

EXPERIMENT-2

Q1

3. Students are required to write both the program and implement it on the following examples.

Take tolerance value $\epsilon = 0.00001$

- Compute $\sqrt{17}$.
- The root of $\exp(-x)(x^2 + 5x + 2) + 1 = 0$. Take initial guess -1.0 .
- Find a non-zero solution of $x = 2\sin x$. (Apply IVT to find an initial guess)

Ans1(i) by Newton Method

SUCCESSFUL:

The image shows a MATLAB script in the Editor and the Command Window. The script implements the Newton-Raphson method to find the root of the function $f(x) = \exp(-x)(x^2 + 5x + 2) + 1$. It uses the Intermediate Value Theorem (IVT) to find an initial guess and then iterates until the tolerance is reached.

```

1 %newton method
2 clc
3 clear
4 f=@(x) x^2-17;
5 fd=@(x) 2*x;
6 tol=0.00001;
7 h=1;
8 flag=0;
9 %ivt
10 for i=-10:h:10
11     if (f(i)*f(i+h)<0)
12         a=i;
13         b=i+h;
14     end
15 end
16
17 N=input('Enter number of iterations \n');
18 x0=(a+b)/2;
19 fprintf('Initial Guess is %f IVT\n',x0);
20 %Newton Formula
21 i=1;
22 while i<=N
23     x1= x0 - (f(x0)/fd(x0));
24     if abs(x1-x0)<=tol || abs(x1-x0)/abs(x1)<=tol
25         flag=1;
26         break
27     else
28         x0=x1;
29         i=i+1;
30     end
31 end

```

The Command Window shows the following output:

```

Enter number of iterations
100
Initial Guess is 4.500000 IVT
successfull
The root is 4.123136
Number of iterations is 3
fx >>

```

The Workspace window shows the following variables and their values:

Name	Value
a	4
b	5
f	@(x)x^2-17
fd	@(x)2*x
flag	1
h	1
i	3
N	100
tol	1.0000e-05
x0	4.1231
x1	4.1231

UNSUCCESSFUL:

The screenshot shows the MATLAB IDE with a script titled 'Newton.m' in the editor. The script implements a Newton's method for finding roots of a function. It starts by defining a function $f(x) = x^2 - 17$ and its derivative $fd(x) = 2x$. It then sets a tolerance $tol = 0.00001$ and an initial guess $x0 = 4$. The script enters a while loop that iterates up to $N = 2$ times. In each iteration, it calculates the next guess $x1 = x0 - f(x0)/fd(x0)$ and checks if the absolute value of the function at $x1$ is less than the tolerance. If not, it updates $x0$ to $x1$ and increments the iteration counter. After 2 iterations, the script prints 'unsuccessful' and 'Failed to achieve tolerance of 0.000010 after 2 iterations'.

The Command Window shows the execution output:

```
Enter number of iterations
2
Initial Guess is 4.500000 IVT
unsuccessful
Failed to achieve tolerance of 0.000010 after 2 iterations
```

The Workspace window shows the following variables and their values:

Name	Value
a	4
b	5
f	@(x)x^2-17
fd	@(x)2*x
flag	0
h	1
i	3
N	2
tol	1.0000e-05
x0	4.1231
x1	4.1231

CODE:

```
%newton method
```

```
clc
```

```
clear
```

```
f=@(x) x^2-17;
```

```
fd=@(x)2*x;
```

```
tol=0.00001;
```

```
h=1;
```

```
flag=0;
```

```
%ivt
```

```
for i=-10:h:10
```

```
    if(f(i)*f(i+h)<0)
```

```
a=i;
b=i+h;
end
end

N=input('Enter number of iterations \n');
x0=(a+b)/2;
fprintf('Initial Guess is %f IVT\n',x0);
%Newton Formula
i=1;
while i<=N

    x1= x0 - (f(x0)/fd(x0));
    if abs(x1-x0)<=tol || abs(x1-x0)/abs(x1)<=tol
        flag=1;
        break

    else
        x0=x1;
        i=i+1;
    end

end

if flag~=1
    fprintf('unsuccessfull\nFailed to achieve tolerance of %f after %d iterations\n',tol,N);
else
    fprintf('successfull\n');
```

```
fprintf('The root is %f',x0);
fprintf('\nNumber of iterations is %d\n',i);
end
```

Ans1(i) by Secant Method

SUCCESSFUL:

The screenshot displays the MATLAB IDE with a script editor on the left and a Command Window on the right. The script implements the Secant Method for finding the root of the function $f(x) = x^2 - 17$. It includes a for-loop for initial bracketing and a while-loop for the iterative Secant method. The Command Window shows the user input for 100 iterations and the resulting root value of 4.123107 after 5 iterations.

```

1  %secant method
2  clc
3  clear
4  f=@(x) x^2-17;
5  fd=@(x) 2*x;
6  tol=0.00001;
7  h=1;
8  flag=0;
9  %ivt
10 for i=-10:h:10
11     if (f(i)*f(i+h)<0)
12         a=i;
13         b=i+h;
14     end
15 end
16
17 N=input('Enter number of iterations \n');
18 x0=a;
19 x1=b;
20 fprintf('Initial Guess\nroot should be in between %f and %f\n',x0,x1);
21 %secant Formula
22 i=1;
23 while i<=N
24
25     x2= x1 - (f(x1)*(x1-x0))/(f(x1)-f(x0));
26     if abs(x1-x0)<=tol || abs(x1-x0)/abs(x1)<=tol
27         flag=1;
28         break
29     else
30
31         x0=x1;

```

Command Window

```

Enter number of iterations
100
Initial Guess
root should be in between 4.000000 and 5.000000
successfull
The root is 4.123107
Number of iterations is 5
fx >>

```

Workspace

Name	Value
a	4
b	5
f	@(x)x^2-17
fd	@(x)2*x
flag	1
h	1
i	5
N	100
tol	1.0000e-05
x0	4.1231
x1	4.1231
x2	4.1231

UNSUCCESSFUL:

The screenshot shows the MATLAB IDE with a script file named 'Exp2' in the 'Numerical Analysis' folder. The script implements the secant method to find the root of the function $f(x) = x^2 - 17$. The user has entered 2 iterations. The Command Window shows the following output:

```

Enter number of iterations
2
Initial Guess
root should be in between 4.000000 and 5.000000
unsuccessfull
Failed to achieve tolerance of 0.000010 after 2 iterations
fx >>

```

The Workspace window shows the following variables and their values:

Name	Value
a	4
b	5
f	@(x)x^2-17
fd	@(x)2*x
flag	0
h	1
i	3
N	2
tol	1.0000e-05
x0	4.1111
x1	4.1220
x2	4.1220

The script code is as follows:

```

12 a=1;
13 b=i+h;
14 end
15 end
16
17 N=input('Enter number of iterations \n');
18 x0=a;
19 x1=b;
20 fprintf('Initial Guess\nroot should be in between %f and %f\n',x0,x1);
21 %secant Formula
22 i=1;
23 while i<=N
24
25     x2= x1 - (f(x1)*(x1-x0))/(f(x1)-f(x0));
26     if abs(x1-x0)<=tol || abs(x1-x0)/abs(x1)<=tol
27         flag=1;
28         break
29
30     else
31         x0=x1;
32         x1=x2;
33         i=i+1;
34     end
35 end
36
37 if flag~=1
38     fprintf('unsuccessfull\nFailed to achieve tolerance of %f after %d iterations\n',tol,N);
39 else
40     fprintf('successfull\n');
41     fprintf('The root is %f',x0);
42     fprintf('\nNumber of iterations is %d\n',i);
43 end

```

CODE:

%secant method

clc

clear

f=@(x) x^2-17;

fd=@(x)2*x;

tol=0.00001;

h=1;

flag=0;

%ivt

for i=-10:h:10

```
if(f(i)*f(i+h)<0)
    a=i;
    b=i+h;
end
end

N=input('Enter number of iterations \n');
x0=a;
x1=b;
fprintf('Initial Guess\nroot should be in between %f and %f\n',x0,x1);
%secant Formula
i=1;
while i<=N

    x2= x1 - (f(x1)*(x1-x0))/(f(x1)-f(x0));
    if abs(x1-x0)<=tol || abs(x1-x0)/abs(x1)<=tol
        flag=1;
        break

    else
        x0=x1;
        x1=x2;
        i=i+1;
    end

end

end

if flag~=1
```

```

fprintf('unsuccessfull\nFailed to achieve tolerance of %f after %d iterations\n',tol,N);
else
    fprintf('successfull\n');
fprintf('The root is %f',x0);
fprintf('\nNumber of iterations is %d\n',i);
end

```

Ans1(ii) by Secant Method

SUCCESSFUL:

The screenshot displays the MATLAB IDE with a script titled 'sem\Numerical Analysis\Exp2\Q1iSecant.m'. The script implements the Secant Method to find the root of the function $f(x) = \exp(-x) * (x^2 + 5x + 2) + 1$. The user has entered 100 iterations and an initial guess of -1. The method successfully converges to a root of approximately -0.579167 after 7 iterations.

```

1 %secant method
2 clc
3 clear
4 f=@(x) (exp(-x)*(x^2+5*x+2))+1;
5 tol=0.00001;
6 h=1;
7 flag=0;
8 %ivt
9 for i=-10:h:10
10     if (f(i)*f(i+h)<0)
11         a=i;
12         b=i+h;
13     end
14 end
15
16 N=input('Enter number of iterations \n');
17 x0=a;
18 x1=b;
19 fprintf('Initial Guess\nroot should be in between %f and %f\n',x0,x1);
20 %secant Formula
21 i=1;
22 while i<=N
23     x2= x1 - (f(x1)*(x1-x0))/(f(x1)-f(x0));
24     if abs(x1-x0)<=tol || abs(x1-x0)/abs(x1)<=tol
25         flag=1;
26         break
27     else
28         x0=x1;
29         x1=x2;
30     end
31 end

```

Command Window Output:

```

Enter number of iterations
100
Initial Guess
root should be in between -1.000000 and 0.000000
successfull
The root is -0.579167
Number of iterations is 7
>>

```

Workspace Variables:

Name	Value
a	-1
b	0
f	@(x)(exp(-x)*(x^2+5*x+2))+1
flag	1
h	1
i	7
N	100
tol	1.0000e-05
x0	-0.5792
x1	-0.5792
x2	-0.5792

UNSUCCESSFUL:

The screenshot shows the MATLAB IDE with a script titled 'Exp2' in the editor. The script implements the secant method to find roots of the function $f(x) = \exp(-x)(x^2 + 5x + 2) + 1$. The user has input 4 iterations, and the Command Window shows the following output:

```

Enter number of iterations
4
Initial Guess
root should be in between -1.000000 and 0.000000
unsuccessfull
Failed to achieve tolerance of 0.000010 after 4 iterations
fx >>

```

The Workspace window shows the following variables and their values:

Name	Value
a	-1
b	0
f	@(x)(exp(-x)*(x^2+5*x+2))+1
flag	0
h	1
i	5
N	4
tol	1.0000e-05
x0	-0.5664
x1	-0.5783
x2	-0.5783

CODE:

```
%secant method
```

```
clc
```

```
clear
```

```
f=@(x)(exp(-x)*(x^2+5*x+2))+1;
```

```
tol=0.00001;
```

```
h=1;
```

```
flag=0;
```

```
%ivt
```

```
for i=-10:h:10
```

```
    if(f(i)*f(i+h)<0)
```



```
a=i;
b=i+h;
end
end

N=input('Enter number of iterations \n');
x0=a;
x1=b;
fprintf('Initial Guess\nroot should be in between %f and %f\n',x0,x1);
%secant Formula
i=1;
while i<=N

    x2= x1 - (f(x1)*(x1-x0))/(f(x1)-f(x0));
    if abs(x1-x0)<=tol || abs(x1-x0)/abs(x1)<=tol
        flag=1;
        break

    else
        x0=x1;
        x1=x2;
        i=i+1;
    end

end

if flag~=1
    fprintf('unsuccessfull\nFailed to achieve tolerance of %f after %d iterations\n',tol,N);
```

else

```
fprintf('successfull\n');
```

```
fprintf('The root is %f',x0);
```

```
fprintf('\nNumber of iterations is %d\n',i);
```

end

Ans1(ii) by Newton Method

SUCCESSFUL:

The image shows a MATLAB script window and a Command Window. The script implements the Newton-Raphson method to find the root of the function $f(x) = \exp(-x) * (x^2 + 5*x + 2) + 1$. The initial guess is $x_0 = -1$ and the tolerance is $1.0000e-05$. The script uses a while loop to iterate until the root is found within the specified tolerance. The Command Window shows the input of 100 iterations, the initial guess, and the successful finding of the root at -0.579159 after 5 iterations.

```

1 %newton method
2 clc
3 clear
4 f=@(x) (exp(-x)*(x^2+5*x+2))+1;
5 fd=@(x) (-exp(-x)*(x^2+5*x+2))+(exp(-x)*(2*x+5));
6 tol=1.00001;
7 h=1;
8 flag=0;
9 %ivt
10
11 N=input('Enter number of iterations \n');
12 x0=-1;
13 fprintf('Initial Guess is %f as given in question\n',x0);
14 %Newton Formula
15 i=1;
16 while i<=N
17     x1= x0 - (f(x0)/fd(x0));
18     if abs(x1-x0)<=tol || abs(x1-x0)/abs(x1)<=tol
19         flag=1;
20         break
21     else
22         x0=x1;
23         i=i+1;
24     end
25 end
26 if flag~=1
27     fprintf('unsuccessfull\nFailed to achieve tolerance of %f after %d :',tol,N);
28 else
29     fprintf('successfull\nThe root is %f\nNumber of iterations is %d\n',x0,i);
30 end

```

Command Window:

```

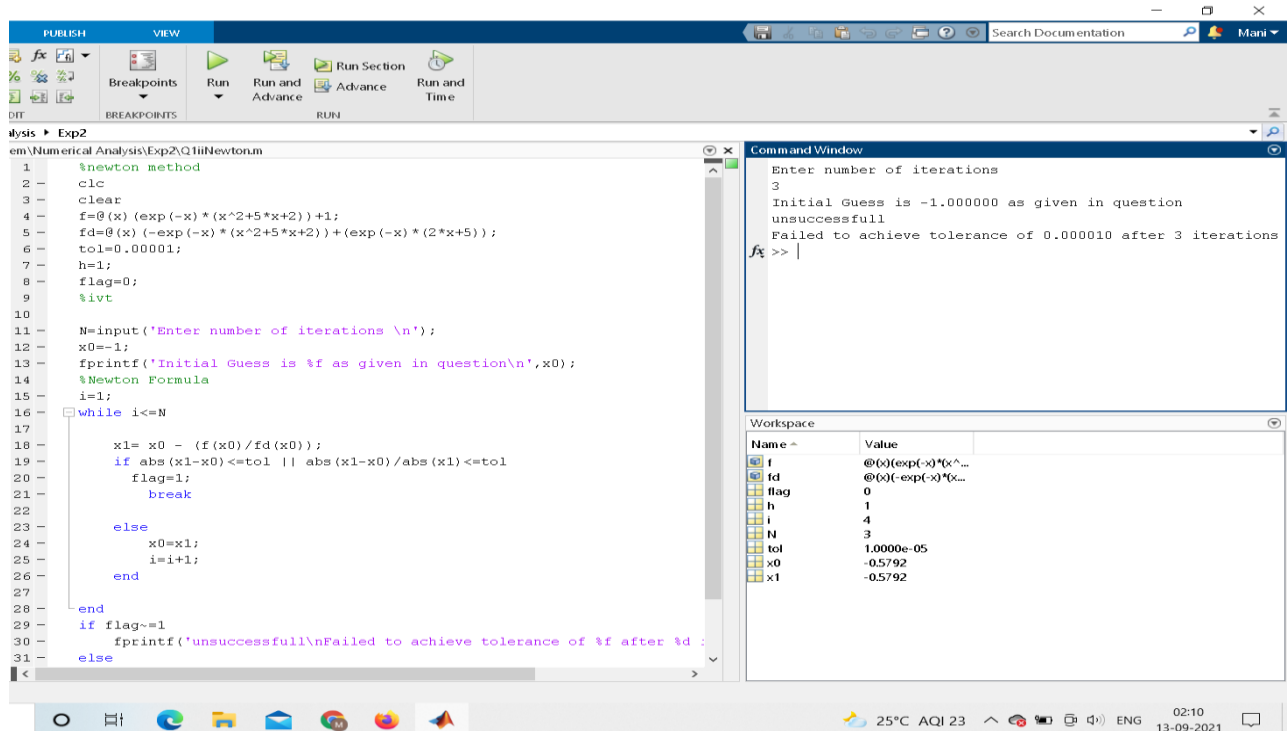
Enter number of iterations
100
Initial Guess is -1.000000 as given in question
successfull
The root is -0.579159
Number of iterations is 5
fx >>

```

Workspace:

Name	Value
f	@(x)(exp(-x)*(x^2+5*x+2))+1
fd	@(x)(-exp(-x)*(x^2+5*x+2))+(exp(-x)*(2*x+5))
flag	1
h	1
i	5
N	100
tol	1.0000e-05
x0	-0.5792
x1	-0.5792

UNSUCCESSFUL:



CODE:

```
%newton method
```

```
clc
```

```
clear
```

```
f=@(x)(exp(-x)*(x^2+5*x+2))+1;
```

```
fd=@(x)(-exp(-x)*(x^2+5*x+2))+(exp(-x)*(2*x+5));
```

```
tol=0.00001;
```

```
h=1;
```

```
flag=0;
```

```
%ivt
```

```
N=input('Enter number of iterations \n');
```

```
x0=-1;
```

```
fprintf('Initial Guess is %f as given in question\n',x0);
```

```
%Newton Formula
```

```
i=1;
while i<=N

    x1= x0 - (f(x0)/fd(x0));
    if abs(x1-x0)<=tol || abs(x1-x0)/abs(x1)<=tol
        flag=1;
        break

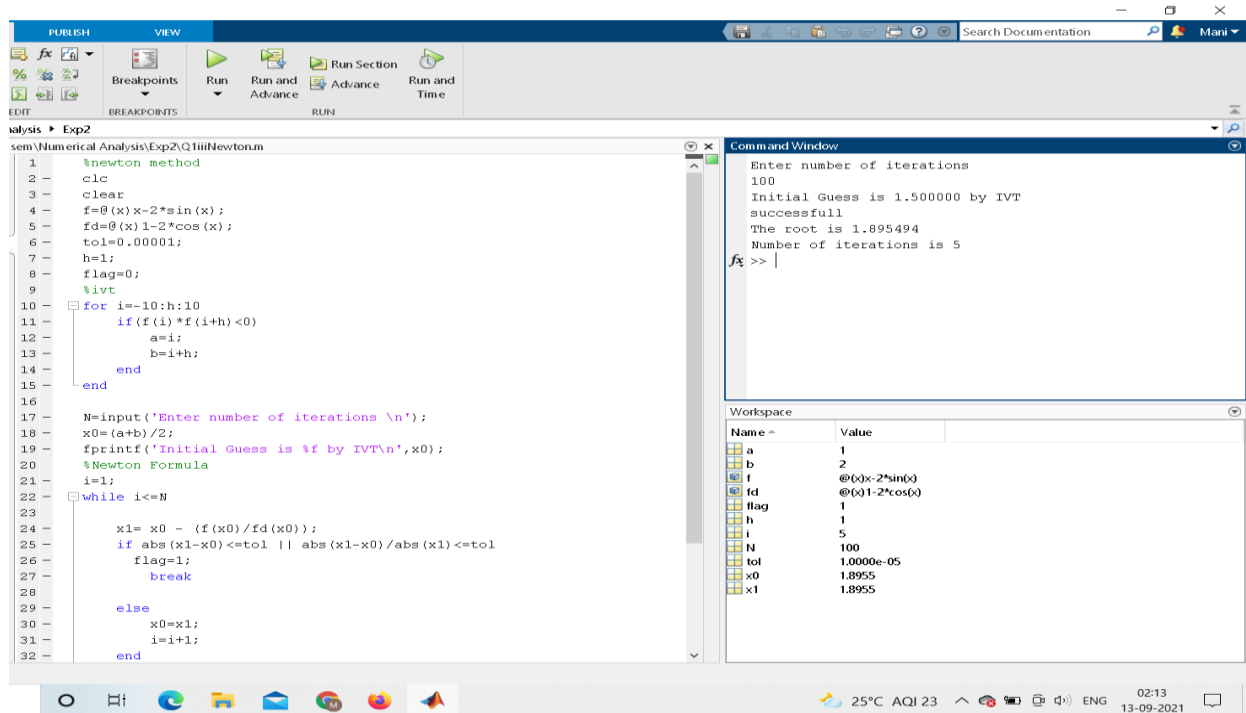
    else
        x0=x1;
        i=i+1;
    end

end

if flag~=1
    fprintf('unsuccessfull\nFailed to achieve tolerance of %f after %d iterations\n',tol,N);
else
    fprintf('successfull\n');
    fprintf('The root is %f',x0);
    fprintf("\nNumber of iterations is %d\n",i);
end
```

Ans1(iii) by Newton Method

SUCCESSFUL:



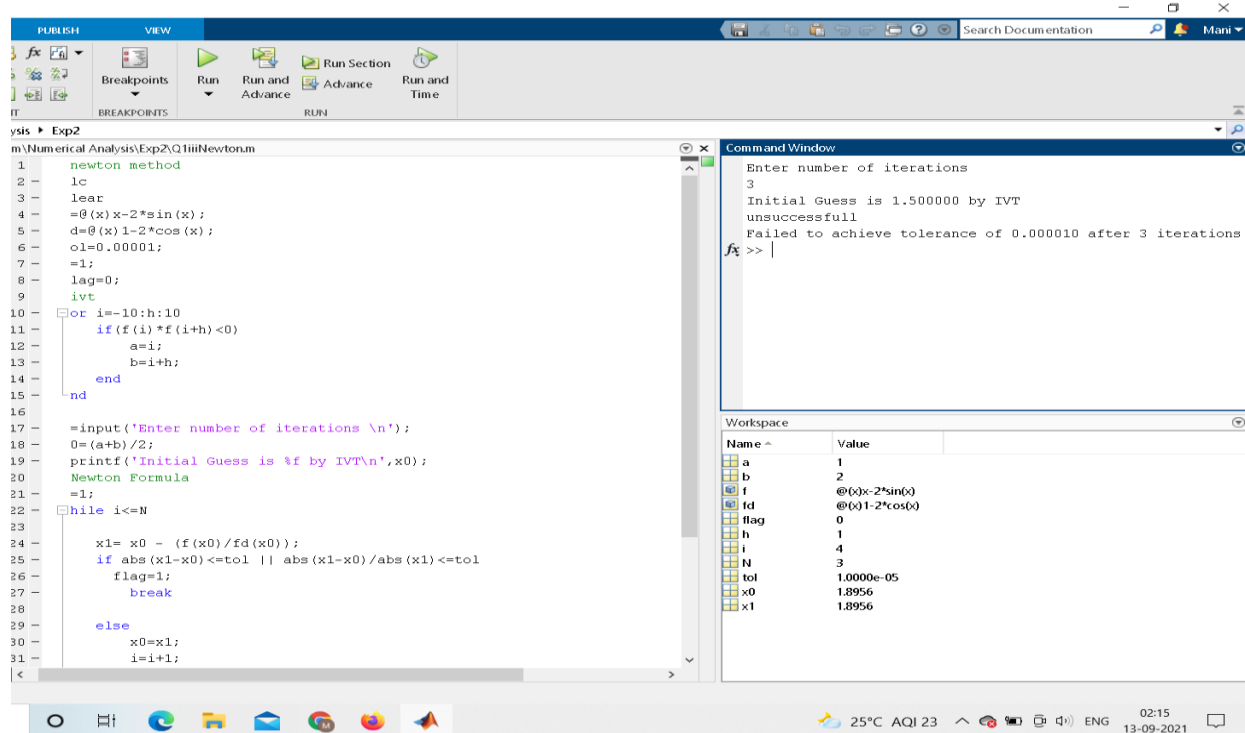
```
sem\Numerical Analysis\Exp2\Q1iiiNewton.m
1 %newton method
2 clc
3 clear
4 f=@(x)x-2*sin(x);
5 fd=@(x)1-2*cos(x);
6 tol=0.00001;
7 h=1;
8 flag=0;
9 %ivt
10 for i=-10:h:10
11     if (f(i)*f(i+h)<0)
12         a=i;
13         b=i+h;
14     end
15 end
16
17 N=input('Enter number of iterations \n');
18 x0=(a+b)/2;
19 fprintf('Initial Guess is %f by IVT\n',x0);
20 %Newton Formula
21 i=1;
22 while i<=N
23     x1= x0 - (f(x0)/fd(x0));
24     if abs(x1-x0)<=tol || abs(x1-x0)/abs(x1)<=tol
25         flag=1;
26         break
27     else
28         x0=x1;
29         i=i+1;
30     end
31 end
32
```

