```
% ------------------------------------------------------------------
% Project #1
% Group 6 (Chloe Blanchard, Carson Cooper, Philip Melton, Trey Nickens,
% Edgar Trujillo, Taylor Gautreaux)
% ME 2543--Simulations Methods
% Spring 2023
% ------------------------------------------------------------------
clear; clc;
tic

%{
INPUT FORMATTING: pipewizard(x_0, Q, d, L, z)

x_0: the initial guess for the solution vector to the system of equations.
It takes the form x_0 = [Q1guess Q2guess Q3guess pressureDropGuess] with
the flow rates in gal/min and pressure drop in psi.

Q: The total flow rate through the network (gal/min).

d: vector of the form [d1 d2 d3], the diameters of the respective pipes
(inches).

L: vector of the form [L1 L2 L3], lengths of the respective pipes (ft).

z: fouling factor (unitless).
%}

d1 = [2 2 2]; %inches
d2 = [2 2.5 1.5];

L1 = [10 10 10]; %ft
L2 = [20 10 30];
% Data from the problem statement from problem 1 and Table 1

% Problem 1 ---------------------------------------------------------

ans1 = pipewizard([300 211 400 10],750 ,d1,L1,1);

% Problem 2 ---------------------------------------------------------

guess = [500 100 150 10];
% initial guess for problem two

ans2 = pipewizard(guess,750,d2,L2,1);

% Problem 3 ---------------------------------------------------------

Q = linspace(100,1500,15);
% evenly spaced array of values for Q in the range 100<Q<1500

for i = 1:length(Q)
ans3(i,:) = pipewizard([10 10 10 10],Q(i),d2, L2, 1);
```

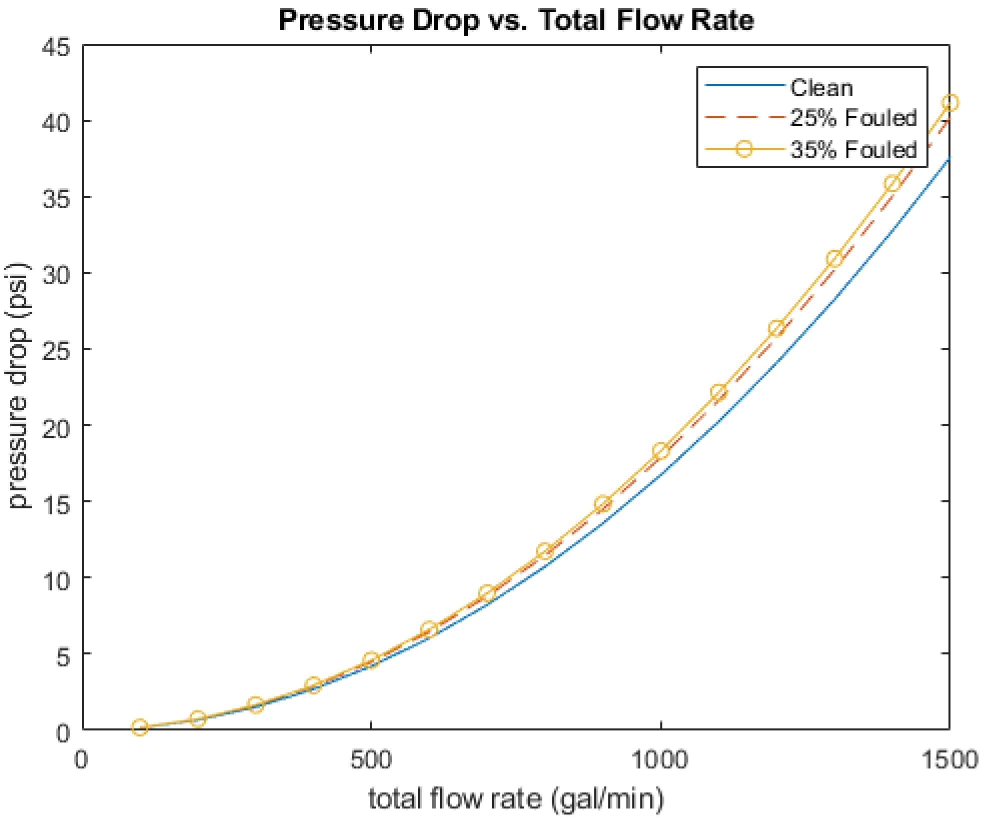
```matlab
end

% the function is evaluated for all values of Q within the range, all
% results (Q1, Q2, Q3, pressureDrop) are stored into the ans3 array

pressureDropClean = (ans3(:,4)');
% pressure drop values are retrieved from the ans3 array. the transpose is
% taken to obtain a row vector instead of a column.

% Problem 4 -------------------------------------------------------

for i = 1:length(Q)
ans4a(i,:) = pipewizard([10 10 10 10],Q(i),d2, L2, 1.25);
end
pressureDropFoul1 = (ans4a(:,4)');
% the function is evaluated for values of Q within the given range. the
% fouling factor "z" is 1.25 to account for the 25% increase in e/d.

for i = 1:length(Q)
ans4b(i,:) = pipewizard([10 10 10 10],Q(i),d2, L2, 1.35);
end
pressureDropFoul2 = (ans4b(:,4)');

plot(Q,pressureDropClean,"-")
hold on
plot(Q,pressureDropFoul1,"--")
hold on
plot(Q, pressureDropFoul2,"-o")

xlabel('total flow rate (gal/min)')
ylabel('pressure drop (psi)')
title('Pressure Drop vs. Total Flow Rate')
legend('Clean','25% Fouled','35% Fouled')

disp('Problem 1: ')
disp(['The flow through pipe 1 is: ' num2str(ans1(1)) ' gal/min'])
disp(['The flow through pipe 2 is: ' num2str(ans1(2)) ' gal/min'])
disp(['The flow through pipe 3 is: ' num2str(ans1(3)) ' gal/min'])
disp(['The pressure drop across the system is: ' num2str(ans1(4)) ' psi'])

disp('-----')
disp('Problem 2: ')
disp(['For an initial guess of Q1 = ' num2str(guess(1)) ...
    ' gal/min, Q2 = ' num2str(guess(2)) ' gal/min, Q3 = ' ...
    num2str(guess(3)) ' gal/min, deltaP = ' num2str(guess(4)) ' psi:'])
disp(['The flow through pipe 1 is: ' num2str(ans2(1)) ' gal/min'])
disp(['The flow through pipe 2 is: ' num2str(ans2(2)) ' gal/min'])
disp(['The flow through pipe 3 is: ' num2str(ans2(3)) ' gal/min'])
disp(['The pressure drop across the system is: ' num2str(ans2(4)) ' psi'])

toc
% -----------------------------------------------------------------

function [x] = pipewizard(x,Q,d,L,z)
```

```
% INPUT GIVEN PARAMETERS HERE
d = (1/12)*d; %diameter (ft)
L = L; %length (ft)
e = [.00085 .00085 .00085]; %roughness factor (ft)
visc = .0000205; %viscosity (lbf*s/ft^2)
rho = 1.94; %density (slug/ft^3) kept in slugs to avoid 32.2 ft/s^2 factor
A = (pi/4)*d.^2; %area (ft^2)
Q = Q * 231/(12^3)/60; %volumetric flow rate (ft^3/s)


% the input to the function is the initial guess with x(1:3) as volumetric
% flow rates (gal/min) and x(4) as a pressure drop (pound-force/square inch
% -- psi)

x = [x(1)/A(1)/60/(12^3)*231 x(2)/A(2)/60/(12^3)*231 ...
    x(3)/A(3)/60/(12^3)*231 x(4)*144];
% converting the initial guess from flow rate in gallon/min to velocity in
% ft/s and pressure drop from psf to psi. Our system is written in terms of
% velocities v1, v2, v3 of the respective pipes and the pressure drop.
% These are our four unknowns.

eq1 = @(x) myf1(x(1),e(1),d(1),rho,visc,z) * (L(1)/d(1))* (x(1)^2)/2 - x(4)/
rho;
eq2 = @(x) myf1(x(2),e(2),d(2),rho,visc,z) * (L(2)/d(2))* (x(2)^2)/2 - x(4)/
rho;
eq3 = @(x) myf1(x(3),e(3),d(3),rho,visc,z) * (L(3)/d(3))* (x(3)^2)/2 - x(4)/
rho;
eq4 = @(x) x(1)*A(1) + x(2)*A(2) + x(3)*A(3) - Q;
F = @(x) [eq1(x); eq2(x); eq3(x); eq4(x)];
% functions are defined to initialize the following loop. note the 'norm'
% command is the 'magnitude' (sqrt(F1^2 + F2^2 + F3^2 + F4^2)) of the F
% vector. If the norm of F is close to zero, the system is solved. These
% functions are repeatedly redefined during the loop (the values of x, and
% subsequently the value of all the friction factors, are changing after
% each fsolve iteration).

