1 Problem Description

We are given a system of two nonlinear equations:

$x1^2 − 10x1 + x2^2 + 8 = 0$

$x1x2^2 + x1 − 10x2 + 8 = 0$

We want to solve this system of equations using the fixed-point method and Newton's method.

We will iterate until the residual error in each of the equations is less than 10e-6.

Fixed Point Method

We can convert the given system of equations into fixed point iteration form by isolating x1 and x2 on one side of the equations. We get the following equations:

$x1 = (x1^2 + x2^2 + 8)/10$

$x2 = (x1x2^2 + x1 + 8)/10$

We can now use these equations to iteratively solve for x1 and x2.

The initial guesses for x1 and x2 are set to (0,0).

The fixed-point method took 15 iterations to converge near the given error bound.

The total wall clock time for the fixed-point method was 0.0751 seconds.

Newton's Method

We can also use Newton's method to solve the given system of equations. The Jacobian matrix for the system of equations is:

[2*x1 - 10, 2*x2]
[x2^2 + 1, 2*x1*x2 - 10]

We will use this Jacobian matrix to iteratively solve for x1 and x2. The initial guesses for x1 and x2 are set to (0,0).

Newton's method took 5 iterations to converge.

The total wall clock time for Newton's method was 0.1126 seconds.

2 Results

Method	Total Iterations	Total Wall Time
Fixed Point	15	0.0751 s
Newton's	5	0.1126 s

As we can see from the table, Newton's method was significantly slower but converge faster and more accurate than the fixed-point method for solving the given system of equations.

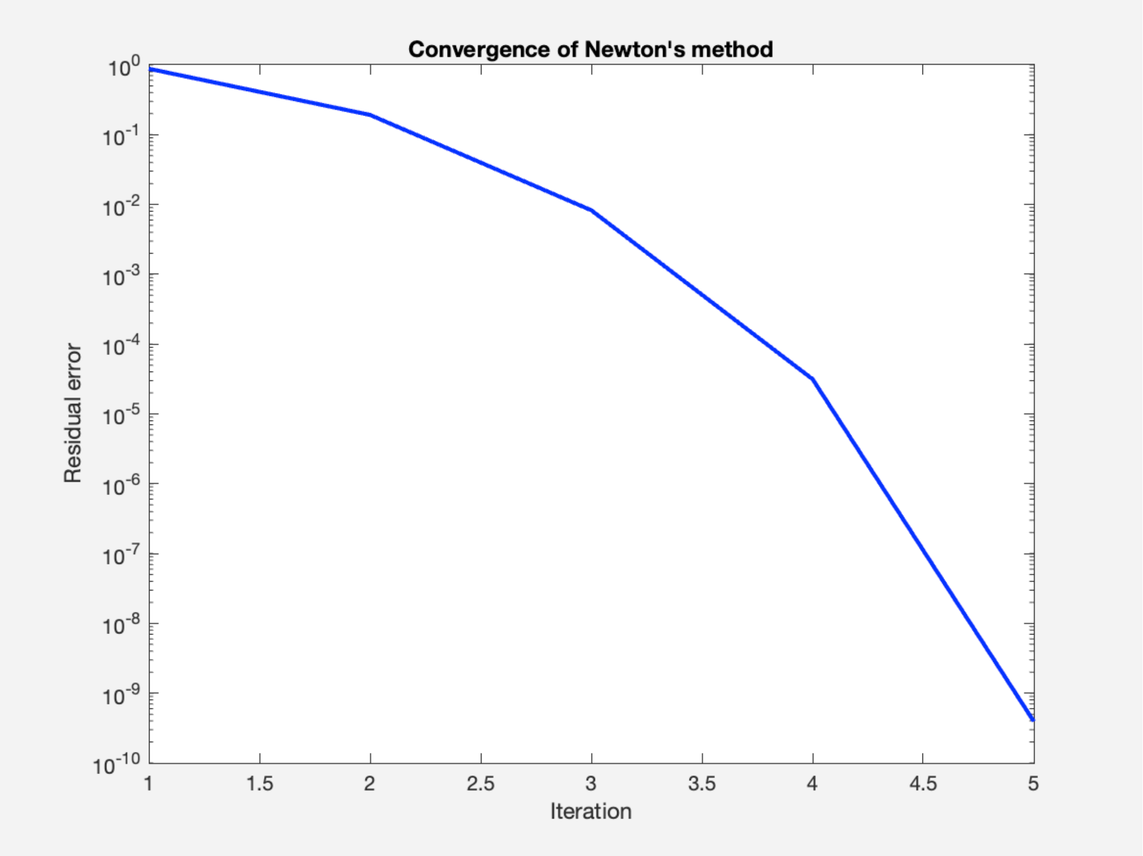

Figure above is Convergence in newtons methods in 5 iterations to given bound.