Command Window

```
Enter number of iterations
100
Initial Guess is 1.500000 by IVT
successfull
The root is 1.895494
Number of iterations is 5
fx >>
```

Name	Value
a	1
b	2
f	@(x)x-2*sin(x)
fd	@(x)1-2*cos(x)
flag	1
h	1
i	5
N	100
tol	1.0000e-05
x0	1.8955
x1	1.8955

UNSUCCESSFUL:



CODE:

```
%newton method
```

```
clc
```

```
clear
```

```
f=@(x)x-2*sin(x);
```

```
fd=@(x)1-2*cos(x);
```

```
tol=0.00001;
```

```
h=1;
```

```
flag=0;
```

```
%ivt
```

```
for i=-10:h:10
```

```
    if(f(i)*f(i+h)<0)
```

```
        a=i;
```

```
        b=i+h;
```

```
    end
```

end

N=input('Enter number of iterations \n');

x0=(a+b)/2;

fprintf('Initial Guess is %f by IVT\n',x0);

%Newton Formula

i=1;

while i<=N

 x1= x0 - (f(x0)/fd(x0));

 if abs(x1-x0)<=tol || abs(x1-x0)/abs(x1)<=tol

 flag=1;

 break

 else

 x0=x1;

 i=i+1;

 end

end

if flag~=1

 fprintf('unsuccessfull\nFailed to achieve tolerance of %f after %d iterations\n',tol,N);

else

 fprintf('successfull\n');

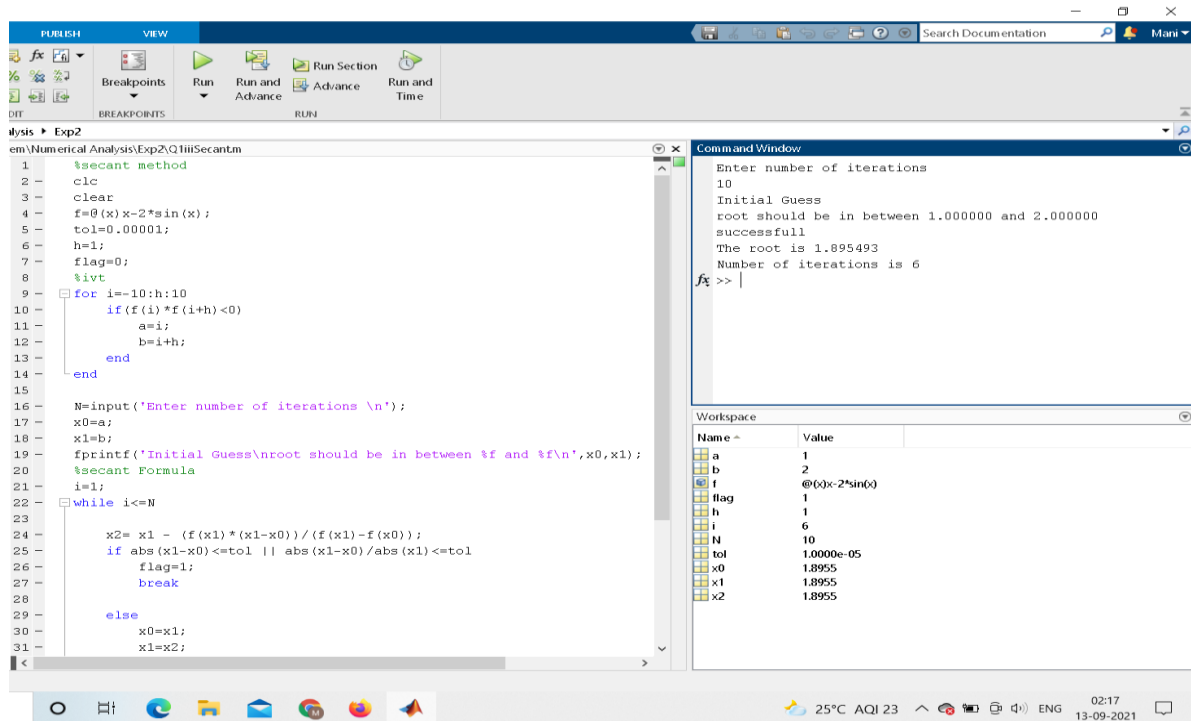
 fprintf('The root is %f',x0);

 fprintf('\nNumber of iterations is %d\n',i);

end

Ans1(iii) by Secant Method

SUCCESSFUL:



```
1 %secant method
2 clc
3 clear
4 f=@(x) x-2*sin(x);
5 tol=0.00001;
6 h=1;
7 flag=0;
8 %ivt
9 for i=-10:h:10
10     if (f(i)*f(i+h)<0)
11         a=i;
12         b=i+h;
13     end
14 end
15
16 N=input('Enter number of iterations \n');
17 x0=a;
18 x1=b;
19 fprintf('Initial Guess\nroot should be in between %f and %f\n',x0,x1);
20 %secant Formula
21 i=1;
22 while i<=N
23
24     x2= x1 - (f(x1)*(x1-x0))/(f(x1)-f(x0));
25     if abs(x1-x0)<=tol || abs(x1-x0)/abs(x1)<=tol
26         flag=1;
27         break
28     else
29
30         x0=x1;
31         x1=x2;
```

