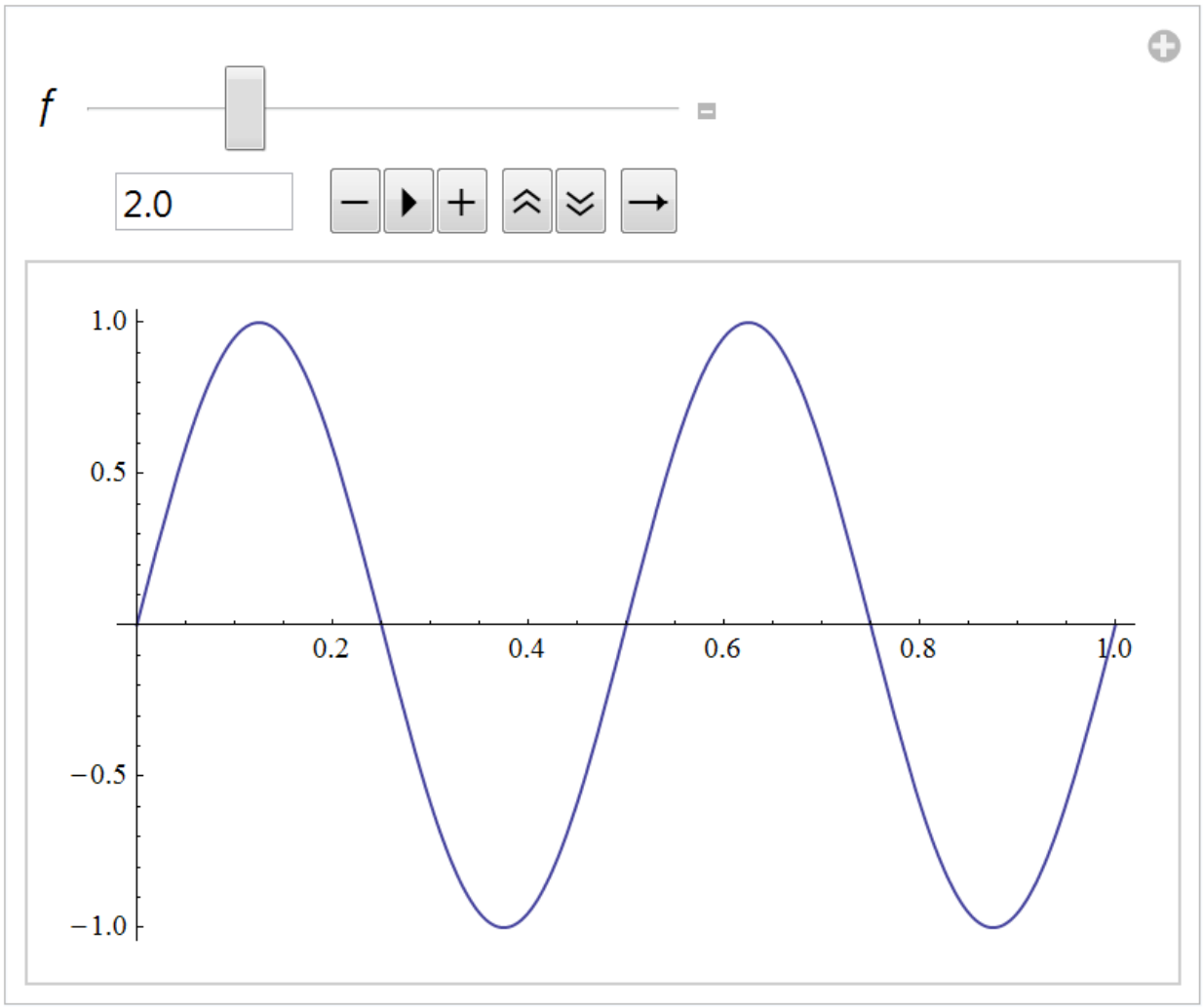
An Introduction to MATHEMATICA



Eric Peasley, Department of Engineering Science, University of Oxford
version 2, 2013

Introduction To Mathematica

Table of Contents

Introduction To Mathematica....1
Introduction....2
Notebooks....3
Algebraic Maths....3
Cells....4
Simple Arithmetic....5
Assignment....6
History....7
Lists....7
Built in Functions....8
Help....9
Palettes....10
Two Dimensional Expression Input....11
Special Numbers, Characters and Symbols....12
Free Format Linguistic Input....13
Operating on Lists....14
Algebraic Manipulate....16
Calculus....19
Graphs....21
Defining your own functions....24
Piecewise Functions....24
Compound Expression....25
Comments....25
Flow Control....26
Interactive Graphs....27
Further Information....28

Version 2, 2013.

Eric Peasley, Department of Engineering Science, University of Oxford.

Introduction

Mathematica is a mathematical software package that can be used by any member of the Engineering Department. This seminar will show you what Mathematica can do, and will let you assess how useful it could be to you. Mathematica is a huge package with far more features than can be covered in a single afternoon. However, the seminar will show you the basics of enter problems and obtaining and displaying results. It will also show you how to explore the package for yourself and find the things that you are interested in. The majority of the seminar is devoted to giving you real hands on experience of using Mathematica.

