Department of Computer Science and Engineering

Assignment

Course Title – Numerical Methods Lab
Course Code- CSE314

Year- 3rd

Semester- 2nd

Submitted by:

Section- A1

Name- Md. Abdur Rashid

ID- 19101008

Submitted to:

Nadeem Ahmed,

Assistant Professor

University of Asia Pacific

Question 1

Take a 2nd/3rd degree equation. To find the root, apply both Bisection and Newton Raphson Algorithm.

Now provide a comparative analysis of the above algorithms.

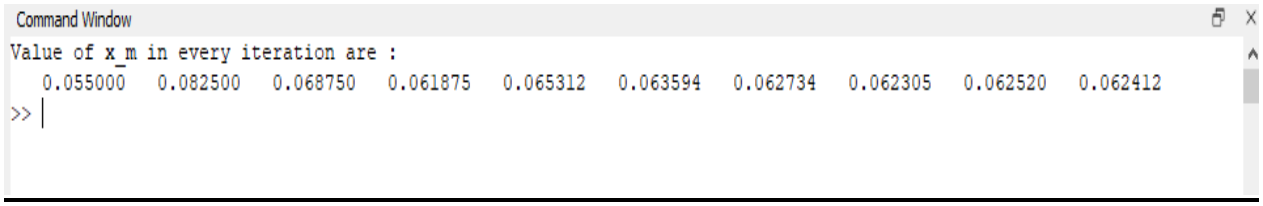
Answer

Let's consider the 3rd degree equation is x^3 - 0.165*x^2 + 3.993*10^-4 for both Bisection Method and Newton Raphson Method. Here I have perform 10 iteration for both Bisection Method and Newton Raphson Method.

Code for Bisection Method:

```
clc
clear
nMax = 10; % total number of iteration
bisec = @(x) x.^3 - 0.165*x.^2 + 3.993*10.^-4;
x_l = 0;
x_u = 0.11;
arr = [];
if bisec(x_l)*bisec(x_u) < 0
  for i = 1:nMax
    x_m = (x_l+x_u)/2;
    arr(i) = double(x_m);
    if bisec(x_l)*bisec(x_m) < 0
      x_u = x_m;
    else
      x_l = x_m;
    endif
  endfor
else
  disp("no root");
end
disp("Value of x_m in every iteration are : ");
disp(arr);
```

Output:

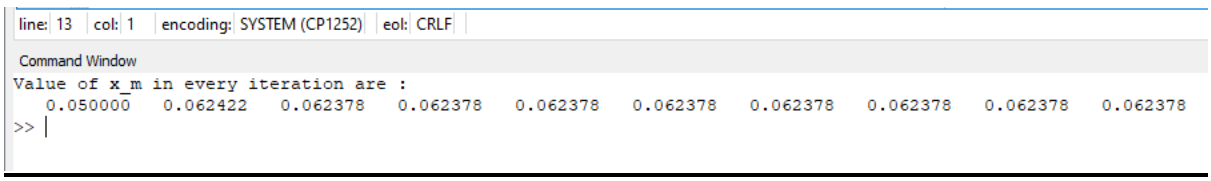


On this problem we used Bisection Method on f(x) = x^3 - 0.165*x^2 + 3.993*10^-4 on the interval 0 ≤ x ≤ 2 using octave. By using octave, it gives more accurate result than traditional manual method.

Code for Newton Raphson Method:

```
clc
clear
nMax = 10; % total number of iteration
f = @(x) x.^3 - 0.165*x.^2 + 3.993*10.^-4;
dfdx = @(x) 3*x.^2 - 0.165*2*x;
arr = [];
arr(1) = 0.05; %initial guess
for i = 1 : nMax-1
  arr(i+1) = arr(i) - (f(arr(i))/dfdx(arr(i)));
end
disp("Value of x_m in every iteration are : ");
disp(arr);
```

Output:



Comparative analysis:

On previous problem we used bisection method for root-finding.but on this problem we have used Newton Raphson Method on f(x) = x^3 - 0.165*x^2 + 3.993*10^-4 on the interval $0 \leq x \leq 2$ using octave. Its shows the new estimate xn and f(xn) at each step.

And we can observe that Newton's method is the more accurate and effective than Bisection method.

Question 2:

Find the below code:

```
function ball_function()
    % This is the main program
    time = 0.6;                     % Just pick some time
    vertical_position = y(time);
    fprintf('%f \n',vertical_position)
    time = 0.9;                     % Pick another time
    vertical_position = y(time);
    fprintf('%f \n',vertical_position)
end

% The function 'y' is a _local_ function in this file
function result = y(t)
    g = 9.81;       % Acceleration of gravity
    v0 = 5;         % Initial velocity
    result = v0*t - 0.5*g*t^2;
end
```

---

a) Change the first line from function ball_function () to ball_function(), i.e. remove the word function.

Command Window

```
>> ans2

error: parse error near line 11 of file H:\CSE 314\ans2.m

  syntax error

>>> end
      ^
>>
```

Here, we can see that when we remove function word from 1st line. We get error. Because without function this code cannot run.

b) Change the first line from function ball_function () to function ball_func (), i.e., change the name of the function.

```
warning: function name 'ball_func' does not agree with function filename
1.234200
0.526950
>>
```

If we change the line ball_function to ball_func (), we can be warning, this function name ball_func does not acceptable in this program and the function in a file should have the same name as the name of the file.

c) Change the line function result = y(t) to function y(t).

```
warning: function name 'ball_func' does not agree with function filename
error: value on right hand side of assignment is undefined
error: called from
    ball_function at line 4 column 21
>>
```

If we change the y(t) function, we cannot solve this code because y(t) function, there is an operator. the calculation (result = v0t - 0.5gt2) is assigned to the variable result. And this result will be returned through the function y(t) from which the function was called.

d) Change the line function result = y(t) to function result = y (), i.e., remove the parameter t.

```
warning: function name 'ball_func' does not agree with function filename
error: 't' undefined near line 14, column 14
error: called from
    ans2>y at line 14 column 10
    ans2 at line 4 column 21
>>
```

Our equation is: (result = v0t - 0.5gt2), where ‘t’ is included, the error is showing because if we remove t parameter and y(t) to y (), then this code will give an error. But in this code, Variable 't' which is not defined. If the 't' variable is defined in the y (t) function, then the code will run perfectly. So there is no problem if there is no argument in the y (t) function.

e) Change the first statement that calls y from vertical_position = y(time); to vertical_position = y ();

```
warning: function name 'ball_func' does not agree with function filename
error: 't' undefined near line 14, column 14
error: called from
    ans2>y at line 14 column 10
    ans2 at line 4 column 21
>>
```

Here we show the same error as the previous one because Octave does not know about the parameter unless it goes through the function. Since vertical_position = y (time) was supposed to pass an argument to y (t), but could not. Could not find any parameter. Octave is looking for where the ‘t’ is defined.