This document will get you started using *Mathematica. Mathematica* is a general purpose CAS, useful for many things, but this document is geared towards linear algebra. Note that there is an extensive help facility built into *Mathematica*, complete with many examples. Commands are entered by pressing shift+enter on the keyboard (just hitting enter will give a new line which is still a continuation of the previous line). Putting a semi-colon at the end of the command will suppress the display of the output. Here are some simple examples.

```
In[1]:= 345 * 57

Out[1]= 19 665

In[2]:= 357^85

Out[2]= 9 479 782 420 272 124 378 632 471 940 724 719 266 375 462 756 417 372 120 723 437 543 034 732 760 935 432 \
  363 579 923 305 468 288 648 220 919 776 974 931 781 570 508 134 746 039 438 628 336 748 119 549 110 589 \
  138 040 211 126 186 217 619 674 031 882 621 081 662 747 182 350 519 014 457 557
```

The above is the traditional syntax. One can also use the 2D input: superscripts are entered by ctrl+^, and subscripts by ctrl+_ (use ctrl+space to leave the superscript/subscript position). We will use both forms in this document.

In[3]:= 357^{85}

```
Out[3]= 9 479 782 420 272 124 378 632 471 940 724 719 266 375 462 756 417 372 120 723 437 543 034 732 760 935 432 \
  363 579 923 305 468 288 648 220 919 776 974 931 781 570 508 134 746 039 438 628 336 748 119 549 110 589 \
  138 040 211 126 186 217 619 674 031 882 621 081 662 747 182 350 519 014 457 557
```

All functions acting on objects are entered using square brackets such as f[x,y]. *Mathematica* will keep exact precision until you tell it to numerically approximate. You use the N command to get a numerical approximation. We can also assign names to expressions so that we can re-use them.

```
In[4]:= a1 = Sin[2 / 5]
```

Out[4]= $\text{Sin}\left[\frac{2}{5}\right]$

```
In[5]:= N[a1, 20]

Out[5]= 0.38941834230865049167
```

In[6]:= a = 45^{31}

```
Out[6]= 1 776 592 919 961 297 100 543 276 866 939 850 151 538 848 876 953 125
```

In[7]:= b = 27^{15}

```
Out[7]= 2 954 312 706 550 833 698 643

In[8]:= N[Sqrt[a / b], 20]
```

Out[8]= $7.7547130537313422191 \times 10^{14}$

In[9]:= N[$b^{35/101}$, 20]

Out[9]= $2.7558602453425081800 \times 10^{7}$

A few more examples.

In[10]:=
```
c = a + b + 2
```
Out[10]= 1 776 592 919 961 297 100 543 276 866 942 804 464 245 399 710 651 770

In[13]:=
```
FactorInteger[c]
```
Out[13]= {{2, 1}, {5, 1}, {173, 1}, {1229, 1}, {67 631, 1}, {318 954 253 070 897, 1}, {38 736 085 833 022 146 668 766 383, 1}}

In[14]:= $p = x^3 - 127 * x^2 + 31 * x + 2$

Out[14]= $2 + 31\,x - 127\,x^2 + x^3$

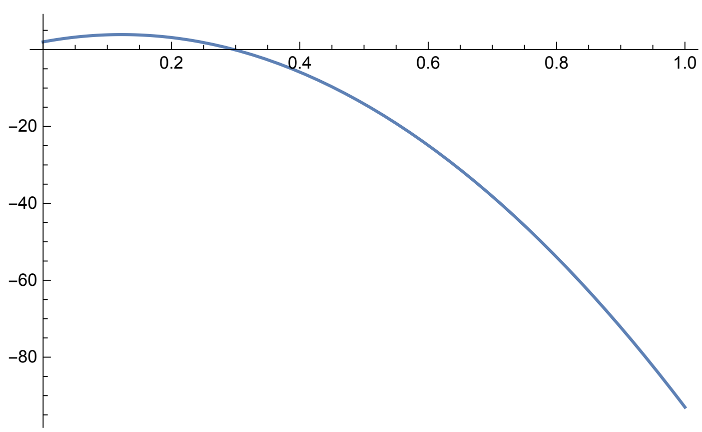
In[15]:=
```
p^3
```
Out[15]= $\left(2 + 31\,x - 127\,x^2 + x^3\right)^3$

In[16]:= Expand$[p^3]$

Out[16]= $8 + 372\,x + 4242\,x^2 - 17\,441\,x^3 - 268\,995\,x^4 + 1\,501\,356\,x^5 - 2\,071\,999\,x^6 + 48\,480\,x^7 - 381\,x^8 + x^9$

Before we get to linear algebra, we consider a few plotting examples.

