```
function [x_sol, iter] = newton_raphson(f, df, x0, tol, max_iter)
 % Initialize the variables
 iter = 0;
 x_sol = x0;

 % Newton-Raphson method loop
 while abs(f(x_sol)) > tol && iter < max_iter
     x_sol = x_sol - f(x_sol) / df(x_sol);
     iter = iter + 1;
 end
end

% Use this function by providing the function f whose root you want to find,
% its derivative df, an initial estimate x0, a tolerance tol for the error,
% and a maximum number of iterations max_iter. For example:

% Define the function to solve and its derivative
f = @(x) x^3 - 2*x - 5;
df = @(x) 3*x^2 - 2;
x0 = 2; % Initial estimate
tol = 1e-6; % Tolerance for the error
max_iter = 1000; % Maximum number of iterations

% Solve the equation
[x_sol, iter] = newton_raphson(f, df, x0, tol, max_iter);
disp(['Approximated solution of the equation: ', num2str(x_sol)]);
disp(['Number of iterations: ', num2str(iter)]);
```