GEBZE TEKNİK ÜNİVERSİTESİ

NUMERİCAL METHODS/MATH214

Project 1

Deadline 28/10/2020 - 17.00

(a) Bisection Method,

(b) Newton's Method,

(c) Secant Method.

Muhammed Cemal Eryiğit
1801022024

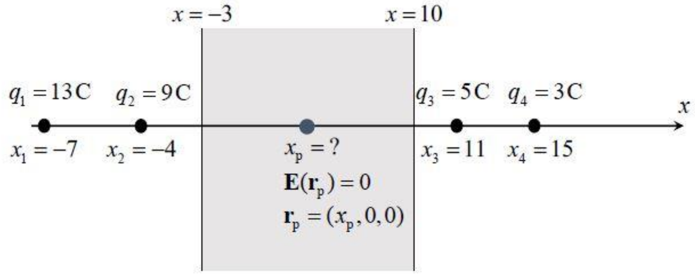


Figure 1: Illustration of the problem.

Question: Four charges are located to the x axis as shown in Figure 1. The coordinates of the charge locations are $(-7, 0, 0)$, $(-4, 0, 0)$, $(11, 0, 0)$, and $(15, 0, 0)$, and the amount of charges are $q_1 = 13$ C, $q_2 = 9$ C, $q_3 = 5$ C, and $q_4 = 3$ C, respectively. It can be predicted that the x component of the electric field intensity in the interval $x \in [-4, 11]$ will be in positive $\hat{\mathbf{x}}$ direction when x is close to $x = -4$, it will be zero at a certain point ($x = x_\mathrm{P}$), and then it will be in negative $\hat{\mathbf{x}}$ direction as x gets closer to 11.

Determine the location where the electric field intensity is zero at the interval $x \in [-3, 10]$ with a tolerance of $tol = 10^{-10}$, using

A- Bisection Method

Bisection method is a root finding method that applies to any continuous function for which one knows to values with opposite signs. The bisection method has a slow convergence rate, as it calculates the root value by dividing it by two continuously between certain ranges.But the bisection method always converges.

B- Newton’s Method

Newton's Method is a recursive algorithm for approximating the root of a differentiable function. To find the root value of the function with Newton's method, it iterates by determining the initial value, using this value and the value of the function at that point and the value of the derivative of the function at this point. Newton's method converges quickly but it is not converge when the initial value is outside the range of the function

C- Secant Method

The derivative required for Newton's method is difficult for some functions. For this reason, finite difference derivative formula is used instead of derivative in this method.

Bisection Method			
İteration	a1	b1	p
1	3.500000	10.000000	3.500000
2	3.500000	6.750000	6.750000
3	3.500000	5.125000	5.125000
4	4.312500	5.125000	4.312500
5	4.718750	5.125000	4.718750
6	4.921875	5.125000	4.921875
7	4.921875	5.023438	5.023438
8	4.972656	5.023438	4.972656
9	4.972656	4.998047	4.998047
10	4.972656	4.985352	4.985352
11	4.972656	4.979004	4.979004
12	4.975830	4.979004	4.975830
13	4.977417	4.979004	4.977417
14	4.977417	4.978210	4.978210
15	4.977417	4.977814	4.977814
16	4.977417	4.977615	4.977615
17	4.977516	4.977615	4.977516
18	4.977566	4.977615	4.977566
19	4.977591	4.977615	4.977591
20	4.977591	4.977603	4.977603
21	4.977591	4.977597	4.977597
22	4.977594	4.977597	4.977594
23	4.977595	4.977597	4.977595
24	4.977595	4.977596	4.977596
25	4.977595	4.977596	4.977596
26	4.977595	4.977595	4.977595
27	4.977595	4.977595	4.977595
28	4.977595	4.977595	4.977595
29	4.977595	4.977595	4.977595
30	4.977595	4.977595	4.977595
31	4.977595	4.977595	4.977595
32	4.977595	4.977595	4.977595
33	4.977595	4.977595	4.977595
34	4.977595	4.977595	4.977595
35	4.977595	4.977595	4.977595
36	4.977595	4.977595	4.977595

Newton Method	
İteration	p
1	4.271521
2	4.997890
3	4.977632
4	4.977595
5	4.977595

