Project Title: Solving Fluid Flow in a Pipe Network using Numerical Methods

By: Seerat E Marryum

Table of Contents

Solving Fluid Flow in a Pipe Network using Numerical Methods

Overview:

Fluid flow in a pipe network is a fundamental topic in engineering, with applications spanning water distribution systems, gas pipelines, and more. These systems can be modeled similarly to electrical grids, as sets of non-linear algebraic equations. This project focuses on developing a mathematical model for a simple two-node pipe network, solving the governing equations using Newton's method, and analyzing its convergence behavior.

Objectives:

1. Model fluid flow and pressure loss across a two-node pipe network.

2. Write the non-linear equations representing the system.

3. Solve these equations numerically using Newton's method.

4. Analyze convergence behavior with different initial guesses.

Project Steps:

1. Model Formulation:

System Description:
A two-node pipe network is considered:

- Supply pressure $Ps=100$ psi (constant).
- Receiving pressure P (to be calculated).
- Flow rate Q, determined by the pressure difference $\Delta P = Ps - P$
- $Q = K\sqrt{\Delta P}$,

K: Flow coefficient based on pipe characteristics.

Substituting $\mathbf{\Delta P=Ps - P}$:

$\mathbf{Q=K\sqrt{Ps-P}}$

2. Derive Equations:

The governing equation becomes:

$$\mathbf{f(P)=K\sqrt{Ps-P} - Q=0}$$

The derivative required for Newton's method:

$$\mathbf{f'(P)=-K/2(\sqrt{Ps-P})}$$

Convergence criterion: $\boldsymbol{\epsilon} = \mathbf{10^{-6}}$.

3. Numerical Solution Using Newton's Method

Newton's method iteratively refines the solution P using:
$\mathbf{P_{n+1}=P_n-f(P_n)/f'(P_n)}$
Initial guess: $\mathbf{P0=90psi}$. and tolerance $\boldsymbol{\epsilon} = \mathbf{10^{-6}}$

4. Implementation

The algorithm was implemented in **MATLAB**, with key features such as error handling, iteration table generation, and visualization.

Code Implementation in MATLAB:

If we take Q=*20 gallon per minute*, K=10 (*Flow coefficient* k=cd^2/l where c depends upon *material of pipe*, d is *diameter* and l is *length of pipe*)

Greater the diameter of pipe and c, result in greater value of k and thus leading to slow convergence.

```
function newton_raphson_pipe_flow()
  % Define the constants
  P_s = 100; % psi (supply node pressure)
  Q = 20;    % Flow rate
  K = 10;    % Flow coefficient
  % Non-linear function and its derivative
  f = @(P) K * sqrt(P_s - P) - Q; % f(P)
  df = @(P) -K / (2 * sqrt(P_s - P)); % f'(P)
  % Initial guess
  P0 = 90; % Initial pressure guess
  % Tolerance and maximum number of iterations
  tol = 1e-6;
  max_iter = 100;
  % Initialize variables for iteration data
  iterations = []; % Iteration numbers
  roots = [];      % Root approximations
  func_values = []; % Function values
  errors = [];      % Errors (differences between successive approximations)
  % Display iteration table header
  fprintf('Iteration\tP_guess (psi)\tf(P)\t\tError\n');
  fprintf('--------------------------------------------------------\n');
    % Newton-Raphson iteration
  for i = 1:max_iter
    % Evaluate function and derivative
    f_val = f(P0);
    df_val = df(P0);
    % Check derivative value to avoid division by zero
    if abs(df_val) < 1e-10
      error('Derivative near zero. Newton-Raphson method fails.');
    end
```

```
        % Compute next approximation
        P1 = P0 - f_val / df_val;
        % Compute error
        error = abs(P1 - P0);
        % Log data
        iterations = [iterations, i];
        roots = [roots, P1];
        func_values = [func_values, f_val];
        errors = [errors, error];
        % Display the current iteration data
        fprintf('%d\t\t%.6f\t\t%.6f\t%.6f\n', i, P1, f_val, error);
        % Check for convergence
        if error < tol
            fprintf('--------------------------------------------------------\n');
            fprintf('Converged to root P = %.6f psi in %d iterations.\n', P1, i);
                        % Plot results
            figure;
            plot(iterations, roots, '-o', 'LineWidth', 1.5);
            title('Newton-Raphson Method: Fluid Flow in Pipe Network');
            xlabel('Iteration Number');
            ylabel('Pressure Approximation (psi)');
            grid on;
            return;
        end
        % Update guess
        P0 = P1;
    end
    % If the loop finishes without convergence
    fprintf('Did not converge after %d iterations.\n', max_iter);
end
```

Output and Results:

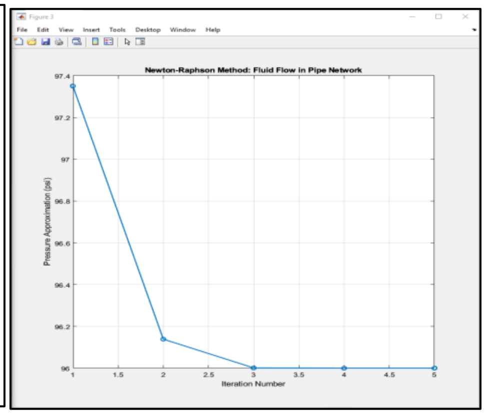
```
>> newton_raphson_pipe_flow
Iteration    P_guess (psi)    f(P)         Error
----------------------------------------------------------
1        97.350889        11.622777    7.350889
2        96.138675        -3.723911    1.212214
3        96.001223        -0.349746    0.137452
4        96.000000        -0.003058    0.001223
5        96.000000        -0.000000    0.000000
----------------------------------------------------------
Converged to root P = 96.000000 psi in 5 iterations.
```

Convergence Behavior (Testing for Different Initial Guesses for P)

Visualization:

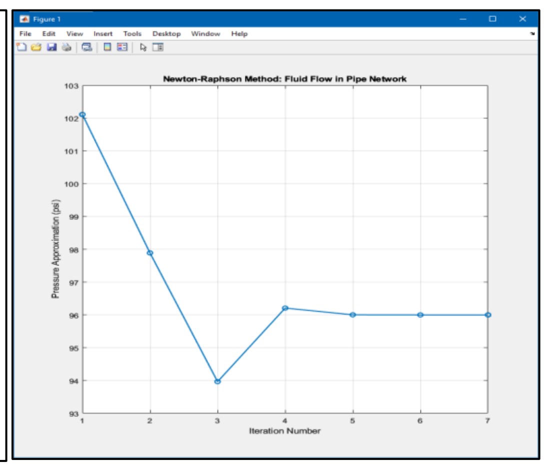
- **With P0=90:** Converges in 5 iterations to P=96 psi.

```
>> newton_raphson_pipe_flow
Iteration    P_guess (psi)    f(P)         Error
----------------------------------------------------------
1        97.350889        11.622777    7.350889
2        96.138675        -3.723911    1.212214
3        96.001223        -0.349746    0.137452
4        96.000000        -0.003058    0.001223
5        96.000000        -0.000000    0.000000
----------------------------------------------------------
Converged to root P = 96.000000 psi in 5 iterations.
```



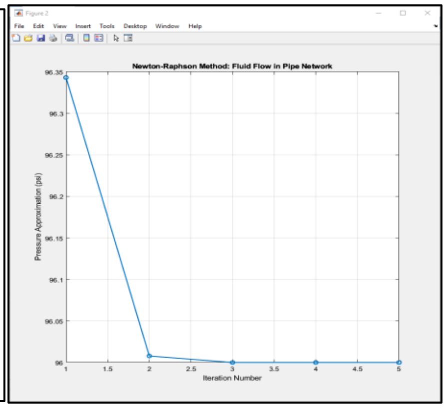
- **With P0=80:** Slower convergence, stops due to invalid range.

```
>> newton_raphson_pipe_flow
Iteration    P_guess (psi)    f(P)         Error
------------------------------------------------------
1        102.111456       24.721360    22.111456
2        97.888544        -20.000000   7.184448
3        93.965080        0.365941     7.099425
4        96.208123        4.566994     2.246460
5        96.002754        -0.527193    0.206235
6        96.000000        -0.006887    0.002804
7        96.000000        -0.000001    0.000000
------------------------------------------------------
Converged to root P = 96.000000 psi in 7 iterations.
Warning: Imaginary parts of complex X and/or Y arguments ignored.
> In newton_raphson_pipe_flow (line 61)
```



- **With P0=98:** Converges in 5 iterations.

```
>> newton_raphson_pipe_flow
Iteration   P_guess (psi)   f(P)        Error
-----------------------------------------------------------
1       96.343146       -5.857864   1.656854
2       96.007693       -0.877097   0.335453
3       96.000004       -0.019242   0.007689
4       96.000000       -0.000009   0.000004
5       96.000000       -0.000000   0.000000
-----------------------------------------------------------
Converged to root P = 96.000000 psi in 5 iterations.
```



5. Extension Task:

- Vary the flow coefficient K and find the maximum K value at which Newton's method stops converging.

Convergence Behavior (Effect of Flow Coefficient K):

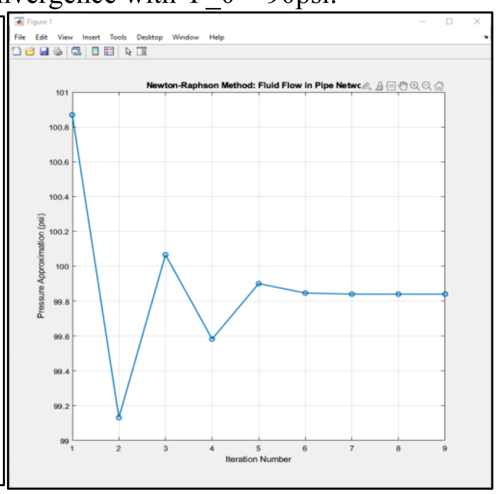
- Increasing K leads to **slower** convergence, as larger K **amplifies *non-linearity***.

  For excessively large K (K>50), Newton's method may fail due to insufficient initial approximation or derivative values approaching zero.

Results and Analysis

Identified maximum K=50 for convergence with P_0 = 90psi.

```
>> newton_raphson_pipe_flow
Iteration   P_guess (psi)   f(P)        Error
-------------------------------------------------------
1       100.868629      50.710678   2.868629
2       99.131371       -20.000000  1.890500
3       100.065959      30.166888   1.516240
4       99.582032       2.000577    0.688061
5       99.900151       12.363561   0.323071
6       99.846295       -4.152876   0.055069
7       99.840036       -0.395647   0.007428
8       99.840000       -0.002220   0.000090
9       99.840000       0.000001    0.000000
-------------------------------------------------------
Converged to root P = 99.840000 psi in 9 iterations.
Warning: Imaginary parts of complex X and/or Y arguments ignored.
> In newton_raphson_pipe_flow (line 61)
```



6. Conclusions

This project successfully applied Newton's method to solve the pressure P in a pipe network, demonstrating the *method's efficiency and limitations*. The model Q=K√ΔP accurately described the system, and the solution converged to P=96.0 psi in 5 iterations for P0=90. Convergence depended on appropriate initial guesses and flow coefficient values (K), with divergence occurring for poor initial guesses (P0=80) or high non-linearity (K>50). The study highlights the importance of informed parameter selection for reliable numerical solutions and provides a basis for extending the model to complex systems.