

---

```

% Program Newton1D.m
%
% Pre -- f : function f(x) of a single variable
% df : the derivative of f(x)
% xold : initial guess
% M : maximum number of iterations
% TOL : nonnegative number
%
% Post -- x : An approximate zero of f
%
%

M=10;
TOL=0.5*10^-6;
xold=3;
list=zeros(M, 4);
xp=3.076421071012352;

format long
%Newton's Method
for k = 1:M
    fx = f(xold);
    dfx = df(xold);
    x = xold - fx=dfx;
    list(k,:,:,:)=[k xold norm(xp-xold) 0];
    if abs(x-xold) < TOL
        break;
    end
%iteration is terminated if |x_k - x_{k-1}| < TOL
xold=x;
end

%Other Updating
% for k = 1:M
%     list(k,:,:,:)=[k xold norm(xp-xold) 0];
%     x = f(xold);
%     if abs(x-xold) < TOL
%         break;
%     end
% %iteration is terminated if |x_k - x_{k-1}| < TOL
% xold=x;
% end

for i=1:M
    list(i,4)=log(list(i+1,3)/list(i+2,3))/log(list(i,3)/list(i+1,3));
end

%5.8a
% function fx=f(x)
%     fx=acos(-1/(1+exp(-2*x)));
% end

```

---

---

```
%5.8b
% function fx=f(x)
%     fx=0.5*log(-1/(1+1/cos(x)));
% end

%5.8c
function fx=f(x)
    fx=cos(x)+1/(1+exp(-2*x));
end

function dfx=df(x)
    dfx=2*exp(-2*x)/(exp(-2*x)+1)^2-sin(x);
end

Index in position 1 exceeds array bounds. Index must not exceed 10.

Error in Newton1dp4 (line 45)
    list(i,4)=log(list(i+1,3)/list(i+2,3))/log(list(i,3)/list(i+1,3));
```

*Published with MATLAB® R2023b*