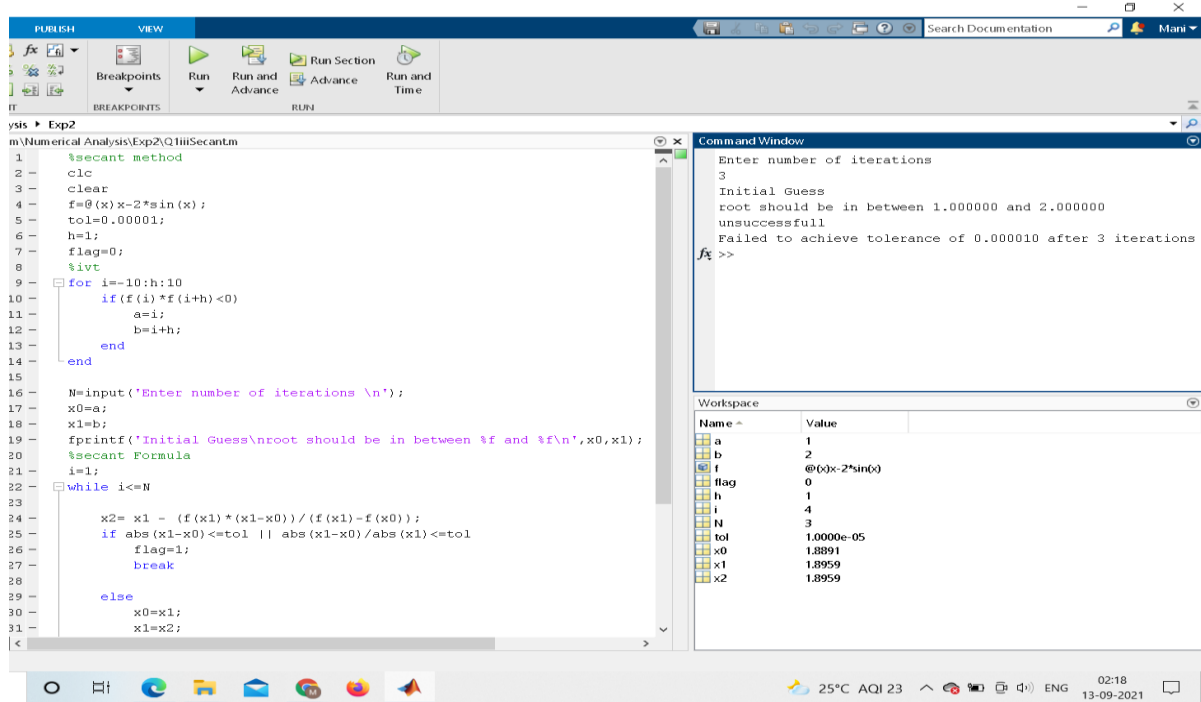
Command Window

```
Enter number of iterations
10
Initial Guess
root should be in between 1.000000 and 2.000000
successful
The root is 1.895493
Number of iterations is 6
fx >> |
```

Workspace

Name	Value
a	1
b	2
f	@(x)x-2*sin(x)
flag	1
h	1
i	6
N	10
tol	1.0000e-05
x0	1.8955
x1	1.8955
x2	1.8955

UNSUCCESSFUL:



CODE:

```
%secant method
```

```
clc
```

```
clear
```

```
f=@(x)x-2*sin(x);
```

```
tol=0.00001;
```

```
h=1;
```

```
flag=0;
```

```
%ivt
```

```
for i=-10:h:10
```

```
    if(f(i)*f(i+h)<0)
```

```
        a=i;
```

```
        b=i+h;
```

```
    end
```

end

N=input('Enter number of iterations \n');

x0=a;

x1=b;

fprintf('Initial Guess\nroot should be in between %f and %f\n',x0,x1);

%secant Formula

i=1;

while i<=N

$x_2 = x_1 - (f(x_1) * (x_1 - x_0)) / (f(x_1) - f(x_0));$

if abs(x1-x0)<=tol || abs(x1-x0)/abs(x1)<=tol

flag=1;

break

else

x0=x1;

x1=x2;

i=i+1;

end

end

if flag~=1

fprintf('unsuccessfull\nFailed to achieve tolerance of %f after %d iterations\n',tol,N);

else

fprintf('successfull\n');

fprintf('The root is %f',x0);

```
fprintf('\nNumber of iterations is %d\n',i);
end
```

Q2

4. An oscillating current in an electric circuit is described by $i = 9e^{-t} \sin(2\pi t)$, where t is in seconds. Determine the lowest value of t such that $i = 3.5$.