The main driving philosophy behind Mathematica is to produce a mathematical software package that is as close as possible to doing mathematics with a pen and paper. This shows itself in two ways. The appearance resembles normal mathematical text much closer than a program using a conventional programming language. You can easily use mathematical symbols like integral signs and Greek characters can be used in variable names. You can include explanatory text, with titles and subsections that make a Mathematica note book look more like a document than a program. However, the most important feature of Mathematica is that it can perform algebraic mathematics in addition to numerical mathematics. Computers mostly do numerical calculations. So for example, the expression $y = x^2$ takes the number x and squares it and puts the value into y. Using computational algebra the result y is a symbolic expression that can then be manipulated algebraically, differentiated and integrated using the same rules you use to manipulate equations.

Notebooks

When you start Mathematica, you get a menu bar right at the top of the screen. To get started you will need to create a Notebook. From the menu bar select **File►New►Notebook(.nb).**

Click in the Notebook and enter a mathematical expression such as :-

```
1 + 2
```

To evaluate the expression, press the **enter** key **on the keypad**. If you don't have a key pad then hold down the **shift** and press E**nter**. The two enter keys have different uses in Mathematica. The enter key on the main part of the key board is used to enter new lines and the enter on the keypad is used for evaluation.

Algebraic Maths

One of the great things about Mathematica is that you can do Mathematics symbolically as well as numerically. Take for example the expression below.

```
y = x^2
```

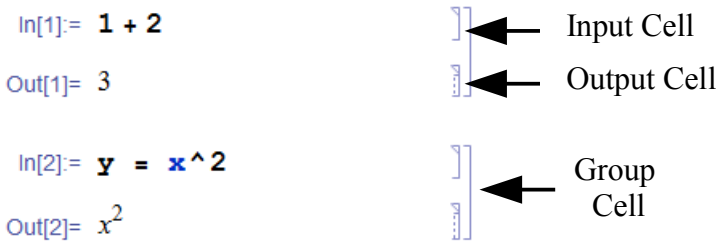
Assuming that x has not been assigned any value beforehand, then what is stored in y is not a number, but the algebraic expression x^2. Because y is symbolic, you can do things that you can't do it if was a number. For example, you can differentiate it.

```
D [ y, x ]
```

This function **D** differentiates y with respect to x. Mathematica knows the rules of Calculus and can differentiate x^2 as easily as you can.

Cells

The notebook is divided into cells. These are shown by square brackets of the right hand side of the notebook.



Input cells are where you entered the expression.
Output cells show the result.
Group cells that groups the input and output cells together.

Hiding Input or Output Cells

Double click on the bracket of the input or output cell that you want to retain.
Double click on the bracket of the group to redisplay the hidden cell.

Adding Other Cell Types

There are also plain text cells, title cells, sections cells and many other types of cells.
Click below any cell in your notebook and you will see a horizontal line.
On the left of the line is a plus sign.

Click on the plus sign to open the **Cell Insertion Assistant**.

Simple Arithmetic

Basic Operators

The basic mathematical operators are as shown below.

( )	
^	Power
*	Multiply
/	Divide
+	Add
-	Subtract

Precedence is the same as normal mathematics, so for example multiplication takes precedence over addition.

Multiplication

You don't have to use * to multiply two numbers. As in normal algebra, objects next to each other signify multiplication. However, in most cases you will need to put in a space to make it clear what you mean.

Expression	Meaning	Comment
A B	$A \times B$	Multiply A by B.
AB	AB	A single variable called AB.
2 3	2×3	The cross is automatically inserted between numbers.
2A	$2 \times A$	Multiply A by 2.
A2	A2	A single variable called A2.

Numerical Evaluation

By default, the output from Mathematica is given in its most exact form. The result may contain fractions or symbols such as **Pi** or ***e***. To get a numerical result, you use the function **N**.

N [22 / 7]	Decimal approximation of 22/7
N [22 / 7 , 50]	numerate to 50 decimal places
22 / 7 // N	postfix form of **N [22 / 7]**

Assignment

These assign values or expressions into variables.

n = 1
y = x² + 8x - 48
u = v = 0

The last example sets both u and v to zero.

Set Delayed

Normal Assignment =	
x = 2 **y = x²**	The expression x^2 is evaluated straight away. y will be set to 4.
x = 3 **y**	No mater what you do with x, y will still be 4.

Delayed Assignment :=	
x = 2 **y := x²**	The expression x^2 is NOT evaluated straight away. y will be defined as the expression x^2, NOT a number.
x = 3 **y**	y will be evaluated when it is used. So y will now evaluate to 9.

It is better not to set the independent variable and use the replacement operator instead.

Replacement Operator

The replacement operator is used to replace a variable with other variables or expressions.

x² /. x → 2

Expression	**Such that**	**Rule**
x²	**/.**	**x → 2** x goes to 2

To get the right arrow, use a minus sign, then greater than (- >).

You can have several rules in a list.

Mass = Density × Volume /. { Density → ρ ,Volume → 4/3 π r³}

Suppressing Output

```
x = 42;
```

A semicolon at the end of an assignment stops the output cell being created.

Clearing Variables

Remove any variable definitions..

```
Clear [ x ]
Clear [ x, y, z ]
x =.
```

History

%	The last output
%%	The last but one output
%%%	The last but two output
%24	The result of the 24th output
Out[24]	The same as above.
In[24]	The 24th input.

You can easily see which input and output are which as it is shown on the far left of each cell.

