FACULTY OF SCIENCE

SCHOOL OF MATHEMATICS
AND STATISTICS

FIRST YEAR
MAPLE NOTES

The information in these notes is correct at the time of printing. Any changes will be announced through the School's Web site at `www.maths.unsw.edu.au`, where an updated version of these notes may be placed if necessary. This version of the notes was produced on February 12, 2018.

Contents

1 INTRODUCTION TO MAPLE 1

2.4.7 Integration 23
2.4.8 Partial Fractions 24
2.5 Collections of Expressions, etc. 24
2.5.1 Sequences 24
2.5.2 Sets and Lists 25
2.5.3 Converting Structures 25
2.5.4 Selecting Operands 26
2.5.5 Sorting 26
2.5.6 Substituting into a Structure 27
2.5.7 Applying Functions to each Entry of a Structure 27
2.5.8 Sums and Products 27
2.6 Equations. 28
2.6.1 Solving Equations. 29
2.7 Complex Numbers. 31
2.8 Plotting. 31
2.8.1 Plotting Piecewise-de ned Functions 33
2.8.2 Plotting Data Points 33
2.8.3 Parametric Plots 34
2.8.4 Polar and Implicit Plots 34
2.8.5 3-D plots 35
2.9 The student Calculus Package. 35
2.9.1 Inert procedures. 35
2.9.2 Change of Variable 36
2.9.3 Integration by Parts 36
2.9.4 Riemann Sums and Simpson's Rule 36
2.10 Vectors and Matrices 37
2.10.1 Vectors. 37
2.10.2 Matrices 37
2.10.3 Selecting Components of Vectors and Matrices 38
2.10.4 Manipulating Vectors and Matrices 39
2.11 Gaussian Elimination. 40
2.12 Vector and Matrix Arithmetic 43
2.13 Vector Geometry 44
2.13.1 Dot and Cross Products, Length 44
2.13.2 Three-dimensional Geometry. 44
2.14 Partial Derivatives 47
2.15 Ordinary Di erential Equations. 47
2.16 Taylor Series. 48
2.17 Discrete Mathematics. 49
2.17.1 Greatest Common Divisors 49
2.17.2 Modular Arithmetic 49
2.17.3 Set Algebra 49
2.17.4 Solving Recurrence Relations 50
2.18 Assuming properties. 50
2.19 Conditions and the if { then construction. 51
2.19.1 Conditions 51
2.19.2 The if { then construction 52

2.20 Looping with for and while 52
2.20.1 Two further examples 54
2.21 Functions and Procedures. 55
2.21.1 Procedures. 55
2.22 Common Mistakes 57
2.23 Reading Files into Maple. 58
2.24 More Maple. 59

3 MORE ON THE MAPLE GUI. 61
3.1 Maple's Default Settings 61
3.2 Maple Settings in the Labs 62
3.3 A Maple Window 62
3.4 Worksheets and Documents 62
3.5 The Menu Bar 63
3.6 The File Menu 64
3.7 The Edit Menu 67
3.8 The View Menu 69
3.9 The Insert Menu 69
3.10 The Format Menu 70
3.11 The Tools menu 70
3.11.1 The Options submenu 70
3.12 The Help Menu 72
3.13 Alternatives to Commonly Used Menu Options 72
3.13.1 The Tool Bar 72
3.13.2 Keystroke Alternatives 72
3.14 2D Math input 72
3.14.1 Symbol recognition 73

A SUMMARY OF MAPLE COMMANDS. 75

INDEX 79

Chapter 1

INTRODUCTION TO MAPLE

Maple is a computer algebra system developed in Canada, initially at the University of Waterloo. It is now owned and developed by a company called Maplesoft. Maple runs on a number of operating systems, including Microsoft Windows, Mac and Linux. The current version of Maple is Maple 2017 and this version is installed in the School of Mathematics and Statistics labs and also available from the bookshop if you wish to buy a copy for your own use. In the School's labs, we have customised the settings of Maple. If you have your own copy of Maple, you should read section 3.2 that explains how to apply the same settings to your copy.

These Notes cover what you need to know about Maple in MATH1131/1141 and MATH1231/1241. They are also useful for other Maple based first year courses, but cover more material than is required. In addition, there are a number of Maple lessons available on Moodle, see section 1.6.

If you want to know more, there is lots of information in Maple's in built help, on the web or in the many books on Maple — some of them are listed in section 2.24.

1.1 What Does Maple Do?

A calculator (or a computer with numerical computation software) can do things like

- evaluating a function at a point
- find an approximate value for a definite integral like $\int_0 x\sin(x)\,dx$,

but it CANNOT tell you that

- $\int x\sin(x)\,dx = \sin(x) - x\cos(x)$
- $\frac{d}{dx}x^x = x^x(1+\ln x)$
- the exact roots of $x^2 - 2x - 2$ are $1 \pm \sqrt{3}$.
- the solution of a general quadratic ax^2+bx+c is $(-b \pm \sqrt{b^2-4ac})/2a$

Maple can do ALL these things — and much more as well. The following list shows just a few (in fact a small fraction) of the things which Maple can do. (Some of these are things which you will not understand now, but you will by the end of the year.)

- differentiate functions;
- find indefinite integrals for many functions;
- evaluate many complicated limits;
- find exact solutions for many algebraic equations, including some with arbitrary coefficients;
- perform algebraic operations on polynomials and rational functions;
- plot graphs of functions of one or two variables;
- solve many classes of differential equations;

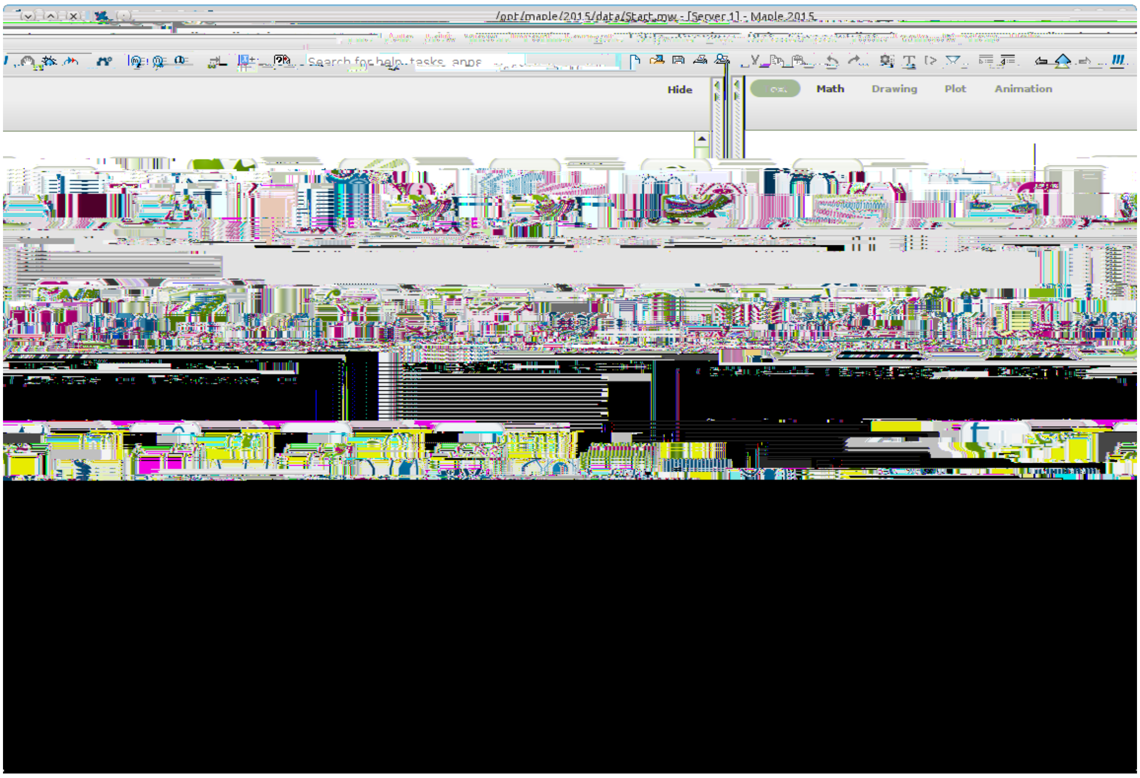

Figure 1.1: Maple in the Red-Centre labs

perform linear algebra operations, such as multiplying matrices.

In addition, it can link to other applications and be used to create interactive documents.

1.2 The Maple Window

This chapter (and chapter 2) will tell you as much about Maple as you need to know in order to complete your rst year Maple assessments.

In your rst year mathematics courses you will use Maple via its Graphical User Interface (GUI). To start Maple click on the Maple Application Icon in the task bar (as described in the lab notes). After some time, a Maple window, similar to that shown in gure 1.1, will appear. Note that the window shown is Maple in the Red-Centre labs. If you have your own copy of Maple it will appear slightly di erent. For more details, see Chapter 3.

To open a blank Maple worksheet, click on the \New Worksheet" tile that is the second from the left in the top row of tiles. Figure 1.2 shows the Maple window with a blank Maple worksheet open in the second tab. This window contains

a menu bar across the top with menus:

File E_dit V_iew I_nsert For_mat . . . . . . Tools Window Help

some of which are described below;

a tool bar immediately below the menu bar, with button-based shortcuts to common operations;

Figure 1.2: Maple in the Red-Centre labs

1.3 Using an Maple Worksheet

For an online or laboratory Maple test you will need to create and save a Maple worksheet that you will use for solving the problems we give you. Commands are typed **t** T][a348n]cursfor(aa348n]ivrticalrnlin)e

you 1..13gingveralrorksheest

will usenl ret orksheet

Figure 1.3: A Maple Worksheet

If you press <Enter> without typing a colon or semicolon, Maple will warn you, insert a semi-colon itself and try to execute your command if it looks complete to Maple This may be what you want but it might not be, so you have to be careful. If the command does not look completed, (for example you are using a loop, see 2.20) then you will get a warning message like

Warning, premature end of input, use <Shift> + <Enter> to avoid this message. which you can ignore if you want and continue with the command. Thus you can enter commands too long to t on a line (for example, a large matrix). Just press <Shift>- <Enter> at a convenient point (such as the end of a column of the matrix | but NOT in the middle of a word or number) and continue on the next line.

