Mathematica

Mathematica is a general system for doing symbolic and numeric mathematics—including root finding, integration, differentiation, matrix algebra, plotting, fitting, ... Note: pay close attention to capitalization as **Mathematica commands are case dependent!**

Starting Mathematica: On a UNIX workstation, open a terminal and at the tcsh (`%`) prompt type:

`math` ... For those using the command-line form.

`mathematica` ... For those using notebooks.

Alternatively, *Mathematica* can be started from the Mandrake yellow star "start" menu: star→CSBSJU Menu→ Physics→ math (or mathematica).

Exiting Mathematica: At the *Mathematica* (`In[`*n*`]:=`) prompt type:

`Quit` ... Note capitalization.
Control-D also quits.
Control-C aborts.

Input and Output:

With notebooks all input and output appears in a browser. You can print selected portions by selecting from a menu. If you run the command-line version, using an editor, cut and paste from an open file.

`In[1]:= <<`*file*`.m` ... *Mathematica* will execute all the commands in *file*`.m` (note: `.m` is the suggested extension).

`In[2]:= Import["`*file.dat*`","Table"]` ... *Mathematica* will create a list with the data from the *file* (note: `.dat` is the suggested extension). Also see `Export`.

`In[3]:= !`*csh command* ... *Mathematica* will execute the csh command (e.g., `ls, kwrite`).

`In[4]:= ?Fi*` ... Help for terms starting `Fi`, e.g., `Fit`.

`In[5]:= Options[Plot]` ... List options for commands, e.g., `AspectRatio -> Automatic`.

Examples:

`In[1]:= Solve[x^2 + b␣x + c == 0, x]` ... *Mathematica* knows the quadratic equation. A space is here denoted "␣" and means multiplication. You could just as well write "`b*x`", but "`bx`" is one variable's name, not the intended "$b \cdot x$".

```
                          2                              2
                 -b + Sqrt[b - 4 c]             -b - Sqrt[b - 4 c]
Out[1]= {{x -> ------------------- }, {x -> ------------------- }}
                       2                              2
```

`In[2]:= x^2 + b␣x + c /.  First[%]` ... Apply the first rule in the set.

```
                        2                            2       2
          b(-b + Sqrt[b - 4 c])    (-b + Sqrt[b - 4 c])
Out[2]=   --------------------- +  --------------------- + c
                     2                           4
```

`In[3]:= Together[%]` ... seek a common denominator, also see `Simplify`

`Out[3]= 0` ... it works!

In *Mathematica*, `%` always stands for the last result. You can type `%%` to use the next-to-last result or `%n` to use the result `Out[n]`.

`In[4]:= Integrate[x^2 Exp[x],x]`	... $\int x^2 e^x \, dx$
`Out[4]= ...`	... $e^x(2 - 2x + x^2)$
`In[5]:= D[%,x]`	... Take the derivative of the previous result.
`Out[5]= ...`	... `Simplify` to get $x^2 e^x$
`In[6]:= N[ Pi, 50 ]`	... 50 accurate digits of π.
`In[7]:= FindRoot[Tanh[y]==1/(2/y-1),{y,.9}]`	... Finds a solution near $y = .9$
`In[8]:= Series[Cos[x],{x,0,6}]`	... Taylor's expansion near $x = 0$ up to x^6
`In[9]:= f[x_ ]:=Re[Exp[I x]]`	... Define the function: $f(x) = \cos(x)$ the hard way
`In[10]:= m={{a,b},{c,d}}`	... Define matrix $m = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$

Mathematica knows lots of matrix operations including: `Det[m], Inverse[m], Eigenvalues[m], Eigenvectors[m], Eigensystem[m], m.n, m+n, ...`

Graphics:

Mathematica can produce both screen and hardcopy plots.

`In[1]:= Plot[Sin[2 Pi x],{x,0,2}]`	... A graph of $\sin(2\pi x)$ appears on your screen.
`In[2]:= PSPrint[%]`	... Prints a copy on the Physics laserprinter.
`In[3]:= Export["file.eps",%%,"EPS"]`	... Saves a file of graphic.

You may want to try some fancy color graphics like:

```
In[4]:= Plot3D[Sin[x + Sin[y]], {x,-6,3}, {y,-9,9}, Lighting->True, Mesh->False,
              PlotRange->All, PlotPoints->90]
```

Including Mathematica Packages:

For example, to load the **Graphics`Animation`** package, at the *Mathematica* prompt type:

```
In[1]:= Needs["Graphics`Animation`"]          ... Note capitalization and odd quote: `
In[2]:= <<Graphics/Polyhedra.m                ... Other ways of adding packages.
In[3]:= <<Graphics`Shapes`
```

```
In[4]:= theta = .3; irat = .3; phidot = 1; psidot = (irat - 1) Cos[theta]
In[5]:= ShowAnimation[Table[RotateShape[AffineShape[Polyhedron[Cube],{1,1,irat}],
        -psidot t,-theta,-phidot t],{t,0,4 Pi,.1 Pi}]]
```

More Information:

For more information about *Mathematica*, please refer to

- *The Mathematica Book, Fifth Edition*, by Stephen Wolfram, ISBN: 1579550223
- *Mathematica 4: Standard Add-on Packages*
- `/usr/local/mathematica_5.2/Documentation/English`