Ans2 by Secant Method

```

1  %secant method
2  clc
3  clear
4  f=@(t) 9*exp(-t) * sin(2*pi*t) - 3.5;
5  tol=0.00001;
6  h=0.1;
7  flag=0;
8  %ivt
9  for i=0:h:5
10     if (f(i)*f(i+h)<0)
11         a=i;
12         b=i+h;
13     end
14 end
15
16 N=input('Enter number of iterations \n');
17 x0=a;
18 x1=b;
19 fprintf('Initial Guess\nroot should be in between %f and %f\n',x0,x1);
20 %secant Formula
21 i=1;
22 while i<=N
23     x2= x1 - (f(x1)*(x1-x0))/(f(x1)-f(x0));
24     if abs(x1-x0)<=tol || abs(x1-x0)/abs(x1)<=tol
25         flag=1;
26         break
27     else
28
29         x0=x1;
30         x1=x2;
31     end
32 end
33 fprintf('The value of t is %f\n',x1);
34 fprintf('Number of iterations is %d\n',i);
35

```

Command Window

```

>>
Enter number of iterations
100
Initial Guess
root should be in between 0.400000 and 0.500000
successfull
The value of t is 0.401342
Number of iterations is 4
>>

```

Name	Value
a	0.4000
b	0.5000
f	@(t)9*exp(-t)*sin...
flag	1
h	0.1000
i	4
N	100
tol	1.0000e-05
x0	0.4013
x1	0.4013
x2	0.4013

CODE:

```
%secant method
```

```
clc
```

```
clear
```

```
f=@(t) 9*exp(-t) * sin(2*pi*t) - 3.5;
tol=0.00001;
h=0.1;
flag=0;
%ivt
for i=0:h:5
    if(f(i)*f(i+h)<0)
        a=i;
        b=i+h;
    end
end

N=input('Enter number of iterations \n');
x0=a;
x1=b;
fprintf('Initial Guess\nroot should be in between %f and %f\n',x0,x1);
%secant Formula
i=1;
while i<=N

    x2= x1 - (f(x1)*(x1-x0))/(f(x1)-f(x0));
    if abs(x1-x0)<=tol || abs(x1-x0)/abs(x1)<=tol
        flag=1;
        break

    else
        x0=x1;
```

```

x1=x2;

i=i+1;

end

end

if flag~=1

    fprintf('unsuccessfull\nFailed to achieve tolerance of %f after %d iterations\n',tol,N);

else

    fprintf('successfull\n');

    fprintf('The value of t is %f',x0);

    fprintf('\nNumber of iterations is %d\n',i);

end

```

Ans2 by Newton Method

The screenshot displays the MATLAB IDE with a script titled 'Exp2' and a Command Window showing the execution results. The script implements Newton's method to find the root of a function. The Command Window shows the user input for iterations (10), the initial guess (0.450000), and the final results: 'successfull', 'The value of t is 0.401344 seconds', and 'Number of iterations is 3'.

Script Code:

```

1 %newton method
2 clc
3 clear
4 f=@(t) 9*exp(-t) * sin(2*pi*t) - 3.5;
5 fd=@(t) (-9*exp(-t) * sin(2*pi*t))+(18*pi*exp(-t) * cos(2*pi*t));
6 tol=0.00001;
7 h=0.1;
8 flag=0;
9 %ivt
10 for i=0:h:10
11     if (f(i)*f(i+h)<0)
12         a=i;
13         b=i+h;
14     end
15 end
16
17 N=input('Enter number of iterations \n');
18 x0=(a+b)/2;
19 fprintf('Initial Guess is %f by IVT\n',x0);
20 %Newton Formula
21 i=1;
22 while i<=N
23
24     x1= x0 - (f(x0)/fd(x0));
25     if abs(x1-x0)<=tol || abs(x1-x0)/abs(x1)<=tol
26         flag=1;
27         break
28     else
29         x0=x1;
30         i=i+1;
31     end
32 end

```

Command Window Output:

```

Enter number of iterations
10
Initial Guess is 0.450000 by IVT
successfull
The value of t is 0.401344 seconds
Number of iterations is 3
fx >>

```

Workspace Variables:

Name	Value
a	0.4000
b	0.5000
f	@(t)9*exp(-t)*sin...
fd	@(t)(-9*exp(-t)*...
flag	1
h	0.1000
i	3
N	10
tol	1.0000e-05
x0	0.4013
x1	0.4013

CODE:

```
%newton method

clc

clear

f=@(t) 9*exp(-t) * sin(2*pi*t) - 3.5;
fd=@(t)(-9*exp(-t) * sin(2*pi*t))+(18*pi*exp(-t) * cos(2*pi*t));

tol=0.00001;

h=0.1;

flag=0;

%ivt
for i=0:h:10
    if(f(i)*f(i+h)<0)
        a=i;
        b=i+h;
    end
end

N=input('Enter number of iterations \n');
x0=(a+b)/2;

fprintf('Initial Guess is %f by IVT\n',x0);

%Newton Formula

i=1;

while i<=N

    x1= x0 - (f(x0)/fd(x0));

    if abs(x1-x0)<=tol || abs(x1-x0)/abs(x1)<=tol

        flag=1;
```

```
break

else

    x0=x1;
    i=i+1;
end

end

if flag~=1
    fprintf('unsuccessfull\nFailed to achieve tolerance of %f after %d iterations\n',tol,N);
else
    fprintf('successfull\n');
    fprintf('The value of t is %f seconds',x0);
    fprintf('\nNumber of iterations is %d\n',i);
end
```

EXTRA QUES GIVEN: $\exp(x)=\cos(x)$

Ans by Newton Method

The screenshot shows the MATLAB IDE with a script titled 'Exp2' and its execution results. The script implements the Newton-Raphson method to find the root of the function $f(x) = \exp(x) - \cos(x)$. It uses the interval bisection method (IVT) to find an initial guess and then iteratively refines it using the Newton-Raphson formula. The script includes comments and uses `fprintf` to display the results.

```
1 %newton method
2 clc
3 clear
4 f=@(x)exp(x)-cos(x);
5 fd=@(x)exp(x)+sin(x);
6 tol=0.00001;
7 h=1;
8 flag=0;
9 %ivt
10 for i=-10:h:10
11     if(f(i)*f(i+h)<0)
12         a=i;
13         b=i+h;
14     end
15 end
16
17 N=input('Enter number of iterations \n');
18 x0=(a+b)/2;
19 fprintf('Initial Guess is %f by IVT\n',x0);
20 %Newton Formula
21 i=1;
22 while i<=N
23     x1= x0 - (f(x0)/fd(x0));
24     if abs(x1-x0)<=tol || abs(x1-x0)/abs(x1)<=tol
25         flag=1;
26         break
27     else
28         x0=x1;
29         i=i+1;
30     end
31 end
```

