Exercises for Programming with Matlab

Obtaining the Examples

1. The course examples are available under the CIC6007 course on the online learning system, see matlab examples under the course content

The matlab examples zip file can also be downloaded from

http://rcg.group.shef.ac.uk/courses/cic6007 /matlab_examples.zip

Iceberg users may extract the examples from a tgz archive in the directory /usr/local/courses on iceberg.

On ShARC, use cd to change directory to a working directory of your choice then use the cp and tar commands to obtain and extract the examples.

The required commands are

cd ~/myworkingdirectory

tar –zxvf /usr/local/courses/matlab_examples.tgz

Practice Session 1

Getting Started

- Startup MATLAB
- Start up and experiment with the following
  - Editor
  - Demos
  - Help
  - Use the file/files File operations to navigate directories
  - Under the applications menu start up the variable browser (windows only).

Practice on the command window

- Enter the following lines into the command window and observe what happens:

```
a = 1.234
b= 5.6
```

```
c = a*b
D = [ a b c ]
E = [ c ; b; a]
F = D*E
G= E*del2
```

• Use the who and whos command to get information about the assigned variables note how the variables are stored and how much storage is used.

Exercises 1: Matrices

1.1 Define the following matrix in MATLAB:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 4 & 5 \\ 2 & 6 & 6 \end{bmatrix}$$

1.2 Define a row vector X which is set to the second row of matrix A

1.3 Define a column vector Y, which is set to the first column of A

1.4 Define a column vector Z, which is set to the first row of A

1.5 Define a column vector V, which contains the third column of A in reverse order

1.6 Find out X*Y and Y*X

```
A = [ 1 2 3 ; 3 4 5 ; 2 6 6 ] ;
X = A( 2, : )  ;
Y =  A( : , 1 )  ;
Z = A(1, : )';
V = A(3:-1:1 , 3)  ;
W1= Y*Z ;
W2 = Z*Y  ;
```

1.7 Find the determinant of A

Determine the solution of the following simultaneous equations

$x + 2y + 3z = 4$

$3x + 4y + 5z = 14$

$2x + 6y + 6z = 20$

```
% w = [%RHS here]
% A=[] %define matrixe of coefficients
% xyz=..... % compute solution using inverse of A

D = det(A)  ;
w = [4;14;20]  ;

xyz = inv(A)*w;
%or
% xyz = A\w
```

Exercises 2 : Functions Applied to Arrays and Matrices

2.1 Using the RAND function define a 16 by 16 matrix R of random numbers, elements of which range between 0 and 10.

2.2 Using the PASCAL function set B to the 5 by 5 Pascal triangle

2.3 Find the inverse of matrix B and store it in C

2.4 Using the SUM function find the sum of each column of matrix C

2.5 Define a new matrix D which is the subset of B and made up of rows 2 to 4 and columns 2 to 4 of the B matrix.

```
R= 10*rand (16 , 16 ) ;
B = pascal ( 5 ) ;
C = inv(B) ;
Sum(C) ;
D= C(2:4,2:4) ;
```

Exercises 3: Data Types

Store the following table -

(a) As a cell array

(b) As a structure

Alkali metal	Standard atomic weight (u)	Density (g·cm^{-3})	Colour Index
Lithium	6.941	0.534	[0.3 0.4 0.3]
Sodium	22.990	0.968	[0.6 0.8 0.7]
Potassium	39.098	0.89	[0.2 0.3 0.2]

```
%alkali metals as a cell array
alkalimetals=cell(3,4);

%alkali metals as a structure
% s = struct(field1,value1,...,fieldN,valueN)
am(1)=struct('name','Lithium','aw',6.941,'density',0.534,'cindex',[0.3 0.4 0.3]);
am(2)=struct('name','Sodium','aw',22.990,'density',0.968,'cindex',[0.6 0.8 0.7]);
```

Exercise 4 : Writing MATLAB Functions

Sin(x) can be represented by the series;