% Equations 1, 2, and 3 are derived from the darcy-weisbach equation, with
% x(1) representing v1, x(2) representing v2, x(3) representing v3, and
% x(4) representing delta P, the pressure drop. All equations are set equal
% to zero, so the pressure drop x(4) is brought over to the same side as
% all of the other terms. Equation 4 comes from the relation Q1 + Q2 + Q3 =
% Q, where Q is brought over to the left side such that V1*A1 + V2*A2 +
% V3*A3 - Q = 0. See derivation of these equations in the pdf in onedrive.

counter = 1;
% initializing loop iteration counter

while norm(F(x)) > 1e-5
% This solution process is looped to bypass limits on fsolve's
% MaxIterations and MaxFunctionCount values, ensuring a more robust solver,
% while keeping track of iterations. Each loop iteration should be about 80
% fsolve iterations, so the cap is >15 loop iterations or ~1200 fsolve
% iterations.
```

```
    eq1 = @(x) myf1(x(1),e(1),d(1),rho,visc,z) * (L(1)/d(1))* (x(1)^2)/2 -
 x(4)/rho;
    eq2 = @(x) myf1(x(2),e(2),d(2),rho,visc,z) * (L(2)/d(2))* (x(2)^2)/2 -
 x(4)/rho;
    eq3 = @(x) myf1(x(3),e(3),d(3),rho,visc,z) * (L(3)/d(3))* (x(3)^2)/2 -
 x(4)/rho;
    eq4 = @(x) x(1)*A(1) + x(2)*A(2) + x(3)*A(3) - Q;
    % all functions are defined with the CURRENT value of x(1),x(2),x(3),x(4)
    % so that the friction factor provided my myf1 is as accurate as possible.
    F = @(x) [eq1(x); eq2(x); eq3(x); eq4(x)];

    x = abs(fsolve(F,x,optimoptions('fsolve','Display','none')));
    % Every fsolve iteration, a new x is used to calculate new friction
    % factors and keep the eq1,2,3,4 as accurate as possible.

    counter = counter + 1;
    % updating iteration counter

    if counter>15
        clc;
        disp('Convergence failed, try another initial guess.')
        return
    end
    % function aborts if the solution is taking too long

end

x = [x(1)*A(1)*60*(12^3)/231, x(2)*A(2)*60*(12^3)/231, ...
    x(3)*A(3)*60*(12^3)/231,x(4)/144];
% velocity results are converted to flow rates. units are converted to
% gal/min and psi

end

% ------------------------------------------------------------------

function fric = myf1(vel, e, d, rho, visc,z)
% this function allows the caluclation of the friction factor given the
% velocity through the pipe (m/s), the roughness factor (ft), the diameter
% (ft), the density (slug/ft^3) and viscosity (lbf*s/ft^2)

fricPipe = @(fric) (fric)^(-1/2) + ....
    2*log10( (z*e/d)/3.7 +2.51/(abs(vel)*rho*d/visc * sqrt(fric)) );
% function in the form fricPipe(fric) = 0 which allows us to solve the
% Colebrook equation numerically

Re = abs(vel)*rho*d/visc;
% Reynold's number is calculated so that the correct friction equation can
% be selected using the if/else statement below.

frApprox = ((log10( (z*e/d)/3.7 + 5.74/(Re)^0.9 ))^-2);
% the approximate solution to the Colebrook equation is calculated as used
% as an initial guess for its solution.
```

```
if Re <= 2300
    fric = 64/Re;
else
    fric = fsolve(fricPipe,frApprox,optimoptions('fsolve','Display','none'));
    % If the Reynolds number is greater than 2300, the Colebrook equation
    % is solved using the function fricPipe(fric) = 0 and the initial guess
    % fApprox
end

end
```

```
Problem 1:
The flow through pipe 1 is: 250 gal/min
The flow through pipe 2 is: 250 gal/min
The flow through pipe 3 is: 250 gal/min
The pressure drop across the system is: 8.1111 psi
-----
Problem 2:
For an initial guess of Q1 = 500 gal/min, Q2 = 100 gal/min, Q3 = 150 gal/min,
 deltaP = 10 psi:
The flow through pipe 1 is: 190.4193 gal/min
The flow through pipe 2 is: 487.322 gal/min
The flow through pipe 3 is: 72.2587 gal/min
The pressure drop across the system is: 9.4337 psi
Elapsed time is 9.646474 seconds.
```



Published with MATLAB® R2022a