



BUBT | BANGLADESH UNIVERSITY OF
BUSINESS AND TECHNOLOGY
Committed to Academic Excellence

Lab Report-04

Course Title : Numerical Analysis Lab

Course Code : CSE 224

Submitted By :

Name : Md. Mursalin Hasan Nirob

ID No : 21225103423

Intake : 49

Section : 10

Program : B.Sc. Engg. in CSE

Submitted To :

Name : Ridwan Mahbub

Lecturer

Department of : CSE

Bangladesh University of Business & Technology

Date of Submission:06/08/2023

Task- 1: Define a function for the following equation as a function file named **functionDemo.m**:

$$F(x) = x^3 - 23x^2 + 142x - 120$$

Solve:

```
function fx = functionDemo(x)
fx = x^3-23*x^2+142*x-120;
endfunction
```

Task 2: Complete the following code for bisection method:

```
function [root, iterations, errors] = bisection_method(guess1, guess2,
max_iterations, tolerance)
if functionDemo(guess1) * functionDemo(guess2) >= 0
error("The guess does not satisfy the required conditions");
end
iterations = 0;
errors = [];
prevGuess = 0;
while iterations < max_iterations
%
% Insert code for bisection method
%
end
endfunction
```

%Example of calling the function in the command window

```
[root, iterations, errors] = bisection_method(1
1, 20, 100, 1e-6);
```

Solve:

```
function [root,iterations,errors]
=nirob1(guess1,guess2,max_iterations,tolerance)

if functionDemo(guess1)*functionDemo(guess2) >=0

    error("The guess does not satisfy the required
conditions");

endif

iterations = 0;

errors = [];

preGuess = 0;
```

```

while iterations <max_iterations

    c=(guess1+guess2)/2;
    if functionDemo(guess1)*functionDemo(guess2)<0
        guess2=c;
    else
        guess1=c;
    endif
    if iterations >1
        error = abs((c-preGuess)/c);
        preGuess = c;
        errors = [errors,error];
    else
        preGuess = c;
    endif
    iterations++;
endwhile

root=c;
endfunction

```

Comand& Output:

```

=> [root, iterations,errors] = nirob1(11,20,100,1e-6)
root = 11.562
iterations = 100
errors =
Columns 28 through 36: 0.000000  0.000000  0.000000  0.000000  0.000000  0.000000
0.000000

    0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000
0.000000

```

Columns 37 through 45:

```
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000
```

Columns 46 through 54:

```
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000      0      0      0
```

Columns 55 through 63:

```
0      0      0      0      0      0      0      0      0
```

Columns 64 through 72:

```
0      0      0      0      0      0      0      0      0
```

Columns 73 through 81:

```
0      0      0      0      0      0      0      0      0
```

Columns 82 through 90:

```
0      0      0      0      0      0      0      0      0
```

Columns 91 through 98:

```
0      0      0      0      0      0      0      0
```

Task-3: Plot the relative errors in a graph with the number of iterations.

Solve:

```
>> X=[1:1:98]
```

```
X =
```

Columns 1 through 20:

```
1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
```

Columns 21 through 40:

```
21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
```

Columns 41 through 60:

41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60

Columns 61 through 80:

61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80

Columns 81 through 98:

81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98

```
>> plot(X,errors,'r:')
```

Graph:

