Contents


```
function [root,it,success]=newton_approx(f,x0,maxit,tol,verbose)
```

```
% root=newton_exact(f,fprime)
%
% finds a set of roots corresponding to the function f (input as a handle)
% given a function which computes the derivative
```

Error checking of input

```
narginchk(3,6);   %check for correct number of inputs to function
if (nargin<4)
    maxit=100;       %maximum number of iterations allowed
end %if
if (nargin<5)
    tol=1e-6;        %how close to zero we need to get to cease iterations
end %if
if (nargin<6)
    verbose=false;
end %if
```

```
Error using newton_approx (line 9)
Not enough input arguments.
```

Newton iterations

```
it=1;
root=x0;
fval=f(root);
converged=false;
eps = 1e-3; % epsilon
derivative = (f(root + eps) - fval)/eps; % Approx. Derivative

% Make sure we don't start at an inflection point with zero derivative
if (abs(derivative)<tol)
    warning(' Attempting to start Newton iterations near an inflection point, you may wish to restart with a different guess...');
    x0=x0+1;   %bump the guess a ways off of initial value to see if we can get anything sensible
end %if

while(~converged && it<=maxit)
    if (abs(derivative)<100*tol)    %this (inflection point) will end up kicking the root really far away...
        converged=false;
        warning(' Derivative close to zero, terminating iterations with failed convergence... ');
        break;
    else
        root=root-fval./derivative;    % update root estimate
        fval=f(root);                  % see how far off we are from zero...
        if (verbose)
            fprintf(' iteration: %d; root:  %f + %f i; function value: %f, derivative:  %f \n',it,real(root),imag(root),fval,derivative);
        end %if
        it=it+1;
        converged=abs(fval)<tol;
    end %if
end %while
it=it-1;

if (~converged)
    warning('Used max number of iterations, or derivative near zero...')
    success=false;
else
    success=true;
end %if
```

```
end %function
```

Published with MATLAB® R2020b