Using *Mathematica* for matrices

Matrices

Matrices are entered in "row form", such that

In[195]:=
```
aa = {{2, 1}, {-1, 2}}
```

Out[195]=
```
{{2, 1}, {-1, 2}}
```

gives the following matrix (the // and "MatrixForm" displays the result so it looks like a matrix)

In[196]:=
```
aa // MatrixForm
```

Out[196]//MatrixForm=

$$\begin{pmatrix} 2 & 1 \\ -1 & 2 \end{pmatrix}$$

Picking out components now requires two indices, which are in standard "row, column" order:

```
aa[[1, 2]]
```

1

In[197]:=
```
bb = {{3, 2}, {-1, -1}}; bb // MatrixForm
```

Out[197]//MatrixForm=

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

There are some canned matrices, in particular the identity
(the argument of IdentityMatrix giving the linear dimension):

```
id = IdentityMatrix[3]; id // MatrixForm
```

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Another predefined set of matrices are the Pauli matrices:

```
{PauliMatrix[1] // MatrixForm,
 PauliMatrix[2] // MatrixForm, PauliMatrix[3] // MatrixForm}
```

$$\left\{\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}\right\}$$

There's a special command to create a diagonal matrix:

```
DiagonalMatrix[{1, 2, 3}] // MatrixForm
```

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

Matrix multiplication is written with a Dot (and is not commutative, as we know)

In[199]:=
```
aa.bb // MatrixForm
bb.aa // MatrixForm
```

Out[199]//MatrixForm=

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

Out[200]//MatrixForm=

$$\begin{pmatrix} 4 & 7 \\ -1 & -3 \end{pmatrix}$$

Whereas a product simply multiplies the corresponding elements, one by one:

In[201]:=
```
aa bb // MatrixForm
```

Out[201]//MatrixForm=

$$\begin{pmatrix} 6 & 2 \\ 1 & -2 \end{pmatrix}$$

Addition and subtraction and multiplication by scalars work:

```
aa + bb // MatrixForm
aa - bb // MatrixForm
3 aa // MatrixForm
```

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

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

$$\begin{pmatrix} 6 & 3 \\ -3 & 6 \end{pmatrix}$$

Multiplication works with any shape matrices, as long as they are conformable. Here's a vector, which, although it's entered as a row-like vector:

```
v5 = {3, 1}
```

{3, 1}

is treated like a column vector under matrix multiply:

```
aa.v5
```

{7, -1}

It is displayed like a column.

```
aa.v5 // MatrixForm
```

$$\begin{pmatrix} 7 \\ -1 \end{pmatrix}$$

However, one can also multiply from the left, in which case the vector is treated as a row

```
v5.aa
```

{5, 5}

Transpose transposes:

```
Transpose[aa] // MatrixForm
```

$$\begin{pmatrix} 2 & -1 \\ 1 & 2 \end{pmatrix}$$

Conjugate conjugates, element by element:

```
cc = {{2 + I, 3 I}, {-3 I, 4}}; cc // MatrixForm
```

$$\begin{pmatrix} 2 + i & 3 i \\ -3 i & 4 \end{pmatrix}$$

```
Conjugate[cc] // MatrixForm
```

$$\begin{pmatrix} 2 - i & -3 i \\ 3 i & 4 \end{pmatrix}$$

To hermitian conjugate use the ConjugateTranspose[ ] function

```
ConjugateTranspose[cc] // MatrixForm
```

$$\begin{pmatrix} 2 - i & 3 i \\ -3 i & 4 \end{pmatrix}$$

or you can make a “dagger” which does the same thing by typing “escape ct escape”

```
cc† // MatrixForm
```

$$\begin{pmatrix} 2 - i & 3 i \\ -3 i & 4 \end{pmatrix}$$

Functions of Matrices

Recall the matrix aa:

```
aa // MatrixForm
```

$$\begin{pmatrix} 2 & 1 \\ -1 & 2 \end{pmatrix}$$

Trace

```
Tr[aa]
```

4

Determinant

```
Det[aa]
```

5

Inverse

```
aainv = Inverse[aa]; aainv // MatrixForm
```

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

Check that inverse “works”

```
aainv.aa // MatrixForm
aa.aainv // MatrixForm
```

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

There's no problem with moving to larger matrices, which would be painful by hand:

```
dd = {{1, 2, 3, 4, 5}, {2, 3, 7, 8, 9}, {-3, 0, 6, 4, 2},
  {6, 2, 4, 5, 1}, {-1, -2, 5, 2, 3}}; dd // MatrixForm
```

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

```
Inverse[dd] // MatrixForm
```

$$\begin{pmatrix} \frac{44}{35} & -\frac{4}{5} & -\frac{3}{35} & \frac{8}{35} & \frac{2}{7} \\ \frac{351}{35} & -\frac{31}{5} & \frac{23}{35} & \frac{32}{35} & \frac{8}{7} \\ \frac{621}{70} & -\frac{28}{5} & \frac{19}{35} & \frac{31}{35} & \frac{19}{14} \\ -\frac{179}{14} & 8 & -\frac{4}{7} & -\frac{8}{7} & -\frac{27}{14} \\ \frac{59}{70} & -\frac{2}{5} & -\frac{4}{35} & -\frac{1}{35} & \frac{3}{14} \end{pmatrix}$$

Rank---it does the row reduction for you:

```
MatrixRank[dd]
```

5

The rank is the same for the transpose, as it should be:

```
MatrixRank[Transpose[dd]]
```

5

Mathematica does row reduction for you. Technically this gives the "reduced row echelon form", with as many off-diagonal zeroes as possible.

```
RowReduce[dd] // MatrixForm
```

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

A simple example discussed in lecture notes:

```
ee = {{2, 2}, {1, 1}}; ee // MatrixForm
```

$$\begin{pmatrix} 2 & 2 \\ 1 & 1 \end{pmatrix}$$

```
MatrixRank[ee]
```

1

```
RowReduce[ee] // MatrixForm
```

$$\begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}$$

More complicated functions of matrices

Mathematica has a built in function for exponentiating a matrix

```
aa = {{1, 1}, {0, 2}}; MatrixExp[aa] // MatrixForm
```

$$\begin{pmatrix} e & -e + e^2 \\ 0 & e^2 \end{pmatrix}$$

Note that this is different from exponentiating in the usual way, which simply exponentiates each element.

```
E^aa // MatrixForm
```

$$\begin{pmatrix} e & e \\ 1 & e^2 \end{pmatrix}$$

There's also a function for taking powers of matrices (which works for all complex powers too)

```
MatrixPower[aa, 10] // MatrixForm
```

$$\begin{pmatrix} 1 & 1023 \\ 0 & 1024 \end{pmatrix}$$

```
MatrixPower[aa, -2] // MatrixForm
```

$$\begin{pmatrix} 1 & -\frac{3}{4} \\ 0 & \frac{1}{4} \end{pmatrix}$$

```
MatrixPower[aa, I]
```

$\{\{1, -1 + 2^{i}\}, \{0, 2^{i}\}\}$

Dimensions (order) of a matrix

In[1]:=
```
mat1 = {{5, 2, 2}, {3, 6, 3}, {6, 6, 9}}
```

Out[1]= {{5, 2, 2}, {3, 6, 3}, {6, 6, 9}}

In[2]:=
```
mat1 // MatrixForm
```

Out[2]//MatrixForm=

$$\begin{pmatrix} 5 & 2 & 2 \\ 3 & 6 & 3 \\ 6 & 6 & 9 \end{pmatrix}$$

In[3]:=
```
Dimensions[mat1]
```

Out[3]= {3, 3}

The command “ Dimensions “ - gives dimension or order of the matrix.

In[4]:=
```
Det[mat1]
```

Out[4]= 126

In[5]:=
```
{λ1, λ2, λ3} = Eigenvalues[mat1]
```

Out[5]= {14, 3, 3}

The command “ Eigenvalues “ - gives eigenvalues of the matrix.

In[6]:=
```
mat2 = Array[Min, {3, 3}]; mat2 // MatrixForm
```

Out[6]//MatrixForm=

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 2 \\ 1 & 2 & 3 \end{pmatrix}$$

In[7]:=
```
Eigenvalues[mat2]
```

Out[7]= {Root[-1 + 5 #1 - 6 #1^2 + #1^3 &, 3], Root[-1 + 5 #1 - 6 #1^2 + #1^3 &, 2], Root[-1 + 5 #1 - 6 #1^2 + #1^3 &, 1]}

The eigenvalues here are returned as Root objects, in this case the three roots of the characteristic polynomial -1 + 5 x - 6 x^2 + x^3. The option setting “Cubics → True” will permit the display of such roots in terms of radicals.

In[8]:=
```
Eigenvalues[mat2, Cubics → True]
```

Out[8]=

$$\left\{2+\frac{7^{2/3}}{\left(\frac{3}{2}\left(9+i\sqrt{3}\right)\right)^{1/3}}+\frac{\left(\frac{7}{2}\left(9+i\sqrt{3}\right)\right)^{1/3}}{3^{2/3}},\right.$$

$$2-\frac{7^{2/3}\left(1-i\sqrt{3}\right)}{2^{2/3}\left(3\left(9+i\sqrt{3}\right)\right)^{1/3}}-\frac{\left(1+i\sqrt{3}\right)\left(\frac{7}{2}\left(9+i\sqrt{3}\right)\right)^{1/3}}{2\times 3^{2/3}},$$

$$\left.2-\frac{7^{2/3}\left(1+i\sqrt{3}\right)}{2^{2/3}\left(3\left(9+i\sqrt{3}\right)\right)^{1/3}}-\frac{\left(1-i\sqrt{3}\right)\left(\frac{7}{2}\left(9+i\sqrt{3}\right)\right)^{1/3}}{2\times 3^{2/3}}\right\}$$

One may also get a numerical approximation of the eigenvalues as follows:

In[9]:=
```
Eigenvalues[mat2] // N
```
Out[9]=
```
{5.04892, 0.643104, 0.307979}
```

In[10]:=
```
{v1, v2, v3} = Eigenvectors[mat1]
```
Out[10]=
```
{{2, 3, 6}, {-1, 0, 1}, {-1, 1, 0}}
```

The command “ Eigenvectors “ - gives eigenvectors of the matrix.

In[11]:=
```
Eigensystem[mat1]
```
Out[11]=
```
{{14, 3, 3}, {{2, 3, 6}, {-1, 0, 1}, {-1, 1, 0}}}
```

The command “ Eigensystem “ - gives both the eigenvalues and the eigenvectors of the matrix. The output is a list whose first item is a list of eigenvalues and whose second item is a list of corresponding eigenvectors.

Solving Systems of Linear Equations

Suppose we want to solve a Nonhomogeneous system of linear equations in the form m x = b, where m is the coefficient matrix, x is a column vector of variables, and b is a column vector. When b is a vector with at least one nonzero entry, then system is called nonhomogeneous.

In[12]:=
```
Clear[m, x, x1, x2, x3, x4, b];
m = {{1, 5, -4, 1}, {3, 4, -1, 2}, {3, 2, 1, 5}, {0, -6, 7, 1}};
x = {{x1}, {x2}, {x3}, {x4}};
b = {{1}, {2}, {3}, {4}};
m.x == b
```
Out[16]=
```
{{x1 + 5 x2 - 4 x3 + x4}, {3 x1 + 4 x2 - x3 + 2 x4}, {3 x1 + 2 x2 + x3 + 5 x4}, {-6 x2 + 7 x3 + x4}} ==
 {{1}, {2}, {3}, {4}}
```

This can be interpreted as a list of four linear equations.

In[17]:=
```
Det[m]
```
Out[17]=
```
35
```

Since matrix m is nonsingular, the system has a unique solution.

In[18]:=
```
ArrayFlatten[{{m, b}}] // MatrixForm
```
Out[18]//MatrixForm=

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

The command “ArrayFlatten”- gives the augmented matrix. The command “RowReduce” is used to find the reduced row echelon form of the matrix. Finding the reduced row echelon form of the matrix is nothing but performing Gaussian Elimination.

In[19]:=
```
RowReduce[%] // MatrixForm
```

Out[19]//MatrixForm=

$$\begin{pmatrix} 1 & 0 & 0 & 0 & -\frac{127}{35} \\ 0 & 1 & 0 & 0 & \frac{141}{35} \\ 0 & 0 & 1 & 0 & \frac{139}{35} \\ 0 & 0 & 0 & 1 & \frac{13}{35} \end{pmatrix}$$

This gives $x_1 = \frac{-127}{35}$, $x_2 = \frac{141}{35}$, $x_3 = \frac{139}{35}$, $x_4 = \frac{13}{35}$.

The command “LinearSolve” provides a quick means for solving systems that have a single solution.

In[20]:=
```
LinearSolve[m, b]
```

Out[20]= $\left\{\left\{-\frac{127}{35}\right\}, \left\{\frac{141}{35}\right\}, \left\{\frac{139}{35}\right\}, \left\{\frac{13}{35}\right\}\right\}$

Notice that in a system having an infinite number of solutions, LinearSolve will return only one of them, giving no warning that there are others.

A homogeneous system of linear equations is of the form m x = 0. If m is nonsingular, a homogeneous system will have only the trivial solution x = 0, while if m is singular the system will have an infinite number of solutions. The set of all solutions to a homogeneous system is called the null space of m.

In[33]:=
```
Clear[m, x, b];
m = {{0, 2, 2, 4}, {1, 0, -1, -3}, {2, 3, 1, 1}, {-2, 1, 3, -2}};
x = {{x1}, {x2}, {x3}, {x4}};
b = {{0}, {0}, {0}, {0}};
Det[m]
```

Out[37]= 0

In[38]:=
```
RowReduce[ArrayFlatten[{{m, b}}]] // MatrixForm
```

Out[38]//MatrixForm=

$$\begin{pmatrix} 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

This tells us that $x_1 = x_3$, $x_2 = -x_3$ and $x_4 = 0$. That is any vector of the form (t, -t, t, 0), where t is a real number, is a solution, and the vector (1, -1, 1, 0) forms a basis for the solution space.

The command “NullSpace”- gives a set of basis vectors for the solution space of the homogeneous equation mx = 0.

In[39]:=
```
NullSpace[m]
```

Out[39]= {{1, -1, 1, 0}}

Nullity of the matrix m can be found using the command “Length”.

In[40]:=
```
Length[NullSpace[m]]
```

Out[40]= 1