Labwork 1

I. Matlab introduction

1. Vector

Exercise 1:

Define a row vector x, one row, five column with value = (1, 2, 3, 4, 5); find the value x(3)+x(2)=?

Exercise 2:

Define a **column** vector x = (1, 2, 3, 4, 5);

Transpose this vector?

2. Matrix

Exercise 1

a. Define matrix $A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ by two approaches: Define element by element and used the function *zeros*
b. Change the diagonal of A to 1 without define the matrix again
c. Using the same procedure, change A to $A = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 4 & 1 \end{bmatrix}$
d. Define B = A transposed
e. Calculate A+B, A-B, A*B and A.*B. Give remark about the results of A*B and A.*B
f. Reduce the size of A to 2x2 by removing row 3 and column 3 of the original matrix. Do the same for B but remove row 1 and column 1. *Hint: use the operator* ":"
g. What's the difference A(1:3) and A(1:3,1) or A(:,1)?

3. Function

Exercise 1

Write a function that calculates the mean of a vector.

Application: vector x = 1:3:100

4. if else structure

Exercise 1

Electricity bill: In Vietnam, the electricity price varies with the number kWh, e.g the first 50 kWh will be cheaper than the next 50 kWh. In particular:

- From 0 to 50 kWh: 1.484 VND / kWh
- From 51 to 100 kWh: 1.533 VND / kWh
- From 101 to 200 kWh: 1.786 VND / kWh
- From 201 to 300 kWh: 2.242 VND / kWh
- From 301 to 400 kWh: 2.503 VND / kWh

- From 400 kWh: 2.587 VND / kWh

Write a script to calculate the price of electricity with input is the number of kWh

5. **For loop**

Exercise 1:

Write a script to calculate these sums:

a. $1^2 + 2^2 + 3^2 + ... + 1000^2$

b. $1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - ... - \frac{1}{1003}$

c. Prove that: $\frac{1}{1^2 \cdot 3^2} + \frac{1}{3^2 \cdot 5^2} + \frac{1}{5^2 \cdot 7^2} + ... = \frac{\pi^2 - 8}{16}$

Exercise 2:

Write a script to calculate:

a. Combination of a set ($C_n^k = \frac{n!}{k!(n-k)!}$)

b. Permutation of a set ($A_n^k = \frac{n!}{(n-k)!}$)

Hint: write a function to calculate the factorial of a number first, then use the function to write the required script)

6. **While loop**

Exercise 1:

Doubling time of an investment: write a script to calculate hold long it will take for our bank investment to double. Suppose that the interest rate is 10% per year and all the interest is added to the initial investment.

7. **Graphic**

Exercise 1:

a. Draw the graph of a linear function y = x+1 in the range of x = -5 to x = 5. *Hint:* consult help to know how to use the function *linspace* and *plot*.
b. Change the color and the pattern of the line using *plot* function (*Hint: the option for colors and patterns can be found in the help sections*)
c. Plot the line y = -x + 2 on the same plot.
d. Add title, legend and labels for each axis.

II. Matlab-numerical method

1. Bisection method

Given an M-file to implement the bisection method:

```
function [root,fx,ea,iter]=bisection(func,xl,xu,es,maxit)
% bisect: root location zeroes
% [root,fx,ea,iter]=bisect(func,xl,xu,es,maxit,p1,p2,...):
% uses bisection method to find the root of func
% input:
% func = name of function
```

```
% xl, xu = lower and upper guesses
% es = desired relative error (default = 0.0001%)
% maxit = maximum allowable iterations (default = 50)
% p1,p2,... = additional parameters used by func
% output:
% root = real root
% fx = function value at root
% ea = approximate relative error (%)
% iter = number of iterations
if nargin<3,error('at least 3 input arguments required'),end
test = func(xl)*func(xu);
if test>0,error('no sign change'),end
if nargin<4||isempty(es), es=0.0001;end
if nargin<5||isempty(maxit), maxit=50;end
iter = 0; xr = xl; ea = 100;
while (1)
xrold = xr;
xr = (xl + xu)/2;
iter = iter + 1;
if xr ~= 0,ea = abs((xr - xrold)/xr) * 100;end
test = func(xl)*func(xr);
if test < 0
xu = xr;
elseif test > 0
xl = xr;
else
ea = 0;
end
if ea <= es || iter >= maxit,break,end
end
root = xr;
fx = func(xr);
```

Exercise 1

Use the function file above to find the square root of 2, taking 1 and 2 as initial values of x_l and x_u. Continue bisecting until the maximum error is less than 0.05

Exercise 2

Use the function file above to Determine the positive real root of $\mathbf{\ln(x^2) = 0.7}$ using three iterations of the bisection method, with initial guesses of $x_l = 0.5$ and $x_u = 2$

2. Newton Raphson method

Newton's method may be used to solve a general equation f(x) = 0 by repeating the assignment

$$\text{x becomes } x - \frac{f(x)}{f'(x)}$$

where $f'(x)$ (i.e., d*f*/d*x*) is the first derivative of f(x). The process continues until successive approximations to x are close enough

Suppose that $f(x) = x^3 + x - 3$

that is, we want to solve the equation $x^3 + x - 3 = 0$ (another way of stating the problem is to say we want to find the zero of f(x)). We have to be able to differentiate f(x), which is quite easy here:

$f'(x) = 3x^2 + 1$. We could write inline objects for both f(x) and f'(x), but for this example we will use function M-files instead.

Here is a structure plan to implement Newton's method:

1. Input starting value $x0$ and required relative error e
2. While relative error $|(x_k - x_{k-1})/x_k| \geq e$, repeat up to, say, $k = 20$:

$$x_{k+1} = x_k - f(x_k)/f'(x_k)$$

Print x_{k+1} and $f(x_{k+1})$

3. Stop

Exercise 1

Use the Editor to create and save (in the current MATLAB directory) the function file **f.m** as follows:

```
function y = f(x)
 y = x^3 + x - 3;
```

Then create and save another function file **df.m**:

```
function y = df(x)
y = 3*x^2 + 1;
```

Now write a separate script file, **newtgen.m** (in the same directory), that will stop either when the relative error in x is less than 10^{-8} or after, say, 20 steps.

Exercise 2

Determine the positive root of $f(x) = x^{10} - 1$ using the Newton Raphson method and an initial guess of $x = 0.5$

Use a simple plot (Matlab) of the first few iterations (helpful in providing insight) to prove that after the first poor prediction, the technique is converging on the true root of 1, but at a very slow rate.