Implementation

(4)

```
function [zero, res, niter] = bisection(f,a,b,tol,nmax)

x = [a (a+b)/2 b]; y = f(x); niter = 0; I = (b-a)/2;
if(y(1)*y(3)>0)
    error('The sign of the function at the extrema must be opposite');
elseif y(1) == 0
    zero = a; res = 0; return
elseif y(3) == 0
    zero = b; res = 0; return
end
while (I>=tol && niter < nmax)
    if sign(y(1))*sign(y(2))<0
        x(3) = x(2); x(2) = (x(1)+x(3))/2;
        y = f(x); I = (x(2)-x(1))/2;
    elseif sign(y(3))*sign(y(2))<0
        x(1) = x(2); x(2) = (x(1)+x(3))/2;
        y = f(x); I = (x(2)-x(1))/2;
    else
        x(2) = x(find(y==0)); I = 0;
    end
    niter = niter + 1;
end

if(niter >= nmax)
    fprintf('bisection method exited without convergence');
end
zero = x(2); res = f(x(2));

end
```

Figure 1: Bisection Method Code

```
function y = f(x)
y = x-exp(-x);
end
```

Figure 2: Equation for (4)

```
function [zero, res, niter] = newtons(f, df, x0, tol, nmax)
niter = 0;
x = x0 - f(x0)/df(x0);

while abs (x0-x) >= tol && niter <= nmax
    x0 = x;
    x = x0 - f(x0)/df(x0);
    niter = niter + 1;
end

if niter > nmax
    fprintf('Newtons method stopped without convergence\n');
end

zero = x; res = f(x);


end
```

Figure 3: Newton's Method Code

```
function y = df(x)
y = 1+exp(-x);
end
```

Figure 4: Derived equation for (4)

(a)

```
Command Window
>> [zero,res,niter] = bisection (@f,0,10,10^(-8),100)

zero =

    0.5671


res =

  -1.6221e-08


niter =

    28

>>
```

Figure 5: Answer for (4)(a)

(b)

```
Command Window
>> [zero, res, niter] = newtons(@f, @df, 10, 10^(-8), 100)

zero =

    0.5671


res =

   1.1102e-16


niter =

     5

>>
```

Figure 6: Answer for (4)(b)

(c)

