want to save your worksheet under a new name.) You may also save a worksheet using the third button from the left in the menu bar.

When you reload a worksheet (with the **Open** item on the **File** menu or with the second button from the left) Maple will not automatically recreate all the values created in the worksheet. You may selectively re-execute commands in the worksheet, or you may use the **Execute** item on the **Edit** menu to execute the entire worksheet.

8

Getting Help

There are (somewhat out of date) manuals for Maple in the lab. You can also get help 'on line' from the Maple program. The help system is very good and always available as you work in Maple.

    `?int`   If you know the command you need help on, just type its name after a question mark.

    You can browse through the entire help system or search it using the **Help** menu.