CHAPTER 4

PIPE NETWORK ANALYSIS

4.1 INTRODUCTION

This chapter describes the analysis of steady flow in pipe systems. In any analysis problem all of the physical features of the network are known, and the solution process endeavors to determine the discharge in every pipe and the pressure, etc. at every node of the network. Therefore in this chapter the diameters of all pipes, their lengths and their roughnesses are known, as well as where reservoirs, pumps, pressure reduction valves, and other fittings are located. The ways in which these devices influence the hydraulics of the system will be specified. Design problems, on the other hand, try to select (wisely!) the diameters of pipes, the capacities of pumps, the water surfaces elevations of reservoirs, and so on. Thus, a design problem is distinguished from an analysis problem by the choice of the variables that are regarded as unknown. At some risk we dare to generalize by saying that design problems are usually more challenging to solve than are analysis problems, and design problems usually require the simultaneous solution of a larger system of equations than do analysis problems. A thorough understanding of the techniques of analysis for large networks that are composed of known physical features is a prerequisite to the understanding of the design of networks. The design of pipe networks is the focus of Chapter 5 and is not discussed directly in this chapter.

The analysis of a pipe network can be one of the more complex mathematical problems that engineers are called upon to solve, particularly if the network is large, as occurs in the water distribution systems of even quite small cities. A significant fraction of the entire set of equations consists of nonlinear equations, and a large number of these equations must be solved simultaneously. Before digital computers were widely used in engineering practice, it simply was not practical to solve such network problems, and consequently many existing water distribution systems have "grown" with time, based primarily on the best professional judgment of engineers, without any thorough or detailed analysis of the pressures and discharges that could exist in the pipes of the network in response to various combinations of demands on the system. The computer has made it possible to solve such large network problems with ease, and as a result many municipalities and water districts have benefited from the results of relatively detailed computer analyses of their systems in recent years. We believe it is important for an engineer to understand what is being accomplished in these computer solutions. To aid engineers in gaining this knowledge, we begin with the basic principles, and the equations that embody them, that interrelate the discharge in each pipe and the pressure at each node of the network.

The same few basic principles of fluid mechanics are the foundation of our work on pipe network analyses. These basic principles are (1) conservation of mass, or the continuity principle, (2) the work-energy principle, and (3) the relation between fluid friction and energy dissipation. Chapter 2 has already introduced these principles. The task here is to employ these ideas effectively in describing a large hydraulic system accurately by means of equations, and then to solve these simultaneous equations efficiently.

The oldest systematic method for solving the problem of steady flow in a pipe network is the Hardy Cross method, which is itself an early adaptation of the method of moment dis-tribution from structural engineering in 1936. Before the ready availability of digital computers in the late 1960s, this method was prized because it is so well suited for hand computations. Then it became the basis of most early computer software, but because of convergence problems for large systems containing pumps and other appurtenances, it will not be discussed herein. Over the past quarter century the Newton method has proven to be superior in solving the nonlinear equations, and now networks of 2500 pipes or more can be analyzed successfully with a desktop computer.

4.1.1. DEFINING AN APPROPRIATE PIPE SYSTEM

The first step in studying pipe systems is to decide what features are important and to retain them in defining the network of pipes. For large water distribution systems some "skeletonization" usually occurs in this process. In other words, not all pipes in the system are included in the analysis. This skeletonization can take on many forms, such as the following:

1. Not all connections to houses are considered as separate nodes or junctions, and all of the distributed demands along one block of a street, or even a small subdivision, may instead be aggregated or lumped at a single node;
2. Only those pipes that carry the water from the supply sources to the areas of demand are included, i.e., only the main transmission system is considered;
3. Only a few pipes and their associated appurtenances are considered; these components are regarded as vital to the proper operation of the system.

Any study of a pipe system may include one or even all of these levels of skeletonization; the first preliminary study may start with a model of type (3), and subsequent, more complete analyses may proceed back to type (1) as the adequacy of each is verified, or as components are adjusted. After these analyses have been obtained and studied, it may then be desirable to study intensively the network of pipes within a city block, or the pipes within the area of a major water user, such as a large structure or a manufacturing facility. Thus analyses can treat an entire delivery system, which is generally skeletonized, or a more detailed analysis of the piping system or plumbing within a large building, or a golf course, etc. When an analysis of a building's piping system is conducted, the exterior pressures that are supplied by the larger system can be specified with some degree of confidence since the analyses of the larger "delivery" networks provide this information. There are no hard rules that dictate which pipes should be omitted. Such decisions are often left to the professional judgment of the supervising engineer, and sometimes these decisions are called "art," but the insight gained from analyses at different levels of skeletonization often indicate which pipes should be included in the next level of analysis. Computers can now analyze a problem consisting of many more pipes (e.g., several thousand) than the human mind can visualize in detail when deciding which features should be changed to improve the performance and reduce the costs of the system.

Another vital part of defining the network problem is to determine which demands should be specified. The demands on an existing system can be obtained from water usage or billing records. Even for existing systems the data are seldom complete in describing how these demands vary during a day, or from day to day. Analyses are usually needed for a range of system demands, from peak hourly demands down to minimal demand periods (e.g., 2-3 a.m.). During above-average demand periods tanks will have their storage volumes partially depleted, but this loss of volume should be recovered when demands are small. Since a water system may be designed for a 50-year life, the specified demands must appropriately reflect future growth and increases (or possibly decreases) in per capita consumption. In the design of a new system, the demands may have to be based on comparisons with similar cities etc. However, if a system is to be designed to deliver known quantities at specified times, then the problem of determining appropriate demands does not exist. So we see that engineering experience, based on sound judgment, is often required in defining the most appropriate piping system problems to analyze.

After the analyst has obtained one or several apparently reasonable solutions, the next step is to verify by measurements in the actual system that reasonable agreement exists between the solution to the mathematical problem and the real system. This process is called network verification. If significant disagreements occur, their causes must be identified.

Are some valves in the real system unknowingly closed or partly closed; do some major leaks exist in the real system; has the skeletonization process inappropriately excluded some pipes that carry large flows? These and other possibilities should be explored until reasonable agreement does exist. Firms specialize in field flow measurements to verify that the actual pipe system is modeled properly.

After analysis has provided solutions to the network problem for various levels of demands, non-ideal or simply inadequate performance parameters can be identified. Some indicators of inadequate or poor performance consist of the following (many other possibilities that are peculiar to an individual system do exist):

1. Pressures at some nodes are too low;
2. Pressures are too high at some nodes (If water is pumped, excessively high pressures cost money, owing to larger power consumption than is needed, more frequent pipe ruptures and the premature replacement of facilities.);
3. Discharges are inadequate and/or pressures are too low to meet emergency demands, such as fire fighting;
4. Pumps are not operating near their peak efficiencies;
5. Some water storage facilities are always nearly empty, while others are nearly full or overtopping (Are the tanks under- or over-sized and located at the best elevations? Unless storage facilities perform near their capacities, they represent investments with cost/benefit ratios that are too large.);
6. Pressure reduction valves, or back pressure valves, are inactive or open (Perhaps they are not needed, or pipes should be removed or added.);
7. Too much of the supply is coming from expensive sources, etc.

Upon identifying deficiencies, the engineer's next task is to determine the best, most economical means of overcoming these deficiencies and improving the performance of the system. How best to accomplish this will again require sound professional judgment, but sound judgment seldom occurs in the absence of relevant information, i.e., the engineer must understand the system. Section 5.7 of the next chapter discusses sensitivity analyses, which could materially aid this evaluation process.

In the following work we will express the head loss in each pipe in a network by an exponential formula $h_f = KQ^n$, Eq. 2.17, so one presentation of the theory covers all cases, regardless of whether the Darcy-Weisbach equation, the Hazen-Williams equation or the Manning equation is used to express the head loss as a function of discharge. Only the values for K and n change, as we saw in Chapter 2.

4.1.2. BASIC RELATIONS BETWEEN NETWORK ELEMENTS

The two basic principles, upon which all network analysis is developed, are (1) the conservation of mass, or continuity, principle, and (2) the work-energy principle, including the Darcy-Weisbach or Hazen-Williams equation to define the relation between the head loss and the discharge in a pipe. The equations that are developed from the continuity principle will be called Junction Continuity Equations, and those that are based on the work-energy principle will be called Energy Loop Equations. The number of these equations that constitutes a non-redundant system of equations is related directly to fundamental relations between the number of pipes, number of nodes and number of independent loops that occur in branched and looped pipe networks. In defining these relations NP will denote the number of pipes in the network, NJ will denote the number of junctions in the network, and NL will denote the number of loops around which independent equations can be written. In defining junctions, a supply source will not be numbered as a junction. A supply source is a point where the elevation of the energy line, or hydraulic grade line, is established; a junction, or node, is a point where two or more pipes join. A node can exist at each end of a "dead end" pipe; this instance is an exception to the usual rule, where only one pipe is connected to a node. In a branched system there are by definition no loops, and thus $NL = 0$ for any branched system. In branched systems the number of nodes is always one larger than the number of pipes, or $NP = NJ - 1$, unless a reservoir is shown at the end of one pipe and this is not considered to be a junction. Then $NP = NJ$. (This situation actually occurs.) Figures 4.1a and 4.1b depict a small branched network and also a small looped network.

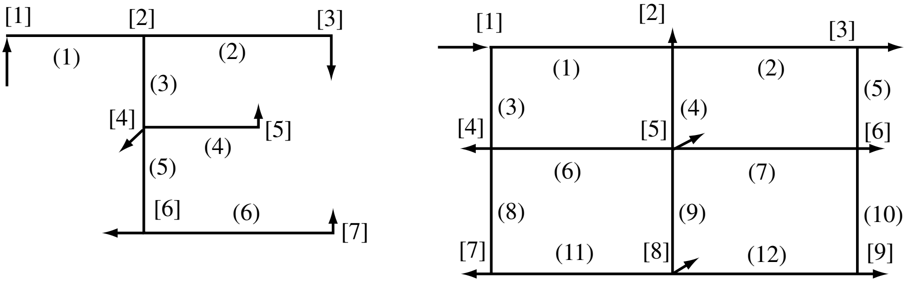


Figure 4.1 (a) A small branched system. 6 pipes, 7 nodes (b) A small looped system. 12 pipes, 9 nodes

In the branched system the number of nodes is 7 and the number of pipes is 6 (one less than the number of nodes), whereas in the looped system there are 12 pipes and 9 nodes, i.e., the number of nodes is less than the number of pipes.

For a looped network the number of loops (around which independent energy equations can be written) is given by

$$NL = NP - NJ \tag{4.1}$$

if the network contains two or more supply sources, or

$$NL = NP - (NJ - 1) = NP - NJ + 1 \tag{4.2}$$

If the network contains fewer than two supply sources and the flow from the single source is determined by adding all of the other demands, then this source is shown as a negative demand and the source is called a node. We note that this is the case in the small looped network in 4.1.b, so we have $NP = 12$, $NJ = 9$ and $NL = 12 - (9 - 1) = 4$.

Equation 4.2 also applies to a branched system with $NL = NP - NJ + 1 = 0$, since a branched system can have at most one supply source. Actually, every pipe system must have at least one supply source, but sometimes the source is not shown since the discharge from this supply source is known, and the source is replaced by a negative demand, which is a flow coming into this junction, equal to the sum of the other demands. When this is done, the elevation of the energy line (or HGL or pressure) must be specified at a node so the other HGL elevations can be determined. Energy loops that begin at one supply source and end at another are called pseudo loops, i.e., these loops do not close on themselves. The number of pseudo loops, which are numbered as part of NL, equal the number of supply sources minus one. In forming pseudo loops all supply sources must be located at the end of a pseudo loop. It is generally possible to form more loops than are needed to produce a set of independent equations. As each new loop is formed, see that at least one pipe in the new loop is not a part of any prior loop; in this way the formation of redundant loops can usually be avoided. For special devices, such as pressure reduction valves, this rule of experience must be modified slightly, as will be described later.

4.2 EQUATION SYSTEMS FOR STEADY FLOW IN NETWORKS

Three different systems of equations can be developed for the solution of network analysis problems. These systems of equations are named after the variables that are regarded as the principal unknowns in that solution method. These systems of equations are called the ***Q*-equations** (when the discharges in the pipes of the network are the principal unknowns), the ***H*-equations** (when the HGL-elevations, also simply called the heads H, at the nodes are the principal unknowns), and the ***ΔQ*-equations** (when corrective discharges, ΔQ, are the principal unknowns). Each of these three systems of equations will be studied separately.

4.2.1. SYSTEM OF *Q*-EQUATIONS

The analysis of flow in pipe networks is based on the continuity and work-energy principles. To satisfy continuity, the volumetric discharge into a junction must equal the volumetric discharge from the junction. Thus at each of the NJ (or NJ - 1) junctions an equation of the form of Eq. 4.3 is obtained:

$$QJ_j - \Sigma Q_i = 0 \tag{4.3}$$

In this equation QJ_j is the demand at the junction j, and each Q_i is the discharge in one of the pipes that join at junction j. These junction continuity equations are the first portion of the Q-equations. The work-energy principle provides additional equations which must be satisfied. These equations are obtained by summing head losses along both real and pseudo loops to produce independent equations. There are NL of these equations, and they are of the form of Eq. 4.4 or 4.5, depending upon whether the loop is a real loop or a pseudo loop, respectively, and they are the second portion of the Q-equations:

$$\Sigma h_{fi} = 0 \tag{4.4a}$$

$$\Sigma h_{fi} = \Delta WS \tag{4.5a}$$

When the head losses are expressed in terms of the exponential formula, then these equations take the forms

$$\sum K_i Q_i^n = 0 \tag{4.4b}$$

$$\sum K_i Q_i^n = \Delta WS \tag{4.5b}$$

in which the summation includes the pipes that form the loop. If the direction of the flow should oppose the direction that was assumed when the energy loop equations were written, such that Q_i becomes negative, then there are two alternatives: One is to reverse the sign in front of this term, i.e., correct the direction of the flow. The second, which is generally preferred when writing a program to solve these equations, is to rewrite the equations as follows:

$$\sum K_i \, Q_i |Q_i|^{n-1} = 0 \tag{4.4c}$$

$$\sum K_i \, Q_i |Q_i|^{n-1} = \Delta WS \tag{4.5c}$$

To illustrate the system of Q-equations, consider the small 5-pipe network shown in Fig. 4.2. Since no supply sources are shown for this network, only NJ - 1 junction continuity equations are available. Writing these continuity equations for nodes 1, 2, and 3 leads to

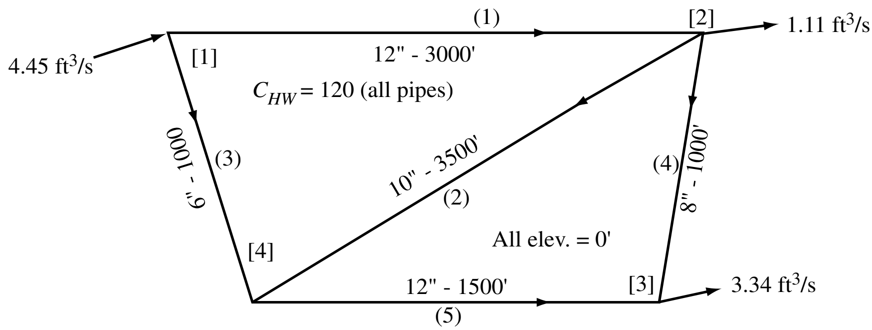


Figure 4.2 Small network.

$$
\begin{aligned}
F_1 &= Q_1 + Q_3 - 4.45 = 0 \\
F_2 &= -Q_1 + Q_2 + Q_4 + 1.11 = 0 \\
F_3 &= -Q_4 - Q_5 + 3.34 = 0
\end{aligned}
\tag{4.6}
$$

In these equations and throughout the text, F_i for any number i is any equation which has been arranged into the form $F_i = 0$; this format is useful for identification purposes and also for subsequent mathematical and numerical developments. The continuity equation at node 4 is $-Q_3 - Q_2 + Q_5 = 0$. However, this equation is not independent of the other three nodal equations since it is, except for sign, the sum of these three equations. Now let us use the Hazen-Williams equation to define the head loss in each pipe. In expressing these head losses, the exponential equation will be used. From Eq. 2.18 the coefficients are the following:

$$
K_1 = 2.018, \; K_2 = 5.722, \; K_3 = 19.674, \; K_4 = 4.847, \; K_5 = 1.009 \tag{4.7}
$$

The energy equations around the two loops may be written as

$$
\begin{aligned}
F_1 &= K_1 Q_1^{1.852} + K_2 Q_2^{1.852} - K_3 Q_3^{1.852} = 0 \\
F_2 &= K_4 Q_4^{1.852} - K_5 Q_5^{1.852} - K_2 Q_2^{1.852} = 0
\end{aligned}
\tag{4.8}
$$

or

$$
\begin{aligned}
F_1 &= 2.108 Q_1^{1.852} + 5.722 Q_2^{1.852} - 19.674 Q_3^{1.852} = 0 \\
F_2 &= 4.847 Q_4^{1.852} - 1.009 Q_5^{1.852} - 5.722 Q_2^{1.852} = 0
\end{aligned}
\tag{4.9}
$$

which might alternatively be written as follows if the directions of the flows are uncertain:

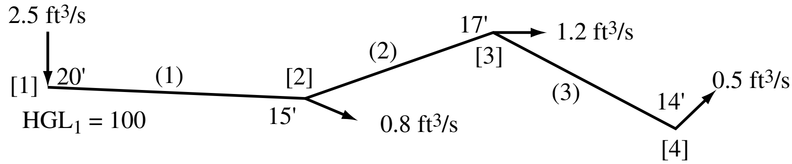
$$
\begin{aligned}
F_1 &= 2.108 Q_1 |Q_1|^{0.852} + 5.722 Q_2 |Q_2|^{0.852} - 19.674 Q_3 |Q_3|^{0.852} = 0 \\
F_2 &= 4.847 Q_4 |Q_4|^{0.852} - 1.009 Q_5 |Q_5|^{0.852} - 5.722 Q_2 |Q_2|^{0.852} = 0
\end{aligned}
\tag{4.10}
$$

These two work-energy equations are obtained by starting at nodes 1 and 2, respectively, and traversing the respective loops I and II in the clockwise direction. If the assumed direction of flow opposes this traverse, a minus sign precedes the head loss term for that pipe. The simultaneous equations, such as those appearing as Eqs. 4.6 and 4.10, are called Q-equations because it is the Q's, the discharges in the pipes, that are the set of primary unknown variables. After the Q's are found (and the head loss in each pipe is therefore also known) for each pipe, the HGL-elevations at the nodes can be found by starting at a known HGL-elevation and repeatedly applying the exponential formula for head loss to each pipe.

If the network is a branched system, then the Q-equations consist of only the junction continuity equations. These can be solved, giving the discharge in every pipe, with a linear algebra solver, i.e. a pocket calculator that implements matrix algebra, a spreadsheet, TK-solver, MathCAD or a solver such as SOLVEQ. Thereafter the individual heads are computed from the head loss equation for each pipe. Methods for solving looped systems are described later.

Example Problem 4.1

The coefficients K and n for the exponential formula are given in the table for each of the three pipes in this branched system. Find the discharge in each pipe and the pressure at each node. The elevation of the HGL at node 1 is $H_1 = 100$ ft.



Pipe	***K***	***n***
1	3.772	1.944
2	5.730	1.926
3	16.29	1.889

A formal method for solving the Q-equations for this network is to use matrix algebra to write the coefficient matrix, an unknown vector, and a known vector in the following way:

$$\begin{bmatrix} \mathbf{1} & \mathbf{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{-1} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{Bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{Bmatrix} = \begin{Bmatrix} 0.8 \\ 1.2 \\ 0.5 \end{Bmatrix}$$

The solution is

$$\begin{Bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{Bmatrix} = \begin{Bmatrix} 2.5 \\ 1.7 \\ 0.5 \end{Bmatrix}$$

We should note that it is easy to obtain this solution by inspection. Starting at the downstream end of a branch, the analyst can progressively satisfy each junction continuity equation while working upstream. After finding the discharges, the elevations H of the HGL are determined by starting where the HGL is known, in this case at node 1, and computing the head losses in the pipes that join this node; then the frictional head losses h_f are subtracted from the known values of H, etc. until all of the nodal heads have been determined. The pressures are then determined by subtracting the nodal elevations z from the heads H and multiplying this by the specific weight, i.e., $p = \gamma(H - z)$. The tables which follow present the computed values for this network:

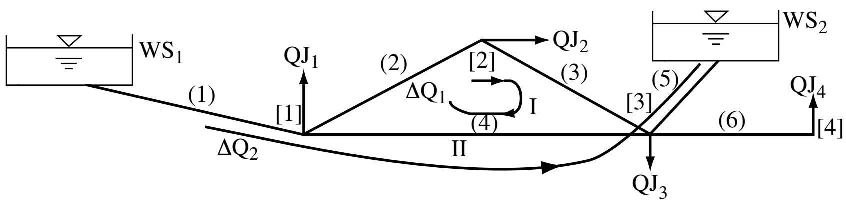
Pipe	Q, ft³/s	$h_f = KQ^n$, ft
1	2.5	22.395
2	1.7	15.922
3	0.5	4.398

Node	$H_{down} = H_{up} - h_f$, ft.	**Pressure**, lb/in^2
1	100.0 Given	34.67
2	100.0 - 22.395 = 77.61	27.13
3	77.61 - 15.922 = 61.69	19.37
4	61.69 - 4.398 = 57.29	18.76

* * *

Example Problem 4.2

Write the system of Q-equations for this network. In these equations use the parameters K_i and n_i, in which i is the pipe number.



Since two supply sources are present, four junction continuity equations are available. They are the following:

$$F_1 = Q_1 - Q_2 - Q_4 - QJ_1 = 0$$
$$F_2 = Q_2 - Q_3 - QJ_2 = 0$$
$$F_3 = Q_3 + Q_4 + Q_5 - Q_6 - QJ_3 = 0$$
$$F_4 = Q_6 - QJ_4 = 0$$

The number of energy loop equations is $NL = NP - NJ = 6 - 4 = 2$ (one pseudo and one real loop). These equations follow:

$$F_5 = K_2 Q_2^{n_2} + K_3 Q_3^{n_3} - K_4 Q_4^{n_4} = 0$$

$$F_6 = K_1 Q_1^{n_1} + K_4 Q_4^{n_4} - K_5 Q_5^{n_5} - WS_1 + WS_2 = 0$$

Since F_4 requires $Q_6 = QJ_4$, this dead end pipe could be removed from the network, and the demand at node 3 would then be changed to $QJ_3 + QJ_4$. These steps would reduce NP to 5 and NJ to 3, and they would eliminate any need for equation F_4. After the HGL elevation, H_3, at node 3 has been determined by solving this equation set, then H_4 can be found by computing h_{f6} and subtracting it from H_3.