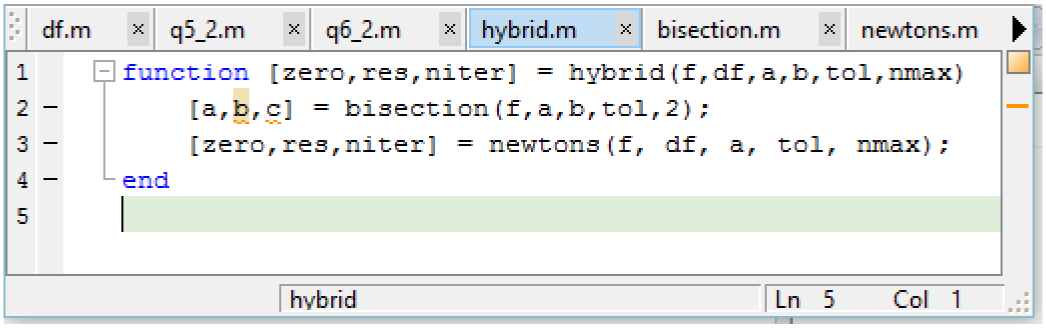

Figure 7: Hybrid Method Code

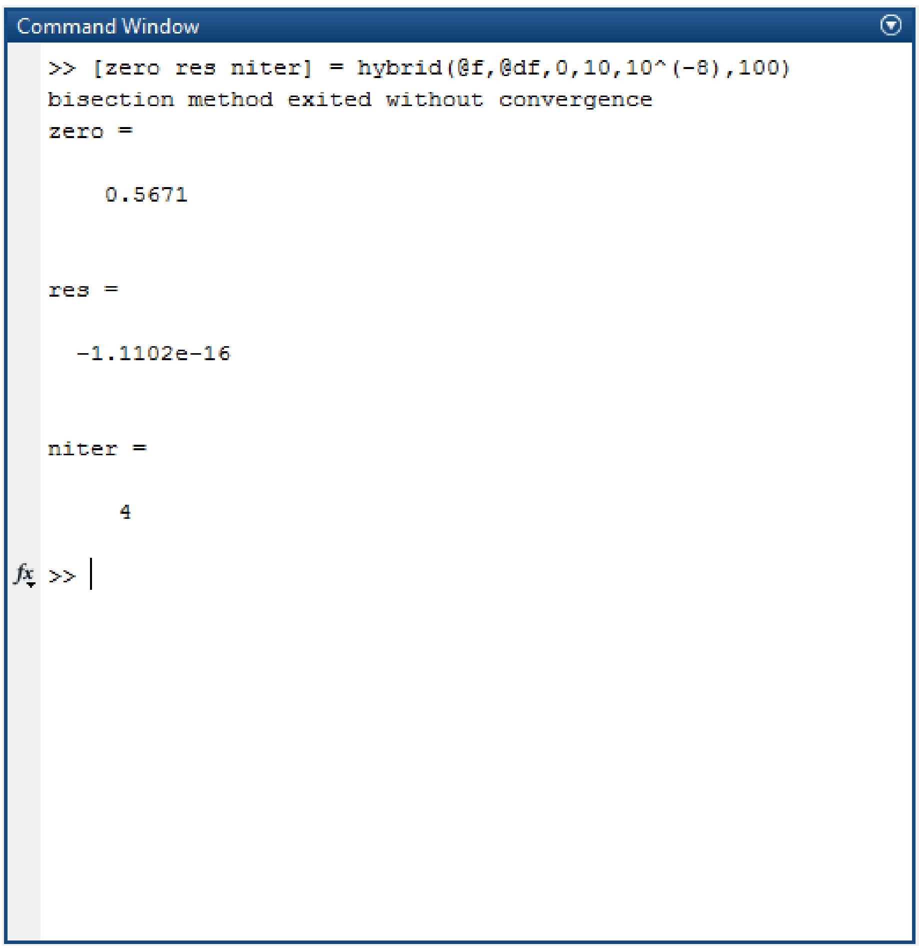

Figure 8: Answer for (4)(c)

(d)

Method	Bisection	Newton	Hybrid
Number of iterations	28	5	4

Fastest algorithm to solve this problem is the Hybrid algorithm.

(5)

```
function y = f5(x)
y = (10*cos((2*pi/5)-x)/(sin((2*pi/5)-x))^(2))- 8*cos(x)/(sin(x))^(2);
end
```

Figure 9: Equation for (5)

```
function y = df5(x)
y = 8/sin(x)-(20*cos(x-(2*pi)/5)^2)/sin(x-(2*pi)/5)^3-10/sin(x-(2*pi)/5)+(16*cos(x)^2)/sin(x)^3;
end
```

Figure 10: Derived equation for (5)

```
>> [zero, res, niter] = newtons(@f5, @df5, 0.01, 10^(-6), 100)

zero =

    0.5963


res =

   1.0658e-14


niter =

    13

>> 
```

Figure 11:Answer for (5)

In 5th question initial point was chosen by considering the alpha's characteristics. Alpha is an acute angle. Therefore minimum value was chosen as the initial point.

(6)

bisection.m × f6.m ×

```matlab
function y = f6(x)
y = (3.5*10.^(7)+0.401.*(10.^6.*(x.^(-2)))).*(x-(42.7).*10.^(-3))- 1.3806503.*10.^(-18).*3;
end
```

f6 Ln 2 Col 89

Figure 12: Equation for (6)

Command Window

```
>> [zero, res, niter] = bisection(@f6,-100,100,10^(-12),100)

zero =

    0.0427


res =

  -3.1248e-04


niter =

    46

>> 
```

Figure 13: Answer for (6)