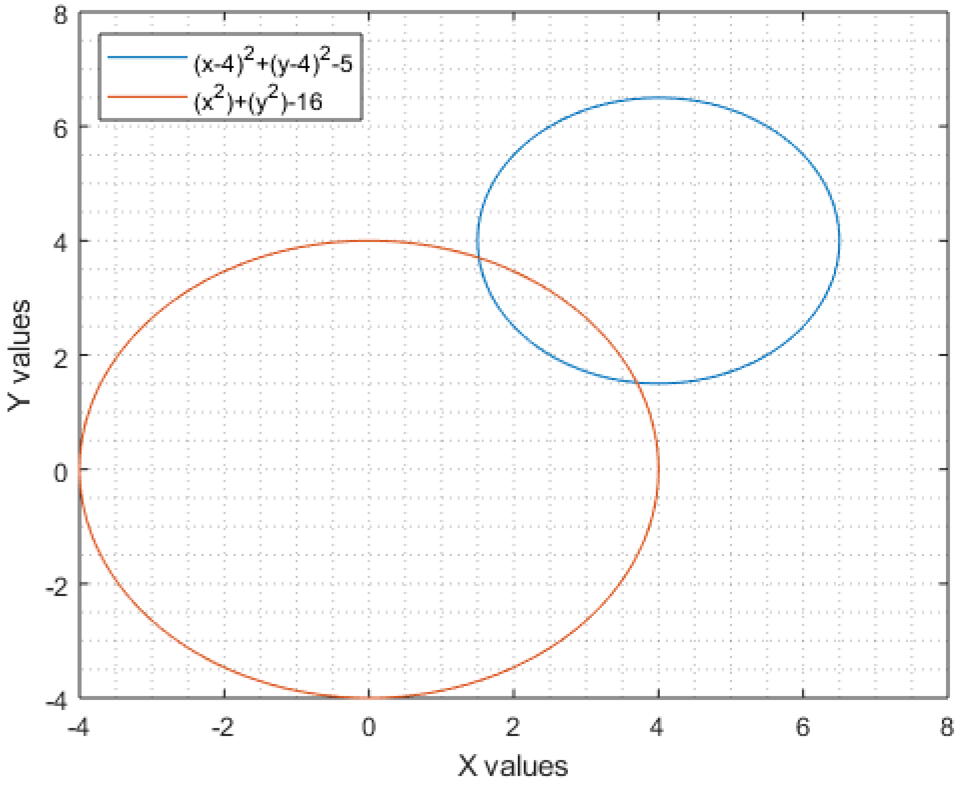
```
func = cell(2,1);
func{1} = @(x,y) (x-4).^2 + (y-4).^2 - 5;
func{2} = @(x,y) (x.^2) + (y.^2) - 16;

%we know these are equations are circle
%where x can be written as x = h + r*cos(t) and y = k +r*sin(t);
%Where (h,k) is the center for circle in which for f1  center is (4,4)
 radius = 2.5 and
%for f2 the center is (0,0), radius = 4

teta = 0:pi/100:2*pi;
x = 4 + 2.5*cos(teta);
y = 4 + 2.5*sin(teta);

%Similarly for second circle

x1 = 4*cos(teta);
y1 = 4*sin(teta);

plot(x,y);
hold on;
grid minor;
plot(x1,y1);
legend('(x-4)^2+(y-4)^2-5','(x^2)+(y^2)-16','location','northwest');
xlabel('X values');
ylabel('Y values');
hold off;

x = [2 5]';

% solutions are (x1,y1) and (y1,x1) so bascially the same values just
% fliped in values

numIter = 25;

%func = handle of function returning [f1,f2,...,fn]
%x = guess solution vector [x1,x2,...,xn]
%tol = error toleance
%numIter = number of iteractions
tol = 1.0e8*eps;

%Ensure x is a column vector
% if size(x,1) == 1
%     x=x';
% end


for i = 1:numIter
    [jac,f0]=findJacobian(func,x);
    if sqrt(dot(f0,f0)/length(x)) < tol
        disp('For the initial guess values as 5 and 2');
        roots = x
        return
    end
    dx = jac\(-f0);
    x = x + dx;
    if sqrt(dot(dx,dx)/length(x)) < tol
        roots=x
        return
    end
end
error('Maximum iterations exceeded')



function [jac, f0] = findJacobian(f,x)
h = 1.0e-06;
n = length(x);
jac = zeros(n);
f0 = [f{1}(x(1),x(2)) f{2}(x(1),x(2))]';
for i = 1:n
    tmp = x(i);
    x(i) = tmp + h;
    f1 = [f{1}(x(1),x(2)) f{2}(x(1),x(2))]';
    x(i) = tmp;
    jac(:,i)= (f1 - f0)/h;
end
end
```

For the initial guess values as 5 and 2

```
roots =

    1.8058
    3.5692
```



Published with MATLAB® R2018b