In[17]:=
```
Plot[p, {x, 0, 1}]
```
Out[17]=

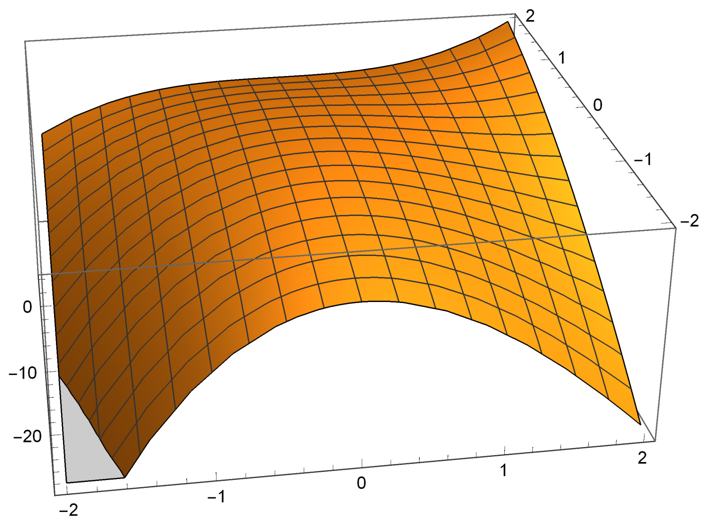


In[18]:= $f = x^3 - 2 * x^2 * (2 - y) - y^2$

Out[18]= $x^3 - 2\,x^2\,(2 - y) - y^2$

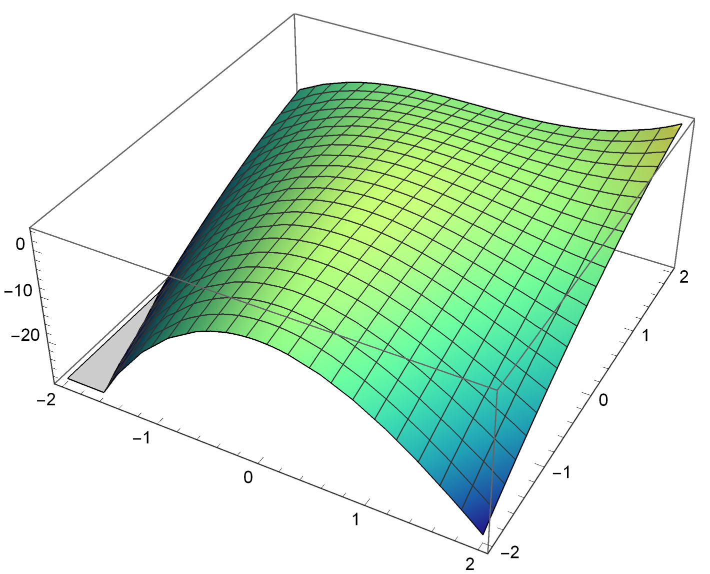
In[20]:=
```
Plot3D[f, {x, -2, 2}, {y, -2, 2}]
```

Out[20]=



In[25]:=
```
Plot3D[f, {x, -2, 2}, {y, -2, 2}, ColorFunction → "BlueGreenYellow", Mesh -> {20, 20}]
```

Out[25]=



Now we turn to linear algebra. We first create a few matrices.

In[26]:=
```
A = {{2, 5, 7}, {-3, 8, 11}, {5, 5, -6}}
```

Out[26]=
```
{{2, 5, 7}, {-3, 8, 11}, {5, 5, -6}}
```

Note that *Mathematica* does not display the matrix in 2D form by default. We must pipe the output to the MatrixForm command.

In[27]:= **A // MatrixForm**

Out[27]//MatrixForm=
$$\begin{pmatrix} 2 & 5 & 7 \\ -3 & 8 & 11 \\ 5 & 5 & -6 \end{pmatrix}$$

In[38]:= **B = {{1, 8, -1}, {0, 3, 7}, {-4, 13, 8}}**

Out[38]= {{1, 8, -1}, {0, 3, 7}, {-4, 13, 8}}

In[39]:= **B // MatrixForm**

Out[39]//MatrixForm=
$$\begin{pmatrix} 1 & 8 & -1 \\ 0 & 3 & 7 \\ -4 & 13 & 8 \end{pmatrix}$$

In[41]:= **A + B // MatrixForm**

Out[41]//MatrixForm=
$$\begin{pmatrix} 3 & 13 & 6 \\ -3 & 11 & 18 \\ 1 & 18 & 2 \end{pmatrix}$$

In[43]:= **A.B // MatrixForm**

Out[43]//MatrixForm=
$$\begin{pmatrix} -26 & 122 & 89 \\ -47 & 143 & 147 \\ 29 & -23 & -18 \end{pmatrix}$$

Here is the 5th power of the matrix A. Note that entering A^5 will take the 5th power of each element, not what you want.

In[46]:= **MatrixForm[MatrixPower[A, 5]]**

Out[46]//MatrixForm=
$$\begin{pmatrix} -11\,288 & 124\,055 & 117\,557 \\ -28\,073 & 124\,158 & 159\,461 \\ 50\,855 & 93\,555 & -64\,596 \end{pmatrix}$$

In[47]:= **A.B - B.A // MatrixForm**

Out[47]//MatrixForm=
$$\begin{pmatrix} 1 & 58 & -12 \\ -73 & 84 & 156 \\ 36 & -147 & -85 \end{pmatrix}$$

We can introduce a shortcut for the MatrixForm command if we like.

In[48]:= **MF = MatrixForm**

Out[48]= MatrixForm

In[49]:= **MF[A + B]**

Out[49]//MatrixForm=
$$\begin{pmatrix} 3 & 13 & 6 \\ -3 & 11 & 18 \\ 1 & 18 & 2 \end{pmatrix}$$

Here is a random 5x5 matrix with integer entries

In[57]:=
```
E1 = RandomInteger[{-10, 10}, {5, 5}]
```

Out[57]=
```
{{-4, 4, 3, 8, 4}, {9, 7, -5, 10, -7},
 {6, 6, 6, -3, 9}, {4, 9, -1, -6, -1}, {10, 7, -1, -2, -4}}
```

In[58]:=
```
E1 // MatrixForm
```

Out[58]//MatrixForm=

$$\begin{pmatrix} -4 & 4 & 3 & 8 & 4 \\ 9 & 7 & -5 & 10 & -7 \\ 6 & 6 & 6 & -3 & 9 \\ 4 & 9 & -1 & -6 & -1 \\ 10 & 7 & -1 & -2 & -4 \end{pmatrix}$$

In[66]:=
```
M = {{1, 2, 4, 3, -1}, {-1, -1, -3, -4, 3}, {-2, -4, -8, -5, 3}, {2, 5, 9, 5, 0}}
```

Out[66]=
```
{{1, 2, 4, 3, -1}, {-1, -1, -3, -4, 3}, {-2, -4, -8, -5, 3}, {2, 5, 9, 5, 0}}
```

In[67]:=
```
M // MF
```

Out[67]//MatrixForm=

$$\begin{pmatrix} 1 & 2 & 4 & 3 & -1 \\ -1 & -1 & -3 & -4 & 3 \\ -2 & -4 & -8 & -5 & 3 \\ 2 & 5 & 9 & 5 & 0 \end{pmatrix}$$

The command RowReduce will put the matrix into reduced row-echelon form.

In[68]:=
```
RowReduce[M] // MF
```

Out[68]//MatrixForm=

$$\begin{pmatrix} 1 & 0 & 2 & 0 & -10 \\ 0 & 1 & 1 & 0 & 3 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Here are a few more common matrix operations.

In[71]:=
```
Det[E1]
```

Out[71]=
```
-36 369
```

In[72]:=
```
A = {{1, 1, 3}, {0, -1, -1}, {0, 0, 2}}
```

Out[72]=
```
{{1, 1, 3}, {0, -1, -1}, {0, 0, 2}}
```

In[75]:=
```
B = {{0, 2, 1}, {1, -1, 0}, {0, 1, 1}}
```

Out[75]=
```
{{0, 2, 1}, {1, -1, 0}, {0, 1, 1}}
```

In[76]:=
```
Det[B]
```

Out[76]=
```
-1
```

```
In[79]:= U = B.A.Inverse[B]
Out[79]= {{-2, 0, 2}, {-1, 1, 5}, {-2, 0, 3}}
```

```
In[80]:= U // MF
```

Out[80]//MatrixForm=

$$\begin{pmatrix} -2 & 0 & 2 \\ -1 & 1 & 5 \\ -2 & 0 & 3 \end{pmatrix}$$

The next command will give the eigenvalues and eigenvectors of the matrix U.

```
In[81]:= Eigensystem[U]
Out[81]= {{2, -1, 1}, {{1, 9, 2}, {4, -3, 2}, {0, 1, 0}}}

In[82]:= Eigenvalues[U]
Out[82]= {2, -1, 1}

In[83]:= Eigenvectors[U]
Out[83]= {{1, 9, 2}, {4, -3, 2}, {0, 1, 0}}

In[85]:= cp = CharacteristicPolynomial[U, x]
```

Out[85]= $-2 + x + 2 x^2 - x^3$

```
In[86]:= Factor[cp]
```

Out[86]= $-(-2 + x)(-1 + x)(1 + x)$

Individual elementary row operations can be done by using the “Part” command to works with the individual rows.

```
In[87]:= M // MF
```