Examples

N[%]	Numerate the last output
%% + %3	Add the last but one output to the 3rd output
Out[2]^2	Square the 2nd output

Lists

A list in Mathematica is a collection of objects that can be used to form a vector, a matrix or a general array. To enter a list you enclose the list within curly braces and separate each item with a comma.

```
x = {10, 11, 12, 13, 14, 15}
```

Two dimensional arrays and matrices are created by entering lists of lists.

```
A = { { 1, 2 } , { 3, 4 } }
```

Three dimensional arrays are list of lists of lists, etc.

Built in Functions

Mathematica has hundreds of in built functions. Here are a few examples.

```
Sin [ x ]
Add [ a, b ]
Eigenvalues [ A ]
```

In general, Mathematica in built functions have the following properties

- The name of the function is the **full English word**.
- The name begins with a **capital letter**.
- Arguments are enclosed in **square brackets** and **separated with commas**.

Postfix Functions

An alternative way of using a function.

Stander function call

function name[expression]

N[%]
Expand [$(1 + x)^2$]
Simplify [$a(1 + a^{-1})$]

The same function calls using postfix calls.

expression // function name

%	**// N**
$(1 + x)^2$	**// Expand**
$a(1 + a^{-1})$	**//Simplify**

Help

Mathematica Documentation Center

Help►Documentation Center from the main Mathematica menu bar.
This is the Mathematica documentation viewer. Instead of going directly to the Documentation Center, I would recommend using the Function Navigator and Virtual Book to help you to find what you are looking for in the documentation.

Function Navigator

Help►Function Navigator.
You can either search for a function, or you can navigate through the various categories. This will then find the full documentation of your function in the Documentation Center. This will have links to other related functions and tutorials.

The Virtual Book

Help►Virtual Book.
This looks very similar to the Function Navigator, except that the virtual book is used to find tutorials. This is the best route into the documentation for beginners.

Find Selected Function

Help►Find Selected Function
Select a function name in your notebook and select **Help►Find Selected Function** on the menu bar. It is worth while remembering the short cut key, **F1** on windows computers.

Video Tutorials

1. **Help►Welcome Screen** from the menu bar.
2. At the bottom of the welcome screen are four icons.
   Click on the icon, **Find Your Learning Path**.
3. This opens the Wolfram learning centre in the web browser.
   Click on **Videos and Screencasts**.
4. Select the **Getting Started with Mathematica** channel on the right.

Palettes

There are numerous characters and symbols that can be inserted into your expression to make it look more like conventional mathematical text. For beginners, the easiest way to obtain these objects is to use one of the palettes. There are many different palettes. Some are specialised for a particular task, such as the Colour Schemes and Special Characters palettes. Others are more general.

Basic Math Assistant Palette
From the main Mathematica menu bar, select **Palette ► Basic Math Assistant.**
I suggest that you start with is the **Basic Math Assistant** palette. To see what a button does, leave the pointer stationary for a while over the top of the button. The Basic Commands section has a selection of tabs for different categories of commands. You can see what each tab is for by placing the pointer over the tab.

Using the palette is quite simple, you just click on the item that you want to insert into your notebook.

Place Holders
Place holders are squares on the palette buttons that will be replaced with an expression in your notebook. Take for example the superscript button ■□.

The filled in square ■ indicates the selection place holder. If you select an expression in your notebook and hit a button on a palette, the selected expression is moved into the selection place holder.

The Tab key is used to alternate between each of the place holders.

Special Characters
Palette ► Special Characters
Addition characters and symbols to those in the typesetting section.

Two Dimensional Expression Input

Entering expressions using the palette can get tedious. Before long you will start looking for short cuts. This is what Wolfram call two dimensional input. For example, *x* squared can be entered in its one dimensional form. $x \wedge 2$ Alternatively, if you hold down the control key instead of the shift key when entering the character ^, you get the two dimensional form. x^2

	Keys	Example	Key sequence
Move to Left	→		
Move to Right	←		
Next item	**Tab**		
Previous Item	**Shift Tab**		
Move to End	**Ctrl Space**		
Higher sub-expression	**Ctrl .**		
Power	**Ctrl ^**	x^2	**x** | **Ctrl ^** | **2**
Subscript	**Ctrl -**	x_k	**x** | **Ctrl -** | **k**
Square root	**Ctrl 2**	$\sqrt{x}$	**Ctrl 2** | **x**
Fraction	**Ctrl /**	$\frac{x}{y}$	**x** | **Ctrl /** | **y**
Switch Position	**Ctrl %**	$\sqrt[n]{x}$	**Ctrl 2** | **x** | **Ctrl %** | **n**
		x_n^2	**x** | **Ctrl ^** | **2** | **Ctrl %** | **n**

There are many more short cuts. To find more just place the pointer over a button on a pallet. This will show you the equivalent key sequence to the button press.

Special Numbers, Characters and Symbols

For numbers like pi, you can use a function to return the value or you can use the Greek character π. You can easily find π on one of the palettes, or you can enter the escape sequence for π into your notebook. To get the character π, press the escape key, then the letter p, then press the escape key again.

Number	Function Name	Symbol Used	Escape Sequence
e	E	e	Esc ee Esc
i	I	i	Esc ii Esc
j	I	j	Esc jj Esc
∞	Infinity	∞	Esc inf Esc
π	Pi	π	Esc p Esc

Mathematica uses a special font for *i* and *e* to stop them being confused with normal single letter variables.

Finding the codes for other characters.

