

GEBZE TECHNICAL UNIVERSITY

ELECTRONIC ENGINEERING MATH214 NUMERICAL ANALYSIS

2020 - 2021 FALL TERM

Project - 1

Student's

Number: 200102002087

Name and Surname : Alican Bayındır

Preparation date: 26.10.2020 -> 28.10.2020

Upload date: 28.10.2020



Used formula;

$$E(x) = \frac{1}{4\pi\epsilon_{0}} \left[\frac{9_{1}(x-x_{1})}{|x-x_{1}|^{3}} + \frac{9_{2}(x-x_{2})}{|x-x_{2}|^{3}} + \frac{9_{3}(x-x_{3})}{|x-x_{3}|^{3}} + \frac{9_{4}(x-x_{4})}{|x-x_{4}|^{3}} \right]$$

This formula is implemented into MATLAB as;

```
% Formula of electrical field

E = @(x) (1/4*pi*(1/36*pi)*10^-9)*((13*(x-(-7)/abs(x-(-7))^3)) ...

+ (9*(x-(-4)/abs(x-(-4))^3)) + (6*(x-11/abs(x-11)^3)) ...

+ (3*(x-14/abs(x-14)^3)));
```

However, this code works fine to handle the bisection method, Newton-Raphson method and the secant method as seen from figures 1, 2 and 3.

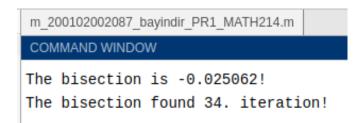


Figure 1: The Bisection Method s result.

The procedure succeded in iteration 7. and the value is: -0.025062 Figure 2: The Newton-Raphson Method's result.

The procedure succeded in iteration 6. and the value is: -0.025062 Figure 3: The Secant Method's result.



And the table's of all methods can be seen below as figures 4, 5, 6.

Iterations	Results	Errors
	3.5	3.5
1 2	0.25	-3.25
3	-1.375	-1.625
4		0.8125
5	-0.5625 -0.15625	0.40625
6	0.046875	0.20313
7		
	-0.054688	-0.10156
8	-0.0039063	0.050781
9	-0.029297	-0.025391
10	-0.016602	0.012695
11	-0.022949	-0.0063477
12	-0.026123	-0.0031738
13	-0.024536	0.0015869
14	-0.02533	-0.00079346
15	-0.024933	0.00039673
16	-0.025131	-0.00019836
17	-0.025032	9.9182e-05
18	-0.025082	-4.9591e-05
19	-0.025057	2.4796e-05
20	-0.025069	-1.2398e-05
21	-0.025063	6.1989e-06
22	-0.02506	3.0994e-06
23	-0.025061	-1.5497e-06
24	-0.025062	-7.7486e-07
25	-0.025062	3.8743e-07
26	-0.025062	1.9372e-07
27	-0.025062	9.6858e-08
28	-0.025062	-4.8429e-08
29	-0.025062	2.4214e-08
30	-0.025062	1.2107e-08
31	-0.025062	6.0536e-09
32	-0.025062	-3.0268e-09
33	-0.025062	1.5134e-09
34	-0.025062	7.567e-10

Figure 4: The Result-Error table of bisection method.



The procedure succeded in iteration 6. and the value is: -0.025062 my table =

6×3 table

Iterations	Results	Errors
1	3.5	3.5
2	-0.013754	3.5138
3	-0.024903	0.011149
4	-0.025059	0.00015611
5	-0.025062	2.2e-06
6	-0.025062	3.1006e-08

Figure 5: The Result-Error table of Newton-Raphson Method.

The procedure succeded in iteration 7. and the value is: -0.025062 my table =

7×3 table

Iterations	Results	Errors
1	0	-5.2289e-11
2	-0.58203	-0.58203
3	0.10329	0.68532
4	-0.025774	-0.12907
5	-0.025062	0.00071162
6	-0.025062	7.0738e-07
7	-0.025062	-4.1046e-12

Figure 6: The Result-Error table of The Secant Method.

As seen in figure 5 using Newton-Raphson's method is far more quick than others but the problem here is that my code does not work well. The reason for that might be the format of numbers. All 3 method's last iterations has some problems in results but error seems work well. The problem in Newton's method might be the derivative of the function but I tried different methods to differentiate it solve the problem. Unfortunately, none of them worked well and solved the problem. You can see my code in APPENDIX – A part and also the code will be delivered with this file to MS-Teams.



