1 Hands-on–Solving Nonlinear Equations with MATLAB II

1. We have given the following function;

$$f(x) = 3x + sin(x) - e^x$$

   (a) To obtain the true value for the root r, which is needed to compute the actual error. MATLAB is used as:

```
>> solve('3*x + sin(x) - exp(x)')
ans=
.36042170296032440136932951583028
```

   (b) Comparing Muller's and Fixed-point Iteration methods. (Download muller.m, fixedpoint.m)
   save with the name *muller.m*. Then;

```
>> fx=inline(' 3 *x + sin ( x) - exp ( x) ');
>> [x,y,err]=muller(fx,0,0.5,1,10^-4,10^-4,15)
>> [x,y,err]=muller(fx,1,1.5,2,10^-4,10^-4,15)
```

   save with the name *fixedpoint.m*. Then;

```
>> gx=inline('sqrt(2*x+3)');
>> [k,x,err,X,F]=fixedpoint(gx,4,10^-4,15)
>> gx=inline('3/(x-2)');
>> [k,x,err,X,F]=fixedpoint(gx,4,10^-4,15)
>> gx=inline('log(3*x+sin(x))')
>> [k,x,err,X,F]=fixedpoint(gx,4,10^-4,15)
```

(c) Plot the behaviours of the errors (may use ratios) for both cases. Compare and discuss the rate of convergence.
Solution:

```
%format long;
realroot=0.36042170296032440136932951583028;
fx=inline('3*x+sin(x)-exp(x)');
[k1,x,y,err,S,F1]=muller(fx,1,1.5,2,10^-4,10^-4,15);
gx=inline('log(3*x+sin(x))');
[k2,x,err,X,F2]=fixedpoint(gx,4,10^-4,15);
if k1>k2
max1=k1;
else
max1=k2;
end
disp('                Muller     Fixed-Point     Muller   Fixed-Point')
disp('iteration      (x-r)          (x-r)         f(x)       f(x)')
for k=1:max1
   if k1>=k& k2>=k
plotyx1(k)=S(k)-realroot;
plotyx2(k)=X(k)-realroot;
plotxx1(k)=k;
plotxx2(k)=k;
   D=[k,plotyx1(k),plotyx2(k),F1(k),F2(k)];
   else if  k1<k& k2>=k
plotyx2(k)=X(k)-realroot;
plotxx2(k)=k;
   D=[k,S(k1)-realroot,plotyx2(k),F1(k1),F2(k)];
   else if  k1>=k& k2<k
plotyx1(k)=S(k)-realroot;
plotxx1(k)=k;
   D=[k,plotyx1(k),X(k2)-realroot,F1(k),F2(k2)];
   end
   end
   end
   disp(D);
end
plot(plotxx1,plotyx1,plotxx2,plotyx2);
%plot(plotxx2,plotyx2);
```

save with the name *main.m*. Then;

```
>> main
```

For the rate of convergence: Muller's method converges much faster than fixed-point iteration.

2. The following **MATLAB** command plots the function
$f_1(x) = x^2 - 3x + 2$

```
x = linspace( 0, 4, 100 );plot(x,x.^2-3*x+2); grid on
```

and the following finds the roots; (What are 1 -3 2?)

```
>> roots([1 -3 2])
ans =
     2
     1
```

These are *distinct real* roots. Apply same procedure for the following functions

$$f_2(x) = x^2 - 10x + 25$$

$$f_3(x) = x^2 - 17x + 72.5$$

comment the outputs. You should observe, repeated real roots, and complex roots.

3. A pair of equations:

   $x^2 + y^2 = 4$

   $e^x + y = 1$

   (a) Write a **MATLAB** program to solve this system by

      i. expanding both functions as a Taylor series expansion (begin with $x_0 = 1, y_0 = -1.7$). See lecture notes.

      ii. and by Iteration (begin with $x = 1$). See lecture notes.

   (b) Tabulate the actual error values as the following; (See Table 1. Number of iterations is not limited to or defined as 15.)

n	Expansion $f(x_n)$	Iteration $f(x_n)$
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
12		
13		
14		
15		

Table 1: The Error Sequences