* * *

4.2.2. SYSTEM OF H-EQUATIONS

If the elevation of the energy line or hydraulic grade line throughout a network is initially regarded as the primary set of unknown variables, then we develop and solve a system of H-equations. One H-equation is written at each junction (or at $NJ - 1$ junctions if fewer than two supply sources exist). Since looped pipe networks have fewer junctions than pipes, there will be fewer H-equations than Q-equations. Every equation in this smaller set is nonlinear, however, whereas the junction continuity equations are linear in the system of Q-equations.

To develop the system of H-equations, we begin by solving the exponential equation for the discharge in the form

$$Q_{ij} = (h_{f\,ij} / K_{ij})^{1/n_{ij}} = [(H_i - H_j)/K_{ij}]^{1/n_{ij}} \tag{4.11a}$$

Here the frictional head loss has been replaced by the difference in HGL values between the upstream and downstream nodes. In addition, in this equation a double subscript notation has been introduced; the first subscript defines the upstream node of the pipe, and the second subscript defines the downstream node. Thus Q_{ij} and K_{ij} denote the discharge and loss coefficient for the pipe from node i to node j. An alternative way of writing Eq. 4.11a is

$$Q_k = (h_{f\,k} / K_k)^{1/n_k} = [(H_i - H_j)/K_k]^{1/n_k} \tag{4.11b}$$

in which k is the pipe number.

Substituting Eq. 4.11 into the junction continuity equations, Eq. 4.3, yields

$$QJ_j - \sum\{[(H_i - H_j)/K_{ij}]^{1/n_{ij}}\}_{in} + \sum\{[(H_j - H_i)/K_{ij}]^{1/n_{ij}}\}_{out} = 0 \tag{4.12a}$$

in which the summations are over all pipes that flow to and from junction j, respectively. If it is desired to automate the choice of sign, then Eq. 4.12a can be written as

$$\begin{aligned} QJ_j &- \sum\{[(H_i - H_j)/K_{ij}]\left|(H_i - H_j)/K_{ij}\right|^{1/n_{ij}-1}\}_{in} \\ &+ \sum\{[(H_j - H_i)/K_{ij}]\left|(H_j - H_i)/K_{ij}\right|^{1/n_{ij}-1}\}_{out} = 0 \end{aligned} \tag{4.12b}$$

As an application of the H-equations with HGL-elevations at the nodes as the unknowns, consider the one-loop network in Fig. 4.3 which consists of three junctions and three pipes. In this network two independent continuity equations are available, and consequently the head at one of the junctions must be specified. In this case at node [1] the head loss in the pipe that connects the reservoir to the network can first be determined, and then this value can be subtracted from the reservoir water surface elevation to determine H_1.

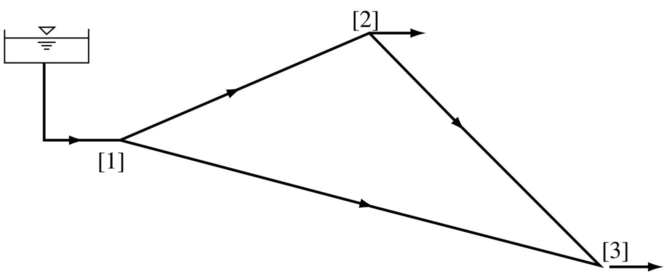


Figure 4.3 A network with three pipes and three junctions.

The two continuity equations are

$$\begin{aligned} &Q_{12} + Q_{13} = QJ_1 = QJ_2 + QJ_3 \\ &Q_{21} + Q_{23} = -\ QJ_2 \quad (or - Q_{12} + Q_{23} = -\ QJ_2) \end{aligned} \tag{4.13}$$

Although in the second equation the flow in pipe 1-2 is toward the junction, the discharge Q_{21} is not preceded by a minus sign since the notation 2-1 takes care of this. Alternatively the equations could have been written at junctions 2 and 3 instead of 1 and 2. Substituting Eq. 4.11 into these continuity equations gives the following two equations to determine H_2 and H_3:

$$\begin{aligned} \left[\frac{H_1 - H_2}{K_{12}}\right]^{1/n_{12}} + \left[\frac{H_1 - H_3}{K_{13}}\right]^{1/n_{13}} &= QJ_2 + QJ_3 \\ -\left[\frac{H_1 - H_2}{K_{12}}\right]^{1/n_{12}} + \left[\frac{H_2 - H_3}{K_{23}}\right]^{1/n_{23}} &= -\, QJ_2 \end{aligned} \tag{4.14}$$

Since a negative value cannot be raised to a power, a minus sign must precede any term in which the subscript notation opposes the direction of flow. Systems of these equations will be called H-equations, since the HGL-elevations are the primary unknowns. After the heads H are found, then the discharge in each pipe can be obtained from Eq. 4.11.

Example Problem 4.3

Write the system of H-equations for the network in Example Problem 4.2.

Refer to the figure in Example Problem 4.2. Only the junction continuity equations are used in forming the H-equations, and each Q_i is replaced by $[(H_{ui} - H_{di})/K_i]^{1/n_i}$, in which subscript u is the upstream node and subscript d is the downstream node. The system is

$$F_1 = \left[\frac{WS_1 - H_1}{K_1}\right]^{1/n_1} - \left[\frac{H_1 - H_2}{K_2}\right]^{1/n_2} - \left[\frac{H_1 - H_3}{K_4}\right]^{1/n_4} - QJ_1 = 0$$

$$F_2 = \left[\frac{H_1 - H_2}{K_2}\right]^{1/n_2} - \left[\frac{H_2 - H_3}{K_3}\right]^{1/n_3} - QJ_2 = 0$$

$$F_3 = \left[\frac{H_2 - H_3}{K_3}\right]^{1/n_3} + \left[\frac{H_1 - H_3}{K_4}\right]^{1/n_4} + \left[\frac{WS_2 - H_3}{K_5}\right]^{1/n_5} - \left[\frac{H_3 - H_4}{K_6}\right]^{1/n_6} - QJ_3 = 0$$

$$F_4 = \left[\frac{H_3 - H_4}{K_6}\right]^{1/n_6} - QJ_4 = 0$$

* * *

4.2.3. SYSTEM OF ΔQ-EQUATIONS

The number of ΔQ-equations is normally about half the number of H-equations for a network. This reduction in number is not necessarily an advantage, since all of the equations are nonlinear and may contain many terms. These equations consider the loop corrective discharges or ΔQ's as the primary unknowns. These corrective discharges or ΔQ's will be determined from the energy equations that are written for NL loops in the network, and thus NL of these corrective discharge equations must be developed. To obtain these equations, we replace the discharge in each pipe of the network by an initial discharge, denoted by Q_{oi}, plus the sum of all of the initially unknown corrective discharges that circulate through pipe i, or

$$Q_i = Q_{oi} + \sum \Delta Q_k \tag{4.15}$$

in which the summation includes all of the corrective discharges passing through pipe i. The initial discharges Q_{oi} must satisfy all of the junction continuity equations. It is not difficult to establish the initial discharge in each pipe so that the junction continuity equations are satisfied. However, these initial discharges usually will not satisfy the energy equations that are written around the loops of the network.

Equation 4.15 is based on the fact that any adjustment can be added (accounting for sign) to the initially assumed flow in each pipe in a loop of the network without violating continuity at the junctions. It is very important to understand the validity of this decompo-sition; it may help to note that any ΔQ entering a junction as it flows around a loop must also leave that junction, and vice versa (See Fig. 4.4). Because of this fact, we decide

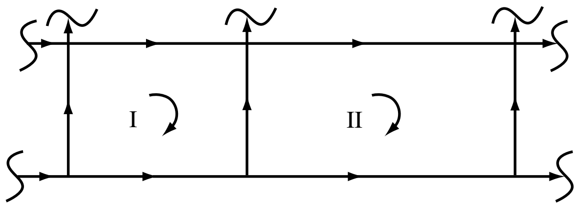


Figure 4.4 A two-loop portion of a network.

to establish NL energy loop equations around the NL loops of the network, in which each initial discharge plus the sum of corrective loop discharges $\Sigma \Delta Q_k$ is used as the discharge. The junction continuity equations are satisfied by the initial discharges Q_{oi} and are not a part of the system of equations. The corrective discharges can be chosen as positive if they circulate around a loop in either the clockwise or counterclockwise direction. It is necessary to be consistent within any one loop, but the sign convention may change from loop to loop, if desired. A corrective discharge adds to the flow Q_{oi} in pipe i if it is in the same direction as the pipe flow, and it subtracts from the initial discharge if it is in the opposite direction.

To summarize how the ΔQ-equations are obtained, replace the Q's in the energy loop equations, Eqs. 4.4 and 4.5, by

$$Q_i = Q_{oi} \pm \sum \Delta Q_k \tag{4.16}$$

Here the summation includes all corrective discharges which pass through pipe i, and the plus sign is used if the net corrective discharge and pipe flow are in the same direction; otherwise the minus sign is used before the summation. Thus Eqs. 4.4 and 4.5 become

$$\sum K_i \left\{ Q_{oi} \pm \sum \Delta Q_k \right\}^{n_i} = 0 \quad \text{for real loops} \tag{4.17a}$$

and

$$\sum K_i \left\{ Q_{oi} \pm \sum \Delta Q_k \right\}^{n_i} = \Delta WS \quad \text{for pseudo loops} \tag{4.18a}$$

To automate the choice of sign, these equations can be rewritten as

$$\sum K_i \left\{ Q_{oi} \pm \sum \Delta Q_k \right\} \left| Q_{oi} \pm \sum \Delta Q_k \right|^{n_i - 1} = 0 \quad \text{for real loops} \tag{4.17b}$$

and $$\sum K_i \left\{Q_{oi} \pm \sum \Delta Q_k\right\} \left|Q_{oi} \pm \sum \Delta Q_k\right|^{n_i - 1} = \Delta WS \quad \text{for pseudo loops} \tag{4.18b}$$

To illustrate the development of the system of ΔQ-equations, consider the network in Fig. 4.5. If the Q-equations were used, there would be five junction continuity equations and two loop equations, a total of seven equations. If the H-equations were used, there would be an equation for the HGL-elevation at each of the five nodes where the head is unknown (The head at one node must be known.). But there will be only two

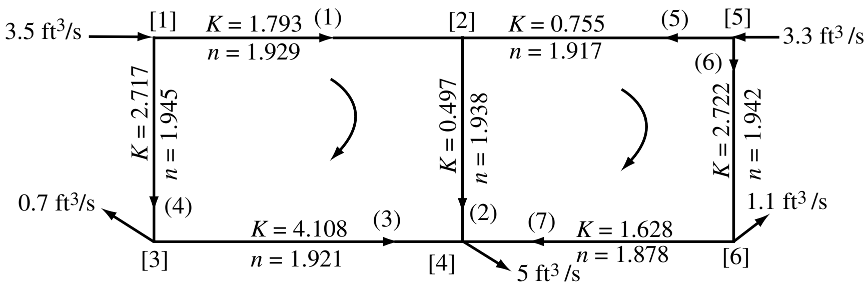


Figure 4.5 A seven-pipe network.

ΔQ-equations, one for each real loop in this network. These two equations are

$$\begin{aligned} F_1 &= K_1\left(Q_{o1} + \Delta Q_1\right)^{n_1} + K_2\left(Q_{o2} + \Delta Q_1 - \Delta Q_2\right)^{n_2} \\ &\quad - K_3\left(Q_{o3} - \Delta Q_1\right)^{n_3} - K_4\left(Q_{o4} - \Delta Q_1\right)^{n_4} = 0 \\ F_2 &= -K_5\left(Q_{o5} - \Delta Q_2\right)^{n_5} + K_6\left(Q_{o6} + \Delta Q_2\right)^{n_6} \\ &\quad + K_7\left(Q_{o7} + \Delta Q_2\right)^{n_7} - K_2\left(Q_{o2} + \Delta Q_1 - \Delta Q_2\right)^{n_2} = 0 \end{aligned} \tag{4.19}$$

The next step is to find an initial estimate for the discharge in each pipe that will satisfy all of the junction continuity equations. One possible set of initial discharges is Q_{o1} = 1.75 ft³/s, Q_{o2} = 3.55 ft³/s, Q_{o3} = 1.05 ft³/s, Q_{o4} = 1.75 ft³/s, Q_{o5} = 1.8 ft³/s, Q_{o6} = 1.5 ft³/s, and Q_{o7} = 0.4 ft³/s. When these initial discharges and the parameters that are listed on the network sketch are substituted into Eqs. 4.19, the following two equations result; their solution will yield values for the two unknowns ΔQ_1 and ΔQ_2:

$$\begin{aligned} F_1 &= 1.793\left(1.75 + \Delta Q_1\right)^{1.929} + 0.497\left(3.55 + \Delta Q_1 - \Delta Q_2\right)^{1.938} \\ &\quad - 4.108\left(1.05 - \Delta Q_1\right)^{1.921} - 2.717\left(1.75 - \Delta Q_1\right)^{1.945} = 0 \\ F_2 &= -0.755\left(1.8 - \Delta Q_2\right)^{1.917} + 2.722\left(1.5 + \Delta Q_2\right)^{1.942} \\ &\quad + 1.628\left(0.4 + \Delta Q_2\right)^{1.878} - 0.497\left(3.55 + \Delta Q_1 - \Delta Q_2\right)^{1.938} = 0 \end{aligned} \tag{4.20}$$

Upon obtaining the solution to these two equations for the two unknowns, ΔQ_1 and ΔQ_2, the discharge in each pipe can easily be determined by adding these loop corrective discharges to the initial discharges. From these discharges the head loss in each pipe can be determined, and from these values the head at each node can be found.

The nonlinearities in these systems of equations create difficulties when we seek the solution. Later in the chapter we apply the Newton method to overcome this problem.

Example Problem 4.4

Write the system of ΔQ-equations for the network depicted in Example Problem 4.2.

The ΔQ-equations are based on the energy loop equations alone. Therefore these equations can be obtained by taking the equations for F_5 and F_6 directly from Example Problem 4.3 and replacing each discharge Q_i by $Q_{oi} \pm \sum \Delta Q_k$. The ΔQ-equations are

$$F_5 = K_2\left(Q_{o2} + \Delta Q_1\right)^{n_2} + K_3\left(Q_{o3} + \Delta Q_1\right)^{n_3} - K_4\left(Q_{o4} - \Delta Q_1 + \Delta Q_2\right)^{n_4} = 0$$

$$F_6 = K_1\left(Q_{o1} + \Delta Q_2\right)^{n_1} + K_4\left(Q_{o4} - \Delta Q_1 + \Delta Q_2\right)^{n_4} - K_5\left(Q_{o5} - \Delta Q_2\right)^{n_5} - WS_1 + WS_2 = 0$$

With the writing of the ΔQ-equations we must also provide values for Q_{oi} that satisfy all of the junction continuity equations. For this purpose we assume that all four demands are equal to 1.0. Then the following values could serve as an acceptable initialization of the discharges: $Q_{o1} = 3$, $Q_{o2} = 1$, $Q_{o3} = 0$, $Q_{o4} = 1$, $Q_{o5} = 1$, and $Q_{o6} = 1$.

* * *

4.3 PRESSURE REDUCTION AND BACK PRESSURE VALVES

A pressure-reducing valve (PRV) is designed to maintain a constant pressure at its downstream side, independent of the value of the upstream pressure at the device. Only two exceptions to the maintenance of this downstream pressure exist: (1) if the upstream pressure becomes less than the valve setting; or (2) if the downstream pressure exceeds the pressure setting of the valve so that flow would occur in the upstream direction if the PRV were not present. If the first exception occurs, the valve has no effect on flow conditions except to create a local loss; generally its effect is then like a globe valve in dissipating additional head beyond the friction loss in that line. If the second condition occurs, then the PRV acts as a check valve, preventing a reverse flow in the line. Then the PRV allows the pressure immediately downstream from the valve to exceed its pressure setting. In this way PRV's reduce pressures in portions of a pipe distribution system if the pressure would otherwise be excessive, and they may also be used to control the choice of a supply source in response to various demand levels. In the latter applications the PRV acts as a check valve until the pressure is reduced to a critical level by large demands, at which time additional supply sources become available.

A back-pressure valve (BPV) is designed to maintain a constant pressure upstream from it, independent of the value of the downstream pressure. Like a PRV there are exceptions to this normal mode of operation. Should the upstream pressure become less than the pressure setting, the valve can not maintain the pressure setting since it is not an energy-creating device, and the most it can do is shut down the flow in its line. Should the flow want to reverse direction from the positive flow direction through the valve, the valve opens completely and acts as a local loss device. A BPV is used in situations where the pressure would otherwise become too low in elevated portions of the network. Such a situation arises, for example, where a pump is needed to sustain adequate pressures in a higher part of a network but is not needed in the lower portions of the network; without a BPV, or possibly several BPV's, the flow pattern might then lead to discharges through pressure relief valves in the lower portions of the network and possibly create excessively large pressures in the lower region.

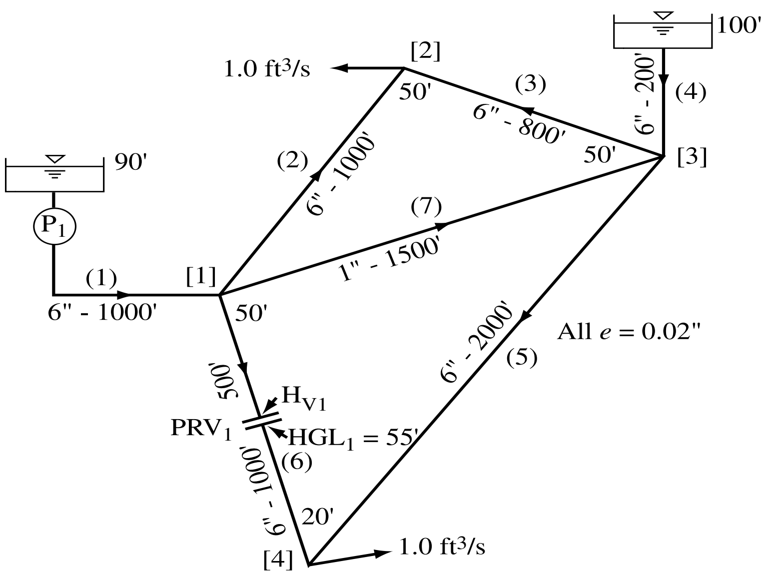
The equations that describe the behavior of a pipe network that contains PRV's or BPV's must include new features to account properly for the effects of these valves on the discharges and pressures throughout the network. Furthermore, the analysis of a pipe network with such devices must be able to determine the correct operational conditions, i.e., determine whether the PRV's and BPV's are operating in their normal modes or in one of their exceptional modes. Methods for adding such devices into network analyses are described in these sections. The discussion begins with pressure reduction valves.

Underlying the writing of the three systems of equations described in Section 4.2 is the basic assumption that a relation exists between the magnitude of the discharge in a pipe and the amount of the head loss, or head difference, between the ends of this pipe. Such a relation does not exist if a PRV (or a BPV) is present in the pipe. Therefore a pipe with a PRV in it should not appear in a normal energy loop equation. However, in the usual mode of operation for a PRV a constant head is maintained at its downstream end; in this way it behaves like a reservoir. Furthermore, regardless of its mode of operation the discharge at the upstream node of a pipe containing a PRV will be the same as the discharge at the downstream node of this pipe. The details of developing a proper system of equations to describe a network containing one or more PRV's are different, depending upon whether a system of Q-equations, H-equations, or ΔQ-equations are desired. Therefore, each of these will be described in a separate section.

4.3.1. *Q*-EQUATIONS FOR NETWORKS WITH PRV'S/BPV'S

The procedure for developing the Q-equation system for a network containing PRV's is as follows: (1) write the junction continuity equations in the usual manner, ignoring the PRV's; (2) replace each PRV with an artificial reservoir which has a water surface elevation equal to the HGL-elevation that is the sum of the pressure head set on the PRV and its elevation in the pipeline; finally (3) write the energy equations around the loops of this modified network. The resulting equations describe the normal mode of operation.

Let's try this procedure on the seven-pipe network shown in Fig. 4.6, in which a PRV exists in pipe 6, located 500 ft. downstream from node 1, the upstream end of this pipe. Since a PRV is a directional device, we must always identify the upstream and downstream



Q ft³/s	Head ft.
1.0	60
1.5	55
2.0	48

Figure 4.6 A seven-pipe network.

ends of the pipe containing it. The system of Q-equations for this network consists of four junction continuity equations and three energy loop equations. According to the usual rules, an independent junction continuity equation can be written for each of the four junctions since there are two supply sources for this network. These junction continuity equations are

$$\begin{aligned} F_1 &= -Q_1 + Q_2 + Q_6 + Q_7 = 0 \\ F_2 &= 1.0 - Q_2 - Q_3 = 0 \\ F_3 &= Q_3 - Q_4 + Q_5 - Q_7 = 0 \\ F_4 &= 1.0 - Q_5 - Q_6 = 0 \end{aligned} \tag{4.21}$$

These continuity equations are unaffected by the presence or absence of a PRV in the network. We next modify the network so the upstream portion of the pipe containing the PRV is removed and the PRV is replaced by a reservoir with a water surface elevation equal to the HGL of the pressure setting of the PRV (see Fig. 4.7). Of the three loops that exist in this modified network, only one is a real loop which traverses pipes 2, 3, and 7. Two pseudo loops also exist. One pseudo loop connects the two original supply sources. This loop can start at the reservoir and end at the source pump so it includes pipes 4, 7, and 1. The second loop must extend from the artificial reservoir created by the PRV to one of the other supply sources (or another artificial reservoir, if two or more PRV's exist in the network). The shortest path for this second pseudo loop will traverse pipes 4, 5, and 6. In writing the head loss in pipe 6, only that portion of the pipe downstream from the PRV is used. A modified loss coefficient K' will be used to denote this change in the exponential formula. The new coefficient K' equals the K for the pipe containing the PRV, multiplied by the ratio of the pipe length from the PRV to the pipe's downstream end divided by the total pipe length, or

$$K' = K(L_d / L) \tag{4.22}$$

or in this example $K_6' = K_6(500/1000) = 0.5K_6$.