As specified in the project manual I changed tolerance value from 10^-10 to 10^-15 and the results can be seen below;

		_	18	-0.025082	-4.9591e-05
Iterations	Results	Errors	19	-0.025057	2.4796e-05
			20	-0.025069	-1.2398e-05
	2.5	2.5	21	-0.025063	6.1989e-06
1	3.5	3.5	22	-0.02506	3.0994e-06
2	0.25	-3.25	23	-0.025061	-1.5497e-06
3	-1.375	-1.625	24	-0.025062	-7.7486e-07
4	-0.5625	0.8125	25	-0.025062	3.8743e-07
5	-0.15625	0.40625	26	-0.025062	1.9372e-07
6	0.046875	0.20313	27	-0.025062	9.6858e-08
7	-0.054688	-0.10156	28	-0.025062	-4.8429e-08
8	-0.0039063	0.050781	29	-0.025062	2.4214e-08
9	-0.029297	-0.025391	30	-0.025062	1.2107e-08
10	-0.016602	0.012695	31	-0.025062	6.0536e-09
11	-0.022949	-0.0063477	32	-0.025062	-3.0268e-09
12	-0.026123	-0.0031738	33	-0.025062	1.5134e-09
13	-0.024536	0.0015869	34	-0.025062	7.567e-10
14	-0.02533	-0.00079346	35	-0.025062	-3.7835e-10
15	-0.024933	0.00039673	36	-0.025062	1.8917e-10
16	-0.025131	-0.00019836	37	-0.025062	-9.4587e-11
17	-0.025032	9.9182e-05	38	-0.025062	4.7294e-11
18	-0.025082	-4.9591e-05	39	-0.025062	2.3647e-11
19	-0.025057	2.4796e-05	40	-0.025062	1.1823e-11
20	-0.025069	-1.2398e-05	41	-0.025062	5.9117e-12
21	-0.025063	6.1989e-06	42	-0.025062	2.9559e-12
22	-0.02506	3.0994e-06	43	-0.025062	-1.4779e-12
23	-0.025061	-1.5497e-06	44	-0.025062	7.3896e-13
24	-0.025062	-7.7486e-07	45	-0.025062	3.6948e-13
25	-0.025062	3.8743e-07	46	-0.025062	1.8474e-13
26	-0.025062	1.9372e-07	47	-0.025062	9.2371e-14
27	-0.025062	9.6858e-08	48		
28	-0.025062	-4.8429e-08		-0.025062	4.6185e-14
29	-0.025062	2.4214e-08	49	-0.025062	-2.3093e-14
gure 7. The	result of bised	ction method	50	-0.025062	-1.1546e-14
•			51 52	-0.025062	-5.7732e-15
TETT LUTELATI	en tolerance value changed to 10^-15.			-0.025062	2.8866e-15

Figure 8: The continued result of bisection method when tolerance value changed to 10^-15

53

-0.025062

-0.025062

-1.4433e-15

-7.2164e-16



Iterations	Results	Errors
1	3.5	3.5
2	-0.013754	3.5138
3	-0.024903	0.011149
4	-0.025059	0.00015611
5	-0.025062	2.2e-06
6	-0.025062	3.1006e-08
7	-0.025062	4.37e-10
8	-0.025062	6.159e-12
9	-0.025062	8.6795e-14
10	-0.025062	1.2247e-15

Figure 9: The result of Newton's method when tolerance changed to 10^-15.

8×3 table

Iterations	Results	Errors
1	0	-5.2289e-12
2	-0.58203	-0.58203
3	0.10329	0.68532
4	-0.025774	-0.12907
5	-0.025062	0.00071162
6	-0.025062	7.0738e-07
7	-0.025062	-4.1046e-12
8	-0.025062	3.4694e-18

Figure 10: The result of Secant method when tolerance changed to 10^-15.

So, as seen in figures 7, 8, 9 and 10 we got closer to the result and error of the procedures became smaller. In the first result of secant method should not be 0 but there is an error here again. In order to approach to the result we need to make our tolerance smaller it causes increase of iterations but it makes our results more precise. Again, the problem in my results segment in table might be because of the format of the numbers. I could not change it because my MATLAB frequently crushes.