Secant Method	
İteration	P
1	5.072320
2	5.025790
3	4.978014
4	4.977597
5	4.977595

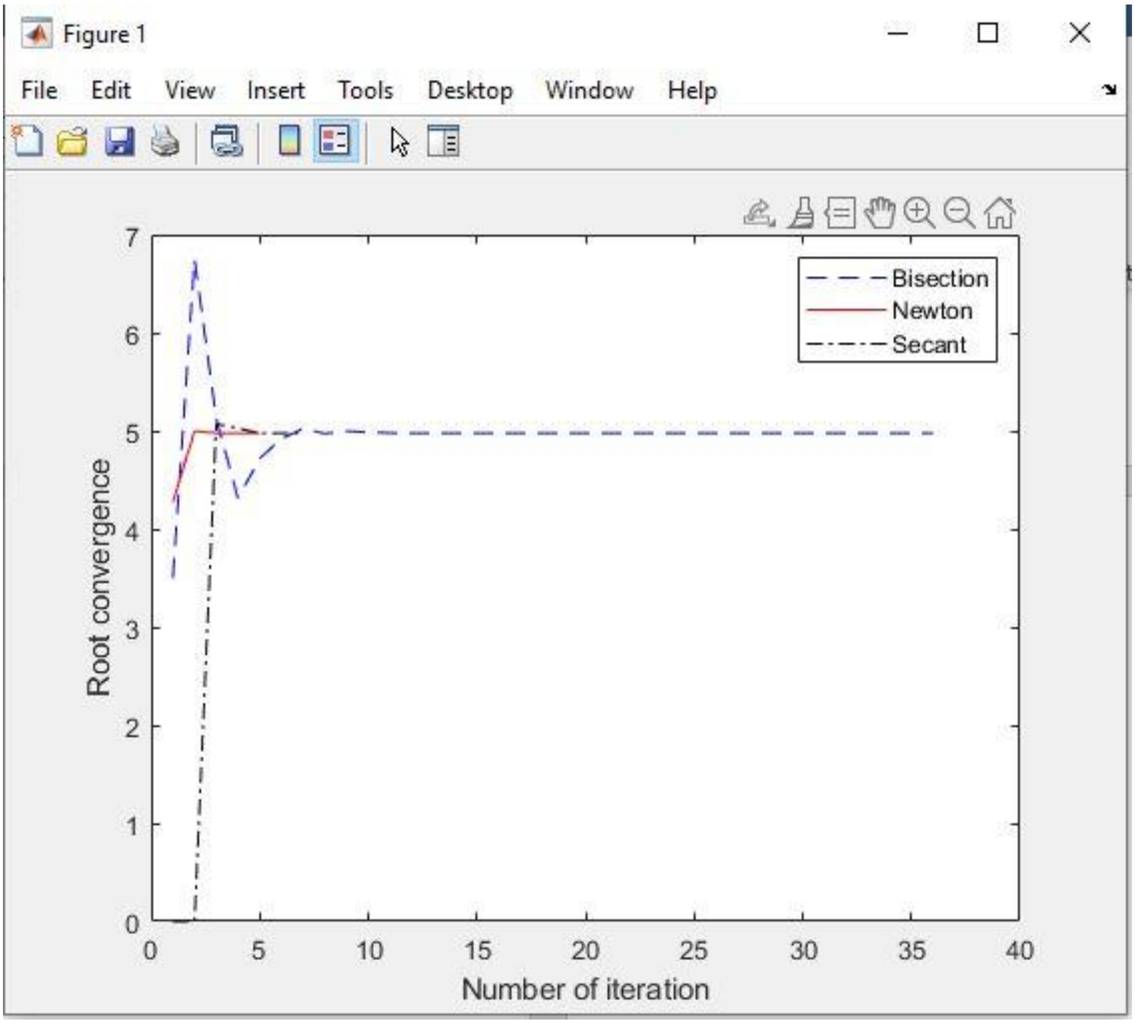
If the tolerance value is decreased, the number of iterations increases. We find the root value in more iterations. For example;

When the tolerance value is 10-15:

The bisection method reached its maximum number of iterations (50) and failed. Newton's method did not change because the error value was greater than the tolerance value. It could not approach the root value. Secant method proggressed 1 more iteration.

When the tolerance value is 10-16:

The bisection method reached its maximum number of iterations (50) and failed again. Newton method could not approach the root value with a low tolerance value and failed at 50 iterations. Secant method proggressed 1 more iteration and total of iteration number was 7.

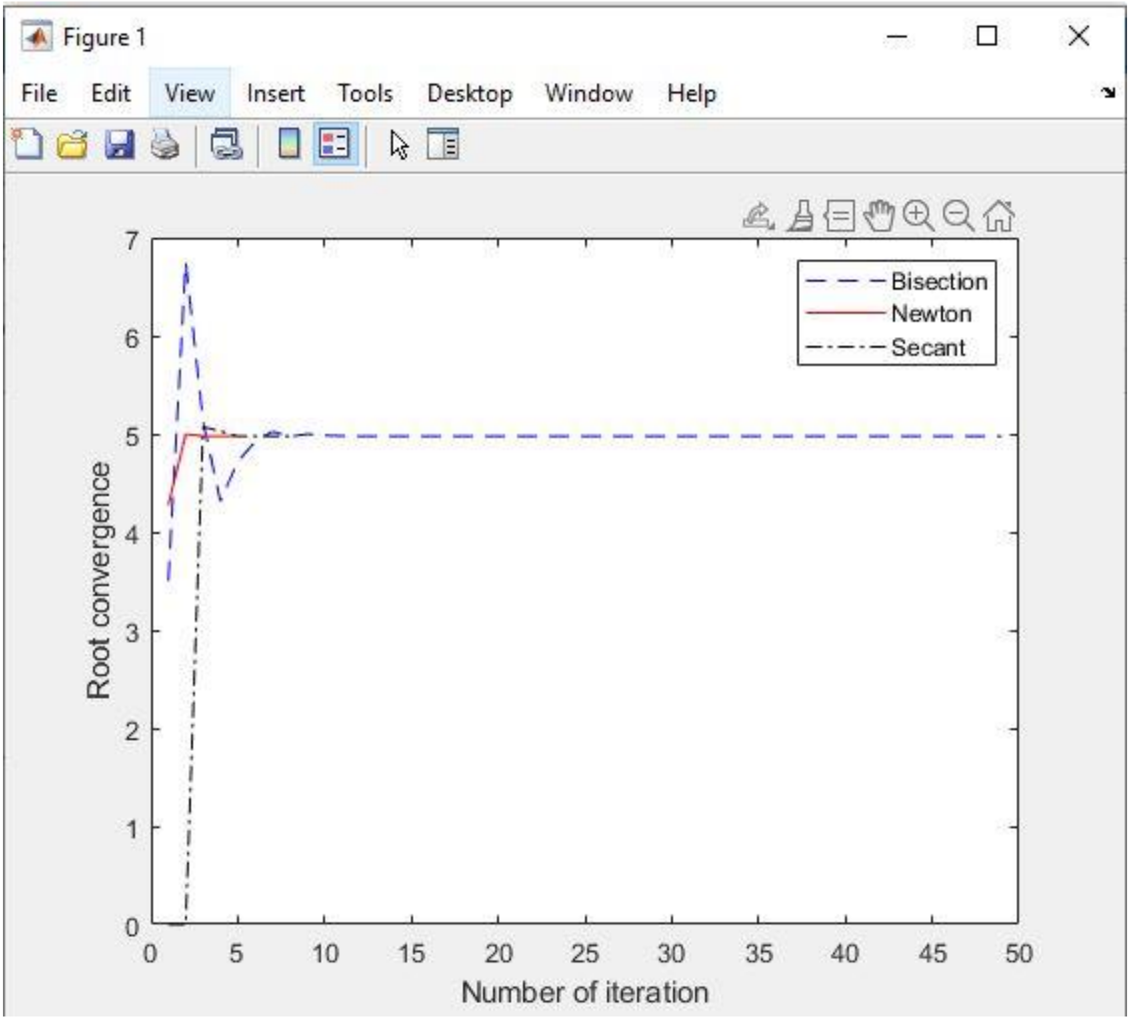


Tolerance value 10^{-10}

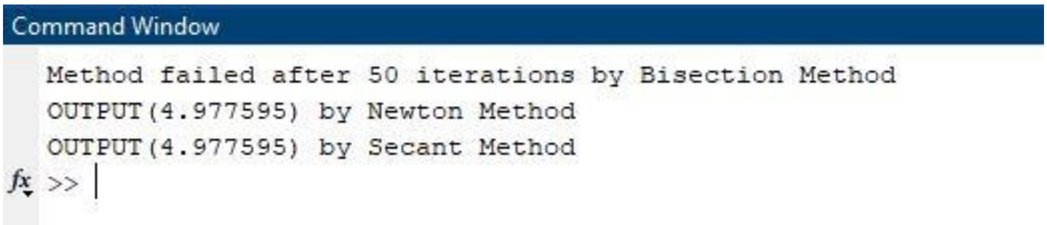
Command Window

```
OUTPUT(4.977595) by Bisection Method
OUTPUT(4.977595) by Newton Method
OUTPUT(4.977595) by Secant Method
fx >>
```

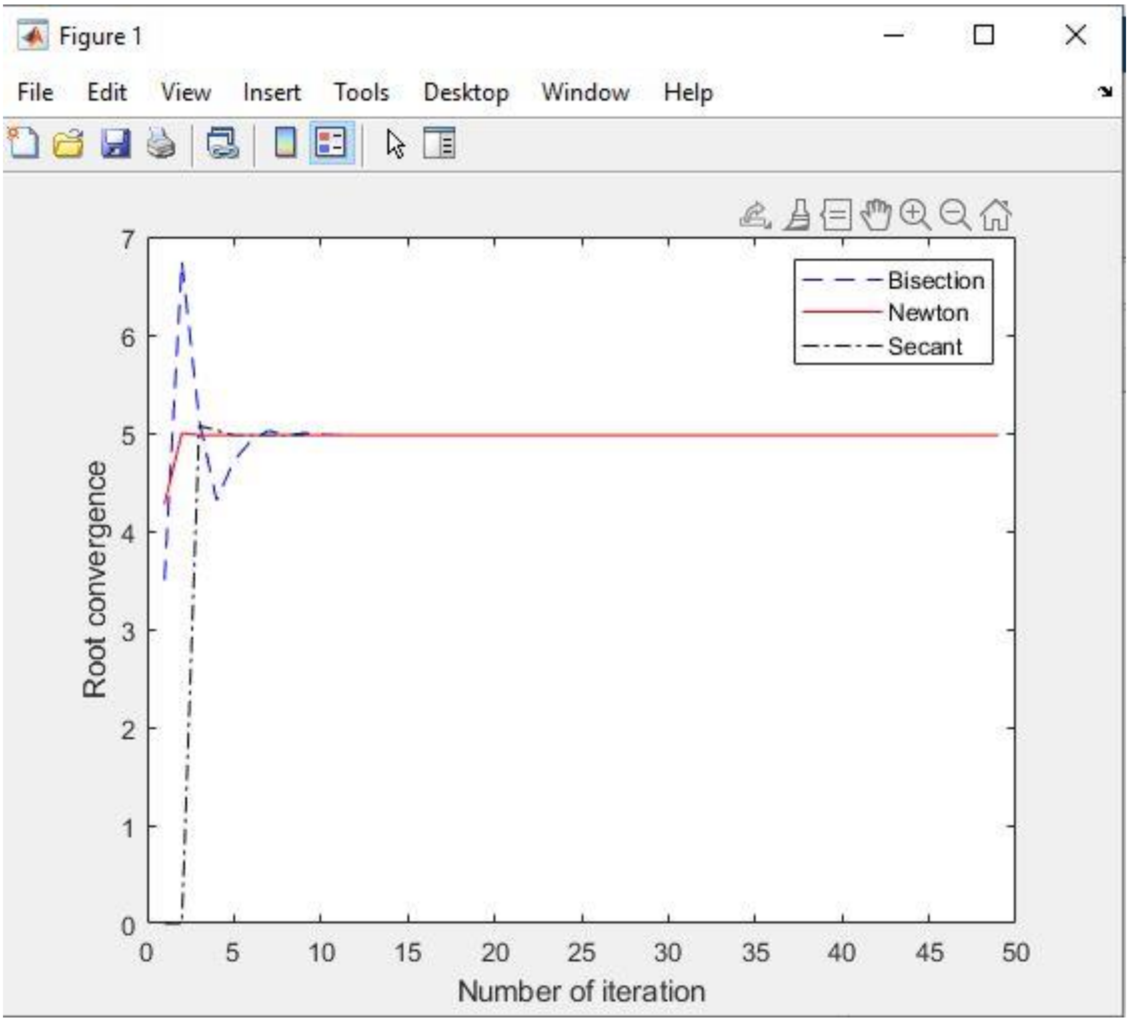
Tolerance value 10^{-10}



Tolerance value 10^{-15}



Tolerance value 10^{-15}



Tolerance value 10^{-16}

Command Window

```
Method failed after 50 iterations by Bisection Method
Method failed after 50 iterations by Newton Method
OUTPUT(4.977595) by Secant Method
>>
```

Tolerance value 10^{-16}

CODE

```
clc
clear
clear all
%The Bisection Method
a1 = -3; %endpoint
b1 = 10; %endpoint
tol = 1.e-10; %tolerance
n0=50; %max iteration
n=1;
e0 = 1/36*pi*10^-9;%definition
E= @(x) (1/4*pi*e0)*[(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)]; %Function
while n<n0
    p=(a1+b1)/2; %bisection method formula
    if(E(p)==0 ||(b1-a1)/2 < tol) %iteration limit point
        fprintf('OUTPUT(%f) by Bisection Method \n',p); %printing the
result to the screen
        break %break of the loop
    end
    root1(n)=p; %root array
    n=n+1; %increase iteration
    if (E(p)*E(b1))< 0
        a1=p; %assigment
    else
        b1=p; %assigment
    end
   if(n==50)
      fprintf('Method failed after %d iterations by Bisection Method
\n',n0) %error massage
   end
end
plot(root1,'b--'); %drawing graphics
    xlabel('Number of iteration')
    ylabel('Root convergence')
    hold on

clear
clear all
%The Newton Method
tol = 1.e-10; %tolerance
n0=50; %max iteration
e0 = 1/36*pi*10^-9;%definition
n=1;
p0=2; %initial approximation
E= @(x) (1/4*pi*e0)*[(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)]; %Function
dE= @(x) (1/4*pi*e0)*[-(26/(abs(x+7)*(x+7)^2))-(18/(abs(x+4)*(x+4)^2))-
(12/(abs(x-11)*(x-11)^2))-(6/(abs(x-14)*(x-14)^2))]; %derivative of the
function
while(n<50)
p=p0-E(p0)/dE(p0); %newton method formula
if(abs(p-p0)<tol)%iteration limit point
    fprintf('OUTPUT(%f) by Newton Method\n',p); %printing the result to the
screen
    break
end
root2(n)=p; %root array
n=n+1; %increase iteration
p0=p; %assigment
```

```
if(n==50)
      fprintf('Method failed after %d iterations by Newton Method\n',n0)
%error massage
end

end
plot(root2,'r-'); %drawing graphics
    hold on

clear
clear all
%The Secant Method
tol = 1.e-10; %tolerance
n0=50; %max iteration
e0 = 1/36*pi*10^-9;%definition
n=2;
p0=-1; %initial approximation
p1=8; %initial approximation
E= @(x) (1/4*pi*e0)*[(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)]; %Function
q0=E(p0); %assigment
q1=E(p1); %assigment
while(n<50)
p=p1-q1*(p1-p0)/(q1-q0); %secant method formula
if(abs(p-p1)<tol) %iteration limit point
    fprintf('OUTPUT(%f) by Secant Method\n',p);
    break
end
n=n+1; %increase iteration
q0=q1; %assigment
p0=p1; %assigment
p1=p; %assigment
q1=E(p); %assigment
root3(n)=p; %root array
if(n==50)
      fprintf('Method failed after %d iterations by Secant Method',n0)
%error massage
end

end
plot(root3,'k-.'); %drawing graphics
legend('Bisection','Newton','Secant')
```