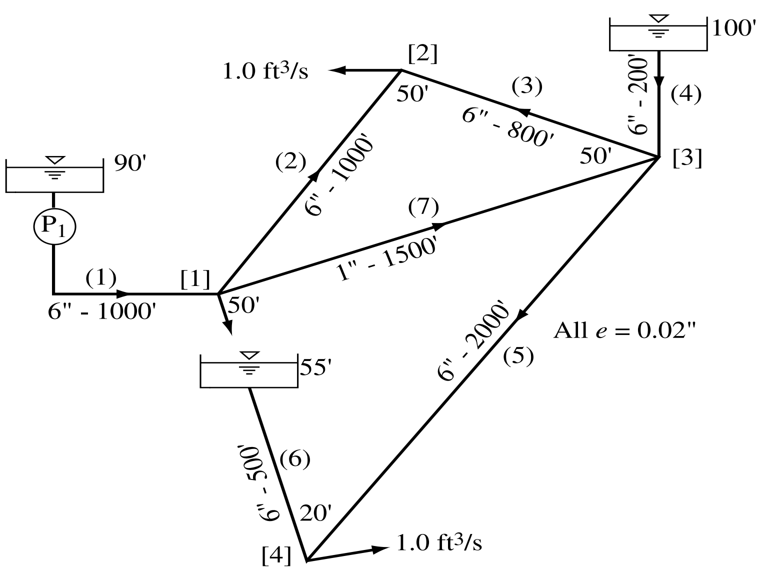


Figure 4.7 The modified seven-pipe network.

The energy equations are

$$F_5 = K_2Q_2^{n_2} - K_3Q_3^{n_3} - K_7Q_7^{n_7} = 0 \ (real \ \ loop)$$

$$F_6 = K_4Q_4^{n_4} - K_7Q_7^{n_7} - K_1Q_1^{n_1} + h_{p1} - 100 + 90 = 0 \ (pseudo \ \ loop) \tag{4.23}$$

$$F_7 = K_4Q_4^{n_4} + K_5Q_5^{n_5} - K_6'Q_6^{n_6} + 55 - 100 = 0 \ (pseudo \ \ loop)$$

The head produced by the pump h_{p1} can be defined by a second-order polynomial passing through three points of the pump curve, or

$$h_{p1} = AQ_1^2 + BQ_1 + C \tag{4.24}$$

We have now formed seven independent equations that contain the seven unknown discharges $Q_1, Q_2, \ldots, Q_7$. In this example the real loop that was lost by having a PRV in pipe 6 is replaced by an additional pseudo loop. We see that the number of equations again equals the number of unknown discharges.

To obtain a solution for this network by using the computer program NETWK, the input data can be prepared (see the NETWK user manual for input data requirements or the condensed description of this input on the CD) as listed in Fig. 4.8. The solution tables from NETWK are reproduced in Fig. 4.9. A study of this output will show that the PRV is operating in its normal mode of operation.

```
Example of a network containing a PRV
/* 6 1 4 1000              RESER
PIPES                      7 1 3 1500 1               4 100
1 0 1 1000 6 0.02          NODES                      PUMPS
2 1 2                      1 0 50                     1 1 60 1.5 55 2 48 90
3 3 2 800                  2 1                        VALVE
4 0 3 200                  3 0                        6 500 55
5 3 4 2000                 4 1 20                     RUN
```

Figure 4.8 Input data for the network shown in Figs. 4.6 and 4.7.

This solution indicates that the PRV is operating in its normal mode of maintaining the set pressure at its downstream end because the reported downstream HGL-elevation equals the value specified in the input data. If this had not been the case, the solution from NETWK would have indicated either that the PRV had shut off the flow in pipe 6 or that it was completely open and replaced by a minor loss. In solving the network equations, if the discharge in pipe 6 had been negative, then the program would have noted that the PRV would act as a check valve, preventing a reverse flow. If this situation should occur, then the network problem would be altered so it would only have six pipes instead of seven (pipe 6 would not exist in this modified network). The equations describing the flows in

```
LOSSES DUE TO FLUID FRICTION IN ALL PIPES
POWER LOSS  =  11.51 H.P.  =  8.585 KWATTS.
ENERGY LOSS  =  206.0  KWHRS/DAY

PUMPS:
PIPE   HEAD     Q   HORSEPOWER   KILOWATT   KWATT-HRS/DAY
 1     59.1   1.11     7.43        5.54        133.0
ELEVATION OF HGL UPSTREAM AND DOWNSTREAM OF PRVS:
```

Figure 4.9 Output tables from NETWK.

```
   PIPE    UPSTREAM  DOWNSTREAM
              HGL       HGL
    6       121.79    55.00
UNITS OF SOLUTION ARE:
DIAMETER,     inch
LENGTH,       feet
HEAD,         feet
ELEVATION,    feet
PRESSURE,     lb/in2
DISCHARGE,    ft3/s
DARCY-WEISBACH FORMULA USED FOR COMPUTING HEAD LOSSES
```

PIPE DATA

PIPE NO.	NODES FROM	TO	L ft.	DIA. in	e x10³ in	Q ft³/s	VEL. ft/s	HEAD LOSS ft.	HLOSS/ 1000
1	0	1	1000	6.0	20.0	1.11	5.65	27.28	27.28
2	1	2	1000	6.0	20.0	1.07	5.43	25.26	25.26
*3	2	3	800	6.0	20.0	0.07	0.34	0.10	0.12
4	0	3	200	6.0	20.0	0.89	4.54	3.55	17.74
5	3	4	2000	6.0	20.0	0.96	4.91	41.47	20.74
6	1	4	1000	6.0	20.0	0.04	0.18	0.04	0.04
7	1	3	1500	1.0	20.0	0.01	1.31	25.56	17.04

AVE. VEL. = 3.19 ft/s, AVE. HL/1000 = 15.46, MAX. VEL. = 5.65 ft/s, MIN. VEL. = 0.18 ft/s
*Flow direction is reversed from input data.

NODE DATA

NODE	DEMAND ft³/s	DEMAND gal/min	ELEV. ft.	HEAD ft.	PRESSURE lb/in²	HGL ELEV. ft.
1	0.00	0.0	50.	71.81	31.1	121.81
2	1.00	448.8	50.	46.55	20.2	96.55
3	0.00	0.0	50.	46.45	20.1	96.45
4	1.00	448.8	20.	34.98	15.2	54.98

AVE. HEAD = 49.95 ft., AVE. HGL = 92.45 ft.
MAX. HEAD = 71.81 ft., MIN HEAD = 34.98 ft.

Figure 4.9, concluded. Output tables from NETWK.

this modified network would consist of the original equations with the last one omitted. If the HGL elevation at node 1 had been lower than the HGL setting of the PRV, then it would be known that the PRV would not be able to sustain its pressure setting, and the network problem must then be solved by using equations that replace the PRV with a minor loss device. For this last mode of operation the last energy equation would be replaced by a real loop equation traversing pipes 5, 6, and 7. Pipe 6 would contain a minor loss device to represent the PRV as being fully open.

The procedure for writing the system of Q-equations should now be apparent for back-pressure valves (BPV's) in networks. As with PRV's, the junction continuity equations are written ignoring the presence of BPV's. The junction continuity equations are unaffected by the existence of a BPV. In writing the energy equations, the upstream side of each BPV is replaced by an artificial reservoir; in each case the pipe segment from the downstream end of the BPV to its downstream node is then removed, and the energy equations are written for this revised network.

The writing of a system of Q-equations will be illustrated with the network in Fig. 4.10, which has 9 pipes and 6 nodes, is supplied by a source pump and has two tanks

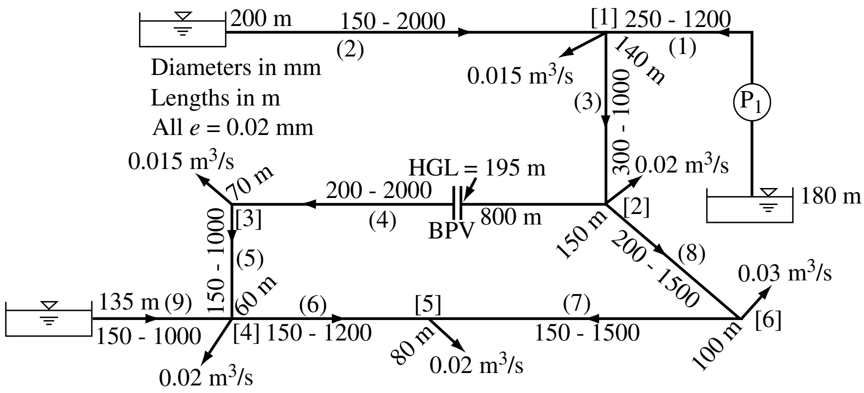


Figure 4.10 A 9-pipe, 6-node network.

(reservoirs) connected to it. Without a BPV (or some other device) this network would cause the lower reservoir at the end of pipe 9 to overflow. There are six junctions in this network. The corresponding six junction continuity equations are

$$
\begin{aligned}
F_1 &= 0.015 - Q_1 - Q_2 + Q_3 = 0 \\
F_2 &= 0.020 - Q_3 + Q_4 + Q_8 = 0 \\
F_3 &= 0.015 - Q_4 + Q_5 = 0 \\
F_4 &= 0.020 - Q_5 + Q_6 - Q_9 = 0 \\
F_5 &= 0.020 - Q_7 - Q_6 = 0 \\
F_6 &= 0.030 - Q_8 + Q_7 = 0
\end{aligned}
\tag{4.25}
$$

Before forming the loops around which the energy equations are written, an artificial reservoir is placed on the upstream side of the BPV with a water surface elevation equal to the HGL resulting from the pressure setting of the valve. The pipe downstream from the BPV is removed. When these changes are completed, the network appears as in Fig. 4.11, and energy equations can next be written around the loops of this modified network. Three loops are needed, since NL = NP - NJ = 9 - 6 = 3. These are all pseudo loops and may be

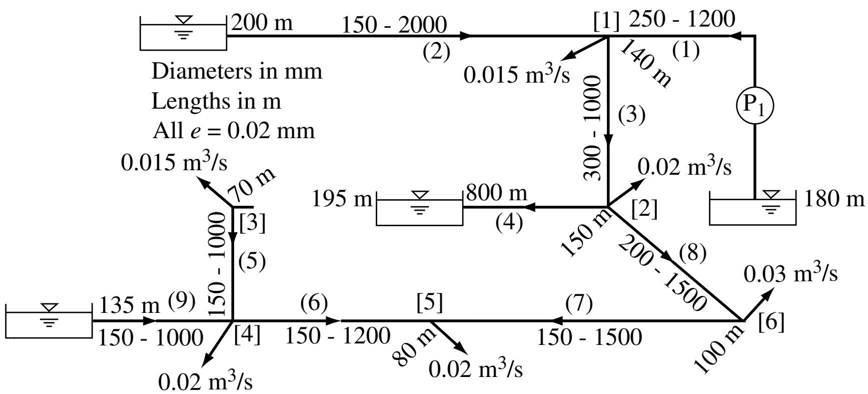


Figure 4.11 The modified 9-pipe, 6-node network.

composed in the following way: the pipes in loop 1 are 1 and 2; the pipes in loop 2 are 1, 3, and 4 (upstream portion); and the pipes in loop 3 are 9, 6, 7, 8, and 4 (upstream portion). It is incorrect to write a loop through pipes 9, 5, and 4 (the upstream portion) because a BPV sets the pressure on its upstream side. Hence the energy equations in the Q-equation system are the following:

$$
\begin{aligned}
F_7 &= K_1 Q_1^{n_1} - h_{p1} - K_2 Q_2^{n_2} - 180 + 200 = 0 \\
F_8 &= K_1 Q_1^{n_1} - h_{p1} + K_3 Q_3^{n_3} + K_4' Q_4^{n_4} - 180 + 195 = 0 \\
F_9 &= K_9 Q_9^{n_9} + K_6 Q_6^{n_6} - K_7 Q_7^{n_7} - K_8 Q_8^{n_8} + K_4' Q_4^{n_4} - 135 + 195 = 0
\end{aligned}
\tag{4.26}
$$

One possible input file to NETWK for the solution of this problem is presented in Fig. 4.12, and the resulting solution tables are presented in Fig. 4.13.

```
Network Containing BPV
/* 2 .02 140
$SPECIF NFLOW=3,NPGPM=3,NUNIT=4 $END
PIPES
1 0 1 1200 250 .02
2 0 1 2000 150
3 1 2 1000 300
4 2 3 2000
5 3 4 1000 150
6 4 5 1200
7 6 5 1500
8 2 6 1500 200
9 0 4 1000 150
NODES
1 .015 140

3 .015 70
4 .02 60
5 .02 80
6 .03 100
RESER
2 200
9 135
PUMPS
1 .1 35 .15 32 .2 28 180
BPVALVE
4 1200 195
RUN
```

Figure 4.12 The input data file to NETWK for the 9-pipe, 6-node network.

From this solution we see that the BPV dissipates 65.88 m of head to sustain the upstream HGL setting of 195 m. This value is obtained by subtracting the downstream HGL from the BPV setting. It is a worthwhile exercise to begin with the head losses in the PIPE DATA table and verify the HGL elevations reported in the NODE DATA table; it will lead to a better understanding of the BPV and its effect on pressures and discharges in this network as the BPV operates in its normal mode. If the solution had shown a negative flow through pipe 4, then the downstream pressure would actually be larger than the BPV setting, and the valve would open up completely. For this occurrence the BPV must be re-placed by a minor loss device, and then this modified network problem could be studied. If the HGL at node 2 (the node immediately upstream from the BPV) were less than the HGL established by the BPV setting, then the BPV would close completely. The pipe

```
LOSSES DUE TO FLUID FRICTION IN ALL PIPES
POWER LOSS  =  65.18 H.P.  =  48.63 KWATTS.
ENERGY LOSS  =  1167.0  KWHRS/DAY

PUMPS:
PIPE   HEAD     Q   HORSEPOWER   KILOWATT   KWATT-HRS/DAY
 1     34.88   10.0    46.9        35.0        839.8
HGL DOWNSTREAM AND UPSTREAM FROM BPV
 4     129.12        195.00
```

Figure 4.13 The output tables from NETWK for the 9-pipe, 6-node network.

PIPE DATA

PIPE NO.	NODES FROM	TO	L	DIA.	e x10³	Q	VEL.	HEAD LOSS	HLOSS/ 1000
			m	mm	mm	m³/s	m/s	m	
1	0	1	1200	250	20.0	0.102	2.09	15.58	12.98
2	0	1	2000	150	20.0	0.004	0.21	0.75	0.37
3	1	2	1000	300.	20.0	0.091	1.29	4.28	4.28
4	2	3	2000	300.	20.0	0.006	0.08	0.06	0.03
5	4	3	1000	150.	20.0	0.009	0.52	1.90	1.89
6	5	4	1200	150.	20.0	0.015	0.86	5.69	4.74
7	6	5	1500	150.	20.0	0.035	2.00	33.11	22.07
8	2	6	1500	200.	20.0	0.065	2.08	25.25	16.83
9	0	4	1000	150.	20.0	0.014	0.79	4.03	4.03

AVE. VEL. = 1.10 m/s, AVE. HL/1000 = 7.47, MAX. VEL. = 2.09 m/s, MIN. VEL. = 0.08 m/s

NODE DATA

NODE	DEMAND m³/s	DEMAND ft³/s	ELEV. m	HEAD m	PRESSURE kPa	HGL ELEV. m
1	0.015	0.53	140.0	59.25	580.7	199.25
2	0.020	0.71	140.0	55.02	539.2	195.02
3	0.015	0.53	70.0	59.08	579.0	129.08
4	0.020	0.71	60.0	70.97	695.6	130.97
5	0.020	0.71	80.0	56.66	555.3	136.66
6	0.030	1.06	100.0	69.78	683.8	169.78

AVE. HEAD = 61.79 m, AVE. HGL = 160.13 m
MAX. HEAD = 70.97 m, MIN. HEAD = 55.02 m

Figure 4.13 (Concluded) The output tables from NETWK for the 9-pipe, 6-node network.

containing the BPV should then be removed from the network, and the problem could then be solved by using the equations for this modified network; then the BPV could not maintain the pressure setting, and it would simply prevent any flow from passing through the pipe in which it is installed.

4.3.2. *H*-EQUATIONS FOR NETWORKS WITH PRV'S/BPV'S

The procedure for writing the *H*-equations for a network that contains PRV's and/or BPV's is described here. First, view the HGL resulting from the pressure setting of the device as a reservoir, since under normal operation the HGL is fixed by the device. Second, place an additional unknown variable on the other side of the device to represent the elevation of the HGL there. We will denote this variable by H_{vi}, in which i is the number of the device. For the first PRV or BPV $i = 1$, for the second $i = 2$, etc. For a PRV the value of H_{vi} is the HGL-elevation immediately upstream from the valve, whereas H_{vi} is the HGL-elevation immediately downstream from the valve for a BPV. Third, the junction continuity equations are written in the usual way, with the difference between the upstream and downstream HGL-elevations, divided by K for this pipe, all raised to the reciprocal of the discharge exponent n, i.e., $Q_k = \{(H_i - H_j)/K_k\}^{1/n_k}$. Finally, since an additional unknown is introduced for each PRV or BPV, one additional equation must be added to the system of continuity equations for each device. These additional equations are obtained by noting that the head losses in the upstream and downstream portions of the pipe containing the device are proportional to these two lengths. For a PRV this equation is

$$(HGL - H_d)L_u - (H_u - H_{vi})L_d = 0 \tag{4.27}$$

in which L_u and L_d are the lengths upstream and downstream from the device, respectively, and H_d and H_u are the HGL-elevations at the downstream and upstream ends of pipe i containing the device.

Using again the network depicted previously in Fig. 4.6 to illustrate the formation of the H-equation system, we would first modify the network as shown in Fig. 4.14:

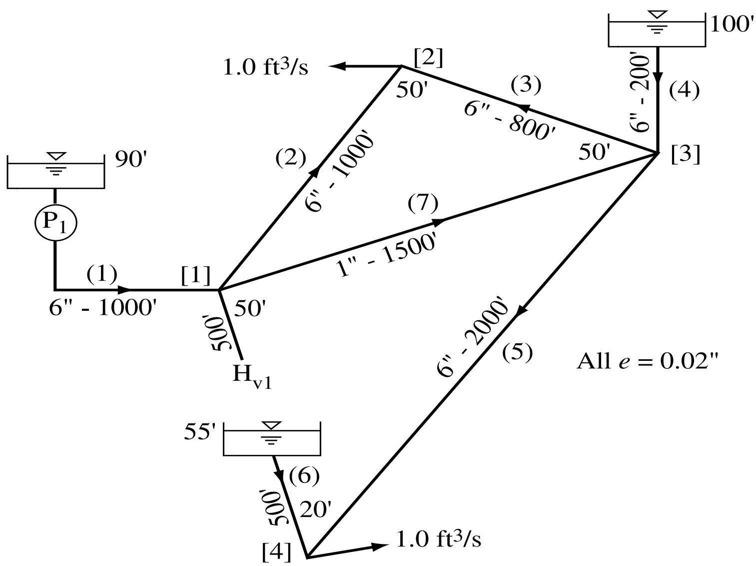


Figure 4.14 A 9-pipe, 6-node network containing a BVP, as modified.

The final H-equations for this network are the following:

$$\begin{aligned}
&-\{(90 + h_{p1} - H_1)/K_1\}^{1/n_1} + \{(H_1 - H_2)/K_2\}^{1/n_2} \\
&\qquad + \{(H_1 - H_{v1})/K_6''\}^{1/n_6} + \{(H_1 - H_3)/K_7\}^{1/n_7} = 0 \\
&1.0 - \{(H_1 - H_2)/K_2\}^{1/n_2} - \{(H_3 - H_2)/K_3\}^{1/n_3} = 0 \\
&\{(H_3 - H_2)/K_3\}^{1/n_3} - \{(100 - H_3)/K_4\}^{1/n_4} \\
&\qquad + \{(H_3 - H_4)/K_5\}^{1/n_5} - \{(H_1 - H_3)/K_7\}^{1/n_7} = 0 \\
&1.0 - \{(H_3 - H_4)/K_5\}^{1/n_5} - \{(HGL_1 - H_4)/K_6'\}^{1/n_6} = 0 \\
&(HGL_1 - H_4)L_u - (H_1 - H_{v1})L_d = (HGL_1 - H_4) - (H_1 - H_{v1}) = 0
\end{aligned} \tag{4.28}$$

in which K_6'' and K_6' are the coefficients for the upstream and downstream portions of pipe 6, respectively.

The network in Fig. 4.10 that contains the BPV should now be viewed as shown in Fig. 4.15, and the H-equation system for this network is presented as Eqs. 4.29.

$$F_1 = 0.015 - \{(180 + h_{p1} - H_1)/K_1\}^{1/n_1}$$
$$- \{(200 - H_1)/K_2\}^{1/n_2} + \{(H_1 - H_2)/K_3\}^{1/n_3} = 0$$
$$F_2 = 0.020 - \{(H_1 - H_2)/K_3\}^{1/n_3}$$
$$+ \{(H_2 - HGL_1)/K_4''\}^{1/n_4} + \{(H_2 - H_6)/K_8\}^{1/n_8} = 0$$
$$F_3 = 0.015 - \{(H_{v1} - H_3)/K_4'\}^{1/n_4} + \{(H_3 - H_4)/K_5\}^{1/n_5} = 0$$
$$F_4 = 0.020 - \{(H_3 - H_4)/K_5\}^{1/n_5}$$
$$+ \{(H_4 - H_5)/K_6\}^{1/n_6} - \{(135 - H_4)/K_9\}^{1/n_9} = 0 \qquad (4.29)$$
$$F_5 = 0.020 - \{(H_4 - H_5)/K_6\}^{1/n_6} - \{(H_6 - H_5)/K_7\}^{1/n_7} = 0$$
$$F_6 = 0.020 - \{(H_2 - H_6)/K_8\}^{1/n_8} + \{(H_6 - H_5)/K_7\}^{1/n_7} = 0$$
$$F_7 = 1200(H_2 - HGL_1) - 800(H_{v1} - H_3) = 0$$

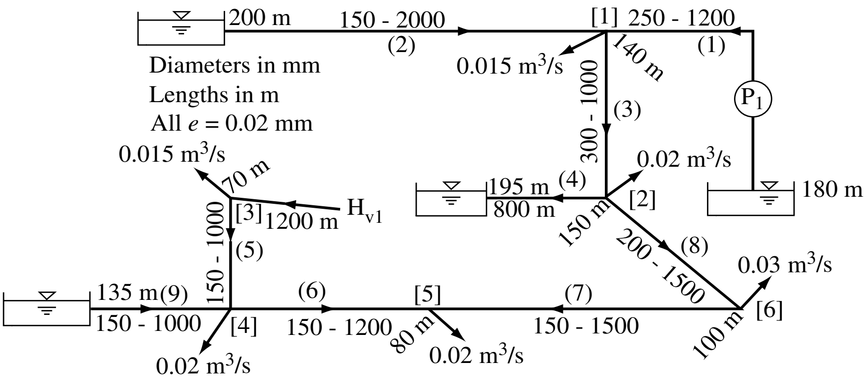


Figure 4.15 A seven-pipe network, modified.