Out[87]//MatrixForm=

$$\begin{pmatrix} 1 & 2 & 4 & 3 & -1 \\ -1 & -1 & -3 & -4 & 3 \\ -2 & -4 & -8 & -5 & 3 \\ 2 & 5 & 9 & 5 & 0 \end{pmatrix}$$

We add 1 times the first row to the second row.

```
In[89]:= M2 = M ; M2[[2]] = M2[[1]] + M2[[2]]
Out[89]= {0, 1, 1, -1, 2}

In[90]:= M2 // MF
```

Out[90]//MatrixForm=

$$\begin{pmatrix} 1 & 2 & 4 & 3 & -1 \\ 0 & 1 & 1 & -1 & 2 \\ -2 & -4 & -8 & -5 & 3 \\ 2 & 5 & 9 & 5 & 0 \end{pmatrix}$$

We add 2 times the first row to the second row.

```
In[92]:= M3 = M2; M3[[3]] = 2 * M3[[1]] + M3[[3]]
Out[92]= {0, 0, 0, 1, 1}
```

In[93]:= `M3 // MF`

Out[93]//MatrixForm=

$$\begin{pmatrix} 1 & 2 & 4 & 3 & -1 \\ 0 & 1 & 1 & -1 & 2 \\ 0 & 0 & 0 & 1 & 1 \\ 2 & 5 & 9 & 5 & 0 \end{pmatrix}$$

We interchange the 1st and 2nd rows (for no particular reason).

In[94]:= `M4 = M3; M4[[{2, 1}]] = M4[[{1, 2}]]`

Out[94]= `{{1, 2, 4, 3, -1}, {0, 1, 1, -1, 2}}`

In[95]:= `M4 // MF`

Out[95]//MatrixForm=

$$\begin{pmatrix} 0 & 1 & 1 & -1 & 2 \\ 1 & 2 & 4 & 3 & -1 \\ 0 & 0 & 0 & 1 & 1 \\ 2 & 5 & 9 & 5 & 0 \end{pmatrix}$$

In[96]:= `M // MF`

Out[96]//MatrixForm=

$$\begin{pmatrix} 1 & 2 & 4 & 3 & -1 \\ -1 & -1 & -3 & -4 & 3 \\ -2 & -4 & -8 & -5 & 3 \\ 2 & 5 & 9 & 5 & 0 \end{pmatrix}$$

In[97]:= `K = Table[0, {i, 4}, {j, 8}]`

Out[97]= `{{0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0},`
`{0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}}`

In[98]:= `K // MF`

Out[98]//MatrixForm=

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

We use some commands that work with parts of a matrix.

In[104]:= `K[[1 ;; 4, 1 ;; 5]] = M`

Out[104]= `{{1, 2, 4, 3, -1}, {-1, -1, -3, -4, 3}, {-2, -4, -8, -5, 3}, {2, 5, 9, 5, 0}}`

In[105]:= `K // MF`

Out[105]//MatrixForm=

$$\begin{pmatrix} 1 & 2 & 4 & 3 & -1 & 0 & 0 & 0 \\ -1 & -1 & -3 & -4 & 3 & 0 & 0 & 0 \\ -2 & -4 & -8 & -5 & 3 & 0 & 0 & 0 \\ 2 & 5 & 9 & 5 & 0 & 0 & 0 & 0 \end{pmatrix}$$

In[106]:= `B = {{1, 2, 1}, {3, 4, 5}, {0, 6, 0}, {-2, 3, 7}}`

Out[106]= `{{1, 2, 1}, {3, 4, 5}, {0, 6, 0}, {-2, 3, 7}}`

In[107]:= `K[[1 ;; 4, 6 ;; 8]] = B`

Out[107]= `{{1, 2, 1}, {3, 4, 5}, {0, 6, 0}, {-2, 3, 7}}`

In[108]:=
```
K // MF
```

Out[108]//MatrixForm=

$$\begin{pmatrix} 1 & 2 & 4 & 3 & -1 & 1 & 2 & 1 \\ -1 & -1 & -3 & -4 & 3 & 3 & 4 & 5 \\ -2 & -4 & -8 & -5 & 3 & 0 & 6 & 0 \\ 2 & 5 & 9 & 5 & 0 & -2 & 3 & 7 \end{pmatrix}$$

In[109]:=
```
J = K[[2 ;; 3, 2 ;; 4]] // MF
```

Out[109]//MatrixForm=

$$\begin{pmatrix} -1 & -3 & -4 \\ -4 & -8 & -5 \end{pmatrix}$$