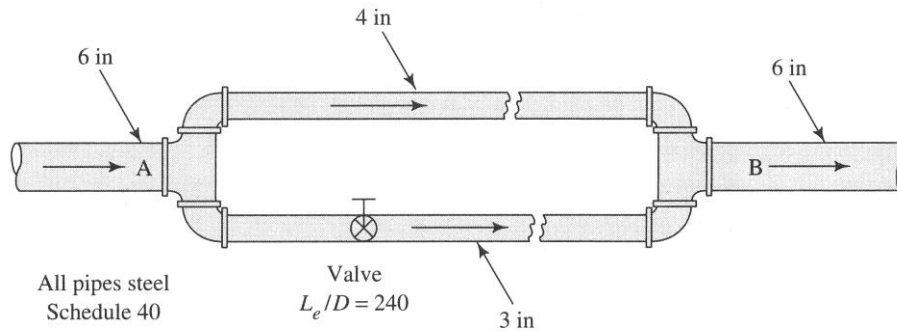


A Two-Pipe Parallel Flow System

Consider the simple parallel flow system shown in the sketch given below. For an incompressible fluid, the total volumetric flow rate in the system is the sum of the volume flow rates in the individual parallel branches. If we refer to the 4" line as pipe 1 (upper path) and the 3" branch as pipe 2 (lower path), then requiring that the conservation of mass be satisfied gives

$$Q_1 + Q_2 = Q_A = Q_B = Q_{\text{tot}} \quad (1)$$

where the flow rates at points A and B clearly must be identical (we will refer to this as the total flow rate, Q_{tot}). Note also that the average fluid velocities at points A and B are the same because both lines contain 6" Sch 40 steel pipe (since $V = Q/A$, where A is the flow area and V is the average velocity).



In addition, the pressure drop (or head loss) in each individual pipe connected in parallel must be the same since $\Delta P = P_A - P_B$ and the junction pressures P_A and P_B are the same for the individual pipes. For horizontal pipes of the same size with no mechanical devices (i.e. no pumps or turbines), the general energy equation reduces to

$$\frac{\Delta P}{\gamma} = \frac{P_A - P_B}{\gamma} = h_L = h_{L_{\text{major}}} + h_{L_{\text{minor}}} \quad (2)$$

where the two loss terms -- the pipe friction losses and the head loss due to acceleration losses in a variety of piping connections/components (i.e. elbows, valves, pipe inlets and exits, etc.) -- are referred to as major and minor losses, respectively. Finally, in eqn. (2), γ is the fluid's specific weight and the $\Delta P/\gamma$ term is the pressure head (with units of distance or "head").

Now, since the head loss in each line must be identical, we have

$$h_{L_1} = h_{L_2} = \frac{\Delta P}{\gamma} \quad (3)$$

where the individual head losses are related to the velocity head, $V^2/2g$, the pipe length-to-diameter ratio, L/D , the dimensionless Darcy friction factor, f , and the sum of the loss coefficients, $\sum K_i$, for each elbow, valve, pipe inlet and exit, etc. by Darcy's equation, or

$$h_{L_j} = \left(f \frac{L}{D} \right) \frac{V^2}{2g} \Big|_j + \left(\sum_i K_i \right) \frac{V^2}{2g} \Big|_j = \left(f \frac{L}{D} + \sum_i K_i \right) \frac{Q^2}{2gA^2} \Big|_j = \kappa_j Q_j^2 \quad (4)$$

where the j subscript refers to the j^{th} pipe segment, g is the gravitational constant, and we have replaced the average velocity with $V = Q/A$ (as noted above). In eqn. (4), κ_j is simply a convenient grouping of terms that multiply Q_j^2 and it gives a measure of the resistance to fluid flow. From eqn. (4), κ_j is given explicitly as

$$\kappa_j = \left(f \frac{L}{D} + \sum_i K_i \right) \left(\frac{1}{2gA^2} \right) \Big|_j \quad (5)$$

Thus, with this definition, we can write the head loss in pipe j in shorthand form as $h_{L_j} = \kappa_j Q_j^2$.