4.3.3. ΔQ-EQUATIONS FOR NETWORKS WITH PRV'S/BPV'S

Let us begin this section by reviewing the underlying concept that is used in writing the ΔQ-equations: if the junction continuity equations are satisfied by the initial discharges Q_{oi}, then a corrective loop discharge, ΔQ, can flow around a loop without violating the principle that the discharge into all junctions will still equal the discharge out of these junctions, regardless of the magnitude of ΔQ. These corrective loop discharges can be regarded as the primary unknowns, and the resulting solution to the system of equations will produce discharges that also satisfy the energy equations around the loops. Therefore the discharges Q_i in the Q-equation loops were replaced by $Q_{oi} \pm \sum \Delta Q_k$. For the junction continuity equations to remain valid for any values of ΔQ_k, these corrective loop discharges must circulate around loops that are formed before any PRV's or BPV's are converted into artificial reservoirs. Thus it is necessary to consider two sets of loops with this method. The first set is the set of loops around which the ΔQ's circulate, and the second set is the set of loops that is used in writing the energy equations. These two sets of loops will be called the corrective discharge or ΔQ loops, and the energy loops. The ΔQ loops are formed while ignoring the existence of PRV's or BPV's. These devices are later replaced by artificial reservoirs, and the energy e quations are written for this modified network. Thus the energy set of loops will always contain more pseudo loops than does the ΔQ set of loops by the number of PRV's and/or BPV's that exist in the network. To track these two separate sets of loops in figures, the ΔQ loops will list ΔQ_i by the arc denoting the loop, and the energy loops will be numbered by roman numerals I, II, etc.

To illustrate the writing of the ΔQ-equations, we examine again the network with a PRV that is in Fig. 4.6. This network is redrawn below in Fig. 4.16 to display both the corrective discharge loops and the energy loops. To emphasize that ΔQ loops are different than energy loops, ΔQ_3 is chosen to pass through pipes 4, 3, 2, and 1. A more effi-cient route for this corrective discharge loop would traverse pipes 4, 7, and 1, coinciding with energy loop II, because one less pipe is in this loop.

To obtain the ΔQ-equations, we replace each Q_i in the energy equation portion of the Q-equations by $Q_i = Q_{oi} \pm \sum \Delta Q_k$, in which the ΔQ's must be those circulating through pipe i, as defined by the ΔQ loops, and the sign before each term in the sum is determined by whether ΔQ agrees with or opposes the direction of the assumed discharge Q_{oi}. If the directions agree, the sign is positive; otherwise the sign is negative. The resulting ΔQ-equations for this network are listed as Eqs. 4.30.

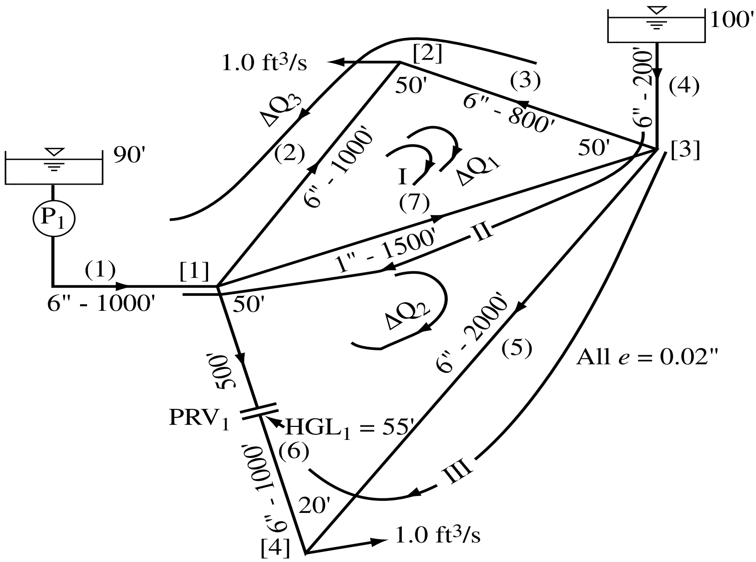


Figure 4.16 The network in Fig. 4.6, modified for solution with the ΔQ-equations.

$$
\begin{aligned}
F_1 &= K_2(Q_{o2} + \Delta Q_1 - \Delta Q_3)^{n_2} - K_3(Q_{o3} - \Delta Q_1 + \Delta Q_3)^{n_3} \\
&\qquad - K_7(Q_{o7} - \Delta Q_1 + \Delta Q_2)^{n_7} = 0 \\
F_2 &= K_4(Q_{o4} + \Delta Q_3)^{n_4} - K_7(Q_{o7} - \Delta Q_1 + \Delta Q_2)^{n_7} \\
&\qquad - K_1(Q_{o1} - \Delta Q_3)^{n_1} + h_{p1} - 10 = 0 \\
F_3 &= K_4(Q_{o4} + \Delta Q_3)^{n_4} + K_5(Q_{o5} + \Delta Q_2)^{n_5} - K_6'(Q_{o6} - \Delta Q_2)^{n_6} - 45 = 0
\end{aligned}
\tag{4.30}
$$

These equations are in a sense incomplete until each Q_{oi} is replaced by a value. When this step is completed, then these three equations contain only three unknowns, ΔQ_1, ΔQ_2 and ΔQ_3. In principle each reader could produce a different set of acceptable values for the initial discharges, so long as they do indeed satisfy each and every junction continuity equation. One valid set of Q_{oi}'s is $Q_{o1} = 1.0$, $Q_{o2} = 1.0$, $Q_{o3} = 0.0$, $Q_{o4} = 1.0$, $Q_{o5} = 1.0$, $Q_{o6} = 0.0$, and $Q_{o7} = 0.0$. Finally, the pump head now becomes $h_{p1} = A(Q_{o1} + \Delta Q_1 - \Delta Q_3)^2 + B(Q_{o1} + \Delta Q_1 - \Delta Q_3) + C$ when the pump curve is fitted with a second-order polynomial. If desired, as an alternative either a linear or a higher-order polynomial could be chosen to describe the operating characteristics of this pump.

Now let us revisit the network in Fig. 4.10 that contains a BPV as a second illustration of forming the ΔQ-equations. In this analysis we can visualize the two sets of loops as shown in Fig. 4.17. The ΔQ loops ignore the presence of the BPV in this network, but the energy loops will be written for the modified network with the BPV converted into an artificial reservoir. The resulting ΔQ-equations for this network appear as Eqs. 4.31. In

$$
\begin{aligned}
F_1 &= K_1(Q_{o1} + \Delta Q_2)^{n_1} - h_{p1} - K_2(Q_{o2} - \Delta Q_2 - \Delta Q_3)^{n_2} + 20 = 0 \\
F_2 &= K_1(Q_{o1} + \Delta Q_2)^{n_1} - h_{p1} + K_3(Q_{o3} - \Delta Q_3)^{n_3} \\
&\qquad + K_4'(Q_{o4} - \Delta Q_1 - \Delta Q_3)^{n_4} + 15 = 0 \qquad (4.31) \\
F_3 &= K_9(Q_{o9} + \Delta Q_3)^{n_9} + K_6(Q_{o6} - \Delta Q_1)^{n_6} - K_7(Q_{o7} + \Delta Q_1)^{n_7} \\
&\qquad - K_8(Q_{o8} + \Delta Q_1)^{n_8} + K_4'(Q_{o4} - \Delta Q_1 - \Delta Q_3)^{n_4} + 60 = 0
\end{aligned}
$$

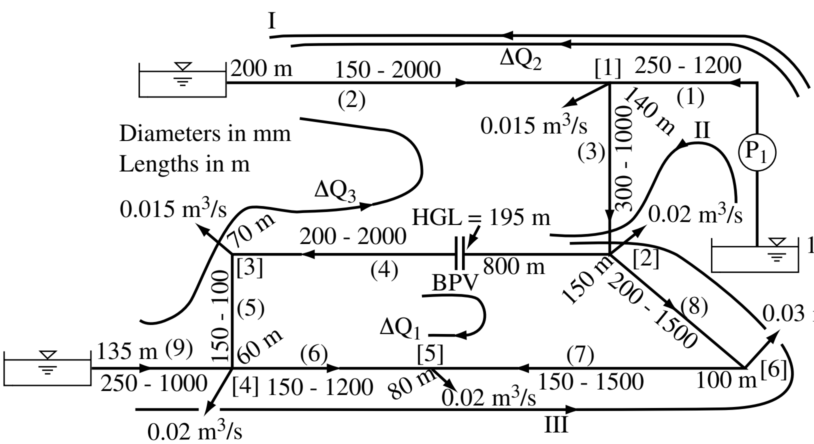
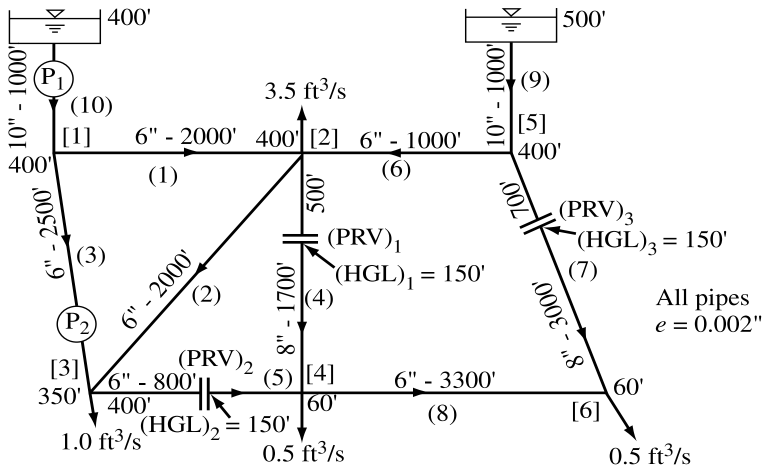


Figure 4.17 The network of Fig. 4.10, modified for the ΔQ-equation system.

these equations $h_{p1} = A(Q_{o1} + \Delta Q_2)^2 + B(Q_{o1} + \Delta Q_2) + C$. The initial flows that satisfy the junction continuity equations are chosen as $Q_{o1} = 0.1$, $Q_{o2} = 0.0$, $Q_{o3} = 0.085$, $Q_{o4} = 0.015$, $Q_{o5} = 0.0$, $Q_{o6} = 0.0$, $Q_{o7} = 0.02$, $Q_{o8} = 0.05$, and $Q_{o9} = 0.02$. The substitution of these values into Eqs. 4.31 yields the final set of ΔQ-equations.

If large differences in ground elevation occur in a network, PRV's are often installed in a sequence of pipes to prevent excessively large pressures in the lower part of the network. Such a series of PRV's may cause pressures in one subregion to be completely independent of the remainder of the network. Such isolation creates what are commonly called separate pressure zones. When separate pressure zones are created, it is normally better to form sub-networks and analyze each one separately, starting with the subnetwork at the lower elevation. The solution from the isolated lower subnetwork can then be used to determine the demands at the nodes of the next higher network, and so on.

The 10-pipe, 6-node network in Fig. 4.18 contains three PRV's in pipes 4, 5, and 7, respectively; it typifies such a situation. In this network the three PRV's cause the pressures at nodes 4 and 6 to be independent of pressures in the remainder of the network. The best analysis, therefore, would begin by studying separately the subnetwork that is composed of pipes 5, 4, 7, and 8 downstream from the PRV's. In this subnetwork the PRV's are modeled as three constant-head reservoirs. The values of Q_4, Q_5, and Q_7 from the solution of the subnetwork are next added to the other demands to determine the demands at nodes 3, 2, and 5, respectively, in organizing the remainder of the network for analysis.



Pump Characteristics

Pump 1 Q ft^3/s	Pump 1 H ft.	Pump 2 Q ft^3/s	Pump 2 H ft.
1.5	110	1.5	15.0
2.5	104	2.0	12.0
3.5	92	3.0	7.5

Figure 4.18 A network with two pressure zones.

While it is generally not difficult to determine by visual examination of a map of the piping system whether PRV's isolate a portion of a network into a separate pressure zone, in computer programs a simple test is needed to identify this situation. Such a test can be based on the fact that no series of connected pipes exists between any of the artificial reservoirs created by the PRV's and any of the other reservoirs and source pumps. That no connection exists in the network example can be seen by resketching the network, as shown in Fig. 4.19. As a consequence, if pseudo loops between artificial reservoirs or source pumps cannot be found by a computer program that uses its own internal loop-finding algorithm, then the PRV's isolate a subnetwork into a pressure zone that is separate from the remainder of the network. One difficulty with this kind of test, which relies on the inability to find paths which connect all supply sources, is that errors in the network input data or an ill-defined network itself can also cause this test to be satisfied; network computer programs are supposed to identify such input errors and terminate if any such errors are found. Thus it is desirable to have an independent verification, i.e., a separate test, that can indicate that separate pressure zones exist.

This alternative or verification test could take the form that is described next. The goal in this "test" is to determine whether the sum $NJ + NL$, determined in the usual way, is equal to, or exceeds, the number of pipes in the network.

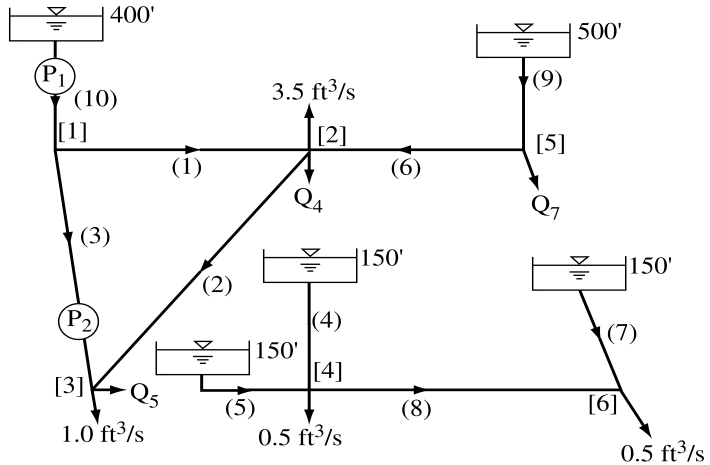


Figure 4.19 PRV's create two separate pressure zones.

To highlight the problem, let us examine closely the network in Fig. 4.19 which is known to have two separate pressure zones. Overall there are six junctions, so $NJ = 6$. There is one real loop, and the usual rule indicates there is N_{res} - 1 = 5 - 1 = 4 pseudo loops, or $NL = 5$. Using these values, we obtain $NJ + NL = 6 + 5 = 11$, which is larger than the actual number of pipes, which is $NP = 10$. In this instance the inequality occurs because only four independent loops exist, one real loop and three pseudo loops. These numbers will be found to be correct when we view the overall network as two separate networks. The higher network in Fig. 4.19 has $NP = 6$, one real loop and one pseudo loop, $N_s = N_{res} - 1$ and $NL = 2$. Since there are four junctions in the network with the higher pressure zone, $NJ = 4$, and $NP = NJ + NL = 6$. For the network with the lower pressure zone $NP = 4$, $NJ = 2$, there are no real loops, and the expected number of pseudo loops is $N_{res} - 1 = 2$, giving $NL = 2$. Again $NP = NJ + NL$.

The verification test to determine whether PRV's isolate a portion of a network into a separate pressure zone might therefore be as follows:

1. Find the real loops which exist after pipes containing PRV's have been disconnected from their upstream junctions.
2. Compute NL_s from $NL_s = N_{res} + N_{pump} - 1$.
3. Add the number of loops that were found in steps 1 and 2 to determine NL, and then determine NP from $NP = NJ + NL$.
4. If this computed NP exceeds the number of pipes in the network, then the PRV's isolate a portion of the network. The amount of the difference between the newly computed NP and the actual number of pipes in the network is the number of additional pressure zones that exist in the network; the total number of zones is one more than this number of additional zones.

4.4 SOLVING THE NETWORK EQUATIONS

4.4.1. NEWTON METHOD FOR LARGE SYSTEMS OF EQUATIONS

In Sections 4.2 and 4.3 we explored the writing of systems of algebraic equations to describe the relations between the discharges, pressures, and other variables and parameters in a pipe network. Many of the equations in these systems of equations are nonlinear. A good method for solving nonlinear equations is therefore needed. Numerous methods exist, but the Newton Method is the method of choice here. Its application to the solution of the Q-equations, the H-equations and the ΔQ-equations will be discussed in this section. To treat the unknown discharges (when using the Q-equations), the unknown heads (when using the H-equations), and the unknown corrective loop discharges (when using the ΔQ-

equations) in a uniform way, the primary unknown variable in this section will be called the vector $\{\boldsymbol{x}\}$.

The Newton iterative formula for solving a system of equations can be written as

$$\{\boldsymbol{x}\}^{(m+1)} = \{\boldsymbol{x}\}^{(m)} - [\boldsymbol{D}]^{-1}\{\boldsymbol{F}\}^{(m)} \tag{4.32a}$$

Here $\boldsymbol{x}$ is an entire column vector $\{\boldsymbol{x}\}$ of unknowns, $\{\boldsymbol{F}\}$ is an entire column vector of equations, and $[\boldsymbol{D}]^{-1}$ is the inverse of a matrix $[\boldsymbol{D}]$ which is the Jacobian. The Jacobian occurs in several applications in mathematics, and it represents the following matrix of derivatives:

$$[\boldsymbol{D}] = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & \frac{\partial F_1}{\partial x_2} & \cdot & \cdot & \frac{\partial F_1}{\partial x_n} \\ \frac{\partial F_2}{\partial x_1} & \frac{\partial F_2}{\partial x_2} & \cdot & \cdot & \frac{\partial F_2}{\partial x_n} \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot & \cdot \\ \frac{\partial F_n}{\partial x_1} & \frac{\partial F_n}{\partial x_2} & \cdot & \cdot & \frac{\partial F_n}{\partial x_n} \end{bmatrix} \tag{4.33}$$

Likewise $\{\boldsymbol{x}\}$ and $\{\boldsymbol{F}\}$ are actually

$$\{\boldsymbol{x}\} = \begin{Bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ x_n \end{Bmatrix} \qquad \{\boldsymbol{F}\} = \begin{Bmatrix} F_1 \\ F_2 \\ \cdot \\ \cdot \\ F_n \end{Bmatrix} \tag{4.34}$$

Equation 4.32a indicates that the Newton method solves a system of nonlinear equations by iteratively solving a system of linear equations because $[\boldsymbol{D}]^{-1}\{\boldsymbol{F}\}$ represents the solution of the linear system of equations

$$[\boldsymbol{D}]\{z\} = \{\boldsymbol{F}\} \tag{4.32b}$$

That is, the vector that is subtracted from the current estimate of the unknown vector $\{\boldsymbol{x}\}$ in Eq. 4.32a is the solution $\{z\}$ to the linear system of equations that is Eq. 4.32b. In practice we therefore see that the Newton method solves a system of equations by the iterative formula

$$\{\boldsymbol{x}\}^{(m+1)} = \{\boldsymbol{x}\}^{(m)} - \{z\} \tag{4.32c}$$

where $\{z\}$ is the solution vector that is obtained by solving $[\boldsymbol{D}]\{z\} = \{\boldsymbol{F}\}$. If the system should actually contain only linear equations, then the first iteration will produce the exact solution.

The development of Eq. 4.32 follows. We begin by using a multi-dimensional Taylor series expansion to evaluate the individual equations F_i in the neighborhood of an initial solution estimate that we call $\{x\}$ which is presumed to be near the actual solution:

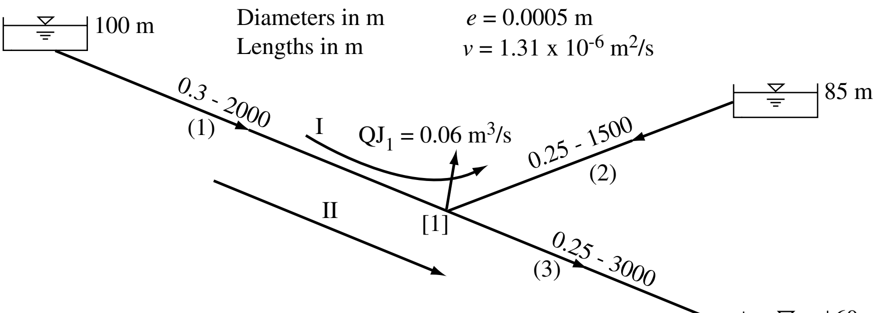
$$
\begin{aligned}
F_1^{(m+1)} &= F_1^{(m)} + \frac{\partial F_1}{\partial x_1}\Delta x_1 + \frac{\partial F_1}{\partial x_2}\Delta x_2 + \cdot\ \ \cdot + \frac{\partial F_1}{\partial x_n}\Delta x_n + \mathrm{O}\left(\Delta x^2\right) = 0 \\
F_2^{(m+1)} &= F_2^{(m)} + \frac{\partial F_2}{\partial x_1}\Delta x_1 + \frac{\partial F_2}{\partial x_2}\Delta x_2 + \cdot\ \ \cdot + \frac{\partial F_2}{\partial x_n}\Delta x_n + \mathrm{O}\left(\Delta x^2\right) = 0 \\
&\cdot \qquad \cdot \qquad \cdot \qquad \cdot \qquad \cdot \qquad \cdot \\
&\cdot \qquad \cdot \qquad \cdot \qquad \cdot \qquad \cdot \qquad \cdot \\
F_n^{(m+1)} &= F_n^{(m)} + \frac{\partial F_n}{\partial x_1}\Delta x_1 + \frac{\partial F_n}{\partial x_2}\Delta x_2 + \cdot\ \ \cdot + \frac{\partial F_n}{\partial x_n}\Delta x_n + \mathrm{O}\left(\Delta x^2\right) = 0
\end{aligned}
\tag{4.35}
$$

When we use matrix notation and make the substitution $\Delta x_i = x_i^{(m+1)} - x_i^{(m)}$, this system of equations becomes

$$
\begin{Bmatrix} F_1 \\ F_2 \\ \cdot \\ \cdot \\ F_n \end{Bmatrix}^{(m)} + \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & \frac{\partial F_1}{\partial x_2} & \cdot & \cdot & \frac{\partial F_1}{\partial x_n} \\ \frac{\partial F_2}{\partial x_1} & \frac{\partial F_2}{\partial x_2} & \cdot & \cdot & \frac{\partial F_2}{\partial x_n} \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot & \cdot \\ \frac{\partial F_n}{\partial x_1} & \frac{\partial F_n}{\partial x_2} & \cdot & \cdot & \frac{\partial F_n}{\partial x_n} \end{bmatrix} \begin{Bmatrix} x_1^{(m+1)} - x_1^{(m)} \\ x_2^{(m+1)} - x_2^{(m)} \\ \cdot \\ \cdot \\ x_n^{(m+1)} - x_n^{(m)} \end{Bmatrix} = 0 \tag{4.36}
$$

which can be written compactly as $\{\boldsymbol{F}\}^{(m)} + [\boldsymbol{D}]^{(m)}(\{\boldsymbol{x}\}^{(m+1)} - \{\boldsymbol{x}\}^{(m)}) = \{\boldsymbol{0}\}$ and solved for $\{\boldsymbol{x}\}^{(m+1)}$ to produce Eq. 4.32a.

