Labwork 2

III. Polynomial

Exercise 1: fzero

Given a quadratic equation

$$f(x) = x^2 - 9$$

a. Find the positive root of this polynomial equation using *fzero* in Matlab
b. If we put in an initial guess of zero, it finds the negative or positive root?
c. Use the *optimset* function of fzero to display all the iteration
d. Use the *optimset* function of fzero to set a tolerance: tolx = 10^{-3}
e. Verify the solution using *roots* in Matlab

Hint: use help (F1)

IV. System non linear

Exercise 1: Newton-Raphson

$$x_1^2 + x_1 x_2 = 10$$
$$x_2 + 3x_1 x_2^2 = 57$$

a. Use the graph to derive good guesses using *ezplot* in Matlab
b. Given a code below, explain the code

```
clc
clear all
close all
%%
 x=[1  1]';
 N = 50;
 for k=1:N,
F=[ x(1)*x(1)+ x(1)*x(2) - 10 ;  x(2) + 3*x(1)*x(2)*x(2)-57];
A=[2*x(1)+x(2), x(1) ; 3*x(2).^2 , 1 + 6*x(1).*x(2)];
dx=A\F;
x=x-dx
k=k+1;
end
k
x'
F'
```

c. Create and save the function file root2d.m as follows:

```
function F = root2d(x)
F(1) = x(1)^2 + x(1)*x(2) - 10;
F(2) = x(2) +3*x(1)*x(2)^2 -57 ;
```

Now write a separate script file (in the same directory), that will use the function *fsolve* to find the solution of the system with initial guess x_0 = [0,0]. Explain the code?

d. Use the function `newtmult.m` to solve the system above, explain the code?

```
function [x,f,ea,iter]=newtmult(func,x0,es,maxit,varargin)
% newtmult: Newton-Raphson root zeroes nonlinear systems
% [x,f,ea,iter]=newtmult(func,x0,es,maxit,p1,p2,...):
% uses the Newton-Raphson method to find the roots of
% a system of nonlinear equations
% input:
% func = name of function that returns f and J
% x0 = initial guess
% es = desired percent relative error (default = 0.0001%)
% maxit = maximum allowable iterations (default = 50)
% p1,p2,... = additional parameters used by function
% output:
% x = vector of roots
% f = vector of functions evaluated at roots
% ea = approximate percent relative error (%)
% iter = number of iterations
if nargin<2,error('at least 2 input arguments required'),end
if nargin<3||isempty(es),es=0.0001;end
if nargin<4||isempty(maxit),maxit=50;end
iter = 0;
x=x0;
while (1)
[J,f]=func(x,varargin{:});
dx=J\f;
x=x-dx;
iter = iter + 1;
ea=100*max(abs(dx./x));
if iter>=maxit||ea<=es, break, end
end
```