GEBZE TECHNICAL UNIVERSITY ELECTRONIC ENGINEERING DEPARTMENT 2020-2021 FALL TERM

MATH 214 - NUMERICAL ANALYSIS - PROJECT 1

As specified in the project manual I changed tolerance value from 10^-10 to 10^-16 and the results can be seen below;

Iterations	Results	Errors	26	-0.025062	1.9372e-07
			27	-0.025062	9.6858e-08
			28	-0.025062	-4.8429e-08
1	3.5	3.5	29	-0.025062	2.4214e-08
2	0.25	-3.25	30	-0.025062	1.2107e-08
3	-1.375	-1.625	31	-0.025062	6.0536e-09
4	-0.5625	0.8125	32	-0.025062	-3.0268e-09
5	-0.15625	0.40625	33	-0.025062	1.5134e-09
6	0.046875	0.20313	34	-0.025062	7.567e-10
7	-0.054688	-0.10156	35	-0.025062	-3.7835e-10
8	-0.0039063	0.050781	36	-0.025062	1.8917e-10
9	-0.029297	-0.025391	37	-0.025062	-9.4587e-11
10	-0.016602	0.012695	38	-0.025062	4.7294e-11
11	-0.022949	-0.0063477	39	-0.025062	2.3647e-11
12	-0.026123	-0.0031738	40	-0.025062	1.1823e-11
13	-0.024536	0.0015869	41	-0.025062	5.9117e-12
14	-0.02533	-0.00079346	42	-0.025062	2.9559e-12
15	-0.024933	0.00039673	43	-0.025062	-1.4779e-12
16	-0.025131	-0.00019836	44	-0.025062	7.3896e-13
17	-0.025032	9.9182e-05	45	-0.025062	3.6948e-13
18	-0.025082	-4.9591e-05	46	-0.025062	1.8474e-13
19	-0.025057	2.4796e-05	47	-0.025062	9.2371e-14
20	-0.025069	-1.2398e-05	48	-0.025062	4.6185e-14
21	-0.025063	6.1989e-06	49	-0.025062	-2.3093e-14
22	-0.02506	3.0994e-06	50	-0.025062	-1.1546e-14
23	-0.025061	-1.5497e-06	51	-0.025062	-5.7732e-15
24	-0.025062	-7.7486e-07	52	-0.025062	2.8866e-15
25	-0.025062	3.8743e-07	53	-0.025062	-1.4433e-15
26	-0.025062	1.9372e-07	54	-0.025062	-7.2164e-16
27	-0.025062	9.6858e-08	55	-0.025062	-3.6082e-16
28	-0.025062	-4.8429e-08	56	-0.025062	-1.8041e-16
29	-0.025062	2.4214e-08	57	-0.025062	9.0206e-17
30	-0.025062	1.2107e-08			
			Figure 11 · 1	The continued re	esult of hisecti

Figure 12: The result of bisection method when tolerance value changed to 10^-16

Figure 11: The continued result of bisection method when tolerance value changed to 10^-16



Iterations	Results	Errors
1	3.5	3.5
2	-0.013754	3.5138
3	-0.024903	0.011149
4	-0.025059	0.00015611
5	-0.025062	2.2e-06
6	-0.025062	3.1006e-08
7	-0.025062	4.37e-10
8	-0.025062	6.159e-12
9	-0.025062	8.6795e-14
10	-0.025062	1.2247e-15

Figure 13: The result of Newton's method when tolerance changed to 10^-16.

8×3 table

Iterations	Results	Errors
1	0	-5.2289e-12
2	-0.58203	-0.58203
3	0.10329	0.68532
4	-0.025774	-0.12907
5	-0.025062	0.00071162
6	-0.025062	7.0738e-07
7	-0.025062	-4.1046e-12
8	-0.025062	3.4694e-18

Figure 14: The result of Secant method when tolerance changed to 10^-16.

The bisection method did more calculation than before since we increased the tolerance of the algorithm to be more precise. On the other hand, other algorithms did not change. I guess my algorithm is not successfull to use. The algorithm failed to calculate the results. As seen above in figures 12 to 14. I used ONLINE MATLAB (can be seen in the pictures in APPENDIX part) to code these three algorithms since I was not able to use downloaded version of MATLAB.