The are hundreds of symbols that you can use. From the menu bar, select **Palette►Special Characters**. Move the pointer over a character and leave it there. You will see a yellow box containing the symbol, the long name and an escape sequence.

Ways of entering the Greek character alpha.	
\[Alpha]	The long name
Esc a Esc	Escape sequence
Esc \alpha Esc	TeX code

Most of the Greek characters use the first letter of the name for an escape sequence. Unfortunately, this cannot be done for all the letters, so some use the first two letters. For example, to insert theta, you use the escape sequence Esc th Esc.

Pictogram key sequences

You can also use pictograms.

Esc* <==> *Esc
Esc* :> *Esc
Esc* :-) *Esc

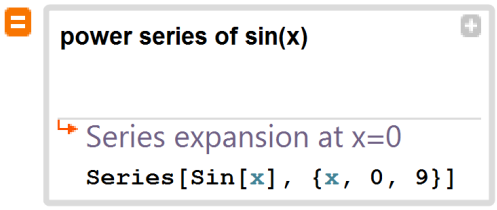
Free Format Linguistic Input

This allows you to use Mathematica even if you don't know the correct syntax.
You can enter plain or abbreviated English. Mathematica will then try to interpret what you have entered.
For example, if you enter the following :-

= Power Series of sin(x)

Then you will get

In[10]:=



Out[10]=

$$x - \frac{x^3}{6} + \frac{x^5}{120} - \frac{x^7}{5040} + \frac{x^9}{362\,880} + O(x^{10})$$

The lower half of the box contains the equivalent command in Mathematica syntax. Just click on this box to replace the input with the Mathematica command.

Hit the plus sign in the right hand corner of the box to select alternative interpretations of the command entered.

Free form linguistic commands use the Wolfram Alpha Web site. http://www.wolframalpha.com/
You are not limited to mathematics.

To enter free form linguistics, you begin the line with an equals sign.

= At start of line	Free form Linguistic Input
= = At start of line	Full Page from Wolfram Alpha
Ctrl =	Conversion into Mathematica Input

Examples

= GDP of UK.

= Date of the Battle of Hastings

= = Wonders of the ancient world Two equals for full page

***σ* = Ctrl= the conductivity of copper.** hold down Ctrl and press =

Operating on Lists

Normal operators (+-*/^) operate on each element by corresponding element.

A = { a1, a2, a3 }	
B = { b1 ,b2, b3 }	
A + B	$\{a1+b1, a2+b2, a3+b3\}$
A - B	$\{a1-b1, a2-b2, a3-b3\}$
A * B	$\{a1\times b1, a2\times b2, a3\times b3\}$
A / B	$\left\{\frac{a1}{b1}, \frac{a2}{b2}, \frac{a3}{b3}\right\}$
A^B	$\{a1^{b1}, a2^{b2}, a3^{b3}\}$

We can also carry out vector operations.

A . B	$a1\,b1+a2\,b2+a3\,b3$	Scalar or dot product
A × B	$\{a2\,b3-a3\,b2, a3\,b1-a1\,b3, a1\,b2-a2\,b1\}$	Vector or cross product

The cross used for the cross product is NOT the same as the cross used for multiplication.

Multiplication	Times	**Esc * Esc**
Cross product	Cross	**Esc cross Esc**

Operations on Matrices

2D lists. Normal operators (+-*/^) operate on element by corresponding element.

A . B	Matrix Multiplication

Indexing Vectors

x = {10,11,12,13,14,15}		Create a list.
First [x]	► **10**	The first item in the list.
Last [x]	► **15**	The last item in the list.
Part [x, 3]	► **12**	The 3rd item of x.
x [[3]]	► **12**	The 3rd item of x. Same as above.
x [[{3, 4, 5}]]	► **{12,13,14}**	The 3rd ,4th and 5th items of x.

Try the escape sequence ***x*** **Esc [[Esc 3 Esc]] Esc** for a neater look.

Indexing Matrices

M = {{1,2,3},{4,5,6},{7,8,9}}		Create a list of lists.
M [[2, 3]]	► **6**	2nd row, 3rd column
M [[{1, 2},{2, 3}]]	► $\begin{pmatrix} 2 & 3 \\ 5 & 6 \end{pmatrix}$	1st and 2nd row 2nd and 3rd column

2D Entry of Lists

There are of course many functions that operate on matrices. To get an idea of what is available, open the **Basic Math Assistant** palette, click on the **Matrix Commands** tab (4th from right) in the **Basic Commands** section.

Also under this tab, you will find the buttons to enter a matrix in 2D form. You can also construct a 2D matrix using short cut keys, which is not too difficult, if you take note of the following points.

- Enclose the matrix in round brackets.
- Use **Ctrl ,** to insert a new column. Hold down the control key and press the comma.
- Use **Ctrl Enter** to insert a new row. Hold down the control key and press Enter.
- To create a matrix, insert a row or column.
- Use the **Tab** key to cycle round unassigned elements.
- Use **Ctrl Space** to leave the matrix before inserting the closing bracket.

Generating Lists

Range produces a vector containing a specified range of numbers.

Range [n]	From 1 to n	Range[5]	{1,2,3,4,5}
Range [m, n]	From m to n	Range[5,10]	{5,6,7,8,9,10}
Range [m, n, s]	From m to n, spacing s	Range[5,11,2]	{6,7,9,11}

The function T**able** applies an expression to a range of inputs.

Table [n! , {n, 5}]	► {1 ,2, 6, 24, 120}
Table [x^2, {x, 0, 4}]	► {0, 1, 4, 9, 16}
Table [x^2, {x, {1, 8, 12} }]	► {1, 64, 144}
Table [{ x, x^2}, {x, 0, 4}]	► {{0, 0}, {1, 1}, {2, 4}, {3, 9}, {4, 16}}

The last example will display the output in a matrix. The function **TableForm** can be used to display to output in tabular form.

Table [{ x, x^2 }, {x, 0, 4}] //TableForm

Grid is another function that can be used with tables. The example below puts a frame around each box.

Grid [Table[{x, Sin[x] }, {x, 0, π, π/3 }], Frame → All]

Algebraic Manipulate

Simplify Tries to convert the expression into a simpler form.

$$\text{Simplify}\left[\frac{x^3+6x^2+11x+6}{x^2+4x+3}\right] \quad \blacktriangleright \quad x+2$$

Limited to pain algebraic manipulation, so it cannot simplify the expression below.

FullSimplify also tries procedures using various functions.

$$\text{FullSimplify}\left[\frac{e^x+e^{-x}}{2}\right] \quad \blacktriangleright \quad \text{Cosh}[x]$$

Assumptions Sometime simplification can be made easier by making assumptions. Simplification of the expression below will not work without the assumption that ***x*** is either positive or negative.

$$\text{Simplify}\left[\sqrt{x^2}, x>0\right] \quad \blacktriangleright \quad x$$

Assumptions can also be used with FullSimplify.

Expand Expand out the brackets.

$$\text{Expand}[(x+1)(x+2)] \quad \blacktriangleright \quad x^2+3x+2$$

Factor Factorise the equation.

$$\text{Factor}\left[x^2+3x+2\right] \quad \blacktriangleright \quad (x+2)(x+1)$$

Apart Partial Fraction Expansion.

$$\text{Apart}\left[\frac{x}{x^2+3x+2}\right] \quad \blacktriangleright \quad \frac{2}{x+2}-\frac{1}{x+1}$$

Together Put over a common denominator.

$$\text{Together}\left[\frac{2}{x+2}-\frac{1}{x+1}\right] \quad \blacktriangleright \quad \frac{x}{(x+1)(x+2)}$$

Cancel Cancel common factors in Numerator and Denominator.

$$\text{Cancel}\left[\frac{x^2-1}{x-1}\right] \quad \blacktriangleright \quad x+1$$

There are many, many other functions for algebraic manipulation.
TrigExpand and **TrigFactor** are specifically for trigonometric functions.
TrigToExp converts trigonometric functions to there exponential equivalent.
ExpToTrig the reverse of the above.
ExpandNumerator, **ExpandDenominator**, **FunctionExpand** and **PowerExpand.**
Then there is the more general function **ExpandAll**.

Then there are functions for extracting parts of an expression.

Exponent[y,x] Largest exponent of x in y

$$\text{Exponent}\left[x^3+6x^2+11x+6, x\right] \quad \blacktriangleright \quad 3$$

Part[y,n] The n^{th} term of y.

$$\text{Part}\left[x^3+6x^2+11x+6, 1\right] \quad \blacktriangleright \quad 6$$

$$\text{Part}\left[x^3+6x^2+11x+6, 2\right] \quad \blacktriangleright \quad 11x$$

$$\text{Part}\left[x^3+6x^2+11x+6, 3\right] \quad \blacktriangleright \quad 6x^2$$

Coefficient[y,x] The coefficient of x in y

$$\text{Coefficient}\left[x^3+6x^2+11x+6, x\right] \quad \blacktriangleright \quad 11$$

$$\text{Coefficient}\left[x^3+6x^2+11x+6, x^2\right] \quad \blacktriangleright \quad 6$$

CoefficientList[y,x] Put all the coefficients of x into a list.

$$\text{CoefficientList}\left[x^3+6x^2+11x+6, x\right] \quad \blacktriangleright \quad \{6, 11, 6, 1\}$$

Numerator[y] The Numerator of y

$$\text{Numerator}\left[\frac{x^3+6x^2+11x+6}{x^2+4x+3}\right] \quad \blacktriangleright \quad x^3+6x^2+11x+6$$

Denominator[y] The Denominator of y

$$\text{Denominator}\left[\frac{x^3+6x^2+11x+6}{x^2+4x+3}\right] \quad \blacktriangleright \quad x^2+4x+3$$

Solvers

Here is an example of how you use the Solve function to find the solutions to a cubic equation.

R = Solve [x^3 - 6x^2 +11x - 6 == 0, x]

The first argument is the equation to be solved. Notice that double equal are used to show that this is a comparison and not an assignment. The second argument is the variable we want to find, in this case *x*.

The result will be a list of rules. In this case, the list contains the three solutions of the cubic equation.

{ {x → 1}, { x → 2}, { x → 3} }

The rules can then be used with the replacement operator. Either individually

x1 = x /. First [R] ► **1**

x2 = x /. R [[2]] ► **2**

x3 = x /. Last [R] ► **3**

or all together into a list.

xall = x /. R ► **{ 1, 2, 3}**

Solve can also solve sets of equations.

R = Solve [x^2 + y^2 == 2 && x - y == 0, { x, y }]

Each solution in R is a list of two rules. A rules for *x* and another rule for *y*.

