Mississippi State University Mathematica Guide

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 example sin(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 lower-case 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 :=
Then entering `f[2]` will yield `7`.

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`
`soln = Solve[f'[x] == 0, x]`
yields `{{x->-3},{x->-1},{x->1}}`
then `f[x] /. soln`
will give `{-9,7,-9}`.

N and //

`N[ ]` forces a numeric approximation of the argument.
`N[`π`]` is equivalent to π `// N`.
`N[`π`,30]` will give 30 digits of π.

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

Row Reduce `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.
`Solve[`$\{1 + x^2 == 0, y^2 == x^2, z == x\}, \{x, y, z\}$`]`
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
`Plot[f[x],{x,-3,3}]`
To plot two or more functions
`Plot[{f[x],g[x]},{x,-3,3}]`
To specify the y axis from -2 to 5 use
`Plot[f[x],{x,-3,3},PlotRange->{-2,5}]`
Three dimensional plotting is done by
`Plot3D[f[x,y],{x,-3,4},{y,0,10}]`
Parametric plots are made by
`ParametricPlot[{`$f_x[t], f_y[t]$`},{t,0,4}]`
`ParametricPlot3D[{`$f_x[t,u], f_y[t,u], f_z[t,u]$`}`
`,{t,0,4},{u,0,10},PlotPoints ->{3,20}]`

Color in plots

`Hue[ ]` as the argument runs from 0 to 1, `Hue[ ]` runs through red, yellow, green, cyan, blue, magenta, and back to red again.
`Plot[{f[x],g[x]},{x,-3,3},`
`PlotStyle->{Hue[.3],Hue[.6]}]`
plots $f(x)$ in green and $g(x)$ in blue.
You can also use `RGBColor[1,0,0]` in place of `Hue[1]`.
`RGBColor[1,1,1]` is white and `RGBColor[0,0,0]` is black.
`ColorFunction->Hue` colors 3D plots.
When using a large number of PlotPoints you may want to set
`Mesh->False`

Combining Plots

You can name plots with an assignment
`a=Plot[f[x],{x,-3,3}]`
`b=Plot[g[x],{x,-3,3}]`
then `Show[a,b]` will combine the plots.

Display function

`DisplayFunction` allows you to suppress the display of intermediate graphics.

```
plot1 = Plot[Sin[x], {x, 0, 2 Pi},
DisplayFunction->Identity];
plot2 = Plot[Sin[2 x], {x, 0, 2 Pi},
DisplayFunction->Identity];
Show[plot1, plot2,
DisplayFunction->$DisplayFunction];
```

Will display one graph.

Axis Labels

`AxesLabel -> label` specifies a label for the a axis of a twodimensional plot, and the z axis of a threedimensional plot.
`AxesLabel -> {xlabel, ylabel, ... }` specifies labels for different axes.

Plot Points

`PlotPoints-> n` specifies n points to be used in the plot. For a 3D plot this would specify n in both the x and y directions.
`PlotPoints ->{nx,ny}` specifies that nx points should be used in the x direction and ny points in the y direction.

Filled Plot

```
<< Graphics`FilledPlot`
FilledPlot[Sin[x], {x, 0, 2 Pi}]
```

ListPlot

`ListPlot[{y1, y2, ... }]` plots a list of values. The x coordinates for each point are taken to be 1, 2,. . . .
`ListPlot[{{x1,y1},{x2,y2},...}]` plots a list of values with specified x and y coordinates.

PlotJoined

`ListPlot[{y1, y2, ... },PlotJoined->True]` connects the points.

ContourPlot

`ContourPlot[f, {x, xmin, xmax},`
`{y, ymin, ymax}]` generates a contour plot of f as a function of x and y.

Differential Equations

DSolve

`DSolve[y'[x] == 2 a x, y[x], x]` yields
`{{y[x] -> a x^2 + C[1]}}` To add initial conditions:
`DSolve[{y'[x]==2 a x,y[0]==2}, y[x], x]` yields
`{{y[x] ->2 + a x^2}}`

NDSolve

`NDSolve[{y'[x]==1/(2y[x]),y[.01]==.1}`
`,y,{x,.01,1}]` numerically solves the BVP for $x \in (0.01, 1)$ the result is an interpolating function.
If you want to graph the interpolating function you can use the following commands

```
soln=
NDSolve[{y'[x]==1/(2y[x],y[.01]==.1},
y,{x,.01,1}];
Plot[Evaluate[y[x] /.  soln], {x, .01, 1}]
```

`soln=` names the interpolating function, `Evaluate` causes the function to be evaluated and `/.` replaces y with the interpolating function.

Loops

`For[i,i<5,i++, Print[i]]` prints 1, 2, 3, 4. You can place commands separated by `;`
`Do[expr ,i,imin,imax,step]`

Table

`Table[Prime[i],{i,1,5}]` yeilds
`{2,3,5,7,11}` while
`Table[Prime[i],{i,1,5,2}]` yeilds
`{2,5,11}`

Important Note

Save often. Mathematica can crash and you will lose all unsaved work.