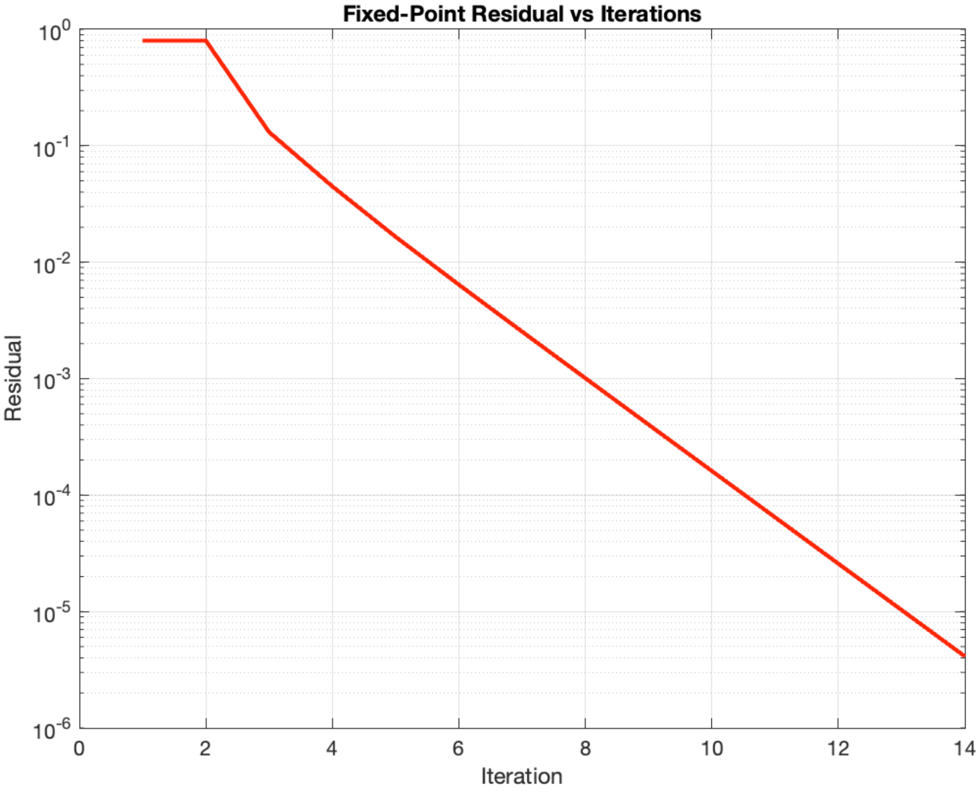

Figure above is Convergence in fixed point methods in 15 iterations to given bound.

3 Collaboration

No collaboration on this project

4 Academic Integrity

On my personal integrity as a student and member of the UCD community, I have not given nor received any unauthorized assistance on this assignment.

5 Appendix

Attached code to call Setup for equations.

```
f = @(x) [x(1)^2 + x(2)^2 - 10*x(1) + 8;
          x(1)*x(2)^2 + x(1) - 10*x(2) + 8];

J = @(x) [2*x(1) - 10,            2*x(2);
          x(2)^2 + 1,         2*x(1)*x(2) - 10];

g = @(x) [(x(1)^2 + x(2)^2 + 8)/10;
          (x(1)*x(2)^2 + x(1) + 8)/10];

x0 = [0; 0];

%Fixed-point method
tol_fp = 10e-6;
max_iters_fp = 100;
x_fp = zeros(2, max_iters_fp+1);
x_fp(:,1) = x0;
fp_resids = zeros(max_iters_fp,1);

tic;
fp_iters = 1;
fp_resids = inf;
while fp_resids(end) > tol_fp && fp_iters < max_iters_fp
    x_fp(:,fp_iters+1) = g(x_fp(:,fp_iters));
    fp_resids(fp_iters) = norm(x_fp(:,fp_iters+1) - x_fp(:,fp_iters), inf);
    if fp_iters > 1
        fp_resids(fp_iters) = norm(x_fp(:,fp_iters) - x_fp(:,fp_iters-1), inf);
    end
    fp_iters = fp_iters + 1;
end

% Plot fixed-point residual vs iterations
if any(isnan(fp_resids))
```

```
    disp('Fixed-point iteration did not converge')
end
figure;
semilogy(1:fp_iters-1, fp_resids, 'r', 'LineWidth', 2);
title('Fixed-Point Residual vs Iterations');
xlabel('Iteration');
ylabel('Residual');
grid on;
fp_time = toc;
fp_total_iters = fp_iters;

%Newton's method
tol_newton = 10e-6;
max_iters_newton = 100;
x_newton = zeros(2, max_iters_newton+1);
x_newton(:,1) = x0;

tic;
for k = 1:max_iters_newton
    Jk = J(x_newton(:,k));
    fk = f(x_newton(:,k));
    delta_x = -Jk\fk;
    x_newton(:,k+1) = x_newton(:,k) + delta_x;
    newton_resids(k) = norm(delta_x, inf);
    if newton_resids(k) < tol_newton
        break;
    end
end
newton_iters = k;

% Plot the convergence graph for Newton's method
figure;
semilogy(1:newton_iters, newton_resids(1:newton_iters), 'b', 'LineWidth', 2);
title('Convergence of Newton''s method');
xlabel('Iteration');
ylabel('Residual error');
newton_time = toc;
newton_total_iters = newton_iters;

% Display the results in a table
fprintf('Method \t\t Total iterations \t\t Wall clock time\n');
fprintf('Fixed-point \t %d \t\t\t %.4f sec\n', fp_total_iters, fp_time);
fprintf('Newton''s \t\t %d \t\t\t %.4f sec\n', newton_total_iters, newton_time);
display(x_newton);
display(x_fp);
display(newton_resids);
display(fp_resids);
```