{ {x → -1, y → -1}, { x → 1, y → 1} }

You can also include inequalities, you can restrict the above to just the positive solution.

R = Solve [x^2 + y^2 == 2 && x - y == 0 && x>0 , { x, y }]

Sometime complex solutions are not appropriate. In the example below the solutions domain has been restricted to real numbers by specifying the domain in a third argument.

Solve [x^3 == 8, x, Reals]

Solve always tries to solve the equations algebraically. This is not always possible, as some equations can only be solved using numerical methods. Then you need to use the numerical version of solve.

NSolve [Cos[x] == x, x, Reals]

For this particular problem, the restriction to real values is essential.

Calculus

Differentiation

Function	Alternative	Meaning
D [f, x]	∂_x **f**	$\frac{d f}{d x}$
D [f, {x, 2}]	$\partial_{x,x}$ **f**	$\frac{d^2 f}{d^2 x}$
D [f, {x, 3}]	$\partial_{x,x,x}$ **f**	$\frac{d^3 f}{d x^3}$

The same function is used for ordinary and partial differentiation.

Function	Alternative	Meaning
D [f, x]	∂_x **f**	$\frac{\partial f}{\partial x}$
D [f, {x, 2}]	$\partial_{x,x}$ **f**	$\frac{\partial^2 f}{\partial x^2}$
D [f, x, y]	$\partial_{x,y}$ **f**	$\frac{\partial^2 f}{\partial x \partial y}$
D [f, {x, 2}, y]	$\partial_{x,x,y}$ **f**	$\frac{\partial^3 f}{\partial x^2 \partial y}$

Integrations

Indefinite integrals

Integrate[f, x] $\int f\, dx$

Definite integrals

Integrate[f, { x, 0, 2}] $\int_0^2 f\, dx$

Multiple integrals

Integrate[f, x, y] $\int\int f\, dx\, dy$

Integrate[f, { x, 0, 2}, { y, 0, 3}] $\int_0^3 \int_0^2 f\, dx\, dy$

Sometime you need to make assumptions to make an integration work.

Assuming [a > 0, Integrate [1/x^2 , {x, a, ∞ }]]

Solving Differential Equations

The function DSolve is used to solve differential equations.

DSolve [y''[x] + 2y'[x]+2y[x] == 0, y[x], x]

The first argument is the equation to be solved. The second argument is function we want to find. The third argument is the independent variable.

The first argument can be a list of equations. For example, we could include the initial conditions.

DSolve [{ y''[x] + 2y'[x]+2y[x] == 0, y[0] == 0, y'[0] == 1}, y[x], x]

If you cannot solve the equation symbolically, you can try a numerical solution.

NDSolve[{y''[x] + 2y'[x] + 2y[x] == 0, y[0]==0, y'[0]==1}, y[x],{x, 0, 2π}]

Notice that the third argument now has to include a range over which the numerical solution will be solved.

Summation

Enter this directly using the palette.

$$\sum_{n=1}^{\infty} x^n$$

or use the Sum function.

Sum [x^n, {n, 1, ∞ }]

The example below sums odd numbers, from one to ***2x+1***.

Sum [n, {n, 1, 2x - 1, 2 }]

The extra two at the end of the list sets the step size to two.
You can also just enter the numbers you want.

Sum [n, { n, {1, 3, 5, 7, 9} }]

Limits

Unfortunately, there is no way to enter the limits such as

$$\lim_{x \to 0} \frac{\sin(x)}{x}$$

using the palette. Instead, you use the Limit function.

Limit [Sin [x] / x, x → 0]

Graphs

Plot

Plot a graph of an expression over a specified range.

```
Plot [ x^2, {x, - 1.2, 1.2 } ]
```

Here the expression, x^2 is plotted from $x = -1.2$ to $x = 1.2$.

You can plot several lines at once by using a list of expressions.

```
Plot [ {x^2, x^3, x^4}, {x, - 1.2, 1.2 } ]
```

There are also logarithmic version of plot.

	X	Y
Plot	Linear	Linear
LogPlot	Linear	Log
LogLinearPlot	Log	Linear
LogLogPlot	Log	Log

Parametric Plot

A parametric plot uses two expressions. One expression is used for the X coordinate, the other expression for the Y coordinate. For example.

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

ListPlot

This is used to plot individual points onto a graph. The argument is a list of x,y coordinates.

```
ListPlot [ { {x1, y1}, {x2, y2}, {x3, y3}, {x4, y4} } ]
```

The **Table** function is useful for generation lists of coordinates.

```
Tz = Table [ { x , x^2 } , { x , -4 , 4 , 1 } ]
ListPlot [ Tz ]
```

ListLinePlot takes the same arguments as ListPlot and plots lines between points instead of individual points. There are also logarithmic versions **ListLogPlot**, **ListLogLinearPlot** and **ListLogLogPlot**.

Plotting Options

Plot options allow you to customize the appearance of a graph to your own requirements. For example, to add a grid to your graph, you use the option GridLines → Automatic as shown below.

```
Plot[ Cos[x], {x , - Pi, Pi },  GridLines → Automatic]
```

To produce the arrow, you enter a minus sign, then a greater than sign (- >). This is automatically converted into the right facing arrow.

