Just Enough Mathematica to Make you Dangerous

Joe St Sauver, Ph.D. (joe@oregon.uoregon.edu)

`% math`	*Use ssh to get to the % prompt*
`In[1]:= Exit` *or hit* `control-d`	*Leave Mathematica (when you're ready to!)*
`% math < sample.m > sample.lst` `% more sample.lst`	*Run Mathematica commands from sample.m (non-interactively) with output to sample.lst*

Using Mathematica like a calculator...

`In[2]:= 27.50-11.92` `Out[2]= 15.58`	*Mathematica as a good old calculator... hit ENTER (or shift-ENTER) after each command*
`In[3]:= 15!` `Out[3]= 1307674368000`	*Large values are no problem; you could even compute 1500 factorial if you wanted to*
`In[4]:= ?Log` `Log[z] gives the natural logarithm of z (logarithm to base e). Log[b, z] gives the logarithm to base b.` `In[5]:= Log[10,3453.8]` `Out[5]= 3.538`	*Need help with a function? Enter a ? followed by the name of a Mathematica function. Not sure of a function's name? You can use a* `*` *to see possible matches, e.g.,* `?L*` *Note that Mathematica functions are case sensitive and begin with a capital letter.*
`In[6]:= (4000/23)^3` `Out[6]=` 64000000000 / 12167 `In[7]:= %//N` `Out[7]= 5.26013 10^6`	*Operations done on whole numbers are always represented exactly when possible.* `%` *means "recall the last result" and* `//N` *means "provide an approximate numerical result"*
`In[8]:= Sin[60 Degree]` `Out[8]=` Sqrt[3] / 2	*Function args must be put in square brackets. Trig functions are in radians by default. Want a numeric value? Remember* `//N` *Inverse functions?* `ArcSin[ ]/Degree`
`In[9]:= Sum[i/(i^i),{i,1,\` `Infinity}]//N` `Out[9]=  1.62847`	*Numerically evaluate an infinite sum. You can continue long Mathematica commands lines with a* `\` *at the end of a line*
`In[10]:= BaseForm[223,2]` `Out[10]//BaseForm= 11011111`$_2$ `In[11]:= 16^^FAE7 + 16^^2C3E` `Out[11]= 75557` `In[12]:= BaseForm[%,16]` `Out12//BaseForm= 12725`$_{16}$	*Convert the value 223 (decimal) to base 2 (binary).* *Add FAE7 (hex) to 2C2E (hex); output by default is in decimal, but you can then force that output into hex, too, if you like.*

Algebra...

`In[1]:= Expand[(x+y)^2]` `Out[1]= x^2 + 2 x y + y^2` `In[2]:= Factor[%]` `Out[2]= (x + y)^2`	*Mathematica can expand an algebraic expression... or factor it back to a compact form.*
`In[3]:= Solve[x^2==81,x]` `Out[3]={{x -> -9}, {x -> 9}}`	*Find the roots of an equation; note use of == (rather than just =) in writing the equation.*
`In[4]:= Solve[x^2==-4,x]` `Out[4]= {{x -> -2I},{x -> 2I}}`	*Imaginary numbers? No problem...*
`In[5]:=Solve[{x+y==1,3x+y==2}]` `Out[5]= {{x -> -, y -> -}}` (1/2, 1/2)	*Mathematica can also solve systems of algebraic equations in multiple variables.*

Calculus...

`In[1]:= Limit[x/(Sqrt[x+1]-1),x->0]` `Out[1]= 2`	*Evaluate a limit*
`In[2]:= Dt[x^3+2x,x]` `Out[2]= 2 + 3 x^2`	*Compute a total derivative*
`In[3]:= D[(x^2)(y^3)+4y+x+2,x]` `Out[3]= 1 + 2 x y^3`	*Partial derivatives work the same way*
`In[4]:= D[x^3+2x,x,x]` `Out[4]= 6 x`	*Take the 2nd derivative with respect to x*
`In[5]:= Integrate[3x^2+2x,x]` `Out[5]= x^2 + x^3`	*Mathematica can also do integrals, just as you'd expect.*
`In[6]:= Integrate[E^x,{x,0,1}]` `Out[6]= -1 + E`	*Definite integral are also easy to evaluate.*
`In[7]:= <<Calculus`VectorAnalysis`` `In[8]:= SetCoordinates[\` `Cylindrical]` `Out[8]= Cylindrical[Rr,Ttheta,Zz]` `In[9]:= Integrate[Sqrt[1+4Rr^2]\` `Rr,{Rr,0,1},{Ttheta,0,2Pi}]//N` `Out[9]= 5.33041`	*Cartesian space is the default, but not our only option. For example, let's find the surface area of the parabola* $z=1+x^2+y^2$ *where* $x^2+y^2 <= 1$. *Because of the nature of that restriction, it is easier to work in cylindrical coordinates. We do so via the vector analysis package (note the backtick marks,* not apostrophes*, used when loading a package!). Package info is at* http://documents.wolfram.com/v4/index20.html

```
In[1]:= w={{a,b},{c,d}}
Out[1]= {{a, b}, {c, d}}
```
Create a 2x2 matrix (we're using symbols, but you could equally easily use numeric values)

```
In[2]:= w.{x,y}=={k1,k2}
Out[2]= {a x + b y, c x + d y} ==
{k1, k2}
```
Use a dot product to apply that matrix of coefficients to two variables to form a system of two equations with constants {k1, k2}

```
In[3]:= Transpose[w]//MatrixForm
Out[3]//MatrixForm= a   c
                    b   d
```
Mathematica can easily do most standard linear algebra operations, for example, we can easily transpose matrix w...

```
In[4]:=Inverse[{{1,-1},{2,2}}]
          1  1      1   1
Out[4]={{-, -},{-(-), -}}
          2  4      2   4
```
Or compute the inverse of a 2x2 numeric matrix...

```
In[5]:= Det[{{a,b,c},{d,e,f},\
{g,h,i}}]
Out[5]= -(c e g) + b f g + c d h
- a f h - b d i + a e i
```
Or compute the determinant of a 3x3 symbolic matrix...