APPENDIX

```
← → Table Tab
            m_200102002087_bayindir_PR1_MATH214.m × +
FOLDER
                 1
                                           %% Alican Bayındır started: 26.10.2020 - finished: 28.10.2020
                                           % Project - 1 MATH214
                  2
CURRENT
                                           % This script contains the Bisection method, the Newton's method
                                           % and the Secant method
                 5 -
                                           clear all
                 6 -
                                           clc
                                           % 1st Bisection method
                 7
                                           % Formula of electrical field
                                            E = @(x) (1/4*pi*(1/36*pi)*10^-9)*((13*(x-(-7)/abs(x-(-7))^3)) ...
             10 -
                                                         + (9*(x-(-4)/abs(x-(-4))^3)) + (6*(x-11/abs(x-11)^3)) ...
              11
                                                          + (3*(x-14/abs(x-14)^3)));
              12
              13
              14
                                           % The closed interval is [-3,10] so a = -3, b = 10;
                                           x_{lower} = -3;
             15 -
                                           x_upper = 10;
             16 -
             17
                                           %tolerance value that is given by project guide
             18
             19 -
                                            tol = 10^{-9};
             20
                                           %c provides us loop
              21
              22 -
                                           c = 0;
              23 -
                                           i = 1;
                                           FA = E(x_{lower});
              24 -
        COMMAND WINDOW
```

```
m_200102002087_bayindir_PR1_MATH214.m × +
CURRENT FOLDER
           i = 1;
   23 -
           FA = E(x_lower);
   24 -
   25
           % Bisection algorithm
   26
   27 - - while c < 1
               % Finding middle point and assigning it to a variable p
   28
   29 -
                p = x_{lower} + (x_{upper} - x_{lower}) / 2;
   30 -
               FP = E(p);
   31
               % The part below is to handle the table
   32
   33 -
               i_for_table = (1:i); % iterations
   34 -
                p for table(i) = p; % results
   35
                %if the result found
   36
   37 -
                if (FP == 0 || (x_upper - x_lower) / 2 < tol)</pre>
                    fprintf('The bisection is %f!\n', p);
   38 -
                    fprintf('The bisection found %d. iteration! \n', i);
   39 -
                    % To a proper result for errors, I needed to change the format
   40
                    format shortEng
   41 -
   42 -
                    o = 1;
   43
                    % calculating and creating errors' array
   44 - -
                    while o <= i
   45 -
                        if o == 1
                        error(1) = p_for_table(1) - E(0);
   46 -
   COMMAND WINDOW
```



```
m 200102002087 bayindir PR1 MATH214.m × +
CURRENT FOLDER
   44 - -
                   while o \le i
   45 -
                       if o == 1
   46 -
                        error(1) = p_for_table(1) - E(0);
   47 -
                        error(o) = p_for_table(o) - p_for_table(o - 1);
   48 -
   49 -
                        end
                    0 = 0 + 1;
   50 -
    51
                    end
    52
                   %Creating table
   53 -
                   my_table = table(i_for_table', p_for_table', error', ...
                        'VariableNames', {'Iterations' 'Results' 'Errors'})
   54
   55 -
                    return
   56 -
               end
   57
               %if not found initialize variables and add 1 to iteration.
               i = i + 1;
   58 -
   59
               %Initializing variables to find new middle point
   60
               if FA*FP > 0
   61 -
   62 -
                    x_lower = p;
    63 -
                    FA = FP;
   64 -
   65 -
                    x_upper = p;
   66 -
                end
   67 -
           end
   COMMAND WINDOW
```

```
m 200102002087 bayindir PR1 MATH214.m × +
          %% 2nd Newton-Raphson method
   68
           clear all
   69 -
   70 -
           clc
   71
   72
           % Formula of electrical field
           E = @(x) (1/4*pi*(1/36*pi)*10^-9)*((13*(x-(-7)/abs(x-(-7))^3)) \dots
   73 -
               + (9*(x-(-4)/abs(x-(-4))^3)) + (6*(x-11/abs(x-11)^3)) ...
   74
   75
               + (3*(x-14/abs(x-14)^3)));
   76
   77
           % Tolerance value
   78 -
           tol = 10^-9:
           c = 0:
   79 -
   80 -
           i = 1;
   81
           % p0 is the initial approximation value. I am going to choose
           % mid point of [-3,10] interval
   82
   83 -
           p0 = 3.5;
           % Derivated version of E
           E_{derivated} = (26249563097690031*sign(p0 - 11))/(19342813113834066795298816* ...
   85 -
               abs(p0\ -\ 11)^4)\ -\ (72385158845145237^*sign(p0\ +\ 7))/(38685626227668133590597632^*\ \dots
               abs(p0 + 7)^4) - (7158971753915463*sign(p0 + 4))/(9671406556917033397649408* \dots
   87
   88
               abs(p0 + 4)^4) + (16704267425802747*sign(p0 - 14))/(19342813113834066795298816* ...
   89
               abs(p0 - 14)^4) + 8219560161902939/38685626227668133590597632;
   90
           % Newton - Raphson method's algorithm
   91
   COMMAND WINDOW
```



```
← 

→ 

→ 

MATLAB Drive → 

And 

And 

MATLAB Drive → 

And 

MATLAB Drive → 

And 

MATLAB Drive → 

          m_200102002087_bayindir_PR1_MATH214.m × +
                                     % Newton - Raphson method's algorithm
           91
          92 - □ | while c < 1
          93 -
                                                   p = p0 - E(p0) / E_derivated;
          94
           95
                                                  % The part below is to handle the table
                                                   i_for_table = (1:i); %iteration' array
           96 -
                                                   p_for_table(i) = p0; %results' array
          97 -
           98
                                                                % if result found
           99 -
                                                                if abs(p - p0) < tol
        100 -
                                                                 fprintf('The procedure succeded in iteration %d. and the value is: %f', i, p);
                                                                % to show proper errors, we need to change the format
        101
        102 -
        103 -
                                                                0 = 1:
        104
                                                                % editing errors' array
