FMCS1-part II
Practical 1. Matlab and vectors and matrices

Mark van Rossum

31st January 2005

1 Getting familiar with matlab

Go through the Matlab tutorial from Kermit Sigmon. Spend not more than hour on this now. Matlab has also an excellent online help. The web has also many matlab resources. You can also use octave, which is free, so you can run it on your home computer.

2 Applying Matlab

As example of matlab's capabilities, we will develop a graphical representation of a two-dimensional matrix transformation. Define a matrix of choice such as

`mat=[cos(phi), sin(phi); -sin(phi), cos(phi)];` Of course, ϕ needs to be defined in your code.

Next define a colum vector such as `v=[0 1]';`

Now check that `mat*v` calculates the matrix product.

2.1 A nice plot

To visualize the action of the matrix we let it operate on many vectors and plot the result. We first create a whole bunch of vectors (named 'data' here) that we want to transform.

```
xmin=-1; xmax=1;
ymin=-1; ymax=1;
dx=0.2;
data=[];
for x=xmin:dx:xmax
for y=ymin:dx:ymax
data=[data [x y]'];
end
end
```

Examine 'data'. What are its rows and columns?

To help the visualisation we split the data in the four quadrants. The points in the first quadrant can be found with:

```
q1=(find(data(1,:)>0 & data(2,:)>0));
```

Examine the 'find' function in the help system, and inspect the result for `q1.` Generate code to find the other three quandrants. To plot the data we can use:

```
plot(data(1,q1),data(2,q1),'*r');
```

This plots the list of the first arguments against the second one. Examine the 'plot' function in the help system. The final argument tells matlab to draw the data with red stars (rather than to connect the data point, which is the default). Plot each quadrant with a different colour. You can use 'hold on' to prevent erasing the old plot.

The data transformed by the matrix can easily be obtained by:

```
transdata=mat*data;
```

This transforms all data points at once. Check what `mat*dat(1,:)` and `mat*dat(:,1)` represent.

Plot the transformed data points using the above technique.

```
plot(transdata(1,q1),transdata(2,q1),'*r');
```

And similar for the other quadrants.

3 To hand in

(it is much faster to write and sketch by hand here).

Take to two 2-dimensional matrices of choice. Write down the matrices. For each matrix sketch how the transformed quadrants look.