Sin(x) = x – x3/3! + x5/5! – x7/7! + …

- Write a MATLAB function named mysine that will calculate sin(x) to the power 11.
- Improve your function by putting checks for the range of x and n supplied by the user.
- Modify the function so that the calculations are carried out to any user specified number of terms *(n).*

```
%you may write the following in a separate function file called mysine

function y = mysin(x)
    % MYSIN : Calculates the sine of an angle expressed
    %           in radians.
    sum = x ;
    nexterm = x ;
    for i =3:2:11
      nexterm = -nexterm.*x.^2./(i*(i-1));
      sum = sum+nexterm;
    end
    y = sum;
end
```

```
%improved function
function y = mysin(x)
% MYSIN : Calculates the sine of an angle expressed
%          in radians using 5 terms of the expansion.
%         Range must be between +- pi.
```

```
%
if x < pi & x > -pi
  sum = x ;
  nexterm = x ;
  for i =3:2:11
    nexterm = -nexterm.*x.^2./(i*(i-1));
    sum = sum+nexterm;
  end
else
   sum = nan;

  disp(['Outside the permited range of ' , num2str(-pi),...
        ' to ' , num2str(pi) ] );
end
y = sum;
```

```
%compute to any number of terms
function y = mysin(x,nseries)
% MYSIN : Calculates the sine of an angle expressed
%          in radians.
%          Range must be between +- pi.
% Second parameter is optional which determines the number of
% terms in the series. Default is 5.
if exist('nseries','var' )
   nn = nseries*2 + 1;
else
   nn = 11;
end

if x < pi & x > -pi
  sum = x ;
  nexterm = x ;
  for i =3:2:nn
    nexterm = -nexterm.*x.^2./(i*(i-1));
    sum = sum+nexterm;
  end
else
   sum = nan;
   disp(['Outside the permited range of ' , num2str(-pi),...
        ' to ' , num2str(pi) ] );
end
y = sum;
```

Exercise 5: Data Import

Import the data gasprices.csv in the exercise folder into a table.

```
G = readtable('gasprices.csv', 'Delimiter', ',', 'Headerlines', 4);
```

Extract the data for Japan, and compute mean and std.

```
JP = G.Japan;
JP_mean = mean(JP);
JP_std = std(JP);
```

Extract the data for Europe, and compute the mean European price in each year.

```
Europe = [G.France, G.Germany, G.Italy, G.UK];
annualEuroMeans = mean(Europe, 2);
```

Compute the return series for Europe.

```
Euro_Returns = log( Europe(2:end, :) ./ Europe(1:end-1, :) );
```

Compute and visualise correlation.

```
C = corrcoef(Euro_Returns);
figure
imagesc(C, [-1, 1])
set(gca, 'XTick', 1:4, 'XTickLabel', {'France', 'Germany', 'Italy', 'UK'}, ...
    'YTick', 1:4, 'YTickLabel', {'France', 'Germany', 'Italy', 'UK'})
colorbar
```

Exercises 6 : Importing Data and Plotting in MATLAB

In the exercise directory you will find a data file named *field.dat* .

6.1 Read it into MATLAB and store columns 1 to 3 of the data as; temperature, pressure and density respectively

```
%Use Import Data from the home bar  or  Start -> Import Wizard .  This should read all the dat
%pres = FIELD(:,2) ;  dens = FIELD(:,3);
field=readtable('field.dat');
[frows,fcols]=size(field)
temp=field.Var1
pres=field.Var2
dens=field.Var3
```

6.2 Using the PLOT command plot temperature variation

6.3 Using the PLOT command plot pressure variation

6.4 By making use of the HOLD command, plot both temperature and pressure in the same graph.

```
plot(temp) ;
plot(pres) ;
hold on ; plot(temp , 'g' ) ;
```

```
Warning: MATLAB has disabled some advanced graphics rendering features by switching to software
OpenGL. For more information, click here.
```

6.5 Fit a 5th degree polynomial to the data, display the equation on the graph and save the fitting parameters into the workspace via the Tools > Basic Fitting menu of the Figure window

```
%for help use
%e.g. use [p,S] = polyfit(x,y,n)
help polyfit
[p,S] = polyfit(1:frows,temp,5)
```

6.6 Compute the max, min, mean and median values of the plot variables

6.7 Plot the mean value of y as a line on the current plot (Hint: contract two end points of a line using values from the basic statistics and use them in a call to line command but don't forget to use the hold command first)

6.8 Using the edit-plot icon from the figure window toolbar edit this graph to change the colour of the mean-value line and the legend for it.

```
% eg. use
field=temp;
y = [mean(field),median(field),max(xfield),min(field)]
```

6.9 Using the SUBPLOT command. Plot three separate plots of all three variables on the same screen. Experiment with editing other aspects of the plot.

```
subplot(2,2,1); plot(temp)
subplot(2,2,2); plot(dens)
subplot(2,2,3); plot(pres)
```