You can put in as many options as you like, but it can get a bit messy putting all the options on one line. If you use the enter key (not the enter key on the key pad) you can insert new lines into your command that can make it more readable.

```
Plot [ {Cos[x], Sin[x], Sin[x]-Cos[x] },{ x, - Pi, Pi },
  Gridlines → Automatic,
  PlotStyle →  { Red, {Green,Thick }, { Blue, Dashed, Thin} } ]
```

The cosine is plotted in red, the sine with a thick green line, the difference between the two is plotted with a thin, blue dashed line.

Here are some examples of the options you can use.

PlotStyle →	Set colour, line width, etc.	(Notes 1 & 2)
GridLines → Automatic	Add a grid to the graph.	
PlotLabel → Title	Add text above the graph.	
Frame → True	Add a frame.	
FrameLabel → {Bottom,Side}	Add Labels .	
AxesLabel → {X axis, Y axis}	Add labels to the Axes.	
PlotRange → Full	Include full range of data.	(Note 3)
PlotRange → {MinY, MaxY}	Explicitly set the Y limits.	
Background → Colour	Set the background colour of the graph.	(Note 1)

The above list is a small example of the different options available. For more information see **Visualization and Graphics►Graphics Options and Styling** in the Function Navigator.

(Note 1) For the colour specification you can use named colours like **LightBrown**, or a colour function such as **RGBColor[0.9 , 1 , 0.9]**. To see the functions and names available, search for **colors** in the Function Navigator.

(Note 2) Search for **GraphicsDirectives** in the Function Navigator. Look for key words like **Thick**,**Thin**, **Dashed** and **Dotted** etc.

(Note 3) By default, outlying points are excluded from the plot. Try **Plot[Exp[-x],{x,0,20}]**. Then add the option **PlotRange → Full**.

Style

The Style function allows you change the appearance of various objects. Supposing that I want to emphasise a particular expression. I could increase the font size to 32 points and use a bold font.

```
Style [ B = Sqrt [ 1/(1+ω^2n)], 32, Bold]
```

In the next example, the string "Hello World" is displayed in 64 point, red italics, using the "Impact" font.

```
Style [ Hello World , 64, Red, Italic, FontFamily → "Impact" ]
```

Style is useful for configuring the appearance of objects within a graph.

```
PlotLabel → Style [Hello World, 36, Blue],
AxesLabel→ {Style [Input, 14, Red], Style [ Value, 14, Blue]}
```

See **Font Options** and **Color** in the Function Navigator.

Combining Multiple Graphs

The Show function allows you to plot graphs of different types on top of each other.
For example, you could combine a Parametric and List plot.

```
Show [ ParametricPlot [ {Cos[ϴ], Sin[ϴ]} , {ϴ, 0, 2 Pi} ],
ListPlot [ { {  −1/2 , −√3/2  },{  −1/2 , √3/2  } } ,
PlotMarkers → Style [ "×", Red, 16] ] ]
```

It is perhaps easier to see what is going on if we assign each of the plots to variables and combine them later.

```
G1 = ParametricPlot [ { Cos[θ], Sin[θ] },{θ, 0, 2 Pi}];

G2 = ListPlot [ { {  −1/2 , −√3/2  },{  −1/2 , √3/2  } } ,
   PlotMarkers → Style[ "×", Red, 16] ];

Show[G1,G2]
```

Note that the assignments of G1 and G2 are terminated with a semicolon. This stops the graphs from being displayed. They are only displayed when they are combined in Show.

Graphs can also be displayed in an array of graphics objects.

GraphicsRow[{G1,G2,G3,G4}]	4 graphs in a row.
GraphicsColumn[{G1,G2,G3,G4}]	4 graphs in a column
GraphicsGrid[{{G1,G2},{G3,G4}}]	4 graphs in a two by two grid

Defining your own functions

Defining your own functions is fairly straight forward.

f [x_] := x^2

The function name is **f**.		It is recommended that you use function names that start with a lower case letter, so that they don't get confused with names of built in functions.
x_	Is a pattern.	The underscore (_, called blank in Mathematica) means put anything you like here. The **x** defines a name for this particular anything.
:=	Set delay assignment	The set delay assignment is essential for a function.

When using a function, the pattern is replaced by an expression. In other words, you fill in the blanks.

f [2]	**4**
f [a]	a^2
f [a+b]	$(a+b)^2$

It is possible to define functions that have several blanks.

r [x_ , y_] := Sqrt [x^2 + y^2]

Evaluate

y = x^2
f [x_] := Evaluate [y]

Without the **Evaluate**, **f** would be defined as ***y***. The Evaluate forces ***y*** to be evaluated to x^2, before it is used to define the function.

Piecewise Functions

A piecewise definition is used when different expressions are required for different pieces of the input range.

$$y[x_] := \begin{cases} x^2 & x \geq 0 \\ -x & x < 0 \end{cases}$$

You can enter the above using a template in the **Typesetting** section of the **Basic Math Assistant** palette, or you can use the **Piecewise** function.

y [x_] := Piecewise[{ { x^2, x >= 0 }, { -x, x <0 } }]

Compound Expression

A compound expression is a sequence of expressions. The value of the whole sequence is the same as the value of the last expression in the sequence.

```
c = x = a; y = b; x + y
```

If you are using a compound expression as part of a function definition, enclose the compound expression in round brackets, else only the first expression is used to define the function.

```
c [ a_, b_ ] := ( x = a ; y = b ; x + y)
```

Module

Anything being stored in *x* will be overwritten every time the function is used.

