A Brief Introduction to *Mathematica* for 231M

Mathematica is a general purpose computer algebra system. That means it can do algebraic manipulations (including calculus and matrix manipulation) and it can also be used to draw graphs.

Starting Mathematica

Starting Mathematica from an X−terminal(Maths)

The command "mathematica" will open a *Mathematica* notebook interface to the system, or you should have the possibility of starting it with one of the items under the middle mouse button on your desktop.

The notebook interface is the most sophisticated interface available. (It was used to create these notes, for instance.)

When the system starts you get two windows immediately. One is the main window where you do your work (called a notebook window) and the other is a palette of symbols that you can use to input some (but not all) instructions. By using the palette, you can have your input look somewhat like standard mathematical notation (with superscript for powers, fractions that look like fractions, and a few other nice things). However, everything can be done without using the palette and the palette does not provide access to the full range of *Mathematica* features. For that reason, these notes largely ignore the palette options in favour of text−styleinput.

Using Mathematica from outside the Mathematics department

Mathematica is available on PC and Macintosh computers around College. Notebooks created on these computers should (in principle) be readable by the *Mathematica* on the mathematics computer system, provided you can transfer the notebook file (the .nb file) from one system to the other. ftp is one way to do that. Email attachments is another.

Mathematica commands

Mathematica has a large collection of Mathematical functions it knows about. There is a whole book about them, in fact, and all we can hope for is to learn enough to do for our purposes.

There is a method in the way *Mathematica* chooses its names for things −generally it uses one of the commonly used names with the *first letter capitalised.* Also *arguments of* *functions have to be in square brackets*, rather than round brackets as is more common in mathematical usage like f(x). Following this logic, Cos[x], Sin[x], Tan[x], etc, are the trig functions in *Mathematica*−speak.The command for drawing a graph is Plot[ ] and the command for drawing parametric graphs is ParametricPlot[ ] (inside the square brackets, you must specify at least the thing to be plotted and the range for the plot −we will come back to this later).

After this explanation of the way *Mathematica* does things, and with some experience, you may be able to guess the right *Mathematica* name for a function you want.

In the notebook interface, you type instructions and press **Shift−Return** at the end (hold down the Shift button and press the return key). Another way that should work is to press the Enter key on the numeric keypad (extreme right of most keyboards). In the text interface, type the instructions and just press enter.

Online help (text mode)

If you think you remember, or just guess, the name for a function, you can get a cryptic summary of what the system knows about the name with the questionmark command. For example, suppose I wanted to find out if a number is prime and I imagined that the command for this is called Prime. I can type in ?Prime (and press Shift−Return)and here is what I get.

```
?Prime
```

```
Prime[n] gives the nth prime number.
```

This is clearly desigend to find the 10th prime, or the 100th one. It might be useful, but is not what I wanted. There is another form of the question mark command with a * at the end which will tell me about anything *Mathematica* knows about beginning with Prime.

```
?Prime*
```

```
Prime   PrimePi PrimeQ
```

We can then look for information on PrimePi and PrimeQ to see if they are what we want.

```
?PrimePi
```

```
PrimePi[x] gives the number of primes less than or equal to x.
```

```
?PrimeQ
```

```
PrimeQ[expr] yields True if expr is a prime number, and yields False otherwise.
```

There is a slightly longer (usually more technical) description available using a double question mark. Thus ??PrimeQ would tell you some more intricacies about the PrimeQ function.

The help menu

There is a more convenient (and more informative) help system available through the "Help" menu button at the top of the notebook window. Choose the item "Help ..." in that menu and you will get a new window which provides online access to the whole *Mathematica* book (if you have the ‘Built−inFunctions’ button checked). It starts with an organised way to go through the summary at the back of the book. The summary gives a brief account of what each *Mathematica* function does and provides references to the section of the book where the function is explained. Also there are sometimes examples in the summary. If you click on the links (highlighted) you should call up the page of the book referred to.