The Command Window shows the following output:

```
Enter number of iterations
10
Initial Guess is -1.500000 by IVT
successfull
The root is -1.292696
Number of iterations is 4
fx >>
```

The Workspace window shows the following variables and their values:

Name	Value
a	-2
b	-1
f	@(x)exp(x)-cos(x)
fd	@(x)exp(x)+sin(x)
flag	1
h	1
i	4
N	10
tol	1.0000e-05
x0	-1.2927
x1	-1.2927

CODE:

```
%newton method
```

```
clc
```

```
clear
```

```
f=@(x)exp(x)-cos(x);
```

```
fd=@(x)exp(x)+sin(x);
```

```
tol=0.00001;
```

```
h=1;
```

```
flag=0;
```

```
%ivt
```

```
for i=-10:h:10
```

```
    if(f(i)*f(i+h)<0)
```

```
        a=i;
```

```
        b=i+h;
```

```
    end
```


end

N=input('Enter number of iterations \n');

x0=(a+b)/2;

fprintf('Initial Guess is %f by IVT\n',x0);

%Newton Formula

i=1;

while i<=N

 x1= x0 - (f(x0)/fd(x0));

 if abs(x1-x0)<=tol || abs(x1-x0)/abs(x1)<=tol

 flag=1;

 break

 else

 x0=x1;

 i=i+1;

 end

end

if flag~=1

 fprintf('unsuccessfull\nFailed to achieve tolerance of %f after %d iterations\n',tol,N);

else

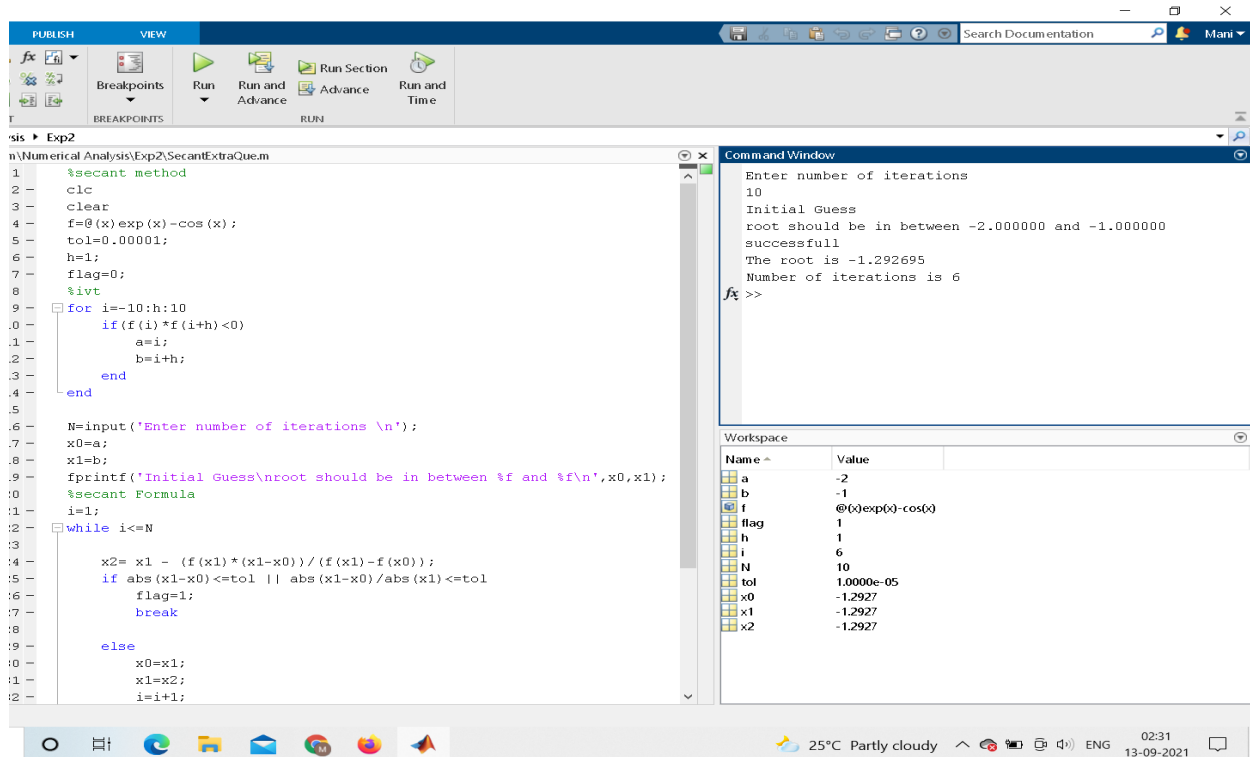
 fprintf('successfull\n');

 fprintf('The root is %f',x0);

 fprintf('\nNumber of iterations is %d\n',i);

end

Ans by Secant Method



CODE:

```
%secant method
```

```
clc
```

```
clear
```

```
f=@(x)exp(x)-cos(x);
```

```
tol=0.00001;
```

```
h=1;
```

```
flag=0;
```

```
%ivt
```

```
for i=-10:h:10
```

```
    if(f(i)*f(i+h)<0)
```

```
        a=i;
```

```
        b=i+h;
```

```
end
end

N=input('Enter number of iterations \n');
x0=a;
x1=b;
fprintf('Initial Guess\nroot should be in between %f and %f\n',x0,x1);
%secant Formula
i=1;
while i<=N

    x2= x1 - (f(x1)*(x1-x0))/(f(x1)-f(x0));
    if abs(x1-x0)<=tol || abs(x1-x0)/abs(x1)<=tol
        flag=1;
        break

    else
        x0=x1;
        x1=x2;
        i=i+1;
    end

end

if flag~=1
    fprintf('unsuccessfull\nFailed to achieve tolerance of %f after %d iterations',tol,N);
else
    fprintf('successfull\n');
```

```
fprintf('The root is %f',x0);  
fprintf('\nNumber of iterations is %d\n',i);  
end
```