EM314

NUMERICAL METHODS

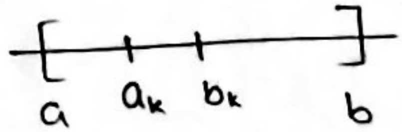
ASSIGNMENT 2

JAYASOORIYA J.K.C.N.

E/15/154

Theory

(1).



Consider after k iterations,

$$b_k - a_k = \frac{b_{k-1} - a_{k-1}}{2}$$

$$b_{k-1} - a_{k-1} = \frac{b_{k-2} - a_{k-2}}{2}$$

$$\vdots$$

$$b_1 - a_1 = \frac{b_0 - a_0}{2}$$

$$\Rightarrow \quad b_k - a_k = \frac{b-a}{2^k} \quad \text{—— (A)}$$

If tolerance τ is satisfied after k iterations,

$$b_k - a_k < 2\tau$$

From (A) $\Rightarrow$ $$\frac{b-a}{2^k} < 2\tau$$

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

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

$$\log_2\left(\frac{b-a}{\tau}\right) < (k+1)\cdot\underbrace{\log_2 2}_{=1}$$

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

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

$G = [\ln 1.1 , \ln 3]$

$= [0.0953 , 1.0986]$

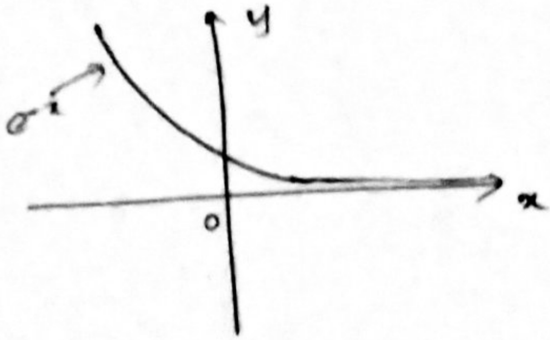
(a) $g(\ln 1.1) = 0.9091$, $g(\ln 3) = 0.3333$

$|g(\ln 1.1) - g(\ln 3)| = 0.5758$ — (A)

$|\ln 1.1 - \ln 3| = 1.0033$ — (B)

(A) < (B)

$\therefore$ g is contraction on G. //

(b). 

e^{-x} is continuous and decreasing in the given interval.
Further,

- g is a contraction on G.

$\therefore$ $g : G \to G$ //

(c) From Banach fixed point th= it can be concluded that,

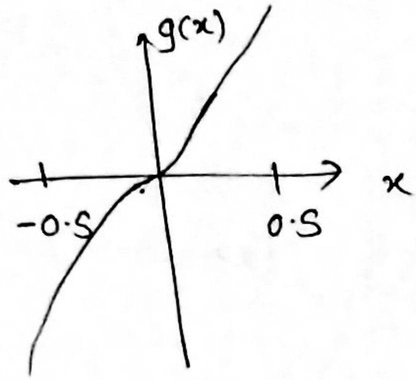
$g(x_k) = x_{k+1}$ converges to the unique fixed point $x_* \in G$ for any $x_0 \in G$. //

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

(a). $x=0$ is a fixed point.

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

consider interval $[-0.5, 0.5]$



when $x = 0.5$

$g'(x) = 1$

$\therefore \max |g'(x)| \geq 1$;

~~x e (-1~~

$x \in [-0.5, 0.5]$

$\therefore$ fixed point iteration will not converge. //

(b). $x_0 = 2$

(i). $res = |g(x_0) - x_0| = \cancel{73.96}\ 0.6741$

$k=1.\quad x_1 = g(x_0) \Rightarrow x_1 = 1.3258$

$\Rightarrow e_1 = |1.16 - 1.3258| = 0.4858$ //

$k=2.\quad x_2 = g(x_1)$

$\Rightarrow x_2 = 1.2101$

$\Rightarrow e_2 = |1.16 - 1.2101|$

$= 0.0501$ //

$k=3:\quad x_3 = g(x_2)$

$= 1.1789$

$e_3 = |1.16 - 1.1789|$

$= 0.018$ //

(ii).

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

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

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

$$\Rightarrow f(x) = 0$$

$$\Rightarrow \tan^{-1}(2x) - x = 0$$

From Newton's method,

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

$$x_{n+1} = x_n - \frac{\{\tan^{-1}(2x_n) - x_n\}}{\left\{\frac{1-4x_n^2}{1+4x_n^2}\right\}}$$

$$x_{n+1} = x_n - \frac{\{\tan^{-1}(2x_n) - x_n\}\{1+4x_n^2\}}{\{1-4x_n^2\}}$$

Let $x_0 = 2$.

$k=1$: $x_1 = 2 - \frac{\{\tan^{-1}(2\times2) - 2\}\{1+4\times2^2\}}{\{1-4\times2^2\}}$

$$x_1 = 1.2359 \;//$$

$k=2$: $x_2 = 1.2359 - \frac{\{\tan^{-1}(2\times1.2359) - 2\}\{1+4\times1.2359^2\}}{\{1-4\times1.2359^2\}}$

$$x_2 = 1.1669 \;//$$

4. (a) Newton's Method (MATLAB code)

```
function [x_n1,res,niter]=newtons(f,diff_f,x0,T,nmax)

niter=0;
res=abs(x0);
x_n=x0;


x_n1 = x_n - (f(x_n)/diff_f(x_n));

if (diff_f(x_n) == 0)
    disp('diff_f(x) must be none zero value.');
    return
end


while(res>=T) && (niter<nmax)

    x_n1 = x_n - (f(x_n)/diff_f(x_n));
    res=abs(x_n1-x_n);
    x_n=x_n1;


    niter=niter+1;
end
```

testNewtons.m

```
x0=100;
T=10^(-5);
nmax=100;
[x_n1,res,niter] = newtons(@f,@diff_f,x0,T,nmax)
```

f.m

```
function y=f(x)
    y=x^2+4*x-4;
```

diff_f.m

```
function y=diff_f(x)
    y=2*x+4;
```

(b).

