
```

function [X,infNorm] = NewtonsMethodTemplate(f, J, x0, tol, N)
    % Returns the iterations of the Newton's method
    % f: Function handle
    %     Objective function, i.e. equation f(x)=0
    % J: Function handle
    %     Jacobian of f
    % x0: Initial root estimate, Nx x 1
    % tol: tolerance
    % N: Maximum number of iterations

    if nargin < 5
        N = 100;
    end
    if nargin < 4
        tol = 1e-6;
    end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Define variables
    % Allocate space for iterations (X)
    Nx = size(x0,1);
    X = zeros(Nx, N);
    X(:,1) = x0;
    infNorm = zeros(N,1);
    %r_k = zeros(N,1);
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    xn = x0; % initial estimate
    n = 1; % iteration number, change to 2 if you need x0 in
iter.values
    fn = f(xn); % save calculation
    infNorm(1) = norm(fn,Inf);
    %r_k(1) = 0
    % Iterate until f(x) is small enough or
    % the maximum number of iterations has been reached
    iterate = norm(fn,Inf) > tol;
    while iterate
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        % Calculate and save next iteration value x
        dx = -J(xn)\f(xn);
        xn = xn + dx;

        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        fn = f(xn);
        X(:,n) = xn;
        n = n + 1;
        % save calculation for next iteration
        % Continue iterating?
        infNorm(n) = [norm(fn,Inf)];
        iterate = norm(fn,Inf) > tol && n <= N;
    end
    X = X(:,1:n-1);
    if n > N

```

```
        fprintf('No more iteration left because of Corona-hoarders,  
sorry')  
    end;  
end
```

Not enough input arguments.

Error in NewtonsMethodTemplate (line 20)
 Nx = size(x0,1);

Published with MATLAB® R2019a