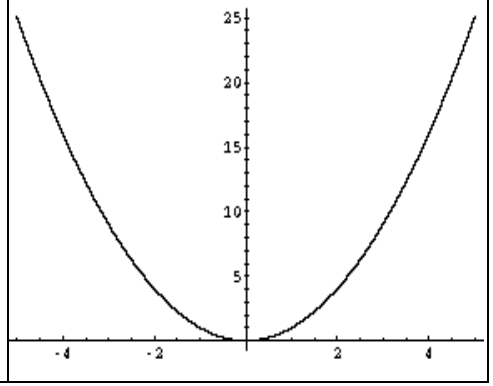
```
In[6]:= Table[If[EvenQ[i]||EvenQ[j]\
,1,0],{i,3},{j,3}]//MatrixForm
Out[6]= 0   1   0
        1   1   1
        0   1   0
```
In addition to entering matrices on an element by element basis, Mathematica will also let us construct matrices using rules, such as this example that sets elements of a 3x3 matrix to be 1 if the column or row is an even number.

Plotting in Mathematica...

```
In[1]:= Plot[x^2,{x,-5,5}]
Out[1]= -Graphics-
In[2]:= Display["a.gif",%,"GIF"]
Out[2]= -Graphics-
```

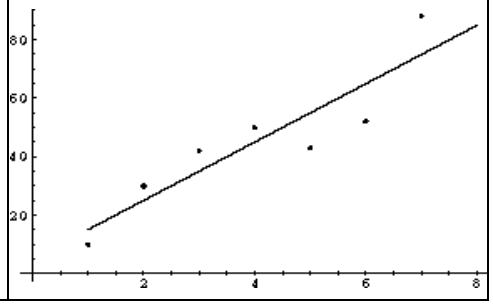
Note: besides GIF format, you can also use the Display function to save Mathematica graphics in PDF, EPS, PCL, PBM and other formats.

Plot a function over an interval. If connecting from a Unix workstation or an X terminal, your graph will be shown in a new window; we also show saving graphic output in gif format.




```
In[3]:=!!mydata.dat
4.1 10.7
[etc]
In[4]:= newvals=ReadList[ \
"mydata.dat",{Number,Number}]
Out[4]= {{4.1,10.7},[etc]}
In[5]:= plot1=ListPlot[newvals]
Out[5]= -Graphics-  [not shown]
In[6]:= Fit[newvals,{1,x},{x}]
Out[6]= 5.14286 + 9.96429 x
In[7]:= plot2=Plot[%,{x,1,8}]
Out[7]= -Graphics-  [not shown]
In[8]:= Show[plot1,plot2]
Out[8]= -Graphics-
In[9]:= Display["b.gif",%,"GIF"]
Out[9]= -Graphics-
```

Work with (x,y) data points from an external file. !!mydata.dat shows us the contents of the file. Read in pairs of numbers from that file, storing the list of values by the name newvals. Plot the dataset. Fit a line to the points & plot that. Finally, overlay both and save as a gif



Mathematica As A Programming Language...

```
(* Approach No. 1 *)
w=Join[Table[0,{7}],Table[5,{7}],\
Table[10,{4}],{25}];
<<DiscreteMath`Combinatorica`
x=Union[KSubsets[w,7]];
Select[x,(Plus@@##)<=45&]\
//TableForm
Print["\n ",Length[%]," soln's"]

(* Approach No. 2 *)
solns=0;
Do[If[((25i+10j+5k<=45)&&\
       (i+j+k<=7)),\
       solns++, Null],\
  {i,0,1},{j,0,5},{k,0,7}];
Print["\n",solns," soln's"]
```

If Mathematica doesn't have precisely what you need (or what it has is overkill), you can always use Mathematica as a programming language and write your own code. For example, assume you have a pile of 5, 10 and 25 pound weights. Using no more than 7 of them in any instance, how many combinations can you form that will total no more than 45 pounds?

We can solve that problem using Mathematica's Combinatorica package, or we can just write a little program to solve that problem directly by looping through a three way nested do loop, using an if statement to tally only solutions that meet the specified restriction.

Mathematica on other platforms...

UO has a site license for Mathematica covering its installation on University owned PC's, Macs, and Unix systems.	*For more information, please see http://darkwing.uoregon.edu/~hak/mathematica*

More Information About Mathematica...

The Mathematica Book, 4th Ed., by Stephen Wolfram [ISBN 0-521-64314-7, 1470 pages] is the definitive reference.	*See also* http://www.wolfram.com/ *and* http://documents.wolfram.com/ *for online copies of many Mathematica documents.*