Now let us see in some detail how the Newton method works in practice. First we shall examine the three-reservoir problem by forming and solving manually the appropriate systems of *Q*-equations, *H*-equations and *ΔQ*-equations. Then we will look at computer programs that could be used to find the solution to the *Q*-equations; the first program is simpler and more specialized, and the second program is longer but more versatile. Finally we examine a third program that will solve any equation system that is supplied to it in a subroutine.



Pipe	***K***	***n***
1	1469	1.974
2	2432	1.927
3	5646	1.971

Figure 4.20 The three-reservoir problem.

Figure 4.20 shows three reservoirs that are connected by three pipes with an external demand at the common junction of the pipes. The highest reservoir has a water surface elevation of 100 m; the middle reservoir water surface elevation is 85 m, and the lowest reservoir has a water surface elevation of 60 m. We will use the data in the figure and table to form and solve the three systems of equations.

The Q-equations are

$$
\begin{aligned}
F_1 &= Q_1 + Q_2 - Q_3 - QJ_1 = 0 & F_1 &= Q_1 + Q_2 - Q_3 - 0.06 = 0 \\
F_2 &= K_1 Q_1^{n_1} - K_2 Q_2^{n_2} - WS_1 + WS_2 = 0 & F_2 &= 1469 Q_1^{1.974} - 2432 Q_2^{1.927} - 15 = 0 \\
F_3 &= K_1 Q_1^{n_1} + K_3 Q_3^{n_3} - WS_1 + WS_3 = 0 & F_3 &= 1469 Q_1^{1.974} + 5646 Q_3^{1.971} - 40 = 0
\end{aligned}
\tag{4.37}
$$

To satisfy the junction continuity equation, equation F_1, and also determine initial values for the Newton method, we can select $Q_1^{(0)} = Q_{o1} = 0.10$ m^3/s, $Q_2^{(0)} = Q_{o2} = 0.05$ m^3/s, and $Q_3^{(0)} = Q_{o3} = 0.09$ m^3/s. The superscript (0) denotes initial values for use by the Newton method in solving the Q-equations, and the subscripts denote initial discharge values for the ΔQ-equations. The initial values for use with the Q-equations are not required to satisfy the junction continuity equations, although this set of values does.

The H-equation is

$$
\begin{aligned}
F_1 &= \left[\frac{WS_1 - H_1}{K_1}\right]^{1/n_1} + \left[\frac{WS_2 - H_1}{K_2}\right]^{1/n_2} - \left[\frac{H_1 - WS_3}{K_3}\right]^{1/n_3} - 0.06 = 0 \\
F_1 &= \left[\frac{100 - H_1}{1469}\right]^{0.507} + \left[\frac{85 - H_1}{2432}\right]^{0.519} - \left[\frac{H_1 - 60}{5646}\right]^{0.507} - 0.06 = 0
\end{aligned}
\tag{4.38}
$$

The ΔQ-equations are

$$
\begin{aligned}
F_1 &= K_1\left(Q_{o1} + \Delta Q_1 + \Delta Q_2\right)^{n_1} - K_2\left(Q_{o2} - \Delta Q_1\right)^{n_2} - WS_1 + WS_2 = 0 \\
F_2 &= K_1\left(Q_{o1} + \Delta Q_1 + \Delta Q_2\right)^{n_1} + K_3\left(Q_{o3} + \Delta Q_2\right)^{n_3} - WS_1 + WS_3 = 0
\end{aligned}
\tag{4.39a}
$$

With the initial discharges that we have chosen, these equations become

$$
\begin{aligned}
F_1 &= 1469\left(0.10 + \Delta Q_1 + \Delta Q_2\right)^{1.974} - 2432\left(0.05 - \Delta Q_1\right)^{1.927} - 15 = 0 \\
F_2 &= 1469\left(0.10 + \Delta Q_1 + \Delta Q_2\right)^{1.974} + 5646\left(0.09 + \Delta Q_2\right)^{1.971} - 40 = 0
\end{aligned}
\tag{4.39b}
$$

We now begin the solution of the Q-equations by the Newton method using the equations $[\boldsymbol{D}]\{z\} = \{\boldsymbol{F}\}$, $\{\boldsymbol{Q}\}^{(m+1)} = \{\boldsymbol{Q}\}^{(m)} - \{z\}$. According to Eq. 4.33, the Jacobian is

$$
[\boldsymbol{D}] = \begin{bmatrix} 1 & 1 & -1 \\ n_1 K_1 Q_1^{n_1 - 1} & -n_2 K_2 Q_2^{n_2 - 1} & 0 \\ n_1 K_1 Q_1^{n_1 - 1} & 0 & n_3 K_3 Q_3^{n_3 - 1} \end{bmatrix}
\tag{4.40a}
$$

or

$$[\boldsymbol{D}] = \begin{bmatrix} 1 & 1 & -1 \\ 2900\boldsymbol{Q}_1^{0.974} & -4686\boldsymbol{Q}_2^{0.927} & 0 \\ 2900\boldsymbol{Q}_1^{0.974} & 0 & 11128\boldsymbol{Q}_3^{0.971} \end{bmatrix} \tag{4.40b}$$

For the first computational cycle we therefore solve the equation set

$$\begin{bmatrix} 1.00 & 1.00 & -1.00 \\ 307.89 & -291.57 & 0.00 \\ 307.89 & 0.00 & 1073.96 \end{bmatrix} \begin{Bmatrix} z_1 \\ z_2 \\ z_3 \end{Bmatrix} = \begin{Bmatrix} 0.00 \\ -6.97 \\ 24.64 \end{Bmatrix} \tag{4.41a}$$

and obtain the results

$$\{z\} = \begin{Bmatrix} -0.0004 \\ 0.0235 \\ 0.0231 \end{Bmatrix} \qquad \{\boldsymbol{Q}\} = \begin{Bmatrix} 0.1004 \\ 0.0265 \\ 0.0669 \end{Bmatrix} \tag{4.42a}$$

We now iterate to obtain the following equation set and updated solution:

$$\begin{bmatrix} 1.00 & 1.00 & -1.00 \\ 309.09 & -161.86 & 0.00 \\ 309.09 & 0.00 & 805.20 \end{bmatrix} \begin{Bmatrix} z_1 \\ z_2 \\ z_3 \end{Bmatrix} = \begin{Bmatrix} 0.000 \\ -1.506 \\ 3.051 \end{Bmatrix} \tag{4.41b}$$

$$\{z\} = \begin{Bmatrix} -0.0017 \\ 0.0061 \\ 0.0044 \end{Bmatrix} \qquad \{\boldsymbol{Q}\} = \begin{Bmatrix} 0.1021 \\ 0.0204 \\ 0.0625 \end{Bmatrix} \tag{4.42b}$$

One more iteration leads to

$$\begin{bmatrix} 1.00 & 1.00 & -1.00 \\ 314.19 & -127.01 & 0.00 \\ 314.19 & 0.00 & 753.42 \end{bmatrix} \begin{Bmatrix} z_1 \\ z_2 \\ z_3 \end{Bmatrix} = \begin{Bmatrix} 0.000 \\ -0.095 \\ 0.151 \end{Bmatrix} \tag{4.41c}$$

$$\{z\} = \begin{Bmatrix} -0.0001 \\ 0.0004 \\ 0.0003 \end{Bmatrix} \qquad \{\boldsymbol{Q}\} = \begin{Bmatrix} 0.1022 \\ 0.0200 \\ 0.0622 \end{Bmatrix} \text{ m}^3/\text{s} \tag{4.42c}$$

This solution is now sufficiently accurate!

The solution of the H-equations by the Newton method uses basically the same equations $[\boldsymbol{D}]\{z\} = \{\boldsymbol{F}\}$ and $\{H\}^{(m+1)} = \{\boldsymbol{H}\}^{(m)} - \{z\}$. These relations lead to a single update equation

$$H_1^{(m+1)} = H_1^{(m)} - F_1 / (\frac{dF_1}{dH_1}) \tag{4.43}$$

The derivative is

$$\frac{dF_1}{dH_1} = -\frac{1}{n_1K_1}\left[\frac{100 - H_1}{K_1}\right]^{1/n_1 - 1} - \frac{1}{n_2K_2}\left[\frac{85 - H_1}{K_2}\right]^{1/n_2 - 1} - \frac{1}{n_3K_3}\left[\frac{H_1 - 60}{K_3}\right]^{1/n_3 - 1} \tag{4.44a}$$

or

$$\frac{dF_1}{dH_1} = -\frac{1}{2900}\left[\frac{100 - H_1}{1469}\right]^{-0.493} - \frac{1}{4686}\left[\frac{85 - H_1}{2432}\right]^{-0.481} - \frac{1}{11128}\left[\frac{H_1 - 60}{5646}\right]^{-0.493} \tag{4.44b}$$

If we initiate the solution procedure with the initial estimate of $H_1^{(0)} = 84$, then the first two iterative cycles produce

$$F_1 = -0.00415, \quad \frac{dF_1}{dH_1} = -0.0136, \quad H_1^{(1)} = 84 - 0.00415/0.0136 = 83.70 \tag{4.45a}$$

and

$$F_1 = -0.000247, \quad \frac{dF_1}{dH_1} = -0.01251, \quad H_1^{(2)} = 83.70 - 0.000247/0.01251 = 83.68\,\mathrm{m} \tag{4.45b}$$

which will be regarded as adequate.

Finally, we now solve the ΔQ-equations by the Newton method using again the equations $[\mathbf{D}]\{z\} = \{F\}$ and $\{\boldsymbol{\Delta Q}^{(m+1)}\} = \{\boldsymbol{\Delta Q}^{(m)}\} - \{z\}$. In this case we solve repeatedly the two-equation system for updated correction vectors $\{z\}$ until it is declared to be sufficiently small. Three cycles of computation yield these results:

$$\begin{bmatrix} 599 & 307 \\ 307 & 1382 \end{bmatrix}\begin{Bmatrix} z_1 \\ z_2 \end{Bmatrix} = \begin{Bmatrix} -6.97 \\ 24.6 \end{Bmatrix} \qquad \begin{Bmatrix} z_1 \\ z_2 \end{Bmatrix} = \begin{Bmatrix} -0.0234 \\ 0.0230 \end{Bmatrix} \qquad \begin{Bmatrix} \Delta Q_1 \\ \Delta Q_2 \end{Bmatrix} = \begin{Bmatrix} 0.0234 \\ -0.0230 \end{Bmatrix} \tag{4.46a}$$

$$\begin{bmatrix} 472 & 309 \\ 309 & 1115 \end{bmatrix}\begin{Bmatrix} z_1 \\ z_2 \end{Bmatrix} = \begin{Bmatrix} -1.523 \\ 3.13 \end{Bmatrix} \qquad \begin{Bmatrix} z_1 \\ z_2 \end{Bmatrix} = \begin{Bmatrix} -0.0062 \\ 0.0045 \end{Bmatrix} \qquad \begin{Bmatrix} \Delta Q_1 \\ \Delta Q_2 \end{Bmatrix} = \begin{Bmatrix} 0.0296 \\ -0.0275 \end{Bmatrix} \tag{4.46b}$$

$$\begin{bmatrix} 441 & 314 \\ 314 & 1068 \end{bmatrix}\begin{Bmatrix} z_1 \\ z_2 \end{Bmatrix} = \begin{Bmatrix} -0.0952 \\ 0.1507 \end{Bmatrix} \qquad \begin{Bmatrix} z_1 \\ z_2 \end{Bmatrix} = \begin{Bmatrix} -0.0004 \\ 0.0003 \end{Bmatrix} \qquad \begin{Bmatrix} \Delta Q_1 \\ \Delta Q_2 \end{Bmatrix} = \begin{Bmatrix} 0.0300 \\ -0.0278 \end{Bmatrix} \tag{4.46c}$$

Now the discharges can be computed as $Q_1 = 0.1 + \Delta Q_1 + \Delta Q_2 = 0.1022$ m^3/s, $Q_2 = 0.05 - \Delta Q_1 = 0.0200$ m^3/s, and $Q_3 = 0.09 + \Delta Q_2 = 0.0622$ m^3/s.

Computer programs of differing complexity and generality can also be developed for the solution of these equation systems by application of the Newton method. We will now look at two programs. The first program is relatively simple but must be recoded in part for each application; it will be applied to the solution of the Q-equations for the three-reservoir problem. The second program is more versatile.

Program 4.2, the FORTRAN program listed in Fig. 4.21, is designed to solve three simultaneous equations with the Newton method. It calls a matrix solver that has the coefficient matrix expanded by one column to contain the known vector, and it places the inverse in additional columns beyond the location of the known vector. The first part of the main program is currently written specifically to solve the Q-equations for the three-reservoir problem. However, the portion that numerically evaluates the derivatives in the Jacobian matrix is written more generally, with N giving the size of the matrix problem to be solved. Careful study of this listing will clarify considerably how the various tasks are performed. The subroutine INVERM employs a common method in linear algebra problems by using an expanded matrix. The coefficient matrix is square, here 3 rows by 3 columns. The known vector is placed in the next column, in this case column 4. The subroutine solves the system of equations and provides the inverse matrix. The solution is returned in the same column that initially contained the known vector, here column 4.

```
************************************************************************
*    PROGRAM NO. 4.2, NEWTON, FORTRAN
*    THIS PROGRAM HAS BEEN INCLUDED FOR THE CONVENIENCE OF THE READER.
*    THE AUTHOR ACCEPTS NO RESPONSIBILITY FOR ITS CORRECTNESS.
*    USERS OF THIS PROGRAM DO SO AT THEIR OWN RISK.
************************************************************************
*    IMPLEMENTS THE NEWTON METHOD IN SOLVING THREE EQUATIONS
      REAL X(3),F(3),D(3,7),RK(3),RN(3),K1,K2,K3,N1,N2,N3
      DATA N,N1/3,4/,DX/.001/,MAX/15/,ERR/.0001/
      WRITE(5,*)' GIVE: K1,K2,K3,N1,N2,N3,Q1,Q2,Q3'
      READ(5,*) RK,RN,X
      M=0
    1 NT=0
    5 F(1)=X(1)+X(2)-X(3)-0.06
      F(2)=RK(1)*X(1)**RN(1)-RK(2)*X(2)**RN(2)-15.0
      F(3)=RK(1)*X(1)**RN(1)+RK(3)*X(3)**RN(3)-40.0
      IF(NT.NE.0) GO TO 15
      DO 10 I=1,N
   10 D(I,N1)=F(I)
      X(1)=X(1)-DX
      NT=1
      GO TO 5
   15 X(NT)=X(NT)+DX
      DO 20 I=1,N
   20 D(I,NT)=(D(I,N1)-F(I))/DX
      NT=NT+1
      IF(NT.GT.N) GO TO 30
      X(NT)=X(NT)-DX
      GO TO 5
   30 CALL INVERM(D,N)
      DIF=0.
      DO 40 I=1,N
      DIF=DIF+ABS(D(I,N1))
   40 X(I)=X(I)-D(I,N1)
      M=M+1
      IF(DIF.GT. ERR .AND. M.LT.MAX) GO TO 1
      WRITE(5,*)' THE SOLUTION IS ',X
      END
```

Figure 4.21 Program 4.2 to use the Newton method to solve three equations.

The listing of the inverse starts in the next column. In solving three equations the array D(3,7) therefore has 7 as its second subscript, and the last three columns contain the inverse.

Currently the input data to this program includes the coefficients K and n for each pipe and an initial estimate of the discharge in each pipe: 1469 2432 5646 1.974 1.927 1.971 0.10 0.05 0.09. The program produces the same solution as we obtained manually.

The next program, listed in Fig. 4.22, is essentially the same as the previous program, except that it calls the linear equation solver GAUSEL, described in Appendix A, rather than INVERM. GAUSEL is a more versatile subroutine that interchanges rows to minimize truncation error, applies one iterative correction to the solution vector, and returns an estimate of the relative error for each unknown in the array ERRNOR, so the user has parameters to determine the accuracy of the solution. However, the relative error is not printed in this program. This subroutine also illustrates the Microsoft FORTRAN ability to allocate the array sizes that are needed.

```
*************************************************************************
*   PROGRAM NO. 4.3, NEWTON, FORTRAN
*   THIS PROGRAM HAS BEEN INCLUDED FOR THE CONVENIENCE OF THE READER.
*   THE AUTHOR ACCEPTS NO RESPONSIBILITY FOR ITS CORRECTNESS.
*   USERS OF THIS PROGRAM DO SO AT THEIR OWN RISK.
*************************************************************************
*   IMPLEMENTS THE NEWTON METHOD IN SOLVING THREE EQUATIONS
      REAL X(3),F(3),D(3,3),F1(3),ERRNOR(3),RK(3),RN(3),K1,K2,K3,N1,N2,N3
      DATA N/3/,DX/.001/,MAX/15/,ERR/.0001/
      WRITE(5,*)' GIVE: K1,K2,K3,N1,N2,N3,Q1,Q2,Q3'
      READ(5,*) RK,RN,X
      M=0
    1 NT=0
    5 F(1)=X(1)+X(2)-X(3)-0.06
      F(2)=RK(1)*X(1)**RN(1)-RK(2)*X(2)**RN(2)-15.0
      F(3)=RK(1)*X(1)**RN(1)+RK(3)*X(3)**RN(3)-40.0
      IF(NT.NE.0) GO TO 15
      DO 10 I=1,N
   10 F1(I)=F(I)
      X(1)=X(1)-DX
      NT=1
      GO TO 5
   15 X(NT)=X(NT)+DX
      DO 20 I=1,N
   20 D(I,NT)=(F1(I)-F(I))/DX
      NT=NT+1
      IF(NT.GT.N) GO TO 30
      X(NT)=X(NT)-DX
      GO TO 5
   30 CALL GAUSEL(3,3,D,F1,DET,ERRNOR)
      DIF=0.
      DO 40 I=1,N
      DIF=DIF+ABS(F1(I))
   40 X(I)=X(I)-F1(I)
      M=M+1
      IF(DIF.GT. ERR .AND. M.LT.MAX) GO TO 1
      WRITE(5,*)' THE SOLUTION IS ',X
      END
```

Figure 4.22 Program 4.3, a more versatile implementation of the Newton method.