Output

```
>> testNewtons

x_n1 =

    0.8284


res =

   3.8464e-06


niter =

     9
```

Expected solution is obtained and it is accurate up to 4 decimal places.

(c).

```
function [x_n1,res,niter]=newtons(f,diff_f,x0,T,nmax)
niter=0;
res=abs(x0);
x_n=x0;
x_true=2*(sqrt(2)-1);
ek_1=abs(x_n-x_true);

x_n1 = x_n - (f(x_n)/diff_f(x_n));

if (diff_f(x_n) == 0)
    disp('diff_f(x) must be none zero value.');
    return
end

fprintf(' __________________________________________________ \n');
fprintf('| k  |       xk       |  ek=|xk - x*| | ek/(ek-1)^2 |\n');
fprintf('|----+--------------+---------------+-------------|\n');

while(res>=T) && (niter<nmax)
    x_n1 = x_n - (f(x_n)/diff_f(x_n));
    res=abs(x_n1-x_n);
    ek=abs(x_n-x_true);
    x_n=x_n1;

    niter=niter+1;
    c=ek/(ek_1)^2;
    fprintf('|%2d  |  %10.6f  |   %10.6f  |  %10f |\n',niter,x_n,ek,c);
    ek_1=ek;

end
```

Output when $\tau = 10^{-5}$

```
>> testNewtons

 ______________________________________________________
| k  |      xk       |  ek=|xk - x*|  | ek/(ek-1)^2  |
|----+---------------+----------------+--------------|
| 1  |   49.039216   |    99.171573   |    0.010084  |
| 2  |   23.597979   |    48.210789   |    0.004902  |
| 3  |   10.955252   |    22.769552   |    0.009796  |
| 4  |    4.786381   |    10.126825   |    0.019533  |
| 5  |    1.982606   |     3.957954   |    0.038594  |
| 6  |    0.995671   |     1.154179   |    0.073677  |
| 7  |    0.833096   |     0.167243   |    0.125546  |
| 8  |    0.828431   |     0.004668   |    0.166908  |
| 9  |    0.828427   |     0.000004   |    0.176485  |

x_n1 =

    0.8284


res =

   3.8464e-06


niter =

     9
```

Quadratic convergence is not obtained. Value of the last column of the table does not converge to a constant.

(d).

Output when $\tau = 10^{-8}$

```
>> testNewtons
```

k	xk	ek=|xk - x*|	ek/(ek-1)^2
1	49.039216	99.171573	0.010084
2	23.597979	48.210789	0.004902
3	10.955252	22.769552	0.009796
4	4.786381	10.126825	0.019533
5	1.982606	3.957954	0.038594
6	0.995671	1.154179	0.073677
7	0.833096	0.167243	0.125546
8	0.828431	0.004668	0.166908
9	0.828427	0.000004	0.176485
10	0.828427	0.000000	0.176766

```
x_n1 =

    0.8284


res =

   2.6155e-12


niter =

    10
```

Here the value of the last column converges to some constant with an accuracy of 0.001. Therefore it can be concluded that quadratic convergence is obtained here.

(5). Kepler's Equation

newtons.m

```
function [x_n1,res,niter]=newtons(f,diff_f,x0,T,nmax)
niter=0;
res=abs(x0);
x_n=x0;

x_n1 = x_n - (f(x_n)/diff_f(x_n));

if (diff_f(x_n) == 0)
    disp('diff_f(x) must be none zero value.');
    return
end

while(res>=T) && (niter<nmax)
    x_n1 = x_n - (f(x_n)/diff_f(x_n));
    res=abs(x_n1-x_n);
    x_n=x_n1;
    niter=niter+1;
end
```

f.m

```
function y=f(x)
    M=3;
    e=0.8;
    y=x-sin(x)*e-M;
```

diff_f.m

```
function y=diff_f(x)
    e=0.8;
    y=1-cos(x)*e;
```

q5.m

```
x0=4;
T=10^(-8);
nmax=100;
[x_n1,res,niter] = newtons(@f,@diff_f,x0,T,nmax)
```

Output

```
>> q5

x_n1 =

    3.0629


res =

   4.0056e-09


niter =

     4
```

Hence, angle E = 3.0629 rad

(6). State Equation of a Gas

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
    disp('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(y ==0); I = 0;
    end
    niter = niter+1;
end
if niter > nmax
    disp('bisection method exited without convergence');
end
    zero = x(2); res = f(x(2));
```

f.m

```
function y = f(x)
    p=3.5*(10^7);
    a=0.401;
    N=1000;
    b=42.7*10^(-6);
    k=1.3806503*(10^(-23));
    T=300;
    y=p*(x.^3)+a*(N^2)*x-a*b*(N^3)-(N*b*p+k*N*T)*x.^2;
```

q6.m

```
a = 0;
b = 1;
tol = 10^(-12);
nmax = 100;
[zero, res, niter] = bisection(@f,a,b,tol,nmax)
```

Output

```
>> q6

zero =

    0.0427


res =

   4.0787e-07


niter =

    39
```

Therefore, volume = 0.0427 m^3