General Mathematica Conventions

**Shift-Enter** evaluates a cell

() are the only grouping symbol

[] are used for function arguments

{} are used for lists

A cell is not evaluated until you Shift-Enter that cell.

Mathematica commands begin with a capital letter for examples in (x) is entered as *Sin[x]*

Do not use E as a variable since it is used as the constant $e$.

If you start your variables and definitions with a lowercase letter then they will not conflict with Mathematica definitions.

% always stands for your last result.

If you enter $2 + 2$ and Shift-Enter the result will be 4. If you then enter % + 2 the result will be 6.

Sessions

When you start Mathematica a session is started. Any functions you define, variable assignments you make, or packages you load are kept for that session.

You should use the *Clear[]* command before you reuse variables.

Palettes

Choose File then Palettes

The *AlgebraicManipulation* and *BasicInput* are useful.

Functions

To enter $f(x) = x^2 + 3$ in Mathematica

$f[x_\_]: = x^2 + 3$

Notice the _ on the left hand side and the :=


Evaluate and /. 

*Evaluate[expr]* causes expr to be evaluated.

*expr /. rules* applies a rule or list of rules to transform each subpart of an expression expr.

$f[x_\_]: = x^4 + 4x^3 - 2x^2 - 12x$

sln = Solve[f'[x] == 0, x]

yields \{\{x -> -3\}, \{x -> -1\}, \{x -> 1\}\}

then $f[x] / . \text{sln}$ will give \{-9, 7, -9\}.

**Integration**

You can use the input palette to enter an integral as you would see it in a Calculus book or you can use

*Integrate[f, \{x, xmin, xmax\}]*

If you only need an approximation you can use

*NIntegrate[f, \{x, xmin, xmax\}]*

Lists

Lists are enclosed in {}.

Let $a = \{1, 3, 5\}$, then $a[[2]] = 3$.

*Flatten[ ]* un-nests one level.

*Join[a, b]* puts list a and list b together.

*Union[a, b]* puts list a and list b together, removes duplicates, and sorts the resulting list.

Matrices

A matrix is represented as a list of lists.

$
\begin{pmatrix}
    a & b \\
    c & d
\end{pmatrix}
$

is entered as \{\{a, b\}, \{c, d\}\}

The command *MatrixForm* formats a matrix in the usual form.

*IdentityMatrix[n]* creates the $n \times n$ Identity Matrix

*RowReduce* *RowReduce[m]* gives the rowreduced form of the matrix m.

Matrix Multiply

*a.b.c* gives products of vectors, matrices and tensors.

$a.b.c$ must have appropriate dimensions.

Eigenvalue

*Eigenvalues[m]* gives a list of the eigenvalues of the square matrix m.

**Eigensystem**

*Eigensystem[m]* gives a list \{values, vectors\} of the eigenvalues and eigenvectors of the square matrix m.

**MatrixPower**

*MatrixPower[mat, n]* gives the n-th matrix power of mat.

Solve and NSolve

*Solve[eqns, vars] attempts to solve an equation or set of equations for the variables vars.*

*NSolve*[\[1 + x^2 == 0, y^2 == x^2, z == x\], \{x, y, z\}]

Notice the double equal signs.

If you don’t need exact solutions to a polynomial you can use

*NSolve*[lhs==rhs, var] which gives a list of numerical approximations to the roots of a polynomial equation.

*FindRoot*[lhs==rhs, \{x, x0\} searches for a numerical solution to the equation lhs==rhs, starting with x=x0.

Simplify

*Simplify[expr]* performs a sequence of algebraic transformations on expr, and returns the simplest form it finds.

Packages

Some commands require the loading of a package before they can be used.

<<package

If you forget to load the package first you can enter

*Remove["Global`*"]* (where ` is to the left of 1 and on the same key as the tilde) or save your work, exit Mathematica, and then open a new session.

For example you must enter

<< Graphics`ImplicitPlot`

before using

*ImplicitPlot*[x^2 + 2 y^2 == 3, \{x, -2, 2\}]*
Plotting
The basic 2-dimension plot command is
\[
\text{Plot}[f[x],\{x,-3,3\}]
\]
To plot two or more functions
\[
\text{Plot}\{\{f[x],g[x]\},\{x,-3,3\}\}
\]
To specify the y axis from -2 to 5 use
\[
\text{Plot}[f[x],\{x,-3,3\},\text{PlotRange}\rightarrow \{-2,5\}]
\]
Three dimensional plotting is done by
\[
\text{Plot3D}[f[x,y],\{x,-3,4\},\{y,0,10\}]
\]
Parametric plots are made by
\[
\text{ParametricPlot}\{\{f[t],g[t]\},\{t,0,4\}\}
\]
\[
\text{ParametricPlot3D}[\{f[t,u],g[t,u],f[t,u]\},\{t,0,4\},\{u,0,10\},\text{PlotPoints}\rightarrow \{3,20\}]
\]
Color in plots
\[
\text{Hue}[\ ] \text{ as the argument runs from 0 to 1, Hue}[\ ] \text{ runs through red, yellow, green, cyan, blue, magenta, and back to red again.}
\]
\[
\text{Plot}\{\{f[x],g[x]\},\{x,-3,3\},\text{PlotStyle}\rightarrow \{\text{Hue}[.3],\text{Hue}[.6]\}\}
\]
plots \( f(x) \) in green and \( g(x) \) in blue.
You can also use \text{RGBColor}[1,0,0] in place of \text{Hue}[1]. \text{RGBColor}[1,1,1] is white and \text{RGBColor}[0,0,0] is black.
\[
\text{ColorFunction}\rightarrow \text{Hue} \text{ colors 3D plots.}
\]
When using a large number of \text{PlotPoints} you may want to set
\[
\text{Mesh}\rightarrow \text{False}
\]
Combining Plots
You can name plots with an assignment
\[
a=\text{Plot}[f[x],\{x,-3,3\}]
b=\text{Plot}[g[x],\{x,-3,3\}]
\]
then \text{Show}[a,b] will combine the plots.
Display function
\[
\text{DisplayFunction} \text{ allows you to suppress the display of intermediate graphics.}
\]
\[
\text{plot1} = \text{Plot}[\text{Sin}[x],\{x,0,2\text{ Pi}\},\text{DisplayFunction}\rightarrow \text{Identity}];
\]
\[
\text{plot2} = \text{Plot}[\text{Sin}[2 x],\{x,0,2\text{ Pi}\},\text{DisplayFunction}\rightarrow \text{Identity}];
\]
\[
\text{Show}[\text{plot1},\text{plot2},\text{DisplayFunction}\rightarrow \$\text{DisplayFunction}];
\]
Will display one graph.

Axis Labels
\[
\text{AxesLabel} \rightarrow \text{label} \text{ specifies a label for the a axis of a two dimensional plot, and the z axis of a three dimensional plot.}
\]
\[
\text{AxesLabel} \rightarrow \{\text{xlabel},\text{ylabel},\ldots\} \text{ specifies labels for different axes.}
\]
Plot Points
\[
\text{PlotPoints}\rightarrow n \text{ specifies n points to be used in the plot. For a 3D plot this would specify n in both the x and y directions.}
\]
\[
\text{PlotPoints} \rightarrow \{nx,ny\} \text{ specifies that nx points should be used in the x direction and ny points in the y direction.}
\]
Filled Plot
\[
\text{FilledPlot}[\text{Sin}[x],\{x,0,2\text{ Pi}\}]
\]

ListPlot
\[
\text{ListPlot}\{\{y_1,y_2,\ldots\}\} \text{ plots a list of values. The x coordinates for each point are taken to be 1,2,\ldots.}
\]
\[
\text{ListPlot}\{\{x_1,y_1\},\{x_2,y_2\},\ldots\}\text{ plots a list of values with specified x and y coordinates.}
\]

ListJoined
\[
\text{ListPlot}\{\{y_1,y_2,\ldots\},\text{ListJoined}\rightarrow \text{True}\} \text{ connects the points.}
\]
ContourPlot
\[
\text{ContourPlot}[f,\{x,\text{xmin},\text{xmax}\},\{y,\text{ymin},\text{ymax}\}] \text{ generates a contour plot of f as a function of x and y.}
\]

Differential Equations

\text{DSolve}
\[
\text{DSolve}[y'[x]==2a\ x,\ y[x],\ x] \text{ yields}
\]
\[
\{(y[x] \rightarrow a x^2 + C[1])\} \text{ To add initial conditions:}
\]
\[
\text{DSolve}\{\{y'[x]==2a\ x,\ y[0]==2\},\ y[x],\ x\} \text{ yields}
\]
\[
\{(y[x] \rightarrow 2 + a x^2)\}
\]

\text{NDSolve}
\[
\text{NDSolve}\{y'[x]==1/(2y[x]),y[.01]==.1\},\ y,\{x,.01,1\}\} \text{ numerically solves the BVP for } x \in (0.01,1) \text{ the result is an interpolating function.}
\]
If you want to graph the interpolating function you can use the following commands
\[
\text{soln}=\text{NDSolve}\{y'[x]==1/(2y[x]),y[.01]==.1\},\ y,\{x,.01,1\}\};
\]
\[
\text{Plot}[\text{Evaluate}[y[x]/.\text{soln}],\{x,.01,1\}];
\]
soln= names the interpolating function. \text{Evaluate} causes the function to be evaluated and \(/.\) replaces \( y \) with the interpolating function.

Loops
\[
\text{For}[i, i<5, i++, \text{Print}[i]] \text{ prints 1, 2, 3, 4. You can place commands separated by ;}
\]
\[
\text{Do}[\text{expr}, i, \text{imin}, \text{imax}, \text{step}]
\]
Table
\[
\text{Table}[\text{Prime}[i],\{i,1,5\}] \text{ yields \{2,3,5,7,11\} while}
\]
\[
\text{Table}[\text{Prime}[i],\{i,1,5,2\}] \text{ yields \{2,5,11\}}
\]
Important Note
Save often. Mathematica can crash and you will lose all unsaved work.