        105 -
                                                                while o <= i
                                                                              if o == 1
        106 -
        107 -
                                                                               error(1) = abs(p_for_table(1) - E(0));
        108 -
                                                                              else
                                                                               error(o) = abs(p_for_table(o) - p_for_table(o - 1));
        109 -
        110 -
                                                                              end
        111 -
                                                                 o = o + 1;
        112 -
                                                                 end
                                                                 %create table
        113
        114
                                                                my_table = table(i_for_table', p_for_table', error', 'VariableNames', {'Iterations' 'Results' 'Errors'})
I▶ COMMAND WINDOW
```

```
← 

→ 

MATLAB Drive >

   m 200102002087 bayindir PR1 MATH214.m × +
FOLDER
  113
                    %create table
                    my_table = table(i_for_table', p_for_table', error', 'VariableNames'
  114 -
  115 -
                    return
  116 -
                    end
  117
                %initialize variables
  118 -
                i = i + 1;
  119 -
                p0 = p;
  120 -
           %% 3rd The Secant Method
  121
  122 -
           clear all
           clc
  123 -
  124
           % Formula of electrical field
  125
           E = @(x) (1/4*pi*(1/36*pi)*10^-9)*((13*(x-(-7)/abs(x-(-7))^3)) ...
  126
  127
                + (9*(x-(-4)/abs(x-(-4))^3)) + (6*(x-11/abs(x-11)^3)) ...
  128
                + (3*(x-14/abs(x-14)^3)));
  129
  130
           %tolerance value
  131 -
           tol = 10^-9;
  132 -
           c = 0;
           i = 2;
  133 -
  134
           % upper point and lower point
   135
           p0 = -3;
   136 -
   COMMAND WINDOW
```



```
← → Table Trive → MATLAB Drive 
          m 200102002087 bayindir PR1 MATH214.m × +
        134
                                            % upper point and lower point
         135
         136 -
                                            p0 = -3;
         137 -
                                            p1 = 10;
        138
                                            q0 = E(p0);
         139 -
         140 -
                                            q1 = E(p1);
         141
                                            %The secant method's algorithm
         142
         143 - - while c < 1
                                                            p = p1 - q1 * (p1-p0) / (q1-q0);
         144 -
         145
                                                            % The part below is to handle the table
          146
         147 -
                                                            i_for_table = (1:i); %iterations
         148 -
                                                            p_for_table(i) = p; %results
         149
         150
                                                            %if results found
                                                            if abs(p-p1) < tol
         151 -
                                                                             fprintf('The procedure succeded in iteration %d. and the value is: %f', i, p)
         152 -
                                                                             % to show proper errors we need to change the format
         153
                                                                             format shortEng
         154 -
                                                                             0 = 1;
          155
          156
                                                                             %editing errors' array
          157 - |
                                                                             while o <= i
          COMMAND WINDOW
```

```
← → Table Tab
            m 200102002087 bayindir PR1 MATH214.m × +
FOLDER
                                                                           fprintf('The procedure succeded in iteration %d. and the value is: %f', i, p)
         152 -
         153
                                                                           % to show proper errors we need to change the format
         154 -
                                                                          format shortEng
          155 -
                                                                           0 = 1;
          156
                                                                          %editing errors' array
                                                                          while o <= i
           157 - -
                                                                                          if o == 1
           158 -
           159 -
                                                                                          error(1) = p_for_table(1) - E(0);
           160 -
                                                                                          else
                                                                                          error(o) = p_for_table(o) - p_for_table(o - 1);
           161 -
           162 -
                                                                                          end
                                                                           o = o + 1;
          163 -
                                                                           end
           164 -
                                                                           %create table
           165
                                                                           my_table = table(i_for_table', p_for_table', error', ...
           166 -
                                                                                           'VariableNames',{'Iterations' 'Results' 'Errors'})
           167
          168 -
                                                                           return
           169 -
                                                           end
                                                           %initialize variables
           170
           171 -
                                                           i = i + 1;
           172 -
                                                           p0 = p1;
                                                           q0 = q1;
           173 -
          174 -
                                                           p1 = p;
          175 -
                                                           q1 = E(p);
I▶ COMMAND WINDOW
```