With this introduction to the Newton method, let us look further at the structure of a computer program that would solve any system of equations. This program should consist of two primary elements: First, a main or driver program that accomplishes the following tasks: (a) it allows the user to assign values to known variables and initial estimates for unknown variables; (b) it creates the Jacobian matrix and the known vector and supplies values to these arrays; (c) it calls a linear algebra solver; (d) it implements the Newton iteration; and (e) it prints the solution. Second, it must contain a subroutine (or function subprogram) that defines the system of equations to be solved. The equations in this subroutine will change, depending upon the nature of the problem that is being solved, and therefore the statements in this subroutine would be changed as different types of problems are to be solved. A listing of such a general purpose program EQUSOL1.FOR will be found in Fig. 4.23. The subroutine FUNCT provides the equations for this program. The main program calls on the linear algebra solver SOLVEQ that is described in Appendix B.

```
*********************************************************************
*   PROGRAM NO. 4.4, EQUSOL1, FORTRAN
*   THIS PROGRAM HAS BEEN INCLUDED FOR THE CONVENIENCE OF THE READER.
*   THE AUTHOR ACCEPTS NO RESPONSIBILITY FOR ITS CORRECTNESS.
*   USERS OF THIS PROGRAM DO SO AT THEIR OWN RISK.
*********************************************************************
*   THIS EQUATION SOLVER IMPLEMENTS THE NEWTON METHOD
      LOGICAL IV[ALLOCATABLE](:)
      INTEGER*2 INDX[ALLOCATABLE](:)
      CHARACTER*3 SYMB[ALLOCATABLE](:),CH
      REAL X[ALLOCATABLE](:),F[ALLOCATABLE](:),F1[ALLOCATABLE](:),
     &D[ALLOCATABLE](:,:)
*
      WRITE(*,*)' GIVE (1) NO. OF EQS., (2) NO. OF VARIABLES,',
     &' (3) INPUT UNIT, (4) OUTPUT UNIT'
      READ(*,*) N,NV,IN,IOUT
      ALLOCATE(X(NV),F(N),F1(N),D(N,N),SYMB(NV),IV(NV),INDX(N))
      IF(IN.EQ.0 .OR. IN.EQ.5) WRITE(IN,100) NV
  100 FORMAT(' GIVE',I3,' LINES WITH:',/,3X,'(1) SYMBOL FOR VAR.(3 CH)',
     &/,3X,'(2) K OR U FOR KNOWN OR UNKNOWN AND',/,3X,'(3) VALUE')
      J=0
      DO 10 I=1,NV
      READ(IN,110) SYMB(I),CH,X(I)
  110 FORMAT(A3,1X,A1,1X,F10.0)
      IF(CH.EQ.'U' .OR.  CH.EQ.'u') THEN
      IV(I)=.TRUE.
      J=J+1
      ELSE IF(CH.EQ.'K' .OR. CH.EQ.'k') THEN
      IV(I)=.FALSE.
      ELSE
      WRITE(*,*)' ERROR IN INPUT FOR VARIABLE', I
      STOP
      ENDIF
   10 CONTINUE
      IF(J.EQ.N) GO TO 12
      WRITE(*,*)' YOU GAVE',N,' EQS. BUT',J,' UNKNOWNS'
      STOP
   12 NCT=0
   15 CALL FUNCT(X,F)
      J=0
      DO 30 JJ=1,NV
      IF(IV(JJ)) THEN
      XX=X(JJ)
      X(JJ)=1.005*X(JJ)
      J=J+1
      CALL FUNCT(X,F1)
      DO 20 I=1,N
   20 D(I,J)=(F1(I)-F(I))/(X(JJ)-XX)
      X(JJ)=XX
      ENDIF
   30 CONTINUE
      CALL SOLVEQ(N,1,N,D,F,1,DD,INDX)
      NCT=NCT+1
      SUM=0.
```

Figure 4.23 A listing of the program EQUSOL1.FOR.

```
      J=0
      DO 40 I=1,NV
       IF(IV(I)) THEN
      J=J+1
       X(I)=X(I)-F(J)
       SUM=SUM+ABS(F(J))
      ENDIF
   40 CONTINUE
       WRITE(*,*)' NCT =',NCT,' SUM =',SUM
       IF(NCT.LT.20 .AND. SUM.GT. 0.0001) GO TO 15
      WRITE(IOUT,120)(I,SYMB(I),X(I),I=1,NV)
  120 FORMAT(I5,1X,A3,' =',F10.3)
      END
*
      SUBROUTINE FUNCT(X,F)
      REAL F(22),X(41)
      DATA E,G2,P,AP/0.005,64.4,8.6714174E-6,5.4541539E-3/
      F(1)=X(41)-X(1)-X(3)-X(35) !Unknowns
      F(2)=X(1)-X(2)-X(36)       !  1=Q2   15=H4   29=L1
      F(3)=X(3)-X(4)-X(37)       !  2=Q3   16=H5   30=L2
      F(4)=X(2)+X(4)-X(5)-X(38)  !  3=Q4   17=f1   31=L3
      F(5)=X(12)-X(40)+X(6)      !  4=Q5   18=f2   32=L4
      F(6)=X(13)-X(12)+X(7)      !  5=Q6   19=f3   33=L5
      F(7)=X(14)-X(12)+X(9)      !  6=hf1  20=f4   34=L6
      F(8)=X(15)-X(13)+X(8)      !  7=hf2  21=f5   35=QJ1
      F(9)=X(16)-X(15)+X(11)     !  8=hf3  22=f6   36=QJ2
      F(10)=X(7)+X(8)-X(10)-X(9) !  9=hf4 Knowns   37=QJ3
      RF1=1./SQRT(X(17))         ! 10=hf5  23=D1   38=QJ4
      RF2=1./SQRT(X(18))         ! 11=hf6  24=D2   39=QJ5
      RF3=1./SQRT(X(19))         ! 12=H1   25=D3   40=WS1
      RF4=1./SQRT(X(20))         ! 13=H2   26=D4   41=Q1
      RF5=1./SQRT(X(21))         ! 14=H3   27=D5
      RF6=1./SQRT(X(22))         ! 15=H4   28=D6
      F(11)=X(6)-X(17)*X(29)*12./X(23)*(X(41)/(AP*X(23)**2))**2/G2
      F(12)=X(7)-X(18)*X(30)*12./X(24)*(X(1)/(AP*X(24)**2))**2/G2
      F(13)=X(8)-X(19)*X(31)*12./X(25)*(X(2)/(AP*X(25)**2))**2/G2
      F(14)=X(9)-X(20)*X(32)*12./X(26)*(X(3)/(AP*X(26)**2))**2/G2
      F(15)=X(10)-X(21)*X(33)*12./X(27)*(X(4)/(AP*X(27)**2))**2/G2
      F(16)=X(11)-X(22)*X(34)*12./X(28)*(X(5)/(AP*X(28)**2))**2/G2
      F(17)=RF1-1.14+2.*ALOG10(E/X(23)+P*X(23)*RF1/X(41))
      F(18)=RF2-1.14+2.*ALOG10(E/X(24)+P*X(24)*RF2/X(1))
      F(19)=RF3-1.14+2.*ALOG10(E/X(25)+P*X(25)*RF3/X(2))
      F(20)=RF4-1.14+2.*ALOG10(E/X(26)+P*X(26)*RF4/X(3))
      F(21)=RF5-1.14+2.*ALOG10(E/X(27)+P*X(27)*RF5/X(4))
      F(22)=RF6-1.14+2.*ALOG10(E/X(28)+P*X(28)*RF6/X(5))
      RETURN
      END
```

Figure 4.23 (Concluded) A listing of the program EQUSOL1.FOR.

Let's examine how the main program does its tasks of providing values to the Jacobian matrix $[\boldsymbol{D}]$ and equation vector $\{\boldsymbol{F}\}$ and then carrying out a Newton solution. The key portion of the main program that implements the Newton method appears in **bold** characters in Fig. 4.23. Three tasks are accomplished by these statements: (1) defining the equation vector; (2) numerically evaluating the elements of the Jacobian matrix; and (3) solving the resulting linear system of equations and subtracting this solution from the current vector of unknowns, as described by Eq. 4.32a.

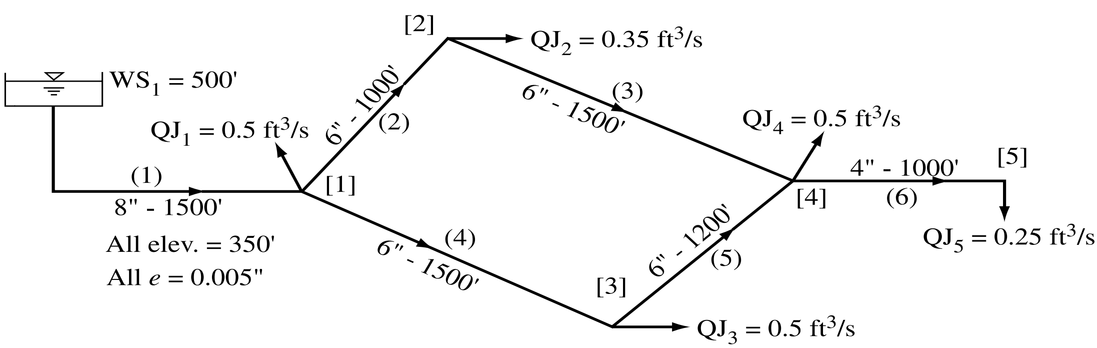
The FORTRAN integer NCT is the iteration counter; it is set to 0 before beginning the Newton iteration. Statement `15 CALL FUNCT(X,F)` has two arguments, an array `X` for the variables and an array `F` for the equations. The array `X` includes both the known and unknown variables of the problem. Upon returning from `CALL FUNCT`, the array `F` contains a set of equations that have been evaluated by using the initial estimates of the unknowns. Since the initial estimates are incorrect, the individual elements of $\{\boldsymbol{F}\}$ will not be zero, but subsequent Newton iterations will drive these elements progressively closer to zero. Statement `DO 30 JJ=1,NV`, in which `NV` is the total of all variables, evaluates individual columns of the Jacobian matrix `D(I,J)` by using a first-order numerical evaluation of the derivatives. Since `IV(JJ)` is `.FALSE.` for known variables and `.TRUE.` for unknown variables, we note that nothing happens in loop `30` if `IV` is `.FALSE.` Hence `J`, which identifies the column in which the Jacobian derivatives are entered, is incremented only for unknown variables. When an unknown is encountered, x_j, which is `X(JJ)`, is incremented by multiplying its current value by 1.005 before the equation is evaluated again by calling `FUNCT`. Upon returning from `FUNCT`, the array `F1` now contains equation values based on $1.005x_j$, and then the statement `D(I,J) = (F1(I) - F(I))/(X(JJ) - XX)` numerically evaluates the derivatives of the equations by using a first-order approximation. The statement `DO 20 I=1,N` fills all row entries for column `J` of the Jacobian matrix $[\boldsymbol{D}]$.

Upon completing the `DO 30` loop, the equation vector $\{\boldsymbol{F}\}$ and the Jacobian matrix $[\boldsymbol{D}]$ have been fully evaluated. The next statement `CALL SOLVEQ` calls a linear equation solver, which upon return has replaced the elements of the array `F` with the solution vector $\{z\}$ found in Eq. 4.32b. The statement `DO 40 I=1,NV` implements Eq. 4.32c with `SUM` accumulating the absolute sum of the corrections applied to the unknown vector $\{\boldsymbol{x}\}$. If this `SUM` is larger than the allowable error and fewer than 20 iterations have been completed, then `GO TO 15` at the end of this code segment will begin another Newton iteration.

In our example it would be relatively easy to derive the actual partial derivatives of each equation with respect to each unknown, and the elements of the Jacobian could be evaluated by using these derivatives. The length of the program would be longer if these derivative expressions were used. The numerical approximation of the derivatives requires extra arithmetic, particularly since many derivatives are zero, but the advantage of a shorter code makes the numerical approximation of the derivatives attractive.

Example Problem 4.5

Use program EQUSOL1 to solve the 6-pipe, 5-node network shown below. Obtain this solution in four ways: (1) use the program as it now exists with subroutine FUNCT; (2) use the Q-equations; (3) use the H-equations; (4) use the ΔQ-equations.



1. The existing subroutine FUNCT explicitly defines the equations that we want to solve; there are 22 equations, `F(1)` through `F(22)`. There are 41 variables associated with the solution; therefore 41 - 22 = 19 of these variables are known. These equations are as follows:

Junction continuity equations:

$$Q_1 - Q_2 - Q_4 - QJ_1 = 0 \tag{1}$$
$$Q_2 - Q_3 - QJ_2 = 0 \tag{2}$$
$$Q_4 - Q_5 - QJ_3 = 0 \tag{3}$$
$$Q_3 + Q_5 - Q_6 - QJ_4 = 0 \tag{4}$$

(The junction continuity equation at node 5 is not included here, but this simple equation Q_6 - $QJ_5 = 0$ establishes the discharge in pipe 6 as 0.25 ft^3/s.)

Head loss equations giving the HGL at a downstream node relative to the upstream node:

$$H_1 = WS_1 - h_{f1} \tag{5}$$
$$H_2 = H_1 - h_{f2} \tag{6}$$
$$H_3 = H_1 - h_{f4} \tag{7}$$
$$H_4 = H_2 - h_{f3} \qquad (\text{or} \quad H_4 = H_3 - h_{f5}) \tag{8}$$
$$H_5 = H_4 - h_{f6} \tag{9}$$

Energy equation around a loop:

$$h_{f2} + h_{f3} - h_{f5} - h_{f4} = 0 \tag{10}$$

Darcy-Weisbach equations to define the frictional head losses (pipe numbers i = 1, 6):

$$h_{fi} = f_i \frac{L_i}{D_i} \frac{Q_i^2}{2gA_i^2} \tag{11-16}$$

Colebrook-White equations (pipe numbers i = 1, 6):

$$\frac{1}{\sqrt{f_i}} = 1.14 - 2\,log_{10}\left\{\frac{e_i}{D_i} + \frac{9.35\nu D_i}{(4/\pi)Q_i\sqrt{f_i}}\right\} \tag{17-22}$$

In the program listing the integer within `X( )` identifies the variable in the array, as is seen by the comments following the exclamation points there. We note there are 22 equations: a continuity equation for NJ - 1 = 4 junctions, a separate head difference equation for each pipe, a Darcy-Weisbach equation for each pipe, a companion Colebrook-White equation for each pipe, and finally an energy loop equation, for a total of $3NP + NJ$ = 18 + 5 = 22 equations. Since the entire system demand must come from pipe 1, its discharge must be $Q_1 = 2.1$ ft^3/s, and the unknowns are 5 unknown discharges Q_2 .. Q_6, 6 unknown head losses h_{f1} .. h_{f6}, 5 unknown heads H_1 .. H_5, and 6 unknown friction factors f_1 .. f_6, for a total of 22. The input and solution files for the program now follow:

Input File			Output File	
Q2 U 0.8	f6 U 0.020		1 Q2 = 0.820	22 f6 = 0.024
Q3 U 0.5	D1 K 8.0		2 Q3 = 0.470	23 D1 = 8.000
Q4 U 0.8	D2 K 6.0		3 Q4 = 0.780	24 D2 = 6.000
Q5 U 0.3	D3 K 6.0		4 Q5 = 0.280	25 D3 = 6.000
Q6 U 0.3	D4 K 6.0		5 Q6 = 0.250	26 D4 = 6.000
hf1 U 24.0	D5 K 6.0		6 hf1 = 23.952	27 D5 = 6.000
hf2 U 11.0	D6 K 4.0		7 hf2 = 11.279	28 D6 = 4.000
hf3 U 0.2	L1 K 1500.0		8 hf3 = 5.872	29 L1 = 1500.0
hf4 U 0.15	L2 K 1000.0		9 hf4 = 15.368	30 L2 = 1000.0
hf5 U 2.0	L3 K 1500.0		10 hf5 = 1.783	31 L3 = 1500.0
hf6 U 6.0	L4 K 1500.0		11 hf6 = 9.117	32 L4 = 1500.0
H1 U 476.0	L5 K 1200.0		12 H1 = 476.048	33 L5 = 1200.0
H2 U 465.0	L6 K 1000.0		13 H2 = 464.769	34 L6 = 1000.0
H3 U 460.0	QJ1 K 0.50		14 H3 = 460.680	35 QJ1 = 0.50
H4 U 458.0	QJ2 K 0.35		15 H4 = 458.897	36 QJ2 = 0.35
H5 U 450.0	QJ3 K 0.50		16 H5 = 449.780	37 QJ3 = 0.50
f1 U 0.020	QJ4 K 0.50		17 f1 = 0.019	38 QJ4 = 0.50
f2 U 0.020	QJ5 K 0.25		18 f2 = 0.021	39 QJ5 = 0.25
f3 U 0.020	WS1 K 500.0		19 f3 = 0.022	40 WS1 = 500.0
f4 U 0.020	Q1 K 2.1		20 f4 = 0.021	41 Q1 = 2.100
f5 U 0.020			21 f5 = 0.024	

2. The Q-equations are

$$F_1 = Q_1 - Q_2 - Q_4 - QJ_1 = 0$$
$$F_2 = Q_2 - Q_3 - QJ_2 = 0$$
$$F_3 = Q_4 - Q_5 - QJ_3 = 0$$
$$F_4 = Q_3 + Q_5 - Q_6 - QJ_4 = 0$$
$$F_5 = K_2 Q_2^{n_2} + K_3 Q_3^{n_3} - K_5 Q_5^{n_5} - K_4 Q_4^{n_4} = 0$$

The K and n for each pipe must now be determined. Program 2.1, PIPK_N, or some other means will provide these values:

Pipe	K	n
1	5.6845	1.9381
2	16.4967	1.9185
3	24.3685	1.8858
4	24.7450	1.9185
5	19.0411	1.8611
6	126.3843	1.8970

To compute the five unknown discharges Q_i ($i = 2, 6$) (with $Q_1 = 2.1$ ft^3/s known), the subroutine FUNCT must be modified as follows:

```
SUBROUTINE FUNCT(X,F)
REAL F(5),X(11)
REAL K2/16.4967/,K3/24.3685/,K4/24.745/,K5/19.0411/
REAL N2/1.9185/,N3/1.8858/,N4/1.9185/,N5/1.8611/
F(1)=X(6)-X(1)-X(3)-X(7)     ! Unknowns           Knowns
F(2)=X(1)-X(2)-X(8)          ! 1 = Q2, 4 = Q5, 6 =  Q1,  9 = QJ3
F(3)=X(3)-X(4)-X(9)          ! 2 = Q3, 5 = Q6, 7 = QJ1, 10 = QJ4
F(4)=X(2)+X(4)-X(5)-X(10)    ! 3 = Q4,         8 = QJ2, 11 = QJ5
F(5)=K2*X(1)**N2+K3*X(2)**N3-K5*X(4)**N5-K4*X(3)**N4
RETURN
END
```

The input data (all in ft^3/s) that were used to solve this problem (with 5 and 11 plus 2 and 3 for I/O units from the keyboard) and the solution are listed now:

Input Data				**Solution**	
Variable	Type	Initial value		Index	Value
Q2	U	0.80		1 Q2	0.82
Q3	U	0.50		2 Q3	0.47
Q4	U	0.80		3 Q4	0.78
Q5	U	0.30		4 Q5	0.28
Q6	U	0.25		5 Q6	0.25
Q1	K	2.10		6 Q1	2.10
QJ1	K	0.50		7 QJ1	0.50
QJ2	K	0.35		8 QJ2	0.35
QJ3	K	0.50		9 QJ3	0.50
QJ4	K	0.50		10 QJ4	0.50
QJ5	K	0.25		11 QJ5	0.25

Since X(11) = QJ5 is not used in the equations, the keyboard input could have been changed to 5 10 2 3, and the last line of input could then be deleted.

3. The number of H-equations could be reduced below five, but we will use five head equations to determine the head at the five nodes. These equations are

$$F_1 = \left[\frac{500 - H_1}{K_1}\right]^{1/n_1} - \left[\frac{H_1 - H_2}{K_2}\right]^{1/n_2} - \left[\frac{H_1 - H_3}{K_4}\right]^{1/n_4} - QJ_1 = 0$$

$$F_2 = \left[\frac{H_1 - H_2}{K_2}\right]^{1/n_2} - \left[\frac{H_2 - H_4}{K_3}\right]^{1/n_3} - QJ_2 = 0$$

$$F_3 = \left[\frac{H_1 - H_3}{K_4}\right]^{1/n_4} - \left[\frac{H_3 - H_4}{K_5}\right]^{1/n_5} - QJ_3 = 0$$

$$F_4 = \left[\frac{H_2 - H_4}{K_3}\right]^{1/n_3} + \left[\frac{H_3 - H_4}{K_5}\right]^{1/n_5} - \left[\frac{H_4 - H_5}{K_6}\right]^{1/n_6} - QJ_4 = 0$$

$$F_5 = \left[\frac{H_4 - H_5}{K_6}\right]^{1/n_6} - QJ_5 = 0$$

These equations are arranged to allow us to find H_i $(i = 1, 5)$ with a demand at each of the five nodes as additional variables, so there are 5 unknowns and 10 variables. The appropriate modifications of subroutine FUNCT are as follows:

```
      SUBROUTINE FUNCT(X,F)
      REAL F(5),X(10)
      REAL K1/5.6845/,K2/16.4967/,K3/24.3685/,K4/24.745/,K5/19.0411/
     &,K6/126.3843/,R1/.515969/,R2/.52124/,R3/.53028/,R4/.52124/
     &,R5/.53732/,R6/.52715/
C UNKNOWNS: 1 = H1, 2 = H2, 3 = H3, 4 = H4, 5 = H5;
```

```
C KNOWNS: 6 = QJ1, 7 = QJ2, 8 = QJ3, 9 = QJ4, 10 = QJ5
      F(1)=(ABS(500-X(1))/K1)**R1-(ABS(X(1)-X(2))/K2)**R2
     &-(ABS(X(1)-X(3))/K4)**R4-X(6)
      F(2)=(ABS(X(1)-X(2))/K2)**R2-(ABS(X(2)-X(4))/K3)**R3-X(7)
      F(3)=(ABS(X(1)-X(3))/K4)**R4-(ABS(X(3)-X(4))/K5)**R5-X(8)
      F(4)=(ABS(X(2)-X(4))/K3)**R3+(ABS(X(3)-X(4))/K5)**R5
     &-(ABS(X(4)-X(5))/K6)**R6-X(9)
      F(5)=(ABS(X(4)-X(5))/K6)**R6-X(10)
      RETURN
      END
```

The input data (all in ft or ft^3/s) for this problem (with an additional 5 and 10 plus 2 and 3 for I/O units from the keyboard) and the solution follow:

Input Data				Solution	
Variable	**Type**	**Initial value**		**Index**	**Value**
H1	U	476.0		1 H1	476.1
H2	U	465.0		2 H2	464.8
H3	U	460.0		3 H3	460.7
H4	U	458.0		4 H4	458.9
H5	U	450.0		5 H5	449.8
QJ1	K	0.50		6 QJ1	0.50
QJ2	K	0.35		7 QJ2	0.35
QJ3	K	0.50		8 QJ3	0.50
QJ4	K	0.50		9 QJ4	0.50
QJ5	K	0.25		10 QJ5	0.25

4. For this problem there is only one ΔQ-equation, which is

$$F_1 = K_2\left(Q_{o2} + \Delta Q_1\right)^{n_1} + K_3\left(Q_{o3} + \Delta Q_1\right)^{n_3} - K_5\left(Q_{o5} - \Delta Q_1\right)^{n_5} - K_4\left(Q_{o4} - \Delta Q_1\right)^{n_4} = 0$$

The input, since it is only two lines, can be given directly from the keyboard as

```
1 1 5 6
DQ1 U 0.1
```

The estimate of 0.1 ft^3/s is used because the main program uses 1.005 times the current value. This might be changed with an IF statement that adds 0.001 to the variable if its value is zero. The solution is DQ1 = 0.020 ft^3/s. The subroutine FUNCT can be modified, with the initial discharge in each pipe chosen to be $Q_{o2} = 0.8$ ft^3/s, $Q_{o3} = 0.45$ ft^3/s, $Q_{o4} = 0.80$ ft^3/s, and $Q_{o5} = 0.30$ ft^3/s to satisfy continuity, as shown:

```
      SUBROUTINE FUNCT(X,F)
      REAL F(1),X(1)
      REAL K2/16.4967/,K3/24.3685/,K4/24.745/,K5/19.0411/,N2/1.9185/
     &,N3/1.8858/,N4/1.9185/,N5/1.8611/
      REAL QO2/0.80/,QO3/0.45/,QO4/0.80/,QO5/0.30/
C UNKNOWN: DQ1
      F(1)=K2*(QO2+X(1))**N2+K3*(QO3+X(1))**N3-
     &K5*(QO5-X(1))**N5-K4*(QO4-X(1))**N4
      RETURN
      END
```

Writing every equation, as was done in the listing of EQUSOL1, makes it easy to follow the computational sequence in subroutine FUNCT. However, since there are as many Darcy-Weisbach equations as there are pipes and the Gauss-Seidel iteration could be used to solve the Colebrook-White equations internally within the system of equations, FUNCT can be simplified. A separate function can be written to evaluate the Colebrook-White equation, and the Darcy-Weisbach equations are in a DO loop. Now the equations to be solved are the four junction continuity equations and the six head loss equations that indicate the difference in head along a pipe is equal to the frictional head loss between the pipe ends. For variety, Q_1 is now unknown, and in its place Q_6 is assumed to be known.

The listing of the modified subroutine FUNCT follows:

```
      SUBROUTINE FUNCT(X,F)
      INTEGER*2 ID(6)/7,8,10,9,10,11/      ! 1 = Q1, 10 = H4,  19 = D3
     &,IU(6)/16,7,8,7,9,10/                ! 2 = Q2, 11 = H5,  20 = D4
      REAL F(10),X(28)                     ! 3 = Q3, 12 = QJ1, 21 = D5
      DATA G2/64.4/,P4/0.7853982/          ! 4 = Q4, 13 = QJ2, 22 = D6
      F(1)=X(1)-X(2)-X(4)-X(12)            ! 5 = Q5, 14 = QJ3, 23 = L1
      F(2)=X(2)-X(3)-X(13)                 ! 6 = Q6, 15 = QJ4, 24 = L2
      F(3)=X(4)-X(5)-X(14)                 ! 7 = H1, 16 = WS1, 25 = L3
      F(4)=X(3)+X(5)-X(6)-X(15)            ! 8 = H2, 17 = D1,  26 = L4
      DO 10 I=1,6                          ! 9 = H3, 18 = D2,  27 = L5
      J=I+16                               !                   28 = L6
   10 F(I+4)=X(ID(I))-X(IU(I))+FR(I,J,X)*X(I+22)
     & /X(J)*(X(I)/(P4*X(J)**2))**2/G2
      RETURN
      END
*
      FUNCTION FR(I,J,X)
      REAL X(28)
      REAL FI(6)/6*.02/
      DATA E/0.0004166667/,CCVISC/1.03543E-4/
      F1=1./SQRT(FI(I))
   10 F2=F1
      F1=1.14-2.*ALOG10(E/X(J)+CCVISC*X(J)*F2/X(I))
      IF(ABS(F1-F2).GT. 1.E-6) GO TO 10
      FR=1./F1/F1
      FI(I)=FR
      RETURN
      END
```

The input data file (all in ft or ft^3/s) and the solution are given next:

Input Data | **Solution**

Variable	Type	Initial value	Index	Value
Q1	U	2.10	1 Q1	2.000
Q2	U	0.80	2 Q2	0.770
Q3	U	0.50	3 Q3	0.420
Q4	U	0.80	4 Q4	0.730
Q5	U	0.30	5 Q5	0.230
Q6	K	0.25	6 Q6	0.250
H1	U	476.0	7 H1	478.2
H2	U	465.0	8 H2	468.2
H3	U	460.0	9 H3	464.7

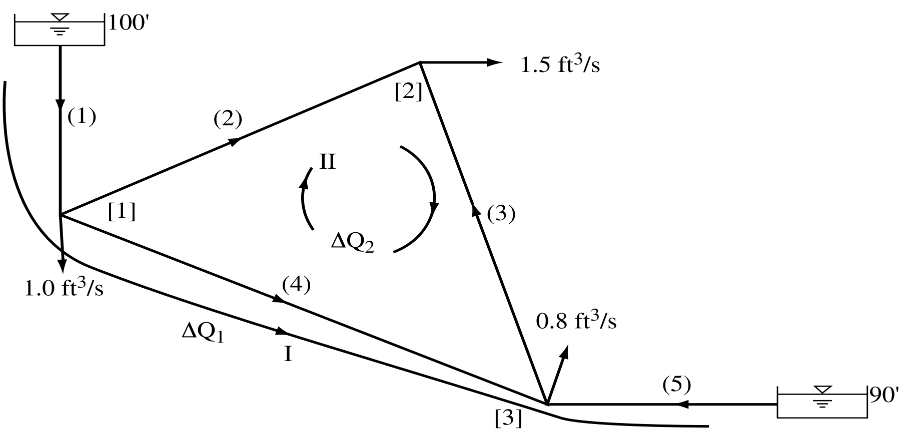
Continued:

Input Data			Solution	
Variable	**Type**	**Initial value**	**Index**	**Value**
H4	U	458.0	10 H4	463.5
H5	U	450.0	11 H5	454.3
QJ1	K	0.50	12 QJ1	0.50
QJ2	K	0.35	13 QJ2	0.35
QJ3	K	0.50	14 QJ3	0.50
QJ4	K	0.50	15 QJ4	0.50
WS1	K	500.0	16 WS1	500.0
D1	K	0.667	17 D1	0.667
D2	K	0.50	18 D2	0.50
D3	K	0.50	19 D3	0.50
D4	K	0.50	20 D4	0.50
D5	K	0.50	21 D5	0.50
D6	K	0.333	22 D6	0.333
L1	K	1500	23 L1	1500
L2	K	1000	24 L2	1000
L3	K	1500	25 L3	1500
L4	K	1500	26 L4	1500
L5	K	1200	27 L5	1200
L6	K	1000	28 L6	1000

* * *

4.4.2. SOLVING THE THREE EQUATION SYSTEMS VIA NEWTON

The Newton method will now be applied in turn to the solution of the Q-equations, the H-equations and the ΔQ-equations for network shown in Fig. 4.24. Considerable detail will be presented in these solutions so the details of applying the Newton method can be



Pipe	K	n
1	7.59	1.936
2	9.63	1.901
3	48.6	1.882
4	39.7	1.768
5	16.5	1.935

Figure 4.24 A 5-pipe, 3-node network.

examined. The reader is encouraged to check numerically some of these steps. In the Q-equations the elements of the Jacobian will either be $\partial F_i / \partial Q_j = \pm 1$ or zero in row i for a junction continuity equation row. The Jacobian terms for the energy loop equation rows will either be $\partial F_i / \partial Q_j = \pm n_j K_j Q_j^{n_j - 1}$ or zero. The non-zero elements of the Jacobian for the H-equations are $\partial F_i / \partial H_j = \pm \{1/(n_m K_m)\}\{(H_j - H_k)/K_m\}^{1/n_m - 1}$ in which the sign is determined by the sign in front of this term in the equation and the sign before H_j within the parentheses. Non-zero terms in the Jacobian for the ΔQ-equations will be of the form $\partial F_i / \partial \Delta Q_j = \pm n_k K_k (Q_{ok} \pm \sum \Delta Q_m)^{n_k - 1}$.

The **Q-equations** are

$$\begin{aligned}
F_1 &= Q_1 - Q_2 - Q_4 - 1.0 = 0 \\
F_2 &= Q_2 + Q_3 - 1.5 = 0 \\
F_3 &= Q_4 - Q_3 + Q_5 - 0.8 = 0 \\
F_4 &= K_1 Q_1^{n_1} + K_4 Q_4^{n_4} - K_5 Q_5^{n_5} - 10 = 0 \\
F_5 &= K_2 Q_2^{n_2} - K_3 Q_3^{n_3} - K_4 Q_4^{n_4} = 0
\end{aligned} \tag{4.47}$$

The Newton method is described by $[\boldsymbol{D}]\{z\}=\{\boldsymbol{F}\}$ and $\{\boldsymbol{Q}\}^{(m+1)} = \{\boldsymbol{Q}\}^{(m)} - \{z\}$ with

$$[D] = \begin{bmatrix} 1 & -1 & 0 & -1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 1 \\ n_1 K_1 Q_1^{n_1 - 1} & 0 & 0 & n_4 K_4 Q_4^{n_4 - 1} & -n_5 K_5 Q_5^{n_5 - 1} \\ 0 & n_2 K_2 Q_2^{n_2 - 1} & -n_3 K_3 Q_3^{n_3 - 1} & -n_4 K_4 Q_4^{n_4 - 1} & 0 \end{bmatrix} \tag{4.48}$$

If we choose the initial estimate of the solution vector to be

$$\{\boldsymbol{Q}\}^{(0)} = \begin{Bmatrix} 2.0 \\ 0.9 \\ 0.6 \\ 0.1 \\ 1.3 \end{Bmatrix} \tag{4.49}$$

with which we have been careful to satisfy the junction continuity equations, so these discharges can be used in the ΔQ-equations, the first evaluation of the Jacobian matrix and right-hand side leads to

$$\begin{bmatrix} 1.0000 & -1.0000 & 0.0000 & -1.0000 & 0.0000 \\ 0.0000 & 1.0000 & 1.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & -1.0000 & 1.0000 & 1.0000 \\ 28.1133 & 0.0000 & 0.0000 & 11.9749 & -40.8039 \\ 0.0000 & 16.6487 & -58.2888 & -11.9749 & 0.0000 \end{bmatrix} \begin{Bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \end{Bmatrix} = \begin{Bmatrix} 0.0000 \\ 0.0000 \\ 0.0000 \\ -7.6935 \\ -11.3783 \end{Bmatrix} \tag{4.50a}$$

with the solution

$$\{z\} = \begin{Bmatrix} -0.1169 \\ -0.1470 \\ 0.1470 \\ 0.0301 \\ 0.1169 \end{Bmatrix} \qquad \{Q\}^{(1)} = \begin{Bmatrix} 2.1169 \\ 1.0470 \\ 0.4530 \\ 0.0699 \\ 1.1831 \end{Bmatrix} \tag{4.51a}$$

The Newton equations for the next cycle are

$$\begin{bmatrix} 1.0000 & -1.0000 & 0.0000 & -1.0000 & 0.0000 \\ 0.0000 & 1.0000 & 1.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & -1.0000 & 1.0000 & 1.0000 \\ 29.6481 & 0.0000 & 0.0000 & 9.0910 & -37.3636 \\ 0.0000 & 19.0804 & -45.4902 & -9.0910 & 0.0000 \end{bmatrix} \begin{Bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \end{Bmatrix} = \begin{Bmatrix} 0.0000 \\ 0.0000 \\ 0.0000 \\ -0.0682 \\ -0.7993 \end{Bmatrix} \tag{4.50b}$$

with the solution

$$\{z\} = \begin{Bmatrix} -0.0022 \\ -0.0111 \\ 0.0111 \\ 0.0089 \\ 0.0022 \end{Bmatrix} \qquad \{Q\}^{(2)} = \begin{Bmatrix} 2.1191 \\ 1.0581 \\ 0.4419 \\ 0.0610 \\ 1.1809 \end{Bmatrix} \tag{4.51b}$$

One more cycle would yield the final solution

$$\{Q\}^{(3)} = \begin{Bmatrix} 2.1191 \\ 1.0583 \\ 0.4417 \\ 0.0608 \\ 1.1809 \end{Bmatrix} \tag{4.51c}$$

Referring again to Fig. 4.24, since we have only three nodes, we must construct three ***H*-equations**. They are

$$\begin{aligned} F_1 &= \left[\frac{100 - H_1}{K_1}\right]^{1/n_1} - \left[\frac{H_1 - H_2}{K_2}\right]^{1/n_2} - \left[\frac{H_1 - H_3}{K_4}\right]^{1/n_4} - 1.0 = 0 \\ F_2 &= \left[\frac{H_1 - H_2}{K_2}\right]^{1/n_2} + \left[\frac{H_3 - H_2}{K_3}\right]^{1/n_3} - 1.5 = 0 \\ F_3 &= \left[\frac{H_1 - H_3}{K_4}\right]^{1/n_4} - \left[\frac{H_3 - H_2}{K_3}\right]^{1/n_3} + \left[\frac{90 - H_3}{K_5}\right]^{1/n_5} - 0.8 = 0 \end{aligned} \tag{4.52}$$

Using $[\mathbf{D}]\{z\} = \{\boldsymbol{F}\}$ and $\{\boldsymbol{H}\}^{(m+1)} = \{\boldsymbol{H}\}^{(m)} - \{z\}$ to implement the Newton method with an initial estimate of the nodal heads as

$$\{\boldsymbol{H}\}^{(0)} = \begin{Bmatrix} 93 \\ 85 \\ 88 \end{Bmatrix} \tag{4.53}$$

successive computational cycles produce

$$\begin{bmatrix} -0.166 & 0.060 & 0.035 \\ 0.060 & -0.100 & 0.040 \\ 0.035 & 0.040 & -0.162 \end{bmatrix} \begin{Bmatrix} z_1 \\ z_2 \\ z_3 \end{Bmatrix} = \begin{Bmatrix} -1.258 \\ -0.365 \\ -0.382 \end{Bmatrix} \quad \{z\} = \begin{Bmatrix} 15.953 \\ 17.241 \\ 10.088 \end{Bmatrix} \quad \{\boldsymbol{H}\}^{(1)} = \begin{Bmatrix} 77.047 \\ 67.759 \\ 77.912 \end{Bmatrix} \tag{4.54a}$$

$$\begin{bmatrix} -0.171 & 0.056 & 0.075 \\ 0.056 & -0.078 & 0.023 \\ 0.075 & 0.023 & -0.134 \end{bmatrix} \begin{Bmatrix} z_1 \\ z_2 \\ z_3 \end{Bmatrix} = \begin{Bmatrix} -0.095 \\ -0.084 \\ -0.499 \end{Bmatrix} \quad \{z\} = \begin{Bmatrix} 8.019 \\ 9.614 \\ 9.830 \end{Bmatrix} \quad \{\boldsymbol{H}\}^{(2)} = \begin{Bmatrix} 69.028 \\ 58.145 \\ 68.083 \end{Bmatrix} \tag{4.54b}$$

$$\begin{bmatrix} -0.158 & 0.052 & 0.072 \\ 0.052 & -0.075 & 0.023 \\ 0.072 & 0.023 & -0.123 \end{bmatrix} \begin{Bmatrix} z_1 \\ z_2 \\ z_3 \end{Bmatrix} = \begin{Bmatrix} -0.120 \\ -0.003 \\ -0.049 \end{Bmatrix} \quad \{z\} = \begin{Bmatrix} 1.547 \\ 1.353 \\ 0.770 \end{Bmatrix} \quad \{\boldsymbol{H}\}^{(3)} = \begin{Bmatrix} 67.480 \\ 56.793 \\ 67.312 \end{Bmatrix} \tag{4.54c}$$

$$\begin{bmatrix} -0.239 & 0.052 & 0.153 \\ 0.052 & -0.075 & 0.022 \\ 0.153 & 0.022 & -0.202 \end{bmatrix} \begin{Bmatrix} z_1 \\ z_2 \\ z_3 \end{Bmatrix} = \begin{Bmatrix} 0.019 \\ 0.000 \\ -0.019 \end{Bmatrix} \quad \{z\} = \begin{Bmatrix} -0.032 \\ 0.002 \\ 0.070 \end{Bmatrix} \quad \{\boldsymbol{H}\}^{(4)} = \begin{Bmatrix} 67.513 \\ 56.791 \\ 67.242 \end{Bmatrix} \tag{4.54d}$$

$$\begin{bmatrix} -0.210 & 0.052 & 0.124 \\ 0.052 & -0.075 & 0.023 \\ 0.124 & 0.023 & -0.174 \end{bmatrix} \begin{Bmatrix} z_1 \\ z_2 \\ z_3 \end{Bmatrix} = \begin{Bmatrix} 0.002 \\ -0.001 \\ 0.002 \end{Bmatrix} \quad \{z\} = \begin{Bmatrix} -0.005 \\ -0.001 \\ 0.006 \end{Bmatrix} \quad \{\boldsymbol{H}\}^{(5)} = \begin{Bmatrix} 67.517 \\ 56.793 \\ 67.236 \end{Bmatrix} \tag{4.54e}$$

Depending on the desired accuracy of this solution, the process might have been terminated one or two cycles sooner. One of the best features of the Newton method is a quadratic convergence rate as the solution is approached; here we can easily see the rapid reduction in the size of the corrections $\{z\}$ in successive cycles.

When this problem is solved by using the ***ΔQ*-equations**, we need only one energy loop and one pseudo loop, as is indicated on Fig. 4.24. The ΔQ-equations are

$$\begin{aligned} F_1 &= K_1(Q_{o1} + \Delta Q_1)^{n_1} + K_4(Q_{o4} - \Delta Q_2 + \Delta Q_1)^{n_4} - K_5(Q_{o5} - \Delta Q_1)^{n_5} - 10 = 0 \\ F_2 &= K_2(Q_{o2} + \Delta Q_2)^{n_2} - K_3(Q_{o3} - \Delta Q_2)^{n_3} - K_4(Q_{o4} - \Delta Q_2 + \Delta Q_1)^{n_4} = 0 \end{aligned} \tag{4.55}$$

The equations for the Newton method are $[\mathbf{D}]\{z\}=\{\boldsymbol{F}\}$ and $\{\boldsymbol{\Delta Q}\}^{(m+1)} = \{\boldsymbol{\Delta Q}\}^{(m)} - \{z\}$ in which the Jacobian and vector of initial discharges are

$$[\boldsymbol{D}] = \begin{bmatrix} \dfrac{\partial \boldsymbol{F}_1}{\partial \Delta \boldsymbol{Q}_1} & \dfrac{\partial \boldsymbol{F}_1}{\partial \Delta \boldsymbol{Q}_2} \\ \dfrac{\partial \boldsymbol{F}_2}{\partial \Delta \boldsymbol{Q}_1} & \dfrac{\partial \boldsymbol{F}_2}{\partial \Delta \boldsymbol{Q}_2} \end{bmatrix} \qquad \{\boldsymbol{Q_o}\}^{(0)} = \begin{Bmatrix} 2.0 \\ 0.9 \\ 0.6 \\ 0.1 \\ 1.3 \end{Bmatrix} \tag{4.56a,b}$$

Three successive solution cycles produce

$$\begin{bmatrix} 80.892 & -11.975 \\ -11.975 & 86.913 \end{bmatrix} \begin{Bmatrix} z_1 \\ z_2 \end{Bmatrix} = \begin{Bmatrix} -7.694 \\ -11.378 \end{Bmatrix} \quad \{z\} = \begin{Bmatrix} -0.117 \\ -0.147 \end{Bmatrix} \quad \{\Delta \boldsymbol{Q}\}^{(1)} = \begin{Bmatrix} 0.117 \\ 0.147 \end{Bmatrix} \tag{4.57a}$$

$$\begin{bmatrix} 76.103 & -9.091 \\ -9.091 & 73.662 \end{bmatrix} \begin{Bmatrix} z_1 \\ z_2 \end{Bmatrix} = \begin{Bmatrix} -0.068 \\ -0.799 \end{Bmatrix} \quad \{z\} = \begin{Bmatrix} -0.002 \\ -0.011 \end{Bmatrix} \quad \{\Delta \boldsymbol{Q}\}^{(2)} = \begin{Bmatrix} 0.119 \\ 0.158 \end{Bmatrix} \tag{4.57b}$$

$$\begin{bmatrix} 75.163 & -8.188 \\ -8.188 & 71.954 \end{bmatrix} \begin{Bmatrix} z_1 \\ z_2 \end{Bmatrix} = \begin{Bmatrix} 0.004 \\ -0.009 \end{Bmatrix} \quad \{z\} = \begin{Bmatrix} 0.000 \\ 0.000 \end{Bmatrix} \quad \{\Delta \boldsymbol{Q}\}^{(3)} = \begin{Bmatrix} 0.119 \\ 0.158 \end{Bmatrix} \tag{4.57c}$$

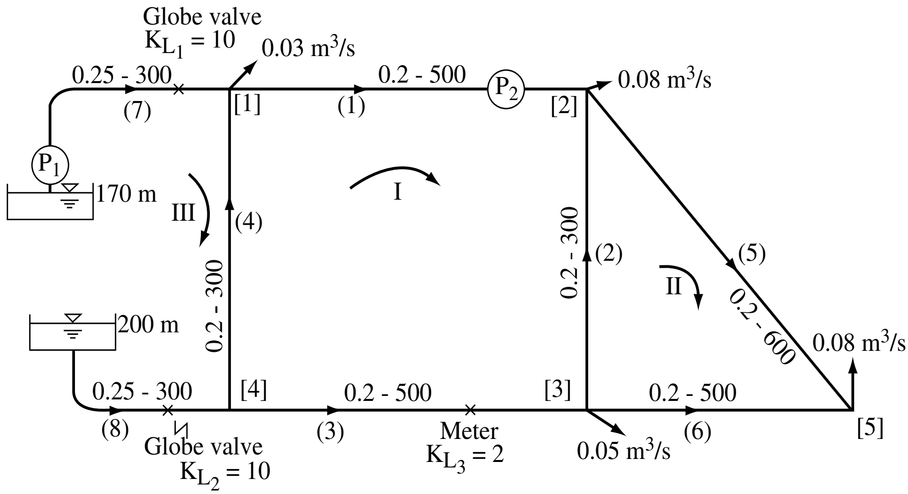
From these results we can compute the discharges themselves in ft^3/s as

$$\begin{aligned} \boldsymbol{Q}_1 &= \boldsymbol{Q_o}_1 + \Delta \boldsymbol{Q}_1 &&= 2.119 \\ \boldsymbol{Q}_2 &= \boldsymbol{Q_o}_2 + \Delta \boldsymbol{Q}_2 &&= 1.058 \\ \boldsymbol{Q}_3 &= \boldsymbol{Q_o}_3 - \Delta \boldsymbol{Q}_2 &&= 0.442 \\ \boldsymbol{Q}_4 &= \boldsymbol{Q_o}_4 + \Delta \boldsymbol{Q}_1 - \Delta \boldsymbol{Q}_2 &&= 0.061 \\ \boldsymbol{Q}_5 &= \boldsymbol{Q_o}_5 - \Delta \boldsymbol{Q}_1 &&= 1.181 \end{aligned} \tag{4.58}$$

Readers will find it instructive to solve this same problem by modifying subroutine FUNCT in program EQUSOL1. For still more experience the reader should consider the use of an equation-solving software program such as MathCAD, TK-Solver, or MathLab.

4.4.3. COMPUTER SOLUTIONS TO NETWORKS

In this section we concentrate on the implementation of solutions to networks using computers, and how pumps, local losses and/or PRV's are readily included. To begin this process consider first the eight-pipe network in Fig. 4.25 that includes a source pump that supplies some of the system demand and a booster pump in pipe 1. There are also local loss devices in pipes 7, 8, and 3, the first two of which have a loss coefficient of 10, and the third has a loss coefficient of 2. (The Roman numeral loop notation will be used in Example Problem 4.6.) After developing and solving the equations for this network, we will place a PRV in pipe 5.



Pump Characteristics
(Q in m³/s and h_p in meters)

Pump No.	Point 1		Point 2		Point 3	
	Q	h_p	Q	h_p	Q	h_p
1	0.025	12.0	0.040	10.5	0.055	8.0
2	0.060	4.0	0.090	3.8	0.120	3.5

Figure 4.25 An eight-pipe network with pumps and local losses.