To complete the above development, we still need an expression for the Darcy friction factor, f . However, for turbulent flow in circular pipes (for $Re > 4000$), this factor can be given approximately by the Swamee-Jain correlation (see any good Fluid Mechanics text),

$$f = \frac{0.25}{\left[\log_{10} \left(\frac{\varepsilon/D}{3.7} + \frac{5.74}{Re^{0.9}} \right) \right]^2} \quad (6)$$

where ε/D is the relative roughness, Re is the Reynolds number,

$$Re = \frac{VD}{\nu} = \frac{D}{\nu A} Q \quad (7)$$

and ν is the fluid's kinematic viscosity (note here that we have again written $V = Q/A$, so that all the quantities of interest can be expressed in terms of the flow rate, Q).

Our development is now essentially complete. To summarize, the conservation of mass and energy equations give two coupled nonlinear equations for the flow rates in each of the two pipe branches. In particular, from eqns. (1), (3), and (4), we have

$$Q_1 + Q_2 = Q_{\text{tot}} \quad \text{and} \quad \kappa_1 Q_1^2 = \kappa_2 Q_2^2 \quad (8)$$

which can be written in two forms,

$$\begin{bmatrix} 1 & 1 \\ \kappa_1 Q_1 & -\kappa_2 Q_2 \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} = \begin{bmatrix} Q_{\text{tot}} \\ 0 \end{bmatrix} \quad \text{or} \quad \mathbf{f}(\mathbf{Q}) = \begin{bmatrix} Q_1 + Q_2 - Q_{\text{tot}} \\ \kappa_1 Q_1^2 - \kappa_2 Q_2^2 \end{bmatrix} = \mathbf{0} \quad (9)$$

where $\kappa_j = f(Q_j)$ via eqns. (5) – (7). Clearly, the two forms in eqn. (9) correspond directly to the discussion in the Lesson #6 Lecture Notes on nonlinear equations, where, as part of this illustrative application, we will use the **Linearized Iteration method** to solve the 1st formulation and we will use Matlab's **fsolve** function to solve the problem using the 2nd form (note that, because of the very complicated functional dependence on Q , the use of Newton's method with evaluation of an exact Jacobian matrix would be somewhat tedious -- thus, we will only use two out of the three techniques discussed in the notes to solve this application problem).

Now, so that we can get an explicit quantitative solution, let's assume that the above system carries oil with a specific weight of 8.80 kN/m^3 and that the total volumetric flow rate is $0.060 \text{ m}^3/\text{s}$. The oil has a kinematic viscosity of about $4.8 \times 10^{-6} \text{ m}^2/\text{s}$ and each pipe branch of the system is 60 m in length. In addition, the specified Sch 40 steel pipes have a nominal roughness of $\varepsilon = 4.6 \times 10^{-5} \text{ m}$ and inside diameters of $D_4'' = 0.1023 \text{ m}$ and $D_3'' = 0.0779 \text{ m}$, respectively, for the 4'' and 3'' segments.

Concerning the minor losses, each segment has two 90° flanged elbows, $K_{\text{elbow}} = 30 f_T$, and the loss coefficient for the valve in the lower branch is given by $K_{\text{valve}} = 240 f_T$, where f_T is the friction factor for fully turbulent flow in each pipe. Numerical values for f_T can be obtained from eqn. (6) with $Re = \infty$ (note that a quick look at a Moody chart from any Fluid Mechanics text shows that the f vs. Re curve for each ε/D value approaches a constant as Re becomes large -- this is the value of f_T that is used in the above expressions for the minor loss coefficients).

Now, with all the needed physical dimensions and fluid properties given, our goal is to compute the flow distribution in the two pipes (i.e. find Q_1 and Q_2) and to determine the pressure drop, ΔP , from point A to point B. This can be accomplished by solving either form of the coupled nonlinear equations given in eqn. (9) for the \mathbf{Q} vector, and then using this to compute h_L and, ultimately, $\Delta P = \gamma h_L$. In practice, both forms of eqn. (9) were used -- primarily to show that the *Linearized Iteration Method* and the use of the *fsolve* function in Matlab give the same results!

In particular, the `parallel_flow_1.m` code, as seen in Table 1, implements the Linearized Iteration method, and the `parallel_flow_2.m` code and associated `parallel_flow_2a.m` function file, which are listed in Table 2, use the built-in *fsolve* function to solve the above problem. Both codes have lots of comments for self-documentation and they follow the same solution logic as the Lesson #6 motivation problems that were discussed previously -- thus, it should be relatively easy to follow the code logic to see exactly how the two separate methods were implemented.

The primary screen outputs from both program versions are given below, and a quick review shows that we do indeed get identical results from both methods (note that all of the input data for the simulations are included as part of the code output -- this is a good habit to develop so that the full details of every case are readily available to the user):

```
Parallel_Flow_1: Summary Results for Simple Parallel Piping System
                  (using Linearized Iteration Method)

Geometry (Sch 40 steel pipe):
Pipe#  Diameter  Length  Flow Area  Surface Roughness  Rel Roughness  Minor Losses
      (m)        (m)      (m^2)      (m)                4.50e-04       SumK
1      0.1023    60.0    8.22e-03    4.60e-05           5.91e-04       0.98
2      0.0779    60.0    4.77e-03    4.60e-05           5.91e-04       5.20

Fluid Properties:
sp weight (N/m^3):      8.800e+03
kinematic viscosity ((m^2/s):  4.800e-06
total flow rate (m^3/s):      0.060

Calculated Parameters for each Pipe Segment:
Pipe#  Length  Flow Rate  Ave Vel  Reynolds#  FricFactor  Kcoeff  Head Loss
      (m)    (m^3/s)   (m/s)    1.08e+05  0.0200     9.59e+03  16.701
1      60.0    0.0417    5.08     6.22e+04  0.0222     5.00e+04  16.701
2      60.0    0.0183    3.83

Total Calculated Flow Rate (m^3/s):      0.0600
Calculated DeltaP across Branched Section (kPa):  146.97
```

Parallel_Flow_2: Summary Results for Simple Parallel Piping System
(using Matlab's fsolve command)

```

Geometry (Sch 40 steel pipe):
Pipe#   Diameter   Length   Flow Area   Surface Roughness   Rel Roughness   Minor Losses
        (m)        (m)      (m^2)      (m)              4.50e-04        SumK
1       0.1023      60.0    8.22e-03    4.60e-05          4.50e-04        0.98
2       0.0779      60.0    4.77e-03    4.60e-05          5.91e-04        5.20

Fluid Properties:
sp weight (N/m^3):      8.800e+03
kinematic viscosity ((m^2/s): 4.800e-06
total flow rate (m^3/s): 0.060

Calculated Parameters for each Pipe Segment:
Pipe#   Length   Flow Rate   Ave Vel   Reynolds#   FricFactor   Kcoeff   Head Loss
        (m)      (m^3/s)    (m/s)     1.08e+05    0.0200      (s^2/m^5) (m)
1       60.0     0.0417     5.08      6.22e+04    0.0222      9.59e+03 16.701
2       60.0     0.0183     3.83      5.00e+04    0.0222      5.00e+04 16.701

Total Calculated Flow Rate (m^3/s):      0.0600
Calculated DeltaP across Branched Section (kPa): 146.97

```

Clearly the lower pipe branch, with the additional valve and the smaller pipe diameter, has a larger resistance to flow (characterized by the κ_2 coefficient). Thus, for equal head loss in each branch, we would expect a lower flow rate in pipe 2 relative to the upper branch (pipe 1). This is exactly the result obtained, with about 30 % of the total flow in the lower branch and about 70 % of the total flow in the upper path. The two individual pipe flow rates, of course, sum to the total specified $0.06 \text{ m}^3/\text{s}$, and the resultant $\Delta P = \gamma h_L$ was computed to be about 147 kPa.

Note: Most piping systems encountered in practical applications (such as typical water distribution systems, for example) involve numerous parallel and series connections, as well as several sources (supply of fluid to the system) and sinks (discharge of fluid from the system). These fluid networks can give rise to large systems of nonlinear equations that can be solved by the methods discussed here. Thus, although the current problem is relatively simple, it represents a nice example of the typical scenario that will be encountered in the design and analysis of practical fluid networks -- real systems will be more complicated, but we will solve them using exactly the same solution methodology as illustrated here...

Table 1 Listing of parallel_flow_1.m.

```

%
% PARALLEL_FLOW_1.M Solve for the Flow Rates in Two Parallel Pipes
% (using the Linearized Iteration Method)
%
% This file computes the flow rates in a relatively simple parallel piping system.
% This problem is derived from Prob. 12.1 in the Fluid Mechanics text by Mott
% (5th Ed.). The problem reduces to a coupled set of 2 nonlinear algebraic
% equations (for steady flow). An iterative hand calculation was done in my
% Fluid Mechanics class (many years ago).
%
% This file simply automates the solution of the two nonlinear equations using
% the "linearized iteration" approach for solving mildly nonlinear problems.
% In this method, the coefficient matrix is a function of the solution vector.
% However, with a given guess for x, A(x) can be determined, and the solution of
% a linear system, A(x) * x = b, can be obtained. This process is continued
% until convergence (or we hit the max number of iterations). Note that a similar
% solution is available using Matlab's fsolve command in file PARALLEL_FLOW_2.M.
%
% Note that one often simply lets the next estimate of the solution be the
% current best guess, xold = xnew. However, for this problem, the solution
% tends to oscillate around the real solution. When this happens, we often
% use the next guess as the average of the previous two solution estimates,
% xold = (xnew + xold)/2. This is the approach taken here.
%
% Also note that the Reynolds number must be positive. Since Q might take on
% negative values (flow direction is opposite to the assumed direction) during
% the iterative process, we will use abs(Q) in the computation of Re.
%
% File prepared by J. R. White, UMass-Lowell (last update: Nov. 2017)
%
%
% clear all; close all; nfig = 0;
%
% set the geometry parameters
% g = 9.8; % gravitational constant (m/s^2)
% D = [0.1023 0.0779]'; % pipe diameter (4" and 3" Sch 40) (m)
% A = pi*D.*D/4; % flow area (m^2)
% L = [60 60]'; % pipe lengths (m)
% e = [4.6e-5 4.6e-5]'; % surface roughness (m)
% eoD = e./D; % relative roughness
% fT = 0.25./(log10(eoD/3.7)).^2; % Darcy friction factor (fully turbulent)
% sumK = [2*30*fT(1) (2*30+240)*fT(2)]'; % minor loss factors
% minor losses: pipe 1 -> 2 standard elbows
% pipe 2 -> 2 standard elbows + valve
%
% set fluid properties (oil)
% gam = 8.80e+3; % fluid specific weight (N/m^3)
% nu = 4.8e-6; % fluid kinematic viscosity (m^2/s)
% Qtot = 0.06; % total flow rate in common pipe (m^3/s)
%
% set reasonable initial guess for flow rates (m^3/s)
% Qold = [Qtot/2 Qtot/2]';
%
% start iteration loop
% itmax = 20; it = 0; tol = 1e-5; emax = 1;
% while emax > tol && it <= itmax
% it = it+1;
%
% set up A and b matrices given current guess and solve for new solution vector
% Re = (D./(nu*A)).*abs(Qold); % Reynolds number
% f = 0.25./(log10(eoD/3.7 + 5.74./Re.^0.9)).^2; % friction factor
% K = (f.*L./D + sumK)./(2*g*A.*A); % resistance coefficients
% AA = [1 1; K(1)*Qold(1) -K(2)*Qold(2)]; % coefficient matrix
% bb = [Qtot 0]'; % RHS vector
% Qnew = AA\b; % find solution vector
%
% calc & edit error (intermediate results)
% emax = max(abs(Qnew-Qold)./Qnew);
% fprintf(1,'\n Intermediate edit for Parallel_Flow_1 Problem \n');

```

```

fprintf(1,' it = %3d      max error = %8.2e \n',it,emax);
fprintf(1,'      Qnew      Qold      \n');
for j = 1:length(Qold)
    fprintf(1,' %10.4g %10.4g      \n',Qnew(j),Qold(j));
end
%
    Qold = (Qnew+Qold)/2; % next guess is average of the two previous estimates
end
%
% print max relative error and iteration count
fprintf(1,'\n Number of iterations to convergence = %3d\n',it);
fprintf(1,' Max relative error at convergence = %8.2e\n',emax);
if it >= itmax
    fprintf(1,' ***** WARNING -- Hit max number of iterations!!! *****\n');
end
%
% now compute a bunch of parameters to edit (for converged solution)
Q = Qnew; % simplify the variable names
v = Q./A; % average flow velocity
Re = abs(v).*D/nu; % Reynolds number
f = 0.25./(log10(eoD/3.7 + 5.74./Re.^0.9)).^2; % Darcy friction factor
K = (f.*(L./D) + sumK)./(2*g*A.*A); % resistance coeff
hL = K.*Q.^2; % head loss
%
% print summary results
fid = 1;
fprintf(fid,' \n');
fprintf(fid,'      Parallel_Flow_1: Summary Results for Simple Parallel Piping System \n');
fprintf(fid,'      (using Linearized Iteration Method) \n');
fprintf(fid,' \n');
fprintf(fid,'      Geometry (Sch 40 steel pipe): \n');
fprintf(fid,'      Pipe#   Diameter   Length   Flow Area   Surface Roughness   Rel Roughness
Minor Losses\n');
fprintf(fid,'      (m)      (m)      (m^2)      (m)
SumK \n');
for n = 1:length(Q)
    fprintf(fid,' %7i      %7.4f %7.1f      %10.2e %10.2e      %10.2e      %7.2f \n', ...
        n,D(n),L(n),A(n),e(n),eoD(n),sumK(n));
end
fprintf(fid,' \n');
fprintf(fid,'      Fluid Properties: \n');
fprintf(fid,'      sp weight (N/m^3):      %10.3e \n',gam);
fprintf(fid,'      kinematic viscosity ((m^2/s): %10.3e \n',nu);
fprintf(fid,'      total flow rate (m^3/s): %10.3f \n',Qtot);
fprintf(fid,' \n');
fprintf(fid,'      Calculated Parameters for each Pipe Segment: \n');
fprintf(fid,'      Pipe#   Length   Flow Rate   Ave Vel   Reynolds#   FricFactor   Kcoeff
Head Loss \n');
fprintf(fid,'      (m)      (m^3/s)      (m/s)      (s^2/m^5)
(m) \n');
for n = 1:length(Q)
    fprintf(fid,' %7i      %7.1f      %7.4f %7.2f %10.2e %7.4f %10.2e %7.3f \n', ...
        n,L(n),Q(n),v(n),Re(n),f(n),K(n),hL(n));
end
fprintf(fid,' \n');
fprintf(fid,'      Total Calculated Flow Rate (m^3/s):      %7.4f \n',sum(Q));
fprintf(fid,'      Calculated DeltaP across Branched Section (kPa): %7.2f
\n',gam*hL(1)/1000);
%
% end of file

```

Table 2 Listing of the parallel_flow_2.m and parallel_flow_2.m files.

```

%
% PARALLEL_FLOW_2.M Solve for the Flow Rates in Two Parallel Pipes
% (using Matlab's fsolve command)
%
% This file computes the flow rates in a relatively simple parallel piping system.
% This problem is derived from Prob. 12.1 in the Fluid Mechanics text by Mott
% (5th Ed.). The problem reduces to a coupled set of 2 nonlinear algebraic
% equations (for steady flow). An iterative hand calculation was done in my
% Fluid Mechanics class (many years ago).
%
% This file simply automates the solution of the two nonlinear equations using
% Matlab's fsolve command. Note that a similar solution is available using
% the Linearized Iteration Method in file PARALLEL_FLOW_1.M.
%
% Also note that the Reynolds number must be positive. Since Q might take on
% negative values (flow direction is opposite to the assumed direction) during
% the iterative process, we will use abs(Q) in the computation of Re.
%
% The function file PARALLEL_FLOW_2A.M evaluates the two nonlinear equations for
% any given flow distribution, Q, for use by the fsolve command. The goal here is
% to find Q such that F(Q) = 0, where Q is the vector of flow rates.
%
% File prepared by J. R. White, UMass-Lowell (last update: Nov. 2017)
%

clear all; close all; nfig = 0;
global D A L eoD sumK nu g Qtot

%
% set the geometry parameters
g = 9.8; % gravitational constant (m/s^2)
D = [0.1023 0.0779]'; % pipe diameter (4" and 3" Sch 40) (m)
A = pi*D.*D/4; % flow area (m^2)
L = [60 60]'; % pipe lengths (m)
e = [4.6e-5 4.6e-5]'; % surface roughness (m)
eoD = e./D; % relative roughness
fT = 0.25./(log10(eoD/3.7)).^2; % Darcy friction factor (fully turbulent)
sumK = [2*30*fT(1) (2*30+240)*fT(2)]'; % minor loss factors
% minor losses: pipe 1 -> 2 standard elbows
% pipe 2 -> 2 standard elbows + valve

%
% set fluid properties (oil)
gam = 8.80e+3; % fluid specific weight (N/m^3)
nu = 4.8e-6; % fluid kinematic viscosity (m^2/s)
Qtot = 0.06; % total flow rate in common pipe (m^3/s)

%
% set reasonable initial guess for flow rates (m^3/s)
Qg = [Qtot/2 Qtot/2]';

%
% call fsolve to find solution to system of unconstrained nonlinear eqns
Q = fsolve(@parallel_flow_2a,Qg);

%
% now compute a bunch of parameters to edit (for converged solution)
v = Q./A; % average flow velocity
Re = abs(v).*D/nu; % Reynolds number
f = 0.25./(log10(eoD/3.7 + 5.74./Re.^0.9)).^2; % Darcy friction factor
K = (f.*(L./D) + sumK)./(2*g*A.*A); % resistance coeff
hL = K.*Q.^2; % head loss

%
% print summary results
fid = 1;
fprintf(fid,' \n');
fprintf(fid,' Parallel_Flow_2: Summary Results for Simple Parallel Piping System \n');
fprintf(fid,' (using Matlab's fsolve command) \n');
fprintf(fid,' \n');
fprintf(fid,' Geometry (Sch 40 steel pipe): \n');
fprintf(fid,' Pipe# Diameter Length Flow Area Surface Roughness Rel Roughness
Minor Losses\n');

```

```

        fprintf(fid,'
                (m)      (m)      (m^2)      (m)
SumK \n');
    for n = 1:length(Q)
        fprintf(fid,' %7i      %7.4f %7.1f      %10.2e %10.2e      %10.2e      %7.2f \n', ...
                n,D(n),L(n),A(n),e(n),eoD(n),sumK(n));
    end
    fprintf(fid,' \n');
    fprintf(fid,' Fluid Properties: \n');
    fprintf(fid,' sp weight (N/m^3): %10.3e \n',gam);
    fprintf(fid,' kinematic viscosity ((m^2/s): %10.3e \n',nu);
    fprintf(fid,' total flow rate (m^3/s): %10.3f \n',Qtot);
    fprintf(fid,' \n');
    fprintf(fid,' Calculated Parameters for each Pipe Segment: \n');
    fprintf(fid,' Pipe# Length Flow Rate Ave Vel Reynolds# FricFactor Kcoeff
Head Loss \n');
    fprintf(fid,'
                (m)      (m^3/s)      (m/s)
                (s^2/m^5)
(m) \n');
    for n = 1:length(Q)
        fprintf(fid,' %7i      %7.1f      %7.4f %7.2f %10.2e %7.4f %10.2e %7.3f \n', ...
                n,L(n),Q(n),v(n),Re(n),f(n),K(n),hL(n));
    end
    fprintf(fid,' \n');
    fprintf(fid,' Total Calculated Flow Rate (m^3/s): %7.4f \n',sum(Q));
    fprintf(fid,' Calculated DeltaP across Branched Section (kPa): %7.2f
\n',gam*hL(1)/1000);
%
% end of file

%
% PARALLEL_FLOW_2A.M Function file to solve for the Flow Rates in
% Two Parallel Pipes using Matlab's fsolve command
%
function F = parallel_flow_2a(Q)
global D A L eoD sumK nu g Qtot
Re = (D./(nu*A)).*abs(Q); % Reynolds number
f = 0.25./(log10(eoD/3.7 + 5.74./Re.^0.9)).^2; % friction factor
K = (f.*L./D + sumK)./(2*g*A.*A); % resistance coeffs
F = [sum(Q)-Qtot; K(1)*Q(1)*Q(1)-K(2)*Q(2)*Q(2)]; % nonlinear function
%
% end of function

```