Assignment#1
DS 288 (AUG) 3:0 Numerical Methods

Somil Jain
CDS(M.Tech)
S.No. 16071

QUESTION 1 : Bessel function :

Exact:

X=1	X=5	X=50
0.765197686560000	-0.177596771310000	0.055812327669000
0.440050585740000	-0.327579137590000	-0.097511828125000
0.114903484930000	0.046565116278000	-0.059712800794000
0.019563353983000	0.364831230610000	0.092734804062000
0.002476638964100	0.391232360460000	0.070840977282000
0.000249757730210	0.261140546120000	-0.081400247697000
0.000020938338002	0.131048731780000	-0.087121026821000
0.000001502325817	0.053376410150000	0.060491201260000
0.000000094223442	0.018405216655000	0.104058563170000
0.000000005249250	0.005520283138500	-0.027192461044000
0.000000000263062	0.001467802647300	-0.113847849150000

A) Recursion in Forward Direction

J2 (approximated) =

0.765198000000000	-0.177597000000000	0.055812300000000
0.440051000000000	-0.327579000000000	-0.097511800000000
0.114904000000000	0.046565400000000	-0.059712772000000
0.019565000000000	0.364831320000000	0.092734778240000
0.002486000000000	0.391232184000000	0.070840945388800
0.000323000000000	0.261140174400000	-0.081400226977792
0.000743999999999	0.131048164800000	-0.087120990784358
0.008604999999989	0.053375421120000	0.060491189189546
0.119725999999847	0.018403014336001	0.104058523757431
1.907010999997570	0.005514224755202	-0.027192461587168
34.206471999956413	0.001448194782725	-0.113847809928812

absolute_error =

0	0	0
0	0	0
0.000000515070000	0.000000283722000	0.000000028794000
0.000001646017000	0.000000089390000	0.000000025822000
0.000009361035900	0.000000176460000	0.000000031893200
0.000073242269790	0.000000371720000	0.000000020719208
0.000723061661997	0.000000566980000	0.000000036036642
0.008603497674172	0.000000989030000	0.000000012070454
0.119725905776406	0.000002202318999	0.000000039412569
1.907010994748320	0.000006055244798	0.000000038412832
34.206471999693349	0.000019605217275	0.000000009928812

relative_error =

1.0e+11 *

0	0	0
0	0	0
0.000000000000000	0.000000000000000	0.000000000000000
0.000000000000001	0.000000000000000	0.000000000000000
0.000000000000038	0.000000000000000	0.000000000000000
0.000000000002933	0.000000000000000	0.000000000000000
0.000000000345329	0.000000000000000	0.000000000000000
0.000000057267855	0.000000000000000	0.000000000000000
0.000012706594408	0.000000000000001	0.000000000000000
0.003632920883456	0.000000000000011	0.000000000000000
1.300319772513451	0.000000000000134	0.000000000000000

B) Recursion in Backward Direction

J2(approximated) =

0.765197656284238	-0.177596689965158	0.055812260393830
0.440050568335090	-0.327578987485696	-0.097511813995781
0.114903480385942	0.046565094970880	-0.059712732953661
0.019563353208678	0.364831063462400	0.092734795359488
0.002476638866126	0.391232181184000	0.070840908396800
0.000249757720330	0.261140426432000	-0.081400250016000
0.000020938337174	0.131048671680000	-0.087120958400000

```
   0.000001502325758   0.053376385600000   0.060491220000000
   0.000000094223438   0.018405208000000   0.104058500000000
   0.000000005249250   0.005520280000000  -0.027192500000000
   0.000000000263062   0.001467800000000  -0.113847800000000



absolute_error =

   1.0e-06 *

   0.343715761963459   0.310034841688456   0.039606169795203
   0.431664909983365   0.012514304148503   0.013995780923093
   0.004544057977118   0.021307120040559   0.067840338555114
   0.000774321998881   0.167147600149686   0.008702512002690
   0.000097974000007   0.179276000156214   0.068885200002478
   0.000009880000030   0.119688000066454   0.002319000005913
   0.000000827999998   0.060100000032204   0.068420999990715
   0.000000059400000   0.024549999998180   0.018739999996242
   0.000000003726000   0.008655000001412   0.063170000008661
                   0                   0                   0
                   0                   0                   0


relative_error =

   1.0e-05 *

   0.044918538987747   0.174572116470692   0.070963156499917
   0.098094291339723   0.003820240048508   0.014352910030471
   0.003954673768064   0.045757686748494   0.113611047636424
   0.003958022737583   0.045815047102797   0.009384299768263
   0.003955925810213   0.045823407845258   0.097239200594683
   0.003955833527796   0.045832790749949   0.002848885687111
   0.003954468584210   0.045860802478499   0.078535575724209
   0.003953869332658   0.045994101006773   0.030979712100104
   0.003954429933039   0.047024711328571   0.060706200512744
                   0                   0                   0
                   0                   0                   0
```

#Is the last value computed by the recurrence relation is having less or more error compared to the forward approach?
ANSWER : No, forward approach having more error in both the cases (absolute and relative error), backward recursion gives more reliable results.