For this network there are five junction continuity equations and three energy loop equations. The Q-equations are

$$
\begin{aligned}
F_1 &= -Q_1 + Q_4 + Q_7 - 0.03 = 0 \\
F_2 &= Q_1 + Q_2 - Q_5 - 0.08 = 0 \\
F_3 &= -Q_2 + Q_3 - Q_6 - 0.05 = 0 \\
F_4 &= -Q_3 - Q_4 + Q_8 - 0.00 = 0 \\
F_5 &= Q_5 + Q_6 - 0.08 = 0 \\
F_6 &= K_1 Q_1^{n_1} - h_{p2} - K_2 Q_2^{n_2} - K_3 Q_3^{n_3} - 2Q_3^2/(2gA_3^2) + K_4 Q_4^{n_4} = 0 \\
F_7 &= K_5 Q_5^{n_5} - K_6 Q_6^{n_6} + K_2 Q_2^{n_2} = 0 \\
F_8 &= K_7 Q_7^{n_7} - h_{p1} + 10Q_7^2/(2gA_7^2) - K_4 Q_4^{n_4} - K_8 Q_8^{n_8} - 10Q_8^2/(2gA_8^2) + 30 = 0
\end{aligned}
\tag{4.59}
$$

In these equations the local loss coefficients 10, 10, and 2 have been inserted in the equations, but the pump heads are written as h_{p1} and h_{p2}. These pump heads can be expressed as a function of discharge by fitting a second-order polynomial through three points on the pump characteristic curve over the range of expected operation. Alternatively, if the power supplied by the pump to the fluid is assumed to be constant, then these pump heads can be defined by $h_p = \text{Power}/(\gamma Q)$.

Example Problem 4.6
Solve the 8-equation system, Eqs. 4.59, for the network shown in Fig. 4.25 by using hand methods. Then verify this solution by using program EQUSOL1 and/or an equation-solving software package such as MathCAD or TK-Solver.

This and the next few Example Problems will be solved by rewriting the subroutine FUNCT for use with EQUSOL1 in each case. MathCAD and TK-Solver models of these problems will be found on the CD that accompanies this book.

The first step is to estimate the pipe discharges; based on these values, we then compute K and n for the 8 pipes. The listed discharges produce the K and n values in the table:

Pipe No.	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Q (m^3/s)	0.100	0.015	0.100	0.080	0.030	0.050	0.070	0.170
K	1160	613	1160	690	1292	1115	322	239
n	1.827	1.788	1.827	1.824	1.801	1.812	1.772	1.832
f	0.0134	0.0314	0.0134	0.0212	0.0168	0.0152	0.0223	0.0127

The next step is to fit the three pairs of points for the two pumps to obtain the coefficients for the polynomials: $A_1 = -2220$, $B_1 = 44.4$, $C_1 = 12.28$ and $A_2 = -55.6$, $B_2 = 1.667$, $C_2 = 4.10$. These values can now be substituted into the equations, the equations can be differentiated to produce the Jacobian matrix and equation vector, and all of the terms can be evaluated by using the data in the table and figure. (The reader should evaluate some terms to verify that the process is fully understood.) The results are

$$[D] = \begin{bmatrix} -1.0 & 0.0 & 0.0 & 1.0 & 0.0 & 0.0 & 1.0 & 0.0 \\ 1.0 & 1.0 & 0.0 & 0.0 & -1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & -1.0 & 1.0 & 0.0 & 0.0 & -1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & -1.0 & -1.0 & 0.0 & 0.0 & 0.0 & 1.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & 1.0 & 0.0 & 0.0 \\ 325 & -40.2 & -337 & 158 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 40.2 & 0.0 & 0.0 & 140 & -178 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & -158 & 0.0 & 0.0 & 370 & -173 \end{bmatrix}$$

and

$$\{F\}^T = \{0.020 \;\; 0.005 \;\; -0.015 \;\; -0.010 \;\; 0.000 \;\; 1.560 \;\; -2.226 \;\; 4.903\}$$

with the superscript T indicating the transpose of the right-side equation vector. The solution to this linear system of equations produces

$$\{z\}^T = \{-0.0029 \;\; 0.0008 \;\; -0.0071 \;\; 0.0021 \;\; -0.0071 \;\; 0.0071 \;\; 0.0150 \;\; -0.0150\}$$

so that the discharges after the first iteration are

$$\{Q\}^T = \{0.1029 \;\; 0.0142 \;\; 0.1071 \;\; 0.0779 \;\; 0.0371 \;\; 0.0429 \;\; 0.0550 \;\; 0.1850\}$$

After two additional iterations, the following solution, in m^3/s, is obtained:

$$\{Q\}^T = \{0.1028 \;\; 0.0142 \;\; 0.1072 \;\; 0.0780 \;\; 0.0370 \;\; 0.0430 \;\; 0.0548 \;\; 0.1852\}$$

The subroutine FUNCT is on the CD under EPRB4_6Q.FOR for use with EQUSOL1, and the TK-Solver model is listed on the CD under EPRB4_6Q.TK. Upon supplying the input file in column one below to EQUSOL1, the solution in the second column is obtained:

From the keyboard: 8 8 2 3	Solution (m^3/s)
Q1 U 0.100	1 Q1 = 0.103
Q2 U 0.015	2 Q2 = 0.014
Q3 U 0.100	3 Q3 = 0.107
Q4 U 0.080	4 Q4 = 0.078
Q5 U 0.030	5 Q5 = 0.037
Q6 U 0.050	6 Q6 = 0.043
Q7 U 0.070	7 Q7 = 0.055
Q8 U 0.170	8 Q8 = 0.185

* * *

Next let us examine the formulation and the solution of the H-equations for the network in Example Problem 4.6. The H-equations are presented as Eqs. 4.60. The pump heads have been added to the upstream HGL-elevations by indicating these heads as h_p. In a similar way the local losses, denoted simply as h_L, have been subtracted from the HGL-elevations in pipes 3, 7, and 8. By using second-order polynomials for the pump characteristics and noting that each Q in these equations can again be replaced by similar

$$
\begin{aligned}
F_1 &= \left[\frac{H_1 + h_{p2} - H_2}{K_1}\right]^{1/n_1} - \left[\frac{H_4 - H_1}{K_4}\right]^{1/n_4} - \left[\frac{170 + h_{p1} - h_{L1} - H_1}{K_7}\right]^{1/n_7} + 0.03 = 0 \\
F_2 &= \left[\frac{H_2 - H_5}{K_5}\right]^{1/n_5} - \left[\frac{H_3 - H_2}{K_2}\right]^{1/n_2} - \left[\frac{H_1 + h_{p2} - H_2}{K_1}\right]^{1/n_1} + 0.08 = 0 \\
F_3 &= \left[\frac{H_3 - H_5}{K_6}\right]^{1/n_6} + \left[\frac{H_3 - H_2}{K_2}\right]^{1/n_2} - \left[\frac{H_4 - h_{L3} - H_3}{K_3}\right]^{1/n_3} + 0.05 = 0 \\
F_4 &= \left[\frac{H_4 - h_{L3} - H_3}{K_3}\right]^{1/n_3} + \left[\frac{H_4 - H_1}{K_4}\right]^{1/n_4} - \left[\frac{200 - h_{L2} - H_4}{K_8}\right]^{1/n_8} + 0.00 = 0 \\
F_5 &= -\left[\frac{H_2 - H_5}{K_5}\right]^{1/n_5} - \left[\frac{H_3 - H_5}{K_6}\right]^{1/n_6} + 0.08 = 0
\end{aligned}
\tag{4.60}
$$

head-difference terms, we find that we cannot create an equation that does not contain the pump discharge in the pipes that contain the pumps. The same is true for pipes that contain local losses because again the magnitude of the loss is a function of the discharge through the pipe. If an iterative approach were used to approximate the discharge in terms of the upstream and downstream nodal heads, the quadratic convergence rate of the Newton method would be sacrificed. This dilemma highlights a deficiency in using the H-equations when pumps, whose heads are strongly dependent upon the discharges passing through them, are present. If many pumps exist in a network and the H-equations are to be used, especially if equation-solving software such as MathCAD or TK-Solver is used, it might be advantageous to write the continuity equations in terms of the discharges, then add additional equations to describe these discharges in terms of nodal heads, and finally solve simultaneously for the heads and discharges. This approach will be taken in

Example Problem 4.7. A successful technique that solves the H-equations in a computer program will be described in a subsequent section. This technique obtains the current value of the discharge in every pipe from the heads that exist during an iteration by calculating $Q = [(H_i - H_j)/K]^{1/n}$, and when a pump exists in a pipe, the Newton method can find the discharge Q from the single equation for that pipe $Q = [(H_i + h_p - H_j)/K]^{1/n}$ in which the pump head is $h_p = AQ^2 + BQ + C$.

Example Problem 4.7

Solve the H-equations for the 8-pipe network shown in Fig. 4.25 that was the subject of study in Example Problem 4.6.

The form of the subroutine FUNCT that is needed in EQUSOL1 to solve the H-equations is on the CD in EPRB4_7H.FOR with input data in EPRB4_7H.DAT; the corresponding TK-Solver model will be found there as EPRB4_7H.TK.

* * *

Finally, we now turn our attention to the ΔQ-equations, which are given in Eq. 4.61 for this same pipe network. In addition, we must select an appropriate set of initial discharges.

$$
\begin{aligned}
F_1 &= K_1\left(Q_{o1} + \Delta Q_1\right)^{n_1} - h_{p2} - K_2\left(Q_{o2} - \Delta Q_1 + \Delta Q_2\right)^{n_2} - K_3\left(Q_{o3} - \Delta Q_1\right)^{n_3} \\
&\quad - 2\left(Q_{o3} - \Delta Q_1\right)^2 / \left(2gA_3^2\right) + K_4\left(Q_{o4} + \Delta Q_1 - \Delta Q_3\right)^{n_4} = 0 \\
F_2 &= K_5\left(Q_{o5} + \Delta Q_2\right)^{n_5} - K_6\left(Q_{o6} - \Delta Q_2\right)^{n_6} + K_2\left(Q_{o2} - \Delta Q_1 + \Delta Q_2\right)^{n_2} = 0 \qquad (4.61) \\
F_3 &= K_7\left(Q_{o7} + \Delta Q_3\right)^{n_7} + 10\left(Q_{o7} + \Delta Q_3\right)^2 / \left(2gA_7^2\right) - K_4\left(Q_{o4} + \Delta Q_1 - \Delta Q_3\right)^{n_4} \\
&\quad - h_{p1} - K_8\left(Q_{o8} - \Delta Q_3\right)^{n_8} - 10\left(Q_{o8} - \Delta Q_3\right)^2 / \left(2gA_8^2\right) + 30 = 0
\end{aligned}
$$

In these ΔQ-equations the pump heads will be replaced by second-order polynomial equations in the forms

$$
\begin{aligned}
h_{p1} &= A_1(Q_{o7} + \Delta Q_3)^2 + B_1(Q_{o7} + \Delta Q_3) + C_1 \\
h_{p2} &= A_2(Q_{o1} + \Delta Q_1)^2 + B_2(Q_{o1} + \Delta Q_1) + C_2
\end{aligned}
\qquad (4.62)
$$

The local losses are replaced by $h_L = K_L Q^2/(2gA^2)$, in which each discharge is written as the algebraic sum of Q_{oi} and the corrective discharges in that pipe. The derivatives of these terms that contribute to elements of the Jacobian are then easily evaluated.

Example Problem 4.8

Solve the ΔQ-equations for the 8-pipe network depicted in Fig. 4.25.

To solve the ΔQ-equations using EQUSOL1, download EPRB4_8D.FOR from the CD; the appropriate input is found in EPRB4_8D.DAT. A TK-Solver model will be found as EPRB4_8D.TK. A set of initial discharges might be selected as follows: Q_{o1} = 0.100 m^3/s, Q_{o2} = 0.015 m^3/s, Q_{o3} = 0.110 m^3/s, Q_{o4} = 0.060 m^3/s, Q_{o5} = 0.035 m^3/s, Q_{o6} = 0.045 m^3/s, Q_{o7} = 0.070 m^3/s, and Q_{o8} = 0.170 m^3/s.

* * *

4.4.4. INCLUDING PRESSURE REDUCING VALVES

To acquire experience in analyzing networks containing pressure reducing valves (and similar appurtenances such as back pressure valves), let us assume that pipe 5 in our 8-pipe network contains a PRV that is 200 m downstream from junction 2, and the pressure setting of this valve is equivalent to a reservoir water surface elevation of 149 m. The five junction continuity equations are unchanged for this network. The energy equations now consist of one real loop and two pseudo loops, as the revised diagram of this network shows in Fig. 4.26. The pump data in Fig. 4.25 are unchanged. In the

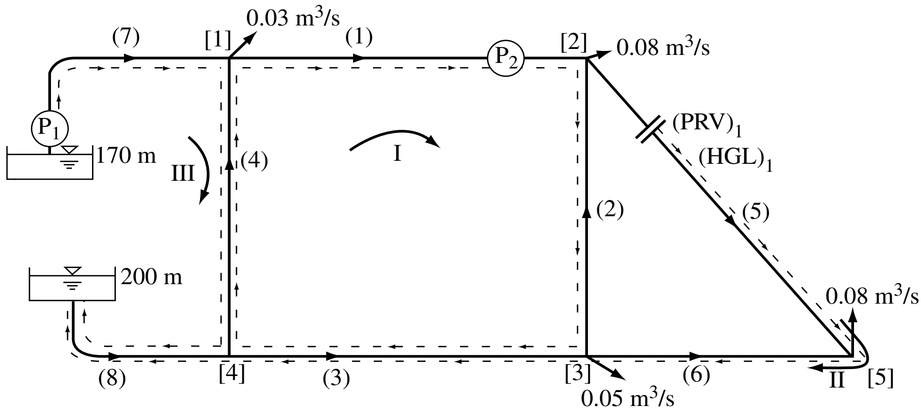


Figure 4.26 An eight-pipe network with pumps and local losses, now including a PRV.

Q-equations, Eqs. 4.72, equations F_6 and F_8 are unchanged, but equation F_7 must now be written around the new pseudo loop; it becomes

$$F_7 = K_5' Q_5^{n_5} - K_6 Q_6^{n_6} - K_3 Q_3^{n_3} - 2Q_3^2/(2gA_3^2) \\ - K_8 Q_8^{n_8} - 10Q_8^2/(2gA_8^2) + 200 - HGL_1 = 0 \tag{4.63}$$

in which K_5' represents the portion of pipe 5 downstream from the PRV. Eight independent equations therefore exist, which may be solved for the discharges Q_i, $i = 1{,}8$. If the PRV does not close completely, the solution is obtained with HGL_1 in F_7 equal to the head that is set at the valve, i.e., 149 m in this example. If the Newton solution process that is based on this assumption should produce a negative value for Q_5, the PRV will close completely. Then the discharge in the pipe containing the PRV is no longer an unknown but is zero, i.e., $Q_5 = 0$, and the value of the HGL immediately downstream from the PRV becomes the unknown. In other words, when this PRV closes, the same system of equations still describes the network operation, but the set of unknowns changes to $\{Q_1, Q_2, Q_3, Q_4, HGL_1, Q_6, Q_7, Q_8\}$. In a computer program this change can be accommodated by dropping the pipe number containing the PRV from the integer arrays that define the junction continuity equations and the energy loop equations. Also, a flag is set in the program to indicate in the solution array that the HGL of the PRV is stored in place of the discharge in that pipe.

Subroutine FUNCT for use with EQUSOL1 for this problem is on the CD under the name EPRB4_VQ.FOR. The reader should study a listing of this file to understand the logic that will model the PRV when it closes. The input from the keyboard for this problem is 8 8 2 3 (Thus there are 8 unknowns and 8 variables, and the 2 and 3 are the input and output units.). The input file that was used in Example Problem 4.6 still applies. There are only two basic changes in the program that was used to solve this network problem in the absence of the PRV: (1) element X(5) is now either Q_5 or HGL, depending upon whether it is negative, and (2) now F(7) has become a pseudo loop from the artificial reservoir at the PRV to the reservoir at the end of pipe 5. The computer provides the following solution (units are m^3/s or m):

$$Q_1 = 0.102,\ \ Q_2 = -\ 0.022,\ \ Q_3 = 0.108,\ \ Q_4 = 0.077,$$
$$Q_5 = 149.227,\ \ Q_6 = 0.080,\ \ Q_7 = 0.055,\ \ Q_8 = 0.185.$$

From this solution we see that the flow has tried to reverse direction in pipe 5, indicating that the PRV must then close. Thus $Q_5 = 0$, and the reported value of 149.227 is the HGL on the downstream side of the PRV, which is slightly above its pressure setting. Instead of operating in its normal mode, the PRV has acted as a check valve, not permitting the flow to reverse its direction.

Let's increase the demand at node 5 to 0.100 m^3/s. To obtain a solution for this demand, we must change the line that defines the continuity equation at node 5 in subroutine FUNCT to `F(5)=Q5+X(6)-0.100`. Now the execution of the program produces the following solution (units are m^3/s):

$$Q_1 = 0.113,\ \ Q_2 = 0.002,\ \ Q_3 = 0.117,\ \ Q_4 = 0.079$$
$$Q_5 = 0.034,\ \ Q_6 = 0.066,\ \ Q_7 = 0.063,\ \ Q_8 = 0.197$$

With this slightly larger demand at node 5 the PRV operates normally, maintaining an HGL = 149 m on its downstream side. From these discharges the pipe head losses, the pump heads, and other quantities can be evaluated, and from these the head at each node can be determined. Upon carrying out such computations, we find that the heads are H_1 = 173.6 m, H_2 = 155.7 m, H_3 = 155.7 m, H_4 = 180.3 m, and H_5 = 147.2 m. The head on the upstream side of the PRV is 154.8 m, so the PRV loss is 154.8 - 149 = 5.8 m.

If the demand at node 5 is $QJ_5 = 0.16\ m^3/s$, the continuity statement in FUNCT for node 5 is changed to `F(5)=Q5+X(6)-0.16`, then the solution (units are m^3/s) becomes

$$Q_1 = 0.145,\ \ Q_2 = 0.050,\ \ Q_3 = 0.145,\ \ Q_4 = 0.088$$
$$Q_5 = 0.115,\ \ Q_6 = 0.045,\ \ Q_7 = 0.087,\ \ Q_8 = 0.233$$

Without a more complete examination of these solutions, we might be inclined to accept all of them as valid. However, upon computing some head losses and pump heads, the following are found: h_{f7} = 6.98 m, h_{p1} = - 28.0 m, h_{p2} = 4.04 m, and h_{f1} = 4.92 m, so the HGL at node 2 is H_2 = 134.2 m. Obviously the negative head for pump 1 is unrealistic; it is caused by operating the pump with a discharge that is outside the range of the three pairs of points that were used to define this pump characteristic curve. Also H_2 = 134.2 m is much smaller than the HGL setting of the PRV, and since this device can not act as a pump to increase the head across it, the most it can do is open completely. If a PRV opens completely, it typically still acts as a local loss device in a way that is similar to a globe valve with a loss coefficient of about 10. Thus the last solution is not valid, and the problem must be solved again with another local loss device in place of the PRV. Currently there exists no simple *a priori* test to learn that the PRV should open fully and that it is unable to maintain its pressure setting; a solution must first be obtained when we use the Q-equations, because the nodal heads are determined as a secondary step after the discharges are found. The same statements apply to the use of the ΔQ-equations.

The three modes in which a PRV may operate are treated most conveniently with the H-equations, since it is then possible to check directly, as the solution is obtained, whether the head at the upstream side of the PRV is less than H_v (the HGL upstream of the PRV) and whether the head at the downstream pipe node is greater than the set HGL. If the head H at the downstream node becomes larger than the HGL setting of the PRV, then it should shut off the flow in the pipe, and if H_v becomes smaller than HGL, then the PRV should open fully.

When the ΔQ-equations are used to analyze networks that contain PRV's, we must work with two different sets of loops, one around which the ΔQ's circulate, and one around which the energy equations are written. For our 8-pipe example, Fig. 4.27, the first and

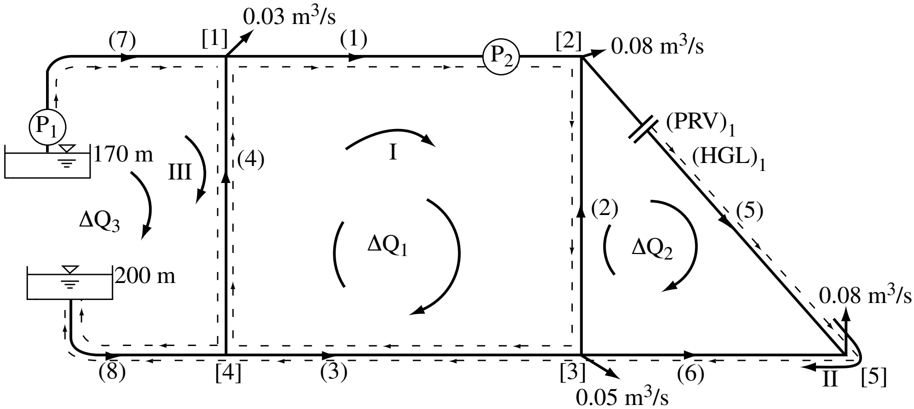


Figure 4.27 The eight-pipe network with a PRV, with loop notation shown.

third equations match the corresponding equations, Eqs. 4.61, for this network without a PRV in pipe 5. The second equation is replaced by

$$
\begin{aligned}
F_2 = K_5'(Q_{o5} + \Delta Q_2)^{n_5} - K_6(Q_{o6} - \Delta Q_2)^{n_6} - K_3(Q_{o3} - \Delta Q_1)^{n_3} \\
- 2(Q_{o3} - \Delta Q_1)^2/(2gA_3^2) - K_8(Q_{o8} - \Delta Q_3)^{n_8} \\
- 10(Q_{o8} - \Delta Q_3)^2/(2gA_8^2) - HGL + 200 = 0
\end{aligned}
\tag{4.64}
$$

It is notable that this equation does not contain ΔQ_2 in every term and that the system of equations does not produce a symmetric Jacobian. To determine the correct operational mode for a PRV when using the ΔQ-equations is much the same as with the Q-equations. Should the flow in a pipe reverse direction, then the PRV should close, and if the HGL at the upstream end of the PRV is less than its setting, then the PRV should open fully; otherwise the PRV is operating in its normal mode. Logic can easily be included in the computer program to check whether the flow is negative in pipes containing PRV's and then change the nature of the problem being solved. The fully-open mode of operation can not be determined until the nodal heads are computed, as with the Q-equations. Should a PRV close, then the discharge in that pipe becomes zero, and the HGL becomes unknown and larger than the setting. If a pipe containing a PRV has only one ΔQ flowing through it, then that corrective discharge becomes known and is $\Delta Q_j = \pm Q_{oi}$, