```
itermax = 50;
iterno = 1;
tol = 10e-5;
% initial guess is x_k
x_k = ones(2,1);
x_n = zeros(2,1);
err = 1/0;
while tol < err
    J = Jacobian(x_k(1),x_k(2));
    F = f(x_k(1),x_k(2));
    s_k = -J\F;
    x_n = x_k + s_k;
    F = f(x_k(1),x_k(2));
    err = sum(sqrt(F.^2));
    if iterno>=itermax
        break
    end
    x_k = x_n;
    iterno = iterno + 1;
end
x_n

% The function and its Jacobian
function F = f(x1,x2)
    F = [ x1 + 2*x2 - 2; x1^2 + 4*x2^2 - 4];
end

function Jcb = Jacobian(x1,x2)
    Jcb = [1 2; 2*x1 8*x2];
end
```

```
x_n =

   -0.0000
    1.0000
```

Published with MATLAB® R2020b