
```
function [x,success] = NewtonRhapsonSolver( fun, dfun, x0, tol )

% Use the Newton-Rhapson method to find the zero crossing for a
% function.
%
% Inputs:
%   fun      Function handle. We want to find x where fun(x) = 0.
%   dfun     Derivative function handle.
%   x0       Initial guess.
%   tol      Tolerance
%
% Outputs:
%   x         Solution to fun(x) = 0.
%   success   Flag indicating whether it returned with a solution
%             (1)
%             or not (0).

if nargin<4
    tol = 1e-8;
end
maxCount = 100;
count = 0;

% initial guess given
x = x0;

dist = tol+1;
while dist>tol && count<maxCount

    % compute function value and function derivative value
    f = feval(fun,x);
    fp = feval(dfun,x);

    % distance between current and next iterate
    dist = abs(f/fp);

    x = x-f/fp;
    count = count + 1;
end

if( dist>tol && count==maxCount )
    success = 0;
    warning('Terminated after %d iterations before reaching desired
tolerance.',count);
else
    success = 1;
end
```

Published with MATLAB® R2019b