You can overcome this problem by placing the expression into a module.

```
c[ a_, b_ ]:= Module[ {x,y}, x = a; y = b; x + y ]
```

The first argument to module is a list of the variables that are to be regarded as local in the expression. The second argument is expression.

Split things up by putting in extra new lines can make your code look clearer.
Using the enter key which is not on the keypad.

```
c[ a_, b_ ]:= Module[ {x,y},
      x = a;
      y = b;
      x + y ]
```

Comments

Comments are enclosed between brackets with stars.

```
(*  This is a comment  *)
```

Flow Control

If Function

This is Mathematica's version of an if statement. The general form is

If [test, then, else]

For example

If [x > 0, x^2, - x^2]

Here the **test** is $x > 0$. If the test is true, **then** we return x^2, **else** we return $-x^2$.

Here is an example of if being used in a function definition.

mysinc [x_] := If [x == 0, 1, Sin [x] / x]

Do Function

The Do function is Mathematica's equivalent of a for loop. The example below prints out 1 to 5.

Do [Print [n], { n, 1, 5 }]

Print[n] is the code to be repeated. It could be a compound expression. The second argument defines what values the loop variable takes on each loop. Here ***n*** goes from 1 to 5 in steps of one.

While Function

While is similar to while in any other language. The general form is

While[test , loop]

Where loop is the code that is repeated and test is the condition for the loop to repeat.

The example below print out the digits, 1 to 5;

n = 1 ; While[n <= 5 , Print [n] ; n = n + 1]

n++ is equivalent to **n = n + 1**. So instead we could have put

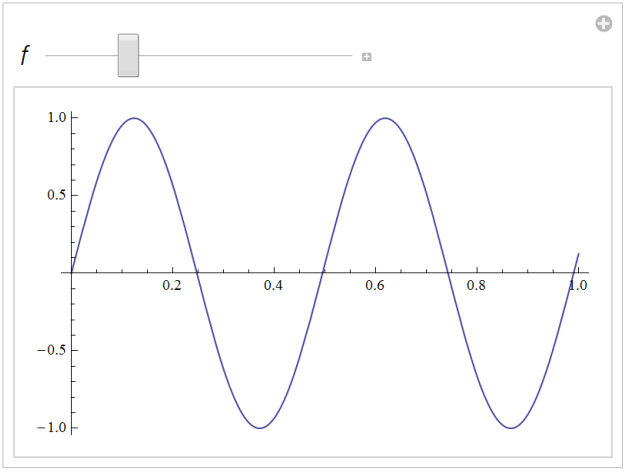
n = 1 ; While[n <= 5 , Print [n++]]

Interactive Graphs

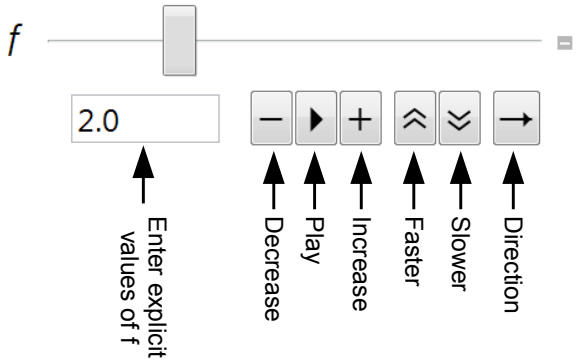
The Manipulate function allows you to change the value of a parameter and to animate your graphics. As an example, we can manipulate the frequency (variable *f*) of a sine wave.

Manipulate[Plot[Sin[2 π f t],{t,0,1}],{f,1,5}]

This produces a graph with a slider so that the value of ***f*** can be varied between 1 and 5. The first argument of Manipulate is a normal plot command. The second argument defines the range of *f*.



Clicking on the plus on the right of the slider opens up more controls.



The **increase and decrease buttons** allow you to incrementally increase or decrease the value of *f*.
The **play button** will start animation where the value of *f* is changed over time.
The **direction button** allows you to choose if the animation goes backwards, forwards or both.
The **faster and slower buttons** let you change the speed of the animation.

You can control as many parameters as you like. For example, you could add a slider to control the amplitude of a sin wave.

Manipulate[Plot[A Sin[2 π f t],{t,0,1}],{f,1,5},{A,1,5}]

Further Information

Licensing Mathematica in the Department
The Department pays for the Mathematica license. There is no charge to your contracts for the license. There are two ways in which your computer can be licensed to use Mathematica. If you are using a computer within the Department, then I suggest that you use the network license. This means that all license updates will be done for you. If you want to use Mathematica on your Laptop or your home computer, then you can use a standalone license.

Links
The Wolfram Learning Center has documents, videos, demos etc.
http://www.wolfram.com/support/learn/

The Mathematica Documentation Center
http://reference.wolfram.com

The Wolfram Library Archive
http://library.wolfram.com/

Wolfram Alpha Computational Knowledge Engine.
http://www.wolframalpha.com/

To add a Wolfram Alpha toolbar to your web browser
http://www.wolframalpha.com/downloads.html

Eric Weisstein MathWorld Mathematical resources,
http://mathworld.wolfram.com/

Mailing List
If you are interested in receiving information about Mathematica, then there is a Mathematica mail list within the Department. See

http://www.eng.ox.ac.uk/ecsg/services/software-mailing-lists