EM314 – NUMERICAL METHODS

ASSIGNMENT 02

DE SILVA K.G.P.M.

E/15/065

SEMESTER 04

12/11/2018

(1)

① $f(x) = 0$

By convergence theorem, error estimate =

$$\underbrace{|x_k - x_*|}_{= e_k} \le \frac{b-a}{2^{k+1}}.$$

$$e_k \le \frac{b-a}{2^{k+1}}$$

when $e^k < \tau$

$$\frac{b-a}{2^{k+1}} < \tau.$$

$$\frac{b-a}{\tau} < 2^{k+1}.$$

$$\log_2\left(\frac{b-a}{\tau}\right) < \log_2 2^{k+1}$$

$$\log_2\left(\frac{b-a}{\tau}\right) < (k+1) \times 1$$

$$\log_2\left(\frac{b-a}{\tau}\right) < k+1.$$

$$k > \log_2\left(\frac{b-a}{\tau}\right) - 1.$$

(2)

② a). $g(x) = e^{-x}$ $\qquad G = [\ln(1.1), \ln(3)]$

$L = |g'(x)| = |-1e^{-x}|$

$= |e^{-x}|.$

We get the largest value for L when $x = \ln(1.1)$.

Because this is a decreasing function.

$L = |e^{-\ln(1.1)}| = 0.9090 < 1.$

Then g is a contraction on $G = [\ln 1.1, \ln 3.]$ //.

b). $G = [\ln(1.1), \ln(3)]$

$G = [0.09531, 1.0986]$

$g(x) = e^{-x}.$

$g(\ln(1.1)) = e^{-\ln 1.1} = 0.9090$ —— ①

$g(\ln(3)) = e^{-\ln 3} = 0.3334$ —— ②.

① and ② falls on. $G = [\ln(1.1), \ln(3)]$. and this is a decreasing function as in the privious one.

So, $g: G \rightarrow G$.

c) $x_{k+1} = g(x_k)$.

Let us use this result $|x_{k+1} - x_k| \leq L^k |x_1 - x_0|$

We have to prove it first.

$x_{k+1} = f(x_k). \qquad x_k = f(x_{k-1})$

$|f(x_k) - f(x_{k-1})| \leq L|x_k - x_{k-1}|$ —— ①

$|x_k - x_{k-1}| = |f(x_{k-1}) - f(x_{k-2})|.$

then $|f(x_{k-1}) - f(x_{k-2})| \leq L|x_{k-1} - x_{k-2}|$ —— ②.

by ①, ②.

$|f(x_k) - f(x_{k-1})| \leq L^2|x_{k-1} - x_{k-2}|.$

$\leq L^3|x_{k-2} - x_{k-3}|.$

⋮

(2) C)

$$|f(x_k) - f(x_{k-1})| \leq L^k |x_1 - x_0|.$$

$$|x_{k+1} - x_k| \leq L^k |x_1 - x_0|. \quad —③$$

by ③

$$|g(x_k) - x_k| \leq L^k |x_1 - x_0|$$

$$L \in [0,1)$$

then. k is large.

$$L^k \to 0.$$

$$|g(x_k) - x_k| \to 0$$

$$g(x_k) \to x_k$$

x_k is the fixed point.

$\therefore$ $x_{k+1} = g(x_k)$ converges to the fixed point $x_* \in G$ for any $x_0 \in G$ //.

(3)

(3) a). Let us take x_k,

$$x_k \in [-0.5, 0.5]$$

$$g(x) = \tan^{-1}(2x)$$

$$g'(x) = \frac{2}{1+4x^2}$$

Consider,

$$e_{k+1} = |x_{k+1} - 0|$$

$$x_{k+1} = g(x_k)$$

$$\alpha = g(0).$$

$$e_{k+1} = |g(x_k) - 0|$$

$$= |g'(\xi)|\,\underbrace{|x_k - 0|}_{= e_k}$$

$$= |g'(\xi)|\,e_k.$$

But $|g'(x)| > 1$ for some values in the interval $(-0.5, 0.5)$

and $e_{k+1} > e_k$.

So, fixed point iteration will not converge to this fixed point.

b)

$x_0 = 2$.

If there is a fixed point. $\tan^{-1}(2x_k) = x_k$

$g(x_k) = x_k$.

Then, $x_{k+1} = g(x_k). \rightarrow x_1 = g(x_0)$.

$x_0 = 2$

$$x_1 = g(2) = \tan^{-1}(2\times 2) = 1.325$$

$$\text{error}_1 = |x_1 - x_*| = |1.325 - 1.66| = 0.334$$

$$x_2 = g(x_1) = \tan^{-1}(2\times 1.325) = 1.209$$

$$\text{error}_2 = |x_2 - x_*| = |1.209 - 1.66| = 0.451$$

(3) b)

b)

ii).

$f(x) = g(x) - x = \tan^{-1}(2x) - x$

$f'(x) = \frac{2}{1+4x^2} - 1 = \frac{2-(1+4x^2)}{1+4x^2} = \frac{1-4x^2}{1+4x^2}$

b) Newton's method

$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} = x_k - \frac{[\tan^{-1}(2x) - x](1+4x^2)}{(1-4x^2)}$

Let $x_0 = 2$,

$x_1 = x_0 - \frac{[\tan^{-1}(2x_0) - x_0]\,((1+4x_0^2))}{(1-4x_0^2)} = 1.2359$

$e_1 = |x_1 - x_*| = |1.2359 - 1.66| = 0.42$

$x_2 = x_1 - \frac{[\tan^{-1}(2(x_1)) - x_1][1+4x_1^2]}{1-4x_1^2} = 1.167$

$e_2 = |x_2 - x_*| = |1.167 - 1.66| = 0.49$

(4) a)

Editor - C:\Users\Prasad-PC\newtons.m*

```matlab
function [sol,res,niter]=newtons(f,Fd,Tol,x0,nmax)
    niter=1;
    x=x0-f(x0)/Fd(x0);
    disp(x);
    while abs(x0-x)>=Tol && niter<=nmax
        niter=niter+1;
        x0=x;
        x=x0-f(x0)/Fd(x0);
    end

    if niter>nmax
        disp('Newtons method stop without convergence');
    end
    res=abs(x0-x);
    sol=x;
end
```

(4) b)

Editor - C:\Users\Prasad-PC\RunNewton.m

```matlab

x0=100;
Tol=10^(-5);
namx=110;
f=@(x) x^2+4*x-4;
Fd=@(x) x*2+4;

[sol,res,niter]=newtons(f,Fd,Tol,x0,namx);

if niter>0
    fprintf('solution is %f\n',sol);
    fprintf('Residual is %f\n',res);
    fprintf('Iterations =%d\n',niter);
end

```

Command Window

```
>> roots([1,4,-4])

ans =

  -4.828427124746190
   0.828427124746190

solution is 0.828427
Residual is 0.000004
Iterations =9
>>
```

Yes. The expected value is 0.828427. And it is obtained.

(4) c)

Editor - C:\Users\Prasad-PC\RunNewton.m

```matlab

x0=100;
Tol=10^(-5);
namx=110;
f=@(x) x^2+4*x-4;
Fd=@(x) x*2+4;

[sol,res,niter]=newtons(f,Fd,Tol,x0,namx);
```

Editor - C:\Users\Prasad-PC\newtons.m

```matlab
function [sol,res,niter]=newtons(f,Fd,Tol,x0,nmax)
    expected_x=0.828427124746190;
    niter=1;
    x=x0-f(x0)/Fd(x0);
    fprintf('k\t  xk\t\tek=|xk-x*|\tek/(ek-1)^2\n\n')
    fprintf('%d\t%f\t%f\t%f\n',niter,x,abs(x-expected_x),abs(x-expected_x)/(x0-expected_x)^2);
    while abs(x0-x)>=Tol && niter<=nmax
        niter=niter+1;
        x0=x;
        x=x0-f(x0)/Fd(x0);
        fprintf('%d\t%f\t%f\t%f\n',niter,x,abs(x-expected_x),abs(x-expected_x)/(x0-expected_x)^2);

    end

    if niter>nmax
        disp('Newtons method stop without convergence');
    end
    res=abs(x0-x);
    sol=x;
end
```

Command Window

```
k     xk          ek=|xk-x*|    ek/(ek-1)^2

1   49.039216   48.210789   0.004902
2   23.597979   22.769552   0.009796
3   10.955252   10.126825   0.019533
4   4.786381    3.957954    0.038594
5   1.982606    1.154179    0.073677
6   0.995671    0.167243    0.125546
7   0.833096    0.004668    0.166908
8   0.828431    0.000004    0.176485
9   0.828427    0.000000    0.176788
>>
```

When going to higher iterations the error goes to 0. And approximate value convergence to expected value as expected.

(4) d)

Command Window

k	xk	ek=|xk-x*|	ek/(ek-1)^2
1	49.039216	48.210789	0.004902
2	23.597979	22.769552	0.009796
3	10.955252	10.126825	0.019533
4	4.786381	3.957954	0.038594
5	1.982606	1.154179	0.073677
6	0.995671	0.167243	0.125546
7	0.833096	0.004668	0.166908
8	0.828431	0.000004	0.176485
9	0.828427	0.000000	0.176788
10	0.828427	0.000000	16228413.531752

```
>>
```

Same the as the previous case in higher iterations error equals to 0. The approximate answer convergence to expected value until 9th step. But in the 10th iteration we got a strange value because of computational limits.

(5)

Editor - C:\Users\Prasad-PC\testKepler.m

```
E0=100;
e=0.8;
Tol=10^(-8);
nmax=100;
M=3;
f=@(E) M-E+e*sin(E);
Fd=@(E) -1+e*cos(E);

[sol,res,niter]=newtons(f,Fd,Tol,E0,nmax);


if niter>0
    fprintf('solution is %f\n',sol);
    fprintf('Residual is %f\n',res);
    fprintf('Iterations =%d\n',niter);

end
```

Command Window

```
solution is 3.062894
Residual is 0.000000
Iterations =16
>>
```

Solution is 3.062894.

(6)

Editor - C:\Users\Prasad-PC\Gas.m

```
N=1000;
T=300;
p=3.5*(10^7);
Tol=10^(-12);
a=0.401;
b=42.7*(10^(-6));
k=1.3806503*(10^(-23));
nmax=100;

f=@(x) p*(x.^3)-(p*N*b+2*k*N*T)*(x.^2)+a*(N^2)*x-a*b*(N^3);

[zero, res, niter]=bisection(f,0,2,Tol,nmax);

if niter>0
    fprintf('solution is %f\n',zero);
    fprintf('Residual is %f\n',res);
    fprintf('Iterations =%d\n',niter);

end
```

Given code for bisection method,

Editor - C:\Users\Prasad-PC\bisection.m

```
function [zero, res, niter] = bisection(f,a,b,tol,nmax)
    x = [a (a+b)/2 b]; y = f(x); niter = 0; I = (b-a)/2;
    if y(1)*y(3)>0
        error('The signs of the function at the extrema must be opposite');
    elseif y(1) == 0
        zero = a; res = 0; return
    elseif y(3) == 0
        zero = b; res = 0; return
    end
    while ( I >= tol && niter <= nmax )
        if sign(y(1))*sign(y(2))<0
             x(3) = x(2); x(2) = (x(1) + x(3))/2;
             y = f(x); I = (x(3)-x(1))/2;
        elseif sign(y(2))*sign(y(3))<0
             x(1) = x(2); x(2) = (x(1) + x(3))/2;
             y = f(x); I = (x(3)-x(1))/2;
        else
             x(2) = x(find(y ==0)); I = 0;
    end
        niter = niter+1;
    end
    if niter > nmax
         fprintf('bisection method exited without convergence');
end
zero = x(2); res = f(x(2));
```

Command Window

```
solution is 0.042700
Residual is 0.000000
Iterations =40
>>
```

Solution is 0.042700.