QUESTION 2 : Using Newton’s method, Secant method, and Modified Newton’s method :

A) f(x)=x*sin(x) + 3*cos(x)-x

CALL :

```
disp("Newton's Method for x*sin(x) + 3*cos(x)-x :::");
[root,tot_iter]= newton_method(-5)
[root,tot_iter]= newton_method(-3)
[root,tot_iter]= newton_method(1)


disp("Modify Newton's Method for x*sin(x) + 3*cos(x)-x :::");
[root,tot_iter]= modify_newton2(-5)
[root,tot_iter]= modify_newton2(-3)
[root,tot_iter]= modify_newton2(1)


disp("Secant Method for x*sin(x) + 3*cos(x)-x :::");

[root,tot_iter]= secant_method(-5,-4)
[root,tot_iter]= secant_method(-3,-1)
[root,tot_iter]= secant_method(1,3)
```

OUTPUT :

```
Newton's Method for x*sin(x) + 3*cos(x)-x :::

root =  -4.7124
tot_iter =  11

root = -3.2088
tot_iter =5

root =  1.5708
tot_iter = 10


Modify Newton's Method for x*sin(x) + 3*cos(x)-x :::

root = -7.0682
tot_iter = 8

root = -2.8523
tot_iter = 3
```

root = 0
tot_iter =5

Secant Method for x*sin(x) + 3*cos(x)-x :::

root = -4.7124
tot_iter = 8

root = -3.2088
tot_iter = 6

root = 1.5708
tot_iter = 6

B) f(x) = sin(x) - 0.1*x

CALL :

```
disp(" Newton's Method for sin(x) - 0.1*x :::");

[root,tot_iter]= newton_method2(3)
[root,tot_iter]= newton_method2(7)
[root,tot_iter]= newton_method2(8)

disp("Modify Newton's Method for sin(x) - 0.1*x :::");

[root,tot_iter]= modify_newton2(3)
[root,tot_iter]= modify_newton2(7)
[root,tot_iter]= modify_newton2(8)


disp("Secant Method for sin(x) - 0.1*x :::");


[root,tot_iter]= secant_method2(3,4)
[root,tot_iter]= secant_method2(7,8)
[root,tot_iter]= secant_method2(8,9)
```

OUTPUT :

Newton's Method for sin(x) - 0.1*x :::

root = 2.8523
tot_iter = 3

root = 7.0682
tot_iter = 4

root = 8.4232
tot_iter = 6

Modify Newton's Method for sin(x) - 0.1*x :::

root = 2.8523
tot_iter = 3

root = 7.0682
tot_iter = 4

root = 8.4232
tot_iter = 6

Secant Method for sin(x) - 0.1*x :::

root = 2.8523
tot_iter = 4

root = 7.0682
tot_iter = 7

root = 8.4232
tot_iter = 6

#observed convergence rates in these cases :

Ans : In Case A Modified Newton's method converges faster than newton's and secant method.
In case B Modified newton's and newton's converges at the same rate.
Secant method convergence rate is slow.
Result agrees with the analysis done in class.

QUESTION 3 : Cubic Newton's Method :

3. Functional form for a cubicly convergent fixed point iteration function $g(p_n)$ to solve the problem $f(x) = 0$

$$g(x) = x - \phi(x) f(x) - \psi(x) f^2(x) \quad \text{— eq (i)}$$

So, for a given condition, p is a fixed point

$$g'(p) = 0 \qquad f(p) = 0$$
$$g''(p) = 0 \qquad g(p) = p$$
$$g'''(p) \neq 0$$

$$g'(x) = 1 - \phi(x) f'(x) - \phi'(x) f(x) - f(x)\{\psi'(x) f(x) + 2f'(x)\psi(x)\}$$

$$g'(p) = 0$$
$$0 = 1 - \phi(p) f'(p) - 0 \qquad // f(p) = 0$$
$$\phi(p) = \frac{1}{f'(p)}$$

let $\phi(x) = \frac{1}{f'(x)}$, then will ensure $\phi(p) = \frac{1}{f'(p)}$ — (ii)

$$g''(x) = -\phi''(x) f(x) - \phi'(x) f'(x) - f'(x)\{\psi'(x) f(x) + 2f'(x)\psi(x)\} - f(x)\{\psi'(x) f(x) + 2f'(x)\psi(x)\}'$$

$$g''(p) = 0$$
$$g''(p) = 0 - \phi'(p) f'(p) - f'(p)\{2f'(p)\psi(p)\}$$
$$0 = + \frac{f''(p)}{(f'(p))^2} f'(p) - 2(f'(p))^2 \psi(p)$$
$$\psi(p) = f''(p) / 2(f'(p))^3$$

let $\psi(x) = \frac{f''(x)}{2(f'(x))^3}$ — (iii)

Putting (ii) & (iii) in eq (i)

$$g(x) = x - \frac{f(x)}{f'(x)} - \frac{f''(x)\, f^2(x)}{2(f'(x))^3}$$

$$g(x) = x - \frac{f(x)}{f'(x)}\left(1 + \frac{f(x) f''(x)}{2|f'(x)|^2}\right)$$

So, the iteration scheme becomes

$$p_{n+1} = p_n - \frac{f(p)}{f'(p)}\left(1 + \frac{f(p) f''(p)}{2|f'(p)|^2}\right)$$

$\alpha = 3$ (given)

$\lambda = \frac{g'''(p)}{6}$ (taylor's equation)

QUESTION 4 : Newton's method for the system :

Solution:

root = [x1 x2]' = [0.791167800382726 1.126737230197029]'
tot_iter = 4