Spring 2022 Class Project

Numerical Methods / Numerical Analysis

MATLAB Implementation

Instructor: Shahzad Ahmad **Roll number: BSEE19047**

Total Marks: 20

Deadline for Submission: 13 June 2022

Instructions:

- Project is assigned to each individual student.
- Plagiarism will be dealt strictly according to ITU Policies.
- The codes will be submitted to check plagiarism.
- Individual viva may be conducted for final evaluation.

1. **Problem Statement:**

   Implement Newton-Raphson method using MATLAB to compute the drag coefficient c needed for a parachutist of mass **m = First Two Digits of Your Registration Number ÷ 2 kg** to have a velocity of **Second Last Digit of Your Registration Number + 40 m/s** after free falling for time **t = Last Digit of your Registration Number + 5 secs**. *Note:* The acceleration due to gravity is 9.81 m/s^2.

   The drag coefficient is given by

$$f(c) = \frac{gm}{c}\left(1 - e^{-(c/m)t}\right) - v$$

   a. **Formulate an iterative formula for the Newton-Raphson method.**

```
Clc
% Setting x as symbolic variable
syms x;
%initializing value
m=9.5;
v=44;
g=9.81;
t=12;

%implementing equation
eq = 1-(exp(-(x/m)*t));
y = (((g*m)/x)*eq)-v;
% Input Section
a = input('Enter the initial guess: ');
e = 1e-4;
N = input('Enter the maximum number of Iterations: ');
% Initializing Iteration counteu
Iteration = 1;
% Finding derivate of given function
g = diff(y,x);
```

```
% Finding Functional Value
fa = eval(subs(y,x,a));
fprintf('\n')
while abs(fa)> e
fa = eval(subs(y,x,a));
ga = eval(subs(g,x,a));
if ga == 0
disp('Division by zero, not approriate.');
break;

end

x_new = a - fa/ga;
Et=x_new-a;
fprintf('Iteration=%d\t\tx_new=%f\t\tEt=%f\n',Iteration,x_new,Et);
a = x_new;
i=i+1;
if Iteration>N
disp('Not convergent');
break;
end
Iteration = Iteration + 1;
end
fprintf('\nThe root is %f\n', a);
```

b. Choose an appropriate initial guess to start iterations to achieve convergence. If the solution diverges re-choose the initial guess.

Choosing the initial guess (1) and number of iterations to 10, we achieved the convergence and the root of the equation/system cam out to be around (1.933999)

c. Calculate the approximated error after every iteration and tabulate your results.

Enter the initial guess: 1

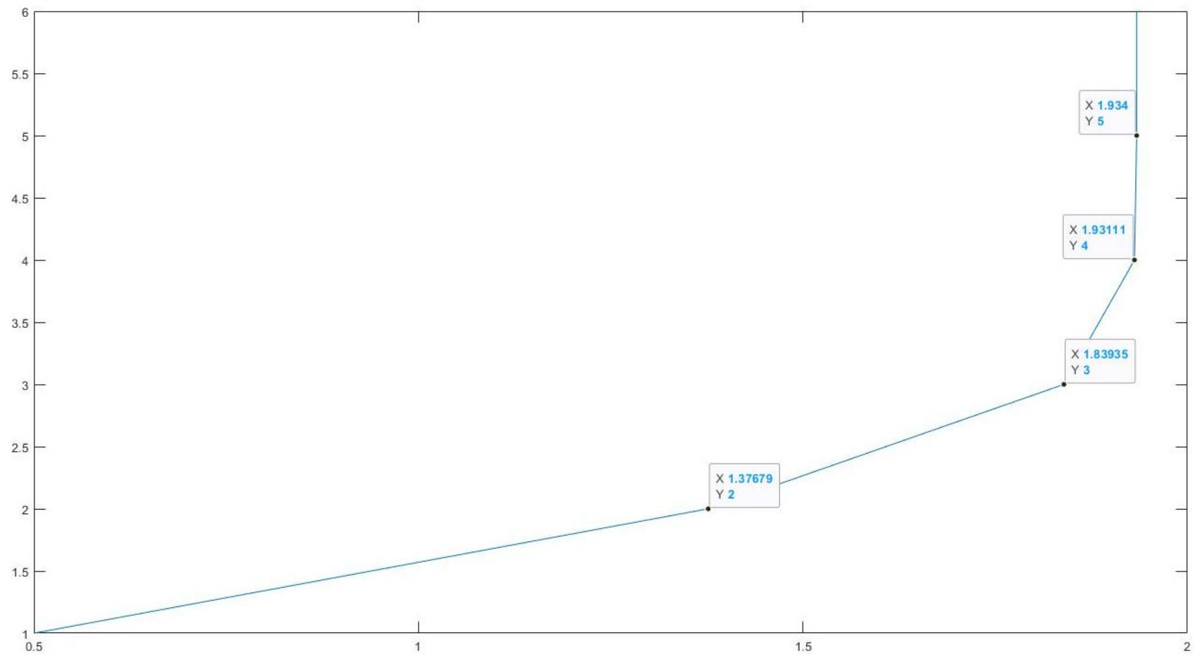
Enter the maximum number of Iterations: 10

Iteration	X new	Error
1	1.680734	0.680734
2	1.913706	0.232972
3	1.933865	0.020159
4	1.933999	0.000134
5	1.933999	0.000000

d. The ending criteria of the numerical computation is such that the consecutive calculations have a precision of 1e-4 (For Even Registration Number) and 1e-5 (For Odd Registration Number).

So, the precision in this case as approximately 100% at the initial guess of 1 with the number of iterations of 10 that means for the last precision if around (6.928648877274e-5

e. **Plot the computed drag coefficient values with respect to the number of iterations to show convergence.**



f. **Validate the computed value.**

```
Enter the initial guess: .5
Enter the maximum number of Iterations: 10

Iteration=1      x_new=1.376793      Et=0.876793
Iteration=2      x_new=1.839349      Et=0.462556
Iteration=3      x_new=1.931112      Et=0.091762
Iteration=4      x_new=1.933996      Et=0.002884
Iteration=5      x_new=1.933999      Et=0.000003

The root is 1.933999
>> val=eval(subs(y,x,x_new));
>> val=eval(subs(y,x,x_new))

val =

   4.1890e-11
```

This (4.1890e-11) is approximately equals to zero (0) that means the computed root is the TRUE ROOT.