1 Hands-on–Solving Nonlinear Equations with MATLAB I

1. We have given the following function;

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

The MATLAB program for the algorithm (see the lecture notes; an algorithm for halving the interval (Bisection)) is given.

```
function rtn=mybisec(fx,xa,xb,n)
%bisec does n bisections to approximate
% a root of fx
x=xa; fa=feval(fx,x);
x=xb; fb=feval(fx,x);
for i=1:n;
  xc=(xa+xb)/2; x=xc; fc=feval(fx,x);
  X=[i,xa,xb,xc,fc];
  disp(X);
  if fc*fa<0
    xb=xc;
    else xa=xc;
  end
end
```

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

```
>> fx=inline(' 3 *x + sin ( x) - exp ( x) ');
>> mybisec(fx,0,1,13)
```

Modify this MATLAB program for the bisection method for using a tolerance value of 1E-4 (see next page).

```
function [c,err,yc]=modbisect(f,a,b,delta)
% nput - f is t e function input as a string 'f'
%     - a and b are t e left and rig t endpoints
%     - delta is t e tolerance
% utput - c is t e zero
%      - yc= f(c)
%       - err is t e error estimate for c

% format long;
disp('iteration        xa          xb          xc        f(c)')
ya=feval(f,a);
yb=feval(f,b);
if ya*yb > 0,end
max1=1+round((log(b-a)-log(delta)) log(2));
for k=1:max1
c=(a+b) 2;
yc=feval(f,c);
if yc==0
a=c;
b=c;
elseif yb*yc>0
b=c;
yb=yc;
else
a=c;
ya=yc;
end
    X=[k,a,b,c,yc];
    disp(X);
    if b-a   delta, break,end
end
c=(a+b) 2;
err=abs(b-a);
yc=feval(f,c);
```

save with the name *modbisect.m*. Then (see next page);

```
>> fx=inline(' 3 *x + sin ( x) - exp ( x) ');
>> [c,err,yc]=modbisect(fx,0,1,10^-4)
>> [c,err,yc]=modbisect(fx,1,2,10^-4)
```

2. Now, we have given the following function;

$$f(x) = x^6 - x - 1 = 0$$

Apply all the procedures you studied above to find the root(s) of this function by modifying the given codes (Interval Halving (bisect.m) and Newton's (newton.m) methods).