AN INTRODUCTION TO USING MATLAB ON A UNIX SYSTEM

Section 1: Getting Started with Matlab

Note:

```
Unix prompt%
```

indicates that this is to be typed in an xterm Unix prompt.

```
>>
```

indicates that this is to be typed in the matlab (window) prompt.

Basic knowledge of Unix is assumed. You may want to work from a directory such as 12.006/ or chaos/

To start matlab type:

```
Unix prompt% matlab &
```

A matlab window should appear on your screen and a welcome message should be displayed. Type

```
>> help
```

This should display a list of online help items available.

```
matlab/general       -  General purpose commands.
matlab/ops           -  Operators and special characters.
matlab/lang          -  Language constructs and debugging.
matlab/elmat         -  Elementary matrices and matrix manipulation.
matlab/matfun        -  Matrix functions - numerical linear algebra.
matlab/datafun       -  Data analysis and Fourier transform functions.
matlab/polyfun       -  Polynomial and interpolation functions.
matlab/funfun        -  Function functions - nonlinear numerical methods.
matlab/sparfun       -  Sparse matrix functions.
matlab/plotxy        -  Two dimensional graphics.
matlab/plotxyz       -  Three dimensional graphics.
```

```
matlab/graphics       -  General purpose graphics functions.
matlab/color          -  Color control and lighting model functions.
--more--
```

For example, if you need help in plotting 2D graphs, type

```
>> help graph2d
```

Section 2: Problem Set1

Download the matlab function **iterate.m**. Make sure it is placed in the working directory where you use matlab. Here is an example session:

1) Create directory called chaos, 12.006J, or 18.353J, etc., under your home directory

```
Unix prompt% cd ~/
Unix prompt% mkdir chaos
Unix prompt% cd chaos
```

2) The last command let you work under your new directory **chaos**, start Matlab by:

```
Unix prompt% matlab &
```

3) You should see a separate window pops up, wait for a while, and Matlab will load. When you see the Matlab prompt, try

```
>> help iterate

  iterate              - iterate a function of an initial value xo
                         N times.

                         mu = function parameter
                         xo = initial value of x
                          N = number of iterations
                          i = running index
                          x = iterated values

  Usage: [i,x] = iterate(mu, xo, N);

>>
```

This explains how to use the matlab function **iterate.m**. You should also try to read **iterate.m** itself to familiarize yourself with writing simple functions in Matlab. For example, if you wish to iterate $x_{i+1} = 2x_i(x_i - 1)$ thirty times with initial value $x_o = 0.1$, type:

```
>> [i,x] = iterate(0.5, 0.1, 30);
>> plot(i,x,'-*');
```

4) The last command plots the value of x_i vs i. To save the figure in postscript format

```
>> print -dps myplot.ps
```

5) Finally, for printing the figure, go to your Unix prompt and type

```
Unix prompt% lpr -P<your_printer_name_here>          myplot.ps
```

Note: To do the problem set, you will only need to modify about 1 or 2 lines of code,