Exercise If you have not already done so, start Maple. Type the following commands (you may omit the words after the # on each line). After each command, press <Enter> and wait for the result to appear before entering the next command.

```
diff(x^3,x);               # a derivative
int(x^2*sin(x),x);   # an integral
solve(x^2-2*x-2,x); # roots of a quadratic
limit((3*x+4)/(5*x+6),x=infinity); # a limit as x goes to infinity
```

The results you get from this exercise should be:

$$3x^2$$

$$x^2 \cos(x) + 2\cos(x) + 2\,x \sin(x)$$

p

1 +

means the last result that Maple calculated and *not* the previous result on the screen — they can be different. Similarly you can use %% for the result before last (i.e. the second last result), and %%% for the one before that (i.e. the third last result). You cannot go further back than that using %. For example, the sequence of commands

```
diff(tan(x),x);
diff(%,x);
```

will give you the second derivative of $\tan x$. Try it (and note that the derivative of $\tan x$ is not expressed in exactly the way that you might have expected).

1.3.4 Context Sensitive Menus

Another way of using a previous result is provided by the *context sensitive menus*. If you right-click on an object in the worksheet (such as a plot, or the result of a command) a menu opens up allowing you to operate on that result. Exactly what you can do depends on the result, which is why they are *context sensitive*. Several of the options have sub-menus.

For example, with a plot the menu will allow you to change the plotting style, the axes, the colours (if a 3-d plot) etc and to output the plot in one of several different forms.

If you have an expression, such as, say $\frac{1}{3}x^3$ then right clicking on it will allow you to choose to do several things to it, such as assign it to a name, differentiate or integrate it, evaluate it or find its zeros.

With a matrix, you get the option to apply many commands from the LinearAlgebra package (see section 2.12). Note that Maple uses the full name of the commands from packages.

Using these menus can save a lot of effort. However, in a first year Maple Lab Test, you need to make sure that you use typed Maple commands so that the marker can see how you have used Maple commands to obtain the result.

1.3.5 Aborting Commands

1.3.7 Changing Maple Commands

Sometimes you will need to change a command that you have already entered. To do so, move the Maple | cursor to the place where you want to make a change by clicking the left mouse button there (or by using the arrow keys). Then

the Delete key deletes the character to the right of the cursor

the Backspace key deletes the character to the left of the cursor

and you can insert new characters at the position of the cursor.

When you have changed a command and want to execute the changed command, make sure that the | cursor is (anywhere) on the line of that command and then press <Enter>. The result of the changed command should appear on the screen, replacing the previous result.

Note: If the command you changed uses % to refer to results of previous commands then you will have to go back and re-execute those commands (by moving the cursor to each of those command lines and pressing <Enter>) before you execute the command you changed. This is because % refers to the result of the command most recently executed by Maple which is NOT necessarily the one on the line above the cursor's position.

Also, if the previous results of the command you changed were used in any subsequent commands then these later commands will have to be executed again. To be safe, it is best to press <Enter> on every command line which comes after any command line that you have changed.

If you want to insert a whole new command among previous commands, move the cursor to the execution groum%714.44546.diterl

1.4 Saving a Maple worksheet

A Maple worksheet can be saved and re-opened at a later time in much the same way as other kinds of documents such as word processer documents or spreadsheets. Saving a Maple worksheet preserves only what you see in the worksheet. When this worksheet is re-opened, Maple will not remember and variables you may have de ned, packages loaded or the results of recently executed Maple commands. This will be become clearer once you have used Maple.

A Maple worksheet can also be exported into other formats such as html, i. e. as a webpage, or as a text le containing only the Maple commands.

It is strongly recommended that you save a Maple worksheet regularly while you are using it.

There are two ways to save your worksheet. You can either click on the Save Icon (picture of a oppy disk) or select either the Save or the Save As ... option from the File menu. If you select Save or click the Save Icon and the active worksheet has a name, it will be saved (as a worksheet) in a le with that name; if it has no name, you get the Save As dialogue box. So you can save an updated version of the same session in a le with the same name simply by clicking on the Save icon: If you wish to save it with a new name, you will need to select the Save As ... from the File menu.

Selecting Save As ... will always bring up the Save As Dialogue Box. At the

Input (.mpl) and then enter the name of the file in the File Name box. If the file does not have the .mpl extension that designates a Maple Input file then Maple will add one. If there is already a file with that name, you will be asked to confirm the name and if you do the old file will be overwritten.

1.5 Maple On-line Help

Maple has built in help that can be accessed from the Help menu. A menu appears looking like this:

Maple Help	Ctrl-F1
Take a Tour of Maple	
Quick Reference	Ctrl-F2
⋮	
Manuals, Dictionary, and more.	
On the Web	
About Maple...	

If you select Maple Help from this menu, a new window will appear. In the right hand section of this window is some text entitled Maple Resources . This window also has several hyperlinks indicated by underlined text.

The left hand section of this window is the Help Browser, which will show the Maple Help Navigator expanded as far as the help page visible in the right hand panel.

1.5.1 The Help Browser

To learn how to use the help browser, click on the Help and move your mouse down to the line \Manual

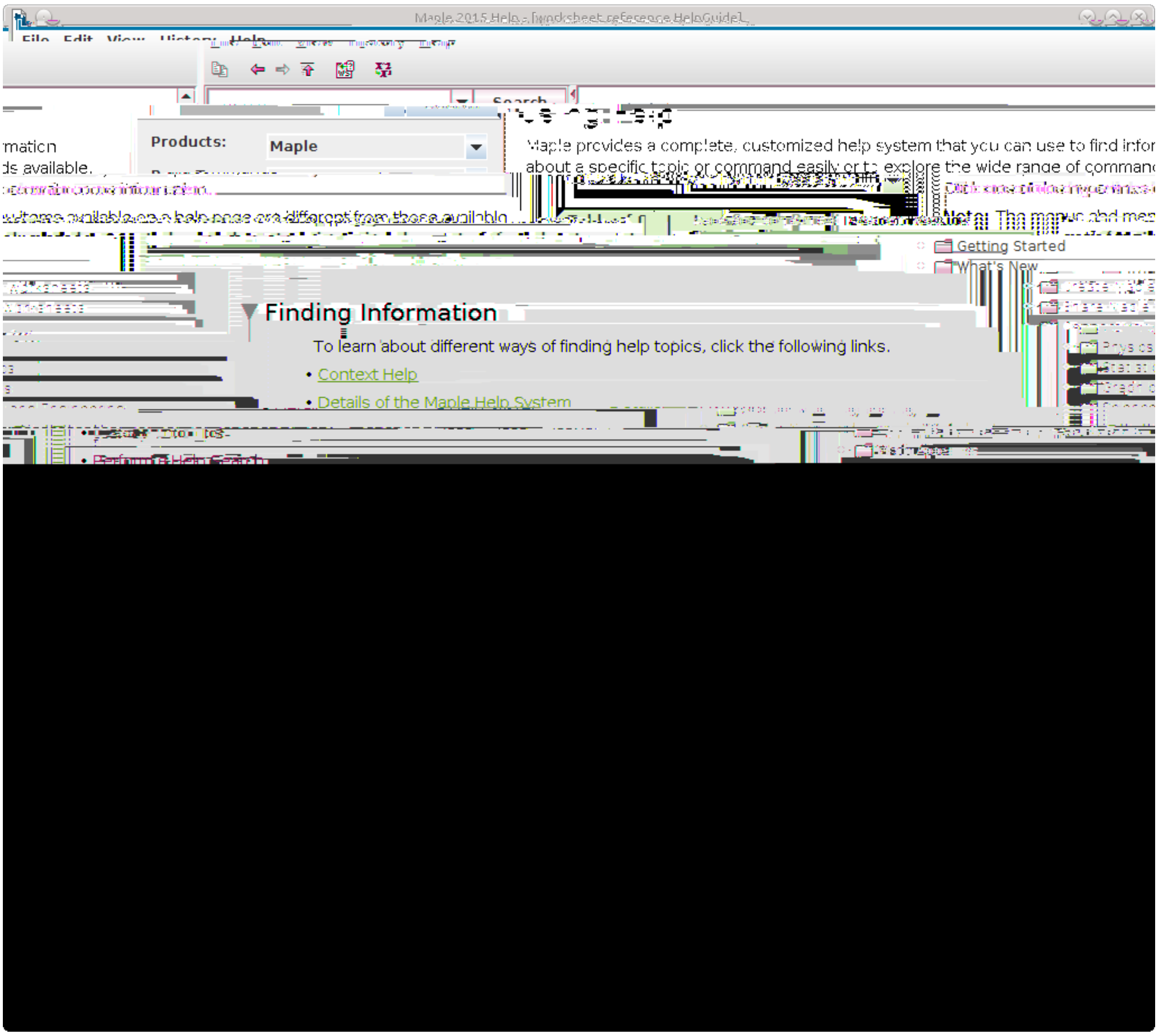

Figure 1.4: Maple Help Browser

For example, with \Text" selected, a search on \di erentiation" gives you a lot of pages; using \Topic" just two help pages and the dictionary de nition page.

If you had selected a help page using the ? (see below) then the \Search Results" tab is open, and lists all the results of a search on your input text.

You can change the relative width of the two panels in the help page by dragging the left boundary of the right panel with the mouse.

1.5.2 Using the Results

The Help Navigator and Search Result panels use di erent icons to tell you about the pages. An icon shaped like a folder tells you that there are subpages (or subfolders) collected under that heading: clicking on a line with a folder icon opens up the subfolders and pages.

A question mark designates a help page: clicking on that opens an actual help page in the right hand side. A letter \D" in a yellow square links to a dictionary de nition page. Both of these open in the right panel of the Help Browser.

The help pages start with a formal statement of the `syntax' of the command (i.e. details of how to enter it). This may be hard to understand, but at the end of the entry will be some examples of usage. Use the scroll bar to move through the entry until you get to the examples.

Exercise Use a topic search to find the help entry for the integration command `int` and read through it trying to understand it. Use the mouse to highlight one of the example command lines from the end of this entry and press <Ctrl>-C to copy

Chapter 2

MAPLE COMMANDS AND LANGUAGE.

This chapter, which contains details of the Maple commands and language, is qu3GE.

Maple you can easily find 1000! because Maple can handle integers of almost any size. Try finding 1000!, but do not get carried away with calculating factorials because they get very large very quickly and can easily lead you to exceed your time and memory limits.

Unlike most calculators and most computer programming languages, Maple does all arithmetic EXACTLY, i.e. as rational numbers (fractions with numerator and denominator having as many digits as is necessary) or as surds or as roots of equations. The only exception is when you deliberately enter numbers as decimals.

If you want to evaluate a fraction as a decimal number, use the command `evalf` (`evaluate as floating point'). This will normally display the answer to 10 significant digits (although it uses more than 10 digits internally when doing its calculations). If you want to use a different number of digits for all your displays of decimal numbers, use the command `Digits` to set the required number. For example, enter

```
Digits := 50;
```

to tell Maple that you want all floating point results displayed to 50 significant digits. If you only want to display one number to a different number of digits (without changing the number of digits for all displays), you can include the number of digits in the `evalf` command itself. For example,

```
evalf(1/17,50);
```

will evaluate $1/17$ to 50 significant digits.

There are several ways to enter a decimal or `floating point' number. For example, $67.2319$ can be entered as `67.2319` or `0.672319*10^2` or `672319*10^(-4)` etc, or in the form `Float(672319,-4)` which stands for 672319×10^{-4}. Note that this always has the form

```
Float(integer,integer);
```

There are limits to the size of the second integer (which specifies the exponent) and the number of digits is governed by the value of the variable `Digits`.

Anything which is entered as a floating point number will stay as a floating point number and will not be converted to a fraction unless you specifically ask Maple to convert it into a fraction. To do that, use the `convert` command as in

```
convert(%,fraction);
```

Arithmetic done on floating point numbers will always give a floating point answer.

2.2 Variables: Assignment and Unassignment.

2.2.1 Assigning

You can assign any expression to a variable for further use, as was done in section 1.3 with the command

```
f := sin(x);
```

This assigns the current `value' of the expression `sin(x)` to the variable `f`. If `x` is an unknown, as was the case in Chapter 6, then `f` stands for the expression $\sin x$ and we can, for example, differentiate this expression with respect to $x$. But if `x` had already been assigned a value then that value will be used to assign a value to `f`. For example, if `x` already had the value 0, then the above assignment would give `f` the value 0 (since $\sin 0 = 0$) and if `x` had the value `a+2` then `f` would be given the value `sin(a+2)`. Notice that the use of the word `value' is not being restricted just to numerical values. Possible

variablename := 'variablename'

(For example, x := 'x' to unassign the variable x.) Notice that both the quote symbols here are forward quote

2.3 Expressions and Functions.

It is important in Maple that you distinguish between expressions and functions .

as well as, for example:

Function	Description	Example
abs	absolute value	abs(-2);
sqrt	square root	sqrt(4);
ifactor	factorise integers (can take a long time)	ifactor(12);
igcd	greatest common divisor of integers	igcd(6,8);
ilcm	least common multiple of integers	ilcm(6,8);
max	maximum of a sequence of numbers	max(132,129,66,120);
min	minimum of a sequence of numbers	min(132,129,66,120);
binomial	binomial coe cient	binomial(4,2);
round	round (up/down) to an integer	

However, if `f` has been defined to be an *expression* in the variable `x` (for example by `f := x^2-x;`), you CANNOT get the value of `f` when `x` is 3 by writing `f(3)` since `f` is not a function. In this situation there are two methods which you can use to evaluate `f` for a particular value of `x`, the first being the preferred one:

1. Use the command `subs` to substitute for `x` as in

   ```
   subs(x=3,f);
   ```

   Note that this does NOT change the value of `x` or `f` — it simply displays the value that `f` would have if `x` were equal to 3. In this case `x` remains an unassigned variable and `f` remains an expression dependent on `x`.

   In general, the command

   subs(*expression1* = *expression2*, *expression3*);

   will substitute *expression2* for *expression1* everywhere that *expression1* appears EXPLICITLY in *expression3*. For example,

   ```
   subs(m=e/c^2,f=m*a);
   ```

   gives the result $f = ae/c^2$.

   Several substitutions can be done in the one command. For example,

   ```
   subs(a=2*d,d=b/c,a*b*c);
   ```

   will substitute `2*d` for `a` and then substitute `b/c` for `d`, giving the result $2b^2$.

2. Assign the desired value to `x` and then ask Maple to display `f`. For example, the sequence of commands

   ```
   f := x^2-x;
   x := 3;
   f;
   ```

   will finally display the value 6, which is the value of $x^2 - x$ when $x = 3$. If you want the value of `f` at a second value of `x`, just assign the second value to `x` and then display `f` again. If you want to return `f` to being an expression in the unknown `x` then you will have to unassign `x` by one of the methods described in section 2.2.3.

2.3.3 Simplifying an Expression.

Maple often leaves an expression in a complicated form rather than in its `simplest' form. To remedy this situation, Maple provides a number of procedures which you can use in an attempt to get an expression into a form which suits you better. Nevertheless, you have to bear in mind that factorising and simplifying expressions (other than very easy ones such as those in high school) is a *very* difficult process, both for humans and for the computer, and it is not always clear what `simplify' means. Consequently, you may have trouble getting Maple to produce what *you* consider to be the nicest form of an expression. This is probably the most frustrating part of using a computer algebra package.

The following is a list of some of the commands you can try if you want to `simplify' an expression. The descriptions given here are only brief, and in each case you should use Maple's Help to find out more about these commands.

normal

2.3.4 Defining functions with the arrow operator.

The arrow operator is used to define functions. For example, the function f which acts as $f(x) = x^2 - x$ is defined by

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

Here we use an arrow (typed as a - immediately followed by >) to show that the function replaces the (dummy) variable x by the expression x^2-x. Then the command

```
f(3);
```

will display 6, which is the value of f when x is equal to 3, and you can write f(2*a-1) for the value of f when x is equal to $2a - 1$.

Note that the x which occurs in the definition of f is a *dummy variable* and has *no* relation to any variable x which might occur anywhere else in your Maple session. It just provides a way of specifying a formula for the function.

You can use f(x) in any situation where Maple will accept an expression in the unknown x. You can also use f on its own to represent the function in some situations, but you must make sure that the situation is one in which this applies. For example, you can differentiate a function using the D operator (see section 2.4.2) or use a function to define a Matrix or Vector (see section 2.10.4).

In MATH1231/1241 you will study functions of more than one variable. These can also be defined with the arrow operator. For example,

```
d := (x,y,z) -> sqrt(x^2+y^2+z^2);
```

is a definition of the 3-dimensional distance function $d(x, y, z) = \sqrt{x^2 + y^2 + z^2}$.

Exercise Enter the above definition of the function f.
What happens when you now enter

```
f;
eval(f);
```

2.4 Elementary Calculus.

2.4.1 Limits

To find the limit of an expression as a variable tends to a value, use

limit(*expression*, *variable*=*value*);

For example, to find the limit of $(\sin x)/x$ as $x \to 0$, type

```
limit( sin(x)/x, x=0 );
```

You can use infinity as a value if you want to find the limit as $x \to \infty$.
For example, to find the limit of $(1 + 1/x)^x$ as $x \to \infty$, type

```
limit( (1+1/x)^x, x=infinity );
```

You can ask for a one-sided limit by inserting left or right. For example, to get the limit of $1/x$ as $x \to 0^+$, try using

```
limit( 1/x, x=0, right );
```

Note that the result is ∞. In this case the lefthand limit gives the result $-\infty$ and the two-sided limit gives the result undefined.

2.4.2 First Derivatives

To differentiate an *expression* with respect to a variable, use

diff(*expression*, *variable*);

For example, to differentiate $e^x \sin x^2$ with respect to x, you can enter

```
y := exp(x)*sin(x^2);    diff(y,x);
```

or just

```
diff( exp(x)*sin(x^2),x );
```

Remember that you should only di erentiate with respect to an *unassigned* variable. There are two ways to di erentiate a (with or without spaces before and after the $) to stand for the sequence consisting of *expression* repeated *number* times. To differentiate an expression n times with respect to x, you can enter

```
diff( expression, x$n );
```

To find a higher derivative, for example the third derivative, of a function f of one variable using the D operator use

```
D[1,1,1](f);
```

This notation looks a little odd, but will make more sense once you learn about partial derivatives in MATH1231/1241 (see 2.14).

2.4.5 Implicit Differentiation

If y related to x by an equation that defines y as a function of x, then Maple can use *implicit differentiation*, to find the derivative of y with respect to x with the Maple command implicitdiff.
For example, to find the slope of (a tangent to) the circle $x^2 + y^2 = 1$, use

```
implicitdiff(x^2+y^2=1,y,x);
```

2.4.6 Maxima and Minima

To find the global minimum value over the whole real line for an expression in one unknown x, use

```
minimize( expression );
```

Use maximize to find the global maximum.

You can also use these commands to find global maximum and minimum values for expressions in several unknowns. For example, to find the smallest value (over all real values of x and y) of $x^2 + 2x + y^2$, try using

```
minimize( x^2 + 2*x + y^2 );
```

(You should get the answer -1, as you can see by completing the square.)

2.4.7 Integration

To find an indefinite integral of an expression with respect to a variable, use

```
int( expression, variable );
```

For example,

```
f := x^2*sin(x);
int(f,x);
```

or just

```
int( x^2*sin(x),x );
```

Note that Maple does NOT show an arbitrary constant C in an indefinite integral.

If f is a function, then you need to put the expression f(x) in the int command, not just f.

You can find a definite integral

$$\int_a^b$$

If you have a variable whose value is a *sequence*, take care that you do not use it as the argument of a Maple function that expects a single object. For example, if you apply `evalf` to a sequence with more than 2 values, you will get an error.

If you want to give Maple a sequence that can be described by a formula, use the command `seq`. For example, the command

```
seq( n*(n+1), n=1..5);
```

generates the finite sequence $2, 6, 12, 20, 30$.

Sequences are used to construction several other types of objects in Maple, for example, sets and lists. We will also use them to enter Vectors and Matrices (section 2.10).

2.5.2 Sets and Lists

A Maple list is a maple sequence enclosed in square brackets and a *set* is a sequence enclosed in a curly brackets. For example, `[1,2,3]` is a list and `{1,2,3}` is a set. Maple treats a list or a set as a single object.

The important differences between sets and lists are

- The order in which the things appear is *not* significant for sets but it is significant for lists. Thus the sets {1,2,3} and {3,2,1} are treated as the *same* set but the lists [1,2,3] and [3,2,1] are different. Of course the contents of a set will be printed in some particular order when Maple displays it, but this order is decided by Maple and may not be the same as the order you entered. Maple keeps the order of

```
op(L);
```

gives a sequence with the same values as Sq. This is useful if you want to add someting at the end of the list. For example, you can change the value of the variable L from the list [1,2] to the list [1,2,3] by

```
L := [ op(L), 3 ];
```

The command op will also convert any expression into a sequence. This is because algebraically any expression consists of an operator acting on a sequence of `parts' or operands, and op extracts the sequence of those operands. For example, the command

```
op(6*(x+y)/z);
```

gives the sequence 6, x + y, 1=z since 6(x + y)=z is the product of 6, (x + y) and 1=z. There is a problem in that it is not always easy to tell what Maple will regard as the operator. For example, Maple regards 1/z as the power of z to the -1, and so

```
op(1/z);
```

gives the sequence z, 1.

2.5.4 Selecting Operands

The command op is actually quite a powerful general procedure for selecting operands in many contexts, but its full description is a bit too advanced for inclusion here (for full details, use Maple's Help). One use is that if n has a positive integer value then

op(n, expression);

selects the nth operand of the expression. For example, the command

```
op(2,1/z);
```

gives 1 (see the example above).

In special cases there are synonyms for op such as:

numer and denom give the numerator and denominator of a quotient expression. For example,

```
numer( sin(x)/(1+cos(x)) );
```

gives the result sin(x).

coeff picks out the coefficient of a specified power of a variable in a polynomial. For example,

coeff(polynomial, x,5);

gives youout);

`sort` can also be used to put the terms of a polynomial intodecreasingpower order. For example,

```
sort(1+x+x^6-x^3);
```

will give the result x^6 x

To get a sum to infinity, use `infinity` as the upper limit. For example, try

```
sum(1/k^2,k=1..infinity);
```

If `f` is a function, you will need to write `f(k)` in the command (not just `f`).

You can use similar methods to evaluate products such as

$$\prod_{k=m}^{n} f(k).$$

Just use `product` instead of `sum`.

Note that there will be many cases where Maple cannot find a simple expression for the answer and it may just return your question as its answer. For example, try asking Maple to find

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

Maple also has the commands `add` and `mul`, which should be used in certain situations in place of `sum` and `product` — see Maple's help pages.

2.6 Equations.

In Maple an expression of the form

leftside `=` rightside `;`

is an *equation*. Note that in this case the `=` symbol is used *alone* and not with the `:` which appears in assignment commands. Make sure that you do not confuse equations with assignment commands.

Maple allows you to perform various operations on equations. You can add two equations, or multiply an equation by a constant, or add a constant to it (i.e. to both sides of it). You can use this to solve simple simultaneous equations. For example,

```
  > e1 := 2*x+y=5:          # e1 is the first equation
k > e2 := x-y=4:            # e2 is the second7111Tf-156.196-25.581
```

2.6.1 Solving Equations.

Maple provides two basic commands for solving equations: solve and fsolve. In general, neither of these procedures tries to find all solutions to an equation. However, both of them will try to find all real solutions to a polynomial equation.

If you give one of these solvers an expression instead of an equation, it will assume that you want to solve the equation expression = 0.

Using solve

This tries to find exact solutions to an equation or set of equations. For example, to solve $3x + 4 = 5x$, use

```
solve( 3*x+4=5*x );
```

which gives the answer 2.

If the equation involves more than one variable, you will have to tell Maple which variable to solve for. For example,

```
solve( a*x^2+b*x+c, x );
```

Notice that in this case we gave solve an expression rather than an equation and Maple will assume that we want to solve the equation $ax^2 + bx + c$

When using solve

The simplest way to get Maple to draw a graph is a using a command like

plot(*expression*, *variable*=*start* .. *end*);

In this case Maple will decide the appropriate vertical scale. If you want to specify min34(aluesT-3343o)-32644ae minimum and maximum values on the vertical scale, use

plot(*expression*, *variable*=*start* .. *end*, *min* .. *max*);

Note that there must be exactly TWO full stops between *start* and *end* and between *min* and *max*. (This notation is used whenever you refer to a *range* of values in Maple.)

This command will plot the *expression* for values of the *variable* over the range min-T-3344a-T-7833(34478334(at for getting approximate coordinates for a point of intersection of two graphs or the intercepts of a graph.

Note: As Maple scales each axis di erently, it may give a misleading impression of the you should first construct a list of the coordinate pairs [x1,y1], [x2,y2], ... , [xn,yn] and then plot this list. (For details on lists, see section 2.5.) For example,

```
datapoints := [ [1,sin(1)], [2,sin(2)], [3,sin(3)], [4,sin(4)],
               [5,sin(5)], [6,sin(6)], [7,sin(7)], [8,sin(8)] ];
```

or (see section 2.5 for the seq command)

```
datapoints := [seq([i,sin(i)],i=1..8)];
```

Note the use of square brackets here. The square brackets in [1,sin(1)] etc. are to indicate the coordinates of a point in the x, y-plane (an ordered pair of numbers — i.e. a list with two members). The outermost square brackets are to make the sequence of coordinate pairs into a list.

When you have created the list of coordinate pairs, execute the plot by

```
plot(datapoints);
```

2.8.3 Parametric Plots

To plot a curve defined by

$$(x, y) = (f(t), g(t))$$

for t from a to b when ft and gt are expressions depending on t, use

```
plot( [ft,gt,t=a..b] );
```

Note the use of square brackets [ ] here. For example, try plotting a cusp with

```
plot( [t^2,t^3,t=-5..5] );
```

or try plotting a circle with

```
plot( [cos(t),sin(t),t=0..2*Pi] );
```

where $\cos(t)$ and $\sin(t)$ are expressions made up using the functions cos and sin.

2.8.4 Polar and Implicit Plots

The package plots provides special plotting procedures. Once you have loaded the package by doing

```
with(plots):
```

you can use polarplot for polar curves and implicitplot for implictly defined curves.

Figure 2.2 is a plot of the the cardioid $r = 1 - \cos\theta$ for θ from 0 to 2π done using the command

```
polarplot(1-cos(t),t=0..2*Pi);
```

For convenience, we have used t as the name for the variable representing θ — we do not have to name it theta.

You can plot the same curve in a parametric format as

```
polarplot( [1-cos(t),t,t=0..2*Pi] );
```

and you can plot several polar curves on the same diagram by using curly brackets in the same way as for the command plot.

Try plotting the implicitly defined ellipse $x^2 + y^2/4 = 1$ to true scale by

```
implicitplot(x^2+y^2/4=1,x=-3..3,y=-3..3,scaling=CONSTRAINED);
```

where the scaling=CONSTRAINED forces Maple to use the same scale on the two axes (otherwise the graph looks like a circle).

2.9. The student

evaluate it. This is an example of what is called an *inert procedure*. Maple also provides the inert procedures `Diff`, `Limit`, `Sum` and `Product` for entering derivatives, limits, sums and products.

After you have entered an expression with one of these commands and applied various procedures to the expression, you can finally evaluate it with the command `value`, as in

```
value(%);
```

Note that `value` only works with `Int`, `Diff`, `Limit`, `Sum` and `Product`.

2.9.2 Change of Variable

To change the variable in an integral (i.e. to apply the method of integration by substitution) use the command `changevar`. The general form of this command is

```
changevar(s,F,u);
```

where

- `u` is the new variable which is being introduced
- `s` is an equation of the mathematical form $h(x) = g(u)$ which implicitly or explicitly defines the old variable x in terms of the new variable u
- `F` is an expression like `Int(f,x)` or `Int(f,x=a..b)` when `f` is an expression in `x`; or `Int(f(x),x)` or `Int(f(x),x=a..b)` when `f` is a function.

The result of the command is an expression for `F` with the variable changed. For example,

```
changevar( 1+x^2=u, Int(x/(1+x^2),x=a..b), u );
```

$$\int_{1+a^2}^{1+b^2} \frac{1}{2u}\, du.$$

If you use `value` to evaluate an indefinite integral in terms of `u` then you cannot always use `changevar` in the reverse direction to get the answer in terms of `x`. You may have to use `subs` to substitute for `u` in terms of `x` (see section 2.3.2).

2.9.3 Integration by Parts

This is integration by means of the formula

$$\int u\, dv = uv - \int v\, du.$$

The general form of the command is

```
intparts(F,u);
```

where `u` is the expression which is to play the role of u in the above formula and `F` is of the form `Int(u*` *expression*`,x)`.

For example, to integrate $x^3 \sin x$ with $u = x^3$ try

```
intparts( Int(x^3*sin(x),x), x^3 );
```

2.9.4 Riemann Sums and Simpson's Rule

The `student` package offers you the opportunity to calculate some Riemann sums and get pictures showing the graph of the integrand together with the rectangles whose areas make up the Riemann sum. It can also show sums corresponding to approximations given by Simpson's Rule.

The command

`rightsum(` *expresssion*`,x=a..b,n);`

will give you a summation expression for the Riemann sum obtained by dividing the interval from a

You create a Matrix using pairs of angle brackets (or the `Matrix` procedure). You enclose the columns of the Matrix as a collection of `Vectors`, separated by vertical lines `|` and linked together with an outer set of angle brackets. For example, you can assign the matrix

$$\begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 5 \\ 3 & 2 & 4 \end{pmatrix}$$

to the variable `A` with the command

```
v := [1,2,-4,9];
```

then v[3] will have the value 4 because the third entry in the list is 4. Note that the same notation also works for lists.

and thesT0909 31(la6(the30and)s)-209 31(sho329(w

You can use these notations in commands to change the value of entries. For example, the commands

```
v[2] := 12:    A[2,1] := 13:
```

will assign the value 12 to the second entry of the list v and the value 13 to the entry in the second row and first column of the Matrix A.

You can do the same thing with a set, but this is dangerous because the order of A

```
v := Column(A,2);
```

We also have `Row` for rows | but this creates a row Vector, which is not something we'll be dealing with.

If you use a range (for example `2..3`) you can extract more than one column (as a sequence).

Note that the things created by `Column` (and `Row`) are Vectors (or sequences of Vectors) and not Matrices

Another procedure which selects parts of matrices is `SubMatrix`. The command

```
SubMatrix( A, a..b, c..d );
```

(where `a..b` and `c..d` are ranges of positive integers) produces a smaller Matrix whose entries are the values of `A[i,j]` for $a \le i \le b$ and $c \le j \le d$, arranged in the same relative positions as they had in `A`.

You can also use `SubMatrix` in the form

```
SubMatrix( A, [1,4..6], [2..4] );
```

to get the submatrix whose entries are the values of `A[i,j]` for `i` in the list `[1,4,5,6]` and `j` in the list `[2,3,4]`.

Maple also let's you define Vectors and Matrices using an *indexing function*. If f is a function that takes one argument, `Vector(5, f)` gives a vector with the 5 entries, $f(1)$, $f(2)$, $f(3)$, $f(4)$, $f(5)$. Similarly, if g is a function that takes two arguments, `Matrix(2, 3, g)` produces a Matrix with 2 rows and 3 columns in which the element in the i^{th} row and j^{th} column is given by $g(i, j)$. See section 2.21 for examples.

2.11 Gaussian Elimination.

The `LinearAlgebra` package provides the procedure `RowOperation` which allows you to solve systems of simultaneous linear equations by going step by step through the steps of Gaussian elimination. The one procedure will do different things, depending on what arguments you give it. So:

`< A | b>`	augment Matrix `A` by Vector `b` to give $(A\|b)$

then you could proceed as follows. (Note that some of the command lines end in a colon. This is to save space by suppressing display of the result.)

```
> with(LinearAlgebra):
> A := < <0,2,1> | <1,-1,2> |< 2,-2,4> | <1,1,0> >:
> b := < 1,-1,5 >:
> m1 := < A | b >;
```

```
            2
             0   1    2   1   1
  m1 := 4 2    1    2  1    1
```

```
> reduced := RowOperation( m8,1,1/2 );
```

$$\text{reduced} := \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 2 & 0 & 2 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

We now have the fully reduced form and we can read o the general solution as

$$[1; 2 \quad 2\ ;\ ; \quad 1]$$

where is an arbitrary parameter representing an arbitrary choice of the third variable.

When Maple is solving systems of linear equations, it uses names like $\_t_1$, $\_t_2$ for arbitrary parameters which may occur in the general solution. So the solution of the above set of equations would be displayed as

$$[1; 2 \quad 2\_t_1; \_$$

2. Maple can solve linear sytems with unknown parameters in them. In particular, it can solve systems in which each of the entries in the righthand side is an unassigned variable, for example

```
b := < b1 , b2 , b3 >;
```

3. If you are doing MATH 1231 or MATH 1241, you should look up the Maple help files for the procedures Rank, RowSpace, ColumnSpace, Transpose, Determinant, Eigenvalues, Eigenvectors, CharacteristicPolynomial. To anyone who has studied the linear algebra section of MATH 1231/1241, the effects of these procedures should be obvious from their names.

2.12 Vector and Matrix Arithmetic

The last section shows how to create and display Vectors and Matrices. However, we cannot do any useful linear algebra without first loading the LinearAlgebra package, which provides many procedures for dealing with Vectors, Matrices and linear equations. (Technically, LinearAlgebra is called a *module*, but you can ignore this distinction.) In these Notes we will only give an outline of some of the basic procedures. To find out more about these and other procedures available in LinearAlgebra, use Maple's Help.

To use the procedures in this package, it is easiest to first load it with the command

```
with(LinearAlgebra):
```

Note the use of the colon here, which supresses the (very long) list of new funcitons in LinearAlgebra.

However, you can use a command directly by giving its *full name*: you do this by adding LinearAlgebra:- to the start of the command. This is what Maple will do if you operate with the context sensitive menus (see section 1.3.4).

Vectors and Matrices can be added, subtracted and multipled by scalars using the usual operators +, - and *. Two Matrices can be multipled using the . operator. The . operator is also used to give the dot product of two Vectors. A square Matrix can be raised to an integer power using ^.

For example, to find the linear combination $2v - 3w$ of the vectors v and w we use

```
2*v - 3*w;
```

and we can enter the formula for a general point on the line through the points with position vectors v and w (i.e. the formula $x = (1 - \lambda)v + \lambda w$, where λ is an arbitrary parameter). We do this by

```
x := (1-lambda)*v + lambda*w;
```

Note that A^(-1) gives the inverse matrix.

When using these operations, you must of course ensure that the Matrices and Vectors are of the appropriate dimensions and, in the case of the negative power of a matrix, that the matrix is invertible.

If you enter an expression involving these operations, Maple will automatically carry out the operations. For example, to enter the mathematical expression $Ax + b$, we use

```
A.x + b ;
```

If values have already been assigned to A, x and b then A.x+b will be evaluated, but if some of these variables are unassigned then you will get an answer involving the unassigned variables.

A useful convention in Maple is that, in expressions involving matrices, a scalar constant can be treated as that scalar multiple of an appropriate identity matrix I. For example, if A is a square matrix then the command

```
1 + A + 2*A^2 ;
```

evaluates the polynomial expression $I + A + 2A^2$, where I is the appropriate identity by giving a cartesian equation for it or three (non-collinear) points on it or a point on it and a normal direction or a point on it and two lines parallel to it. The command `sphere` is used to assign a name to a sphere. A sphere may be specified by giving its cartesian equation or four points on it or the end-points of a diameter or its center and its radius. To display the specifications of one of these things you need to use `detail` (see the example below). For a plane, the detail includes a cartesian equation for the plane. If you want to find a normal to a plane `p` use

```
NormalVector(p);
```

Be warned that if you specify a plane or sphere by means of an equation then Maple will want you to specify the names of the variables which are associated with the three axes. You can do this by listing them as a third argument to the `plane` or `sphere` command, as in

```
plane(P,x+y+z=1,[x,y,z]);
```

If you leave out the `[x,y,z]` then Maple will, rather strangely, prompt you with the request `enter the name of the x-axis`, to which you reply `x;`, and similarly for the other two axes. The semi-colons are *compulsory* here. Depending on the way Maple is set up, this might be done using a pop-up dialogue box.

When you have set up objects of these types you can, for example, use the command `distance` to find the distance between two of them or the command `intersection` to find the intersection of two of them or the command `FindAngle` to find the angle between two of them (where appropriate).

You can also use `Equation` to find a cartesian equation for a line, plane or sphere and `center` for the center of a sphere and `coordinates` to find the coordinates of a point.

Use the Maple `Help` to find out more about any of these commands and to find out about the many other commands available in `geom3d`.

In the following example, we first label the points $A(0;1;2)$ and $B(2;3;1)$ and the line AB through A and B. Applying `detail` to AB shows that the direction of the line is $(2;2;-1)$ and the line can be expressed in parametric vector form as

$$x = \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix} + t\begin{pmatrix} 2 \\ 2 \\ -1 \end{pmatrix}; \quad t \in \mathbb{R}.$$

Then we assign the label P to the plane through $C(4;5;6)$ with normal $(1;1;1)$ and use `detail` to find that P can be described by the cartesian equation

x + y + vow:s8 0 Td [5.244 1.9552 Tf 5.244 65ine and so on.

Notice that we use a colon : to suppress the output of most of the commands because the output would just be an echo of assigned names.

```
with(geom3d):
```

Warning, the name polar has been redefined

2.14 Partial Derivatives

If `f` is an expression which contains several unknowns (one of which is `x`) then the result of the command

```
diff(f,x);
```

is the partial derivative $\partial f/\partial x$. Similarly, the command

```
diff(f,x,x);
```

gives you the second order partial derivative with respect to `x`. If `f` is an expression in unknowns `x` and `y` (and maybe other unknowns as well), the mixed second order partial derivative $\partial^2 f/\partial y \partial x$ is given by

```
diff(f,x,y);
```

If `f` is a function, then you need to use either of the two methods in section 2.4.2. For example, if `f` is a function of two variables, then the first method becomes

```
diff(f(x,y),y);
```

In the second method, use the `D` operator followed by numbers in square brackets to indicate the variable with respect to which you are differentiating. For example, if `f` has been defined as a function of two variabels then

```
D[2](f);
```

gives you the partial derivative with respect to the second variable and

```
D[1,2](f);
```

gives you function of two variables given by the mixed second order partial derivative $\partial^2 f(x, y)/\partial y \partial x$ and

```
D[1,2](f)(3,-2);
```

gives the value of that mixed derivative evaluated at the point $(3, -2)$.

Note: All derivatives in Maple are partial derivatives and this is the reason why Maple insists on a second argument for the `diff` command.

2.15 Ordinary Differential Equations.

The procedure `dsolve` is used to solve ordinary differential equations (ODEs). You enter the ODE as an equation involving the function, say `y(x)`, differentiated appropriately with respect to `x`. For example to solve (i.e. find the general solution of) $y' + y = \sin x + x$ you would use

```
dsolve( diff(y(x),x)+y(x)=sin(x)+x, y(x) );
```

and for the more complicated equation

$$xy' + y(x^2 + \ln y) = 0$$

you would use

```
dsolve( x*diff(y(x),x)+y(x)*(x^2+ln(y(x)))=0, y(x) );
```

and to solve

$$y'' - 2y' - 3y = 2x$$

you would use

```
dsolve( diff(y(x),x,x)-2*diff(y(x),x)-3*y(x)=2*x, y(x) );
```

In the resulting general solutions, arbitrary constants are shown as `_C1`, `_C2` etc.

NOTES:

1. The dependent variable $y(x)$ must always be written completely as $y(x)$ and never just as y

will nd the rst eight terms (i.e. up to and including the term of degree 7) in the Taylor series about the point $x = \frac{}{2}$ (i.e. in powers of $(x \quad \frac{}{2})$) for the the function $e^{x} \sin x$ and it will assign the result to the variable s.

Note that Taylor series are not polynomials and cannot be treated as such, hence

Only one assumption at a time can be made using assume. If you use assume twice on the same variable then the first assumption is lost. However, you can make additional assumptions with the command additionally. It is used in the same way as assume.

To find out what assumptions have been made about a variable, use

```
about(variable);
```

and Maple will display all assumptions about that variable.

For example, to assume that the variable a is a positive integer and check the assumptions on a and then evaluate the limit of e^{-ax} as $x \to \infty$ you could say

```
assume(a,integer);
additionally(a>0);
about(a);
limit( exp(-a*x), x=infinity );
```

To show that a variable has had an assumption made about it, Maple puts a ~ after the variable name. For example, the variable a with an assumption having been made about it will always be displayed and printed as a~. This sometimes looks like a-, so take care when reading the Maple output. There is no need for you to type the ~ when using the variable.

An assumption about a variable can be removed by unassigning the variable in any of the ways described in section 2.2.3.

There are other details about making assumptions, such as the assuming command, and you can check the relevant help entries if you want to know more.

Exercise What happens when you enter

2.19.2 The if – then construction

As an example, the following commands are a procedure to find the absolute value of a real number.

```
absval := proc(x)
  if evalf(x)>=0 then
    x;
  else
    -x;
  end if;
end proc;
```

The general shape of the if command is something like

```
if condition1 then
     commseq1
elif condition2 then
     commseq2
elif condition3 then
     commseq3
else commseq4
end if:
```

where commseq1 etc are sequences of commands (each finishing as usual with : or ;) and condition1 etc are conditions that are tested for their truth. The action of such an if command is to test if condition1 is true and if it is true then execute (carry out) the commands in commseq1, while if it is not true then Maple tests condition2. If this is true then Maple executes commseq2, and if it is not true then Maple tests condition3. If this is true then Maple executes commseq3, and if it is not true then Maple executes commseq4.

An if command needs the first if and then and the final end if, but the other parts are optional, depending on what is required. We have seen this above and there are further examples in section 2.21.1.

You do not have to put : or ; after the then (though it will not matter if you do), but you MUST put one after the end if.

2.20 Looping with for and while.

Suppose that you want to execute a set of Maple commands several times, changing the value of one variable n at each repetition. This is called creating a loop, and is very common in scientific programming.

The way you create the loop depends on whether you know in advance exactly how may times you want to repeat the commands or not. If you know that you want to repeat the commands 100 times then you can use a construction of the type `for n from 1 to 100 do ...'. If you do not know how many repetitions you want to make then you will have to tell Maple to keep repeating until some condition is no longer satisfied, using a construction of the type `while ... do ...'.

As an example of the first type of situation, consider the following piece of Maple.

```
for n from 0 by 2 to 4 do
    print(`(a+b) to the power`,n,`equals`,expand((a+b)^n));
end do:
```

The resulting display will be

$$(a+b) \text{ to the power}, 0, \text{equals}, 1$$
$$(a+b) \text{ to the power}, 2, \text{equals}, a^2 + 2ab + b^2$$
$$(a+b) \text{ to the power}, 4, \text{equals}, a^4 + 4a^3b + 6a^2b^2 + 4ab^3 + b^4$$

This display shows the expansion of $(a+b)^n$ for each of the values $n = 0, 2$ and 4. We terminate this `for–loop' with `end do:`, using a `:` rather than a `;`, so that any intermediate results will not be printed on the screen. We use the `print` procedure to force display of the things that we want to see. In the argument to the `print` procedure, text contained between back-quotes will be echoed in the printout and variables which occur outside back-quotes will have their values printed.

It is possible to print things nicely using a specified *format*. This is done using the `printf` procedure which works the same as it does in the C programming language. We will not go into any details here — use Maple's Help if you want more details.

As an example of a situation where you need to use `while`, here is a loop that finds the first value of n for which the sum

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

is greater than 5.

the condition is satis ed then the body of the loop is carried out. Note that this means that in Maple it is possible for the body of the loop not to be run, unlike some other computer languages in which a loop is always run at least once. Then n is increased by the value of step. If its new value is less than or equal to that of nish and the condition is still satis ed then the body is carried out again. This process continues until the value of n is greater than that of nish OR the condition becomes false (whichever is rst). At this point the loop ends. The nal value of n is the rst of the values start, start+step, start+2*step, ... that exceeds nish OR the rst of these values at which the condition is false (whichever is the lesser).

The start, step and nish must have values that are numbers. If the value of step is negative then the obvious changes to the above description must be made.

If the from start or the by step is omitted then a value of 1 is assumed. If the while condition is omitted, then it is assumed to always be true. If the for variable is omitted, then the loop is carried out for the speci ed number of times and exactly the same unchanged sequence of commands is performed at each repetition. You can leave out everything except the while ::: do ::: end do:, provided that you have assigned an initial value to the variable occurring in the condition and that the value of this variable is changed by commands in the body of the loop. For example, the following commands apply the Euclidean algorithm to nd the greatest common divisor of 3960 and 3780.

```
a := 3960;    b := 3780;
while b<>0 do
  r := a mod b;
  a := b;
  b := r;
end do:
print(‘gcd is‘,a)
```

Take extreme care with looping because you can very easily set Maple o on an in nite loop, or at least a very long loop that will take you past your time limit and completely ruin your Maple session. The STOP button does not always stop a loop.

The commands break and next can be used in for loops to change the normal procedure. If a break is encountered, then that loop is immediately nished, and the command after the end do is rewrite these as `while` loops requires a bit of thought. The following two examples show how to get around this.

Suppose you wish to find the first Fibonacci number that is larger than 1000. The natural way to do this is to calculate each number, check to see that it is smaller than 1000, if not, then calculate the next one and repeat. But we cannot check the size of the number at the end of a loop in Maple, only at the start. To get around this, we do the following:

```
a[0] := 0: a[1] := 1:
for n from 1 while a[n] <= 1000 do
  a[n+1] := a[n]+a[n-1]
end do:
```

At the end of this loop `a[n]` will contain the first Fibonacci number greater than 1000, which is 1597. Note that the colon after the `end do` supresses the printing of the intermediate values. If you enter `op(a)` you will see a table with all the Fibonacci numbers that were calculated.

As a second example, suppose you are trying to use Newton's method to find the square root of 5, starting from a guess of 2. The you would be calculating successive iterates x_n using the formula $x_{n+1} = \frac{1}{2}\left(x_n + \frac{5}{x_n}\right)$. Suppose we decide to find the first x_n with $|x_n - x_{n-1}| < 10^{-6}$. This can be done as follows:

```
x[0] := 2: x[1] := evalf((x[0]+5/x[0])/2):
for n from 1 while abs(x[n]-x[n-1]) >= 10^(-6) do
x[n+1] := evalf((x[n]+5/x[n])/2);
end do:
```

What this loop does is succesively calculate x_n as long as the difference $|x_n - x_{n-1}|$ is greater than or equal to 10^{-6}. Once it is is not, the loop finishes. So at the end of this loop `x[n]` has the desired value.

2.21 Functions and Procedures.

2.21.1 Procedures.

The functions defined by the arrow operator are a special case of a procedure. The general form of a procedure is as follows.

procedurename `:= proc` (variables)
command sequence
`end proc:`

wm7u Td [(x)]TJ8 [(c)580(n)27(um59(c0(se)50(q(c)linJ8 [(c)ple,)6(prin)2r)5+an8TJ - Td5n)2r apro

The procedure is used in the form procedurename(variables) and the value produced

```
f := proc(n)
      option remember:
      if n=0 then return 0
      elif n=1 then return 1
      else return f(n-1)+f(n-2)
      end if:
   end proc:
```

Check that this gives the same value for f(7)

2.23 Reading Files into Maple.

Suppose that you are running a Maple session and you wish to use a sequence of commands in a file. There are two essentially different ways of doing this, each with a different effect.

1. If the commands in a Maple worksheet (.mw) or Maple Input file (.mpl) file then you can open a new worksheet containing the text of the commands by a method similar to that for saving a Maple session given in section 1.4.

   (a) Click on the second button from the left in the tool bar (it looks like a half-open folder): a dialogue box called Open will appear.

   (b) At the bottom of this dialogue box, you will see a box labelled Filetype which looks something similar to:

All Worksheets (mw, mws) r

   (c) If you want to open a Maple Input file, click on the symbol r and a menu will appear. Clicking on the Maple Input line. If you want to open a .mw file, ignore this step.

   (d) The main part of the dialogue box will be a list of all the .mw (or .mpl) files in your home directory. If the file is there then click on it and the name will be transferred to the box beside the word File Name.

   (e) Otherwise, move the mouse pointer into that box and enter the name of the file you want to read including the file extension.

   (f) Click on the word OK.

   (g) If you are opening a Maple Input file then all the commands are placed in one execution group, which is generally unhelpful.

   The commands in this new worksheet will not be executed unless you go through them pressing <Enter> in each execution group. This method is good if you want to modify the commands before executing them. Alternatively, select Execute from the Edit menu (see 3.7) or click on the Execute Icon (three exclamation marks).

2. If the file is a Maple Input file then the command

   ```
   read( filename );
   ```

   will execute the commands in your file and display the results in your Maple worksheet, but it will not display the text of the commands. If filename contains any special characters it must be surrounded with back-quotes ` as in

   ```
   read( `proc.mpl` );
   ```

   The read command is usually used when you have defined a procedure (or something that takes a lot of typing, like a matrix) which you want to use in more than one Maple session (or more than once in a session) since the commands are executed immediately each time the read command is used.

   Warning: If you use a read command in a Maple command file that you are submitting as an assignment, remember that you must submit the file that is read as well as the command file, otherwise we cannot process the submission for marking. Remember that read does not give you a copy of the commands it carries out, only their results.

3. It may happen that there are commands that you want to read in every time you start a Maple session. You can do this automatically by putting the commands in a file called .mapleinit in your home directory. This file will automatically be read in at the start of each and every Maple session (including when you use restart).

Note:

Chapter 3

MORE ON THE MAPLE GUI.

The Maple GUI (Graphical User Interface) was introduced in chapter 2, where some of its features were described. In this appendix, we build on what was said there and give a more complete treatment of Maple's features.

3.1 Maple's Default Settings

Maple is installed on all PCs (both Windows and Linux) in the School of Mathematics and Statistics labs. However, the default settings have been modi ed to simplify the interface. If you buy your own copy of Maple for use at home, you are strongly encouraged to modify the settings of your copy to match those in the lab. This is because all documentation provided in your course assumes you are using a Maple in the School's labs and any Maple tests in our lab will use those settings.

The rst time you start Maple after installing it with the default settings you should see a window like the one in Figure 3.1.

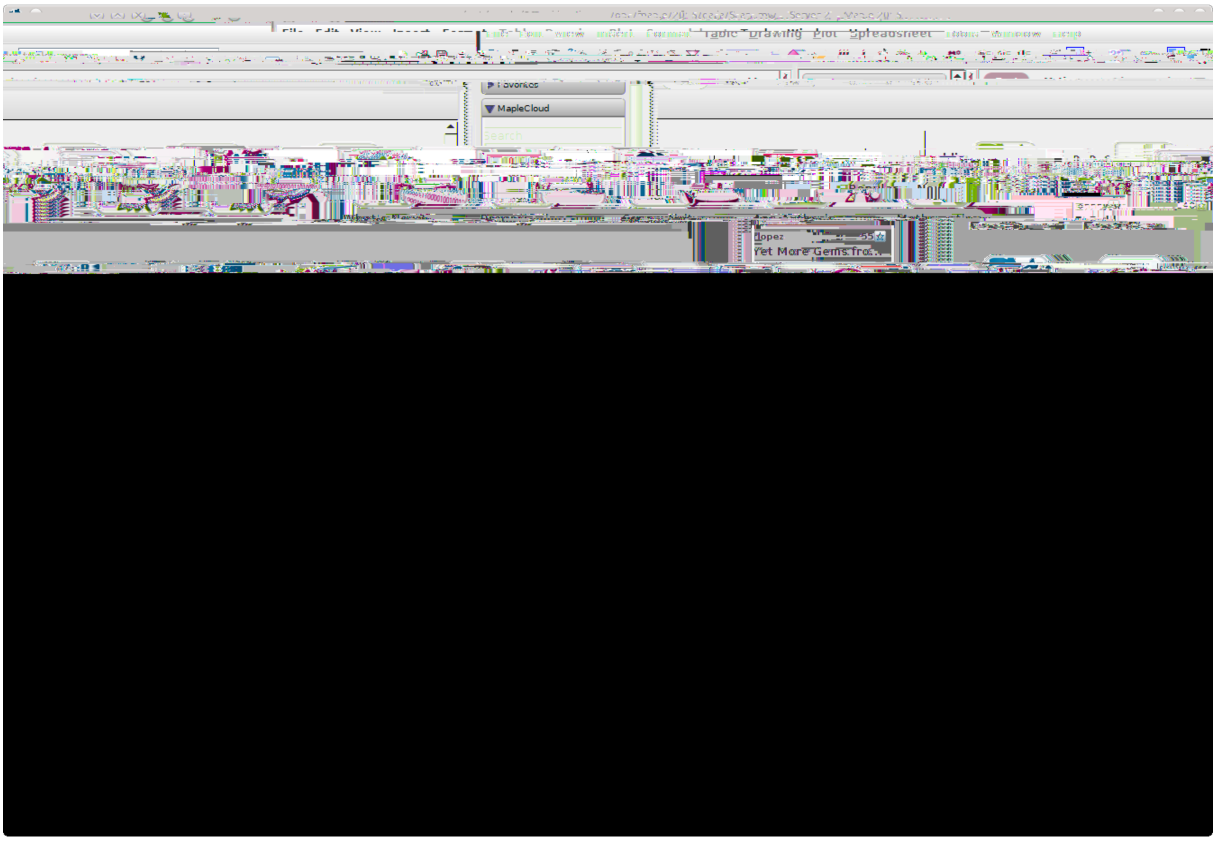

Figure 3.1: Maple with default settings

3.2 Maple Settings in the Labs

To modify the default Maple settings to match the lab settings, do the following:

Close the palette (the region of the left of the Maple window containing lots of wide buttons). To close the palette, click on the triangle pointing left at the top of the bar between the palette and the main document area of the Maple window.

From the `Tools` menu select `Options:::` .

- Click on the `Display` tab and for the rst item called `Input display` select `Maple notation`.
- Click on the `Interface` tab and for the fourth item called `Default format for new worksheets` select `Worksheet`.
- Click on the `Apply Gobally` button to save the new settings and close the options window.

Select `Maple Help` from the Help menu. Then click on the Help window's `View` menu and select `Display examples with 2D math` (this needs to be turned o).

Close all Maple windows to save the changes you've made to the settings.

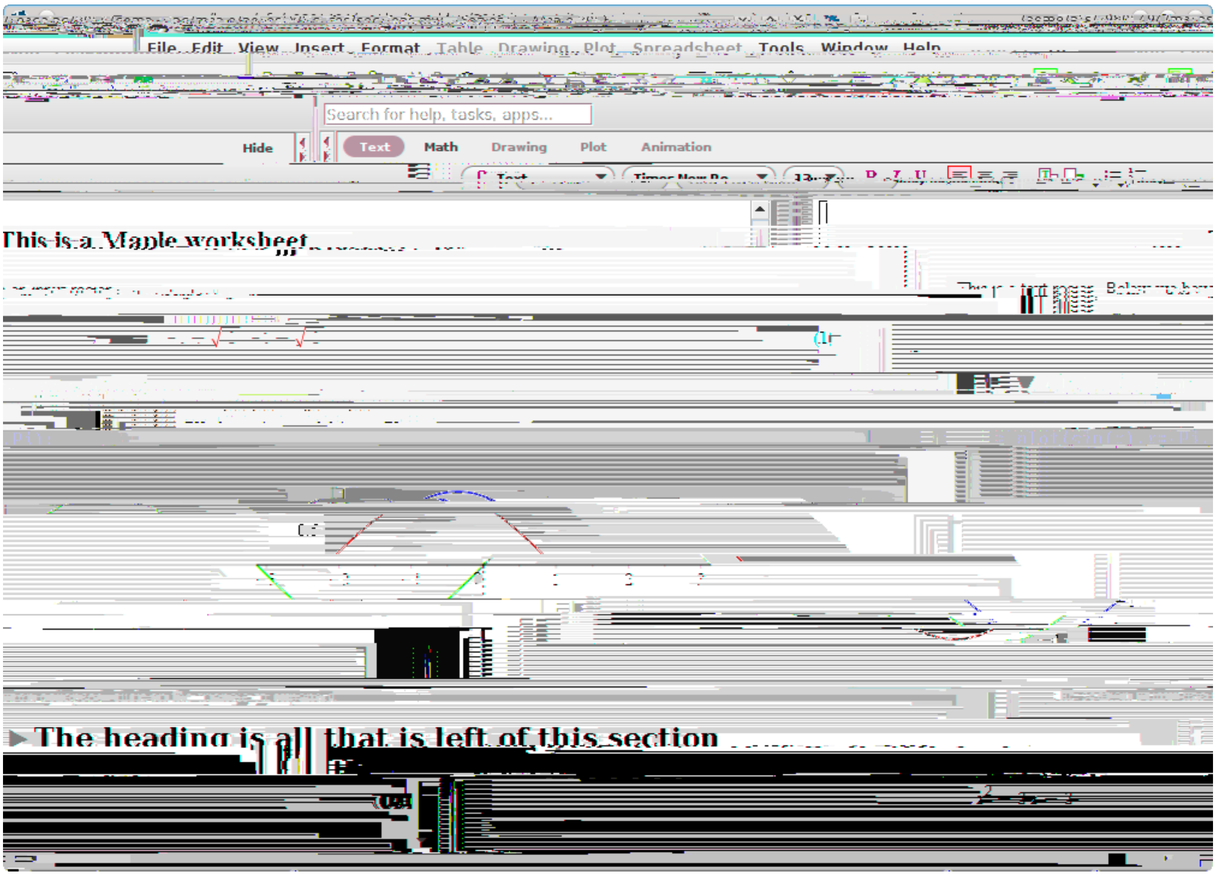

Figure 3.2: The Maple Window

These are collected into *execution groups* consisting of lines of text and/or input commands. When you press <Enter> with the cursor anywhere within an input region of an execution group, all of the commands in that group are executed and their outputs are placed at the end of the group. Each execution group is indicated by a left square bracket [and each region within an execution group is called a *paragraph*.

Execution groups in a worksheet, or material in any Maple window, may also be organised into expandable *sections* indicated by a grey triangles in front of their headings. When you click the left mouse button on the triangle, an expanded section is collapsed (hidden) and a collapsed section is expanded | there is one of each in figure 3.2. We will not discuss the creation of expandable sections in these Notes, but you will see them in the Maple help pages, for example.

It is possible to have several worksheets open at the same time. Each worksheet can be printed as a separate document (see section 2.8), and has its own window manager allowing you to raise, move or resize it. The worksheet which is 'on top' (i.e. which has been raised) is called the *active* worksheet.

3.5 The Menu Bar

The menu bar is at the top of the Maple window. Depending on what type of region the cursor is, some of these menus will be greyed out. For example, the Plot menu is greyed out in figure 3.2, but if the cursor were over the plot of sin x, it would become active.

LaTeX (.tex) which are processed by the mathematical wordprocessor LaTeX (as an example, these notes were prepared using LaTeX).

Maple Input (.mpl) which contain just input and text regions. This is useful if you want to write Maple scripts that can the executed without using the GUI using just the commandline version of Maple. This is not something that you will need to do in any first year mathematics course at UNSW.

Maplet (.maplet) which produces files that be used without by the Mapleviewer. You will not need this.

Maple Text (.txt) which is an ASCII form of the worksheet. It is meaningful to all versions of Maple and humans and contains output. The different types of regions are preserved as follows:

- all lines in input regions are saved as lines starting with >;
- all lines in text regions are saved as lines starting with [illegible]

all lines starting with # will be put into text regions;

all other lines will be ignored.

Warning: In all three cases above, the le is included in the new worksheet and appears on the screen (possibly including output), BUT it has NOT been processed by the Maple processor at this session: it is as though you entered it all using the editing features. If you want to make corrections or changes rst, then you may need to go through the whole text, pressing <Enter> on every command line (after making the changes) as described in section 1.3.7. If you want all the commands to be executed just as they are, then click the Execute Icon (three exclamation marks).

Notes :

1. If you open a Maple Input File or a Maple Text le then all the commands are placed in the same execution group. This is rather annoying.

2. A list of the most recently opened les will also appear on the File menu, if you have ever used this feature. Clicking on the name opens the le.

Exercise

Restart Maple and open the le exercise.mw that you made in the last exercise. Use the mouse pointer in the scroll bar to move around on the screen. Note that the output regions are as you would expect. Move the pointer to the last empty input region (i.e. the last empty line with a >) and click the left mouse button to bring the screen | cursor to that point, then type

c;

You do not get anything except for the unassigned value. Why? Well you have not sent anything to the Maple processor yet. To do so, move the cursor up to the top line (the a := command line) and press <Enter> on each line.

Leave this session on the screen for the next exercise.

`Close...` : Close the active worksheet or document.

If you select the appropriate Close... and have made any changes to the active worksheet or document since the last time you saved it, Maple will ask you if you want to save it before closing. Otherwise it will simply close it.

`Print` : Print the active worksheet.

When you select Print , a dialogue box will appear giving allowing you to print the worksheet directly or to a le. There are tabs to change the appearance of the printout, but we advise you to leave these alone. If you just click on print, then your worksheet is printed on one of the lab printers (see the lab notes for details).

If you tick the Print to F_ile box, then a dialogue box will prompt you for a le name, the default given being out.ps | the `.ps` extension telling you it is a PostScript le. You can then view this le with a postscript viewer. The le manager on the Linux PCs in the lab knows how to open view this le. Note that you might nd the Export as PDF option described above to be a more convenient way to `print to le'.

`Print Preview`

Gives you a preview of the worksheet to be printed before proceeding to the Print Dialogue Box.

`Exit` : Terminate the Whole Maple Session.

If you select Exit, then Maple will close all worksheets and documents before terminating the whole session. If you have made changes to any worksheet or document since the last time you saved it, Maple will ask you if you want to save before closing.

3.7 The Edit Menu

This menu deals mainly with deleting or moving highlighted material using Maple's clipboard. This clipboard is similar to (and uses) the mouse clipboard.

Highlighting Material : `Select All` and `Find ...`

Material in a Maple window can be highlighted using the mouse in the usual way. You can also highlight the whole of a worksheet by selecting the option Select All

You can delete the whole paragraph containing the cursor by selecting Delete Element. Note that input and output regions in an execution group are separate paragraphs and so deleting one of them in this way does not delete the other one.

It is also possible to remove the output regions from either all of the highlighted

Recovering from an Error : `Undo result`

The 2D-Math options are discussed in section 3.14.

Exercises Check that the Maple commands of the previous exercises are on the screen.

1. Move the cursor to the output region of the rst line. Insert a text region above it by using the Paragraph option in the Edit menu. Enter

   this is an example

   (it will printed in black instead of red). Using <Enter> will give you a new line of text, and you can continue adding comments.

2. Create an input region after the above execution group using the Execution Group option in the Insert menu. Now type in the comment

   ```
   # this is an example
   ```

   (it will printed in red) and press <Enter>. This puts you on the line

   ```
   b := expand(%);
   ```

   To add more lines of comments using #, insert another input region and type in the comment beginning with # etc.

Page Break

This puts a page break at the cursor. On the screen this looks like a line across the workshet with the words \page break" in the middle, but when you print the worksheet a new page is taken at that point.

3.10 The Format Menu

The options in this menu are used to control the look of individual characters or words, such as their type-face (font) or size, or the positioning of paragraphs within the worksheet. We advise you not to get too involved with this menu at this stage. Note that any change you make only refers to the highlighted text or the current paragraph, and most8b327(thenb327(thenb327(raph,)-3s7(thenbaJ7.44446 Tume)-287(thad [(6 T,hm..gh)27(ted)]T

General

The two interesting options here are AutoSave and Kernel Mode.

If you tick the Enable AutoSave box then Maple will save your worksheet every 3 minutes (you can change the time gap if you wish). The file is saved as a Maple Worksheet, but is not given the .mw extension, instead it is called something like `file_MAS.bck`, and is overwritten each time.

For Kernel Mode, the default is

Precision

This tab allows you to control the number of digits Maple uses in its calculations and displays: it is similar to resetting the global variable Digits .

Security

This tab is for features you will not have any use for in rst year.

3.12 The Help Menu

We have already looked at the most important options in this menu in section 1.5. To nd out about the rest, select the option \Using the Help System" under \Manuals, Dictionary, and more" and follow the links which interest you.

3.13 Alternatives to Commonly Used Menu Options

There are two shorter alternatives to the menus for many commonly used commands, one of which uses the mouse and one of which uses the keyboard.

3.13.1 The Tool Bar

Some of the options from the menus in the menu bar can be replaced by clicking the left mouse button on one of the buttons in the tool bar. The STOP button is described in section 1.3.3, and the rst 13 buttons are the New, Open, Save, Print and Print Preview

1. Open the palette dock by clicking on the right pointing triangle in the left edge of the worksheet.

2. Open the expression palette by clicking on the appropriate grey triangle

3. Click on the $\int f\,dx$ symbol.

4. Type the expression you want to integrate, using the caret ^ for a power and the down arrow key to go back down.

5. Use the <Tab> key to move the the next part of the expression (the variable of integration and type t .

6. Press <Enter> . There is no need for a colon or semi-colon.

There are a large number of palettes with various things you might wish to enter. If you want to use them, then you may — it is even acceptable to use 2D Math notation in the laboratory test if you wish

3.14.1 Symbol recognition

The top palette (which is open by default) is for symbol recognition. This allows you to draw an approximation to a symbol and ask Maple to make a guess at what you mean. Draw the symbol by holding down the left mouse button in the scratchpad and then click on the right button to ask Maple to guess. The left button clears the scratchpad.

Appendix A

SUMMARY OF MAPLE COMMANDS.

General

<Ctrl>-F4	emergency exit
#	indicates a comment
;	indicates end of a command

Algebra: Simplification

expand(expr)	expand expression
combine(	

Calculus

- diff(expr, vars) — di erentiate expression with repect to vars
- D(function) — derivative of a function of one variable
- D[n](function) — derivative of a function with respect to its n^{th} argument
- D[n_1, n_2, . . .](function) — derivative respect to argument n_1 then n_2, . . .
- int(expr, var) — integral of expr with respect to var
- int(expr, var=lo.. hi) — de nite integral of expr for var from lo to hi
- limit(expr, var=value) — limit of expr as var approaches value
- limit(expr, var=value, left) — limit from the left
- limit(expr, var=value, right) — limit from the right
- sum(expr, var=lo.. hi) — sum of expr for var from lo to hi
- product(expr, var=lo.. hi) — product from lo to hi
- Diff , Int , Limit , Sum, Product — inert versions

Other Calculus

- taylor(function, var=value, order) — Taylor series about var = value
- convert(t-series, polynom) — convert a Taylor series to a polynomial
- dsolve(ODE, y(x)) — solve ODE in derivative of y(x)
- dsolve(f ODE, condsg, y(x)) — solve ODE subject to conditions like y(0)=1, D(y)(0)=1

Student Calculus Package

- with(student) — load student calculus package
- changevar(old-expr=new-expr, inert-expr, new-var) — change of variables
- value(inert-expr) — evaluate an inert integral etc.
- intparts(inert-int , u) — integrate by parts
- convert(integrand, parfrac, var) — convert to partial fractions
- rightbox(expr, var=lo.. hi, n) — Riemann sum plot
- leftbox, middlebox — similar
- rightsum(expr, var=lo.. hi, n) — Riemann sum calculation
- leftsum, middlesum — similar
- simpson(expr, var=lo.. hi, n) — apply Simpson's rule
- maximize(function) — nd maximum value of function
- minimize(function) — nd minimum value of function

LinearAlgebra Package

- with(LinearAlgebra): — load LinearAlgebra Package
- <sequence> — create a Vector
- <Vector1 | Vector2 | ...> — create a Matrix
- <M1 or v1 | M2 or v2 | ...> — augmented matrix (side by side)
- <M1 or v1 , M2 or v2 , ...> — stacked matrix (on top of each other)
- Vector[i] — i^{th} element of vector
- Matrix [i, j] — $(i,j)^{th}$ element of matrix
- Row(Matrix , i) — i^{th}

Index

aborting, 6
about, 51
absolute value, 18, 31
abuse of notation, 57

parametric, 34
polar, 34

D, 22, 48
defaults, 62
de ning functions
arrow, 21
as procedures, 55
denominator, 26
common, 20
rationalising, 20
dependent variable, 17
derivative, 21
higher order, 22
partial, 47
Determinant, 43
dialogue box, 3
dictionary, 9
Di , 36
di
in Maple, 21

general solution, 42
geom3d, 44
geometry, 44
    3D, 44
graphs, 31
    intersections of, 33
greatest common divisors, 49
GUI
    Maple, 61

help
    Maple, 9
Help Browser, 9
Help Navigator, 9
hyperbolic functions, 17

i (square root of 1), 15
identity matrix, 39, 44
if - then, 52
imaginary part, 31

partial fractions, 24
percent, 5
, 15
piecewise functions, 33
plane, 45
plot, 32
    buttons, 32
    scaling, 33
plot3d, 35
plots
    3D, 35
    comparing, 33
    implicit, 34
    package, 34
    parametric, 34
    polar, 34
plotting, 31
point, 45
polarplot, 34
polynomial, 26
    characteristic, 43
    sorting, 27
    Taylor, 49
Power, 49
precedence, 13
prettyprint, 71
prime numbers, 51
principal value, 31
print, 53, 56
printf, 53
printing
    graphs, 33
problems
    common Maple, 57
procedure, 55
    inert, 35, 49
    recursive, 56
Product, 36
product, 28
    cross, 44
    dot, 44
properties, 50
punctuation, 57

radicals, 20
radsimp, 20, 50
range, 30, 32
rank, 43
rational functions, 20
reading in les, 58
real part, 31
recalculation, 57
recurrence relations, 50
regions, 4
remainder, 49
remember, 57
repetition, 22, 25
reserved names, 15
restart, 16
restart icon, 72
result
    last, 5
return, 56
rhs, 26
Riemann sum, 36
rightbox, 36
righthand side, 26
rightsum, 36
root, general, 30
RootOf, 30, 31
round, 18
row operations, 40
RowOperation, 40
RowSpace, 43
rref, 42
rsolve, 50

scalar constants, 44
scaling, 34
sec, 17
sech, 17
section
    expandable, 63
selecting, 26
    from Matrices, 38
    from sets and lists, 38
    from Vectors, 38
    operands, 26
seq, 25, 34
sequence, 24
    of repetitions, 22
series, Taylor, 48
set, 24
set algebra, 49

sets, 25
signi cant digits, 14
simplify, 20
Simpson's Rule, 36
sin, 17
sinh, 17
solutions
  checking, 29
  complex, 27, 30
  exact, 29
   oating point, 30
  general, 42
  not nding, 30
  ranges for, 30
  selecting, 27
  simplifying, 31
  to linear equations, 42
solve, 29
solving, 29
  di erence equations, 50
  di erential equations, 47
  polynomial equations, 29
  recurrence relations, 50
  several equations, 29
  simultaneous linear equations, 40
sort
  in Maple, 26
sorting
  in Maple, 26
sphere, 45
square root, 18
structure, 24
student, 35
SubMatrix, 40
subs, 19
substituting, 19
  in list, 27
  in set, 27
Sum, 36
sum, 27
  Riemann, 36
supressing output, 18, 41, 46, 53, 55
switches, 64
symbol recognition, 73

tan, 17
tanh, 17
Taylor polynomials, 49
Taylor series, 48
text region, 4
tick marks, 33
toggle, 64
Transpose, 43
trigonometric
  expansions, 20
  functions, 17
truncate, 18

unassign, 15
unassignment, 15
underline character, 15
unknown, 14

value, 14, 36, 37
  principal, 31
  returning, 56
variable, 14
  change of, 36
  dependent, 17
  dummy, 21
  independent, 17
  names, 15
Vectors, 24, 37
  arithmetic, 43
  index functions, 37
  operations on, 43
  selecting entries, 38

warning
  assigning values, 16
  defaults, 70
  functions, 17
  reserved symbols, 38
  using read command, 59
  worksheets, 16
while, 52, 53
window
  Maple, 2, 61
with, 18
worksheet
  active, 63
  Maple, 3, 4