If you check the button ‘The *Mathematica* Book’ you can browse the whole book online.

Number Theoretic Examples

We can try out some of these commands a little, now we know about them.

```
Prime[1]
```

```
2
```

```
Prime[1004]
```

```
7949
```

```
PrimePi[7949]
```

```
1004
```

```
PrimeQ[7951]
```

```
True
```

This shows that 7949 and the next odd number 7951 are both prime. Twin primes of this kind are of interest in number theory.

There is a related *Mathematica* command FactorInteger for factoring whole numbers (we will see that Factor[ ] is used for factoring polynomials, not factoring whole numbers).

```
FactorInteger[7953]
```

```
{{3, 1}, {11, 1}, {241, 1}}
```

This means that 7953 has three prime factors, 3, 11 and 241 and that each occurs just once. So if we multiply these 3 primes, we should get 7953.

```
3 * 11 * 241
```

```
7953
```

Algebra

Here are some simple examples of *Mathematica* doing algebraic calculations. Note the use of Expand[ ] and Factor[ ] to tell *Mathematica* what we want it to do. (Remember the Shift−Return.)

```
Expand[ (x + 1) (x + 2) (x + 3)]
```

$$6 + 11\,x + 6\,x^2 + x^3$$

```
Factor[x^3 + 2 x - 3]
```

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

Here we have input the power using the exponentiation symbol ^ (and we could need parenthesis is the exponent is a complicated expression like a+1). Using the palette we could instead input the same thing with a nicer supescript layout in the input side.

Factor[x^3 + 2 x - 3]

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

Or, there is a way to do this whithout using a palette, where we use the key combination Control ^ to start the superscript (power) and the combination Control space to finish the superscript. (Hold down the Control key and then press the other key.)

Using either of these techniques you can make your input look more like the way you are used to writing mathematical formulae, though of course it is an extra complication to have two ways to do things.

```
(x + 1)^10
```

$$(1 + x)^{10}$$

```
Expand[(x + 1)^10]
```

$$1 + 10\,x + 45\,x^2 + 120\,x^3 + 210\,x^4 + 252\,x^5 + 210\,x^6 + 120\,x^7 + 45\,x^8 + 10\,x^9 + x^{10}$$

An important facility in *Mathematica* is the ability to solve equations, both exactly (symbolically) with Solve[ ] and numerically with NSolve[ ].

```
Solve[ x^2 + 5 x + 2 == 0, x]
```

$$\left\{\left\{x \to \frac{1}{2}\left(-5 - \sqrt{17}\right)\right\}, \left\{x \to \frac{1}{2}\left(-5 + \sqrt{17}\right)\right\}\right\}$$

```
NSolve[ x^2 + 5 x + 2 == 0, x]
```

$$\{\{x \to -4.56155\}, \{x \to -0.438447\}\}$$

You might want to note that we need to specify the equation to be solved with a double equals sign == and also we need to say what unknown (in this case x) to solve for. The reason for the == as opposed to = is that *Mathematica* uses a single = to mean an assignment, to make the left hand side of the = have the value of the right hand side. We will come to using this assignment facility later in the course.

You might like to know that there is a general purpose N[ ] command in *Mathematica* to get the numerical value of any expression. So a nested

```
N[Solve[x^2 + 5 x + 2 == 0, x]]
```

```
{{x → -4.56155}, {x → -0.438447}}
```

will also work in this case instead of using NSolve. In fact NSolve[ ] is identical with N[Solve[ ]]. There is another command FindRoot[ ] which can succeed in finding a numerical value for a solution where Solve[ ] and NSolve[ ] may fail. FindRoot gives one solution and you have to tell it a starting value of the unknown where to look for a solution.

```
FindRoot[ Cos[x] == x, {x, 0}]
```

```
{x → 0.739085}
```

Notation and Rules

The remarks above about the difference between == and = indicate that it is time to spell out the basic rules and notation used by *Mathematica* for interpreting what you type.

Addition, subtraction and division are +, −and /. For multiplication there are quite a few options (3 altogether). Looking back over the previous examples, there are examples where we used * for multiplication (as in 3 * 11 * 241) but there are other examples where we did not put in any * (for example the 5 x + 2 in the last equation could be 5*x + 2). So *Mathematica* will interpret a space between two quantities as meaning multiplication in cases standard mathematical notation would take that meaning, except that square brackets are used as indicating arguments of functions and curly brackets { } have another special purpose (for lists of items). Ordinary round brackets ( ) have a grouping affect as in ordinary algebra notation, but not other types of brackets. Thus we were able to write (x +1)(x +2) where we might have been more explicit about the multiplication between the factors with (x + 1)*(x + 2). So we can indicate multiplication with a star *, or a space, or sometimes by juxtaposition without any space needed. This latter possibility (leaving out the space) is available for examples like (x + 1)(x + 2) where we mean to multiply grouped things together and also in the case of numerical quantities times variables, such as 2x. For two times x, we can write 2*x, 2 x (with a space) or just 2x (no space between).

All this looks good, but there is a difference between *Mathematica* notation and ordinary algebra notation when it comes to multiplication. If we have two quantities called x and t, and we want to multiply them, we must put in the * or else a space (x*t or x t are both ok). The catch is that *Mathematica* will treat xt (no space in between) as a new name for a new quantity. You can see this if we divide x*t by x and see what we get with the *, with a space and with no space.

```
x*t/x
```

t

```
x t/x
```

t

```
xt/x
```

$$\frac{xt}{x}$$

This may seem to be a silly way for *Mathematica* to do things, but it has the advantage that you can use names for quantities that are longer than one letter. So, for example, we could use "mass" for mass and "time" for time, and that might be clearer than having to abbreviate them as something like m and t.

```
mass / (mass * time)
```

$$\frac{1}{\text{time}}$$

```
m a s s / (mass * time)
```

$$\frac{a\, m\, s^2}{\text{mass time}}$$

Note that the spaces in the "m a s s" mean that *Mathematica* treats it as m*a*s*s

Observe that we have been using the hat notation for "raised to the power of".

The / notation for division is the standard way to indicate division or fractions, but as with Control ^ for powers it is possible to input fractions so that they look like fractions. You can do that with the palette or by typing Control / to enter the denominator and Contol space to finish the denominator. One advantage of these methods, apart from making it easier to read the input, is that you don't have to put parentheses around your numerators and denominators.

$$m\,a\,s\,\frac{s}{\text{mass} * \text{time}}$$

$$\frac{a\, m\, s^2}{\text{mass time}}$$

Differentiation

$$\text{D}\left[a\,x\,\text{Cos}[x^2] - \frac{x}{x+1},\ x\right]$$

$$\frac{x}{(1+x)^2} - \frac{1}{1+x} + a\,\text{Cos}[x^2] - 2\,a\,x^2\,\text{Sin}[x^2]$$

Integration

```
Integrate[ x Cos[x], x]
```

```
Cos[x] + x Sin[x]
```

```
Integrate[ x Cos[x], {x, 0, Pi} ]
```

```
-2
```

These were examples of indefinite integration (no limits) and definite integration (with limits). Pi (with a capital P) is the text way of inputting the number pi. With the palette you could enter the usual symbol for Pi and you could also make the inetgral look like an integral.

$$\int_0^{\pi} x \, \mathrm{Cos}[x] \, dx$$

```
-2
```

Solving differential equations

```
DSolve[ D[y[x], x] - a y[x] == 0, y[x], x ]
```

$\{\{y[x] \to E^{a\,x}\, C[1]\}\}$

We needed to tell DSolve the differential equation, the unknown function y[x] and the independent variable x.

Plotting in 2D

As mentioned a long while back, the basic commands for ordinary graphs is Plot[ ] and the command for parametric plots is ParametricPlot[ ]. We can use the double question mark help facility to see that there are a huge variety of ways of controlling the way a Plot comes out, depending on what aspect of the graphs you think needs to be shown.

```
?? Plot
```

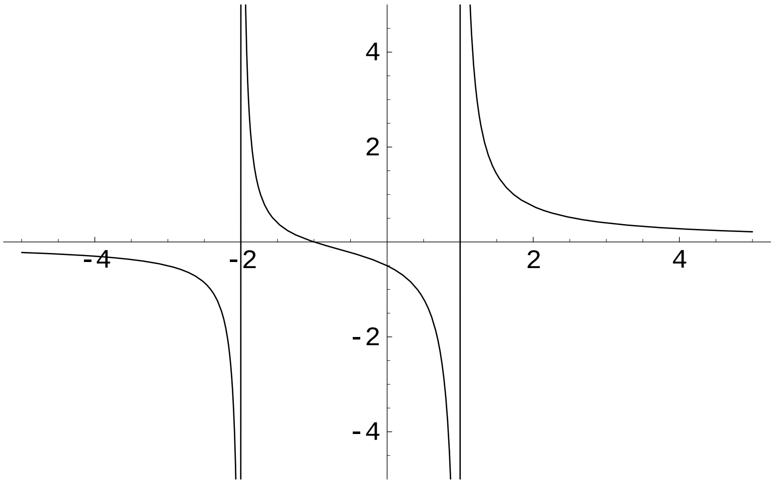
```
Plot[f, {x, xmin, xmax}] generates a plot of f as a function of x from xmin
  to xmax. Plot[{f1, f2, ... }, {x, xmin, xmax}] plots several functions fi.

Attributes[Plot] = {HoldAll, Protected}

Options[Plot] = {AspectRatio -> GoldenRatio^(-1), Axes -> Automatic, AxesLabel -> None,
  AxesOrigin -> Automatic, AxesStyle -> Automatic, Background -> Automatic,
  ColorOutput -> Automatic, Compiled -> True, DefaultColor -> Automatic,
  Epilog -> {}, Frame -> False, FrameLabel -> None, FrameStyle -> Automatic,
  FrameTicks -> Automatic, GridLines -> None, ImageSize -> Automatic,
  MaxBend -> 10., PlotDivision -> 30., PlotLabel -> None, PlotPoints -> 25,
  PlotRange -> Automatic, PlotRegion -> Automatic, PlotStyle -> Automatic,
  Prolog -> {}, RotateLabel -> True, Ticks -> Automatic, DefaultFont :> $DefaultFont,
  DisplayFunction :> $DisplayFunction, FormatType :> $FormatType, TextStyle :> $TextStyle}
```

If we wanted to plot y = (x + 1)/(x^2 + x −2) over the range x = −5to x = 5, the command Plot[(x + 1)/(x^2 + x −2), {x, −5,5}] would do a reasonable job. In the following example, I have added a PlotRange option to control the range on the y−axis.(Without that I think *Mathematica* chooses too big a range.)

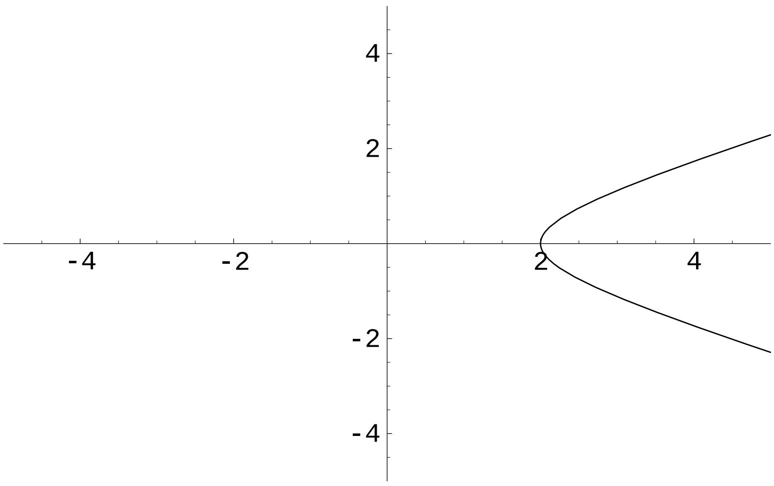
```
Plot[(x + 1) / (x^2 + x - 2), {x, -5, 5}, PlotRange -> {-5, 5}]
```



- Graphics -

For an example of a parametric plot, I use hyperbolic cosine and hyperbolic sine as a parametrization of the hyperbola x^2/4 −y^2 = 1.

```
ParametricPlot[{2 Cosh[t], Sinh[t]}, {t, -2, 2}, PlotRange -> {{-5, 5}, {-5, 5}}]
```



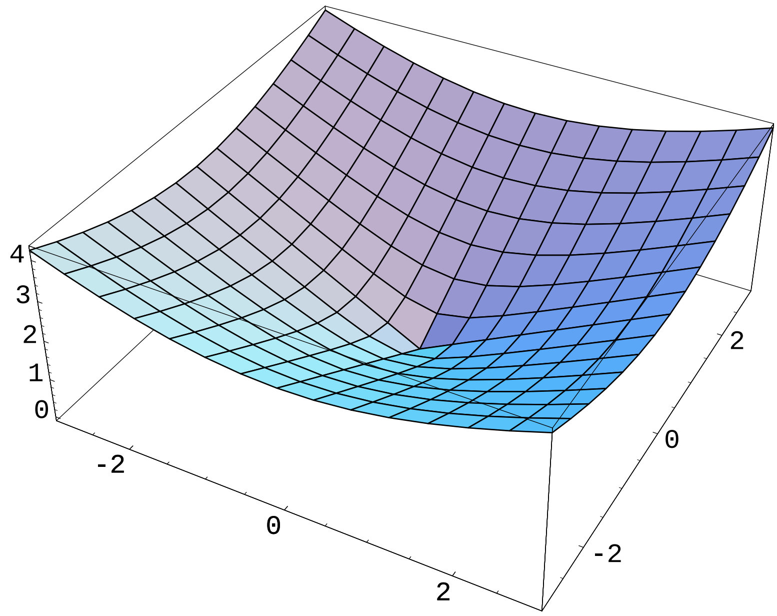
- Graphics -

This graph actually shows only the half of the hyperbola where x > 0. There is a mirror image half for x < 0.

Plotting in 3D

Mathematica is a rather good tool for making 3 dimensional graphs also. To plot a graph of a function z = f(x,y) of two variables, we use the Plot3D[ ] command like this.

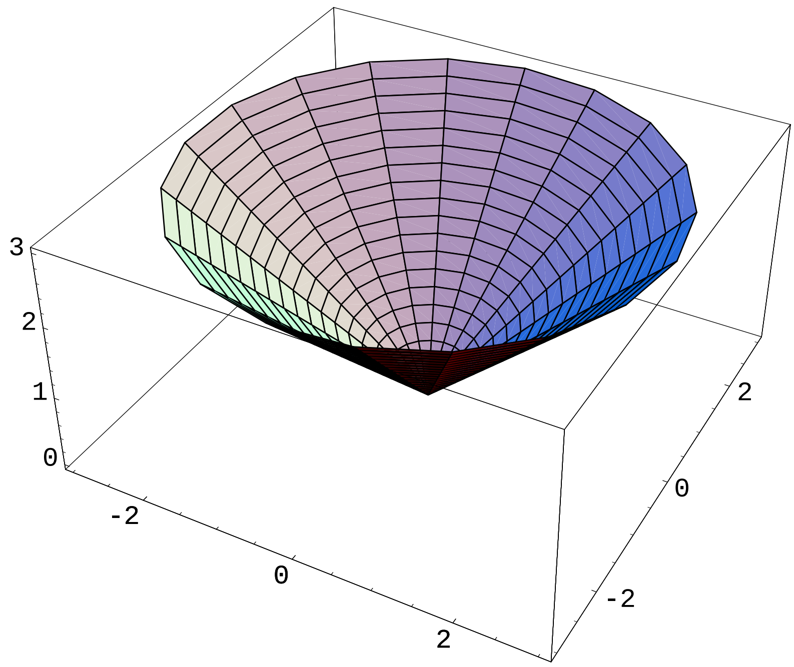
```
Plot3D[Sqrt[x^2 + y^2], {x, -3, 3}, {y, -3, 3}]
```



- SurfaceGraphics -

We know we should see a cone, or half a cone, but the square base in the x−yplane makes it hard to see this. If we plot the same cone using ParametricPlot3D[ ] and basing our parametric equations on cylindrical coordinates (r, θ, z) our cone becomes z = r. Recall that x = r cos θ and y = r sin θ

```
ParametricPlot3D[{r Cos[θ], r Sin[θ], r}, {r, 0, 3}, {θ, 0, 2π}]
```



- Graphics3D -

Note that we have used the palette (the standard ‘Basic Input’ one) to get the θ and π although we could have used any name we wanted in place of θ (for example the word theta) and we could alternatively get 3.14159... as Pi. The picture could be clearer if we left a gap in the cone by putting the range for θ as 0 to $3\pi/2$ for example. By default *Mathematica* puts axes in boxed style on 3D plots and chooses its own viewing angle, but these things can be changed with various options.

ParametricPlot3D can be used for curves in space also.

Printing

The easiest way to print out graphs and other work you do in a *Mathematica* notebook is to use the "Print ..." menu item you will find under the "File" menu. You get a little menu to fill in. You should select "A4 letter" paper size, untick the File box and tick the Print box (just click with the left button of your mouse). Finally you need to add to the "lpr" (that's LPR, by the way) in the Print line to direct your printout to a particular printer. If you are sitting in room 1.4, you should change the "lpr" to "lpr −Plaser14"(that'slaser with an L and not a one, but one four or fourteen). Your printout should come out on the laser printer in 1.4 if you add paper to the papertray. Substitute "lpr −Plaser15"for the printer in room 1.5.

Saving and reloading

With the notebook interface, you can save your work to date. Use the "Save" or "Save As" menu item in the "File" menu. You should save the notebook with a more useful name than Unititled−1.nb(but you should keep the extension nb for saved notebooks).

If you then start *Mathematica* at a later time, you can get back your earlier work. You can do this in two ways. Start *Mathematica* as already indicated and use the "Open ..." item in the "File" menu to open your earlier notebook. Another way is to start *Mathematica* by giving the command "mathematica mywork.nb" (replace mywork.nb with the actual name you saved it under).

When you reload a notebook this way, *Mathematica* will not recompute anything. If you had defined your own variables or functions (we will do this later), these definitions will not be active upon reloading the notebook. You must re−activateor recompute any such definitions by selecting them and pressing Shift−Return.There is also a menu item under 'Kernel' −> 'Evaluation' to recompute everything in the notebook in one fell swoop.

Bug: There is/was a small problem with the implementation we are using on the mathematics system. It causes *Mathematica* to make a mistake about which directory it is working in (sometimes) and the effect is that you cannot save your work when you do "Save As". If you look carefully at where it is trying to save your work, you will see the problem. Most of you have home directories called /u3/maths/2003/xyz/ (where xyz is replaced by your user name). *Mathematica* sometimes replaces the u3 at the beginning with something else. Click on it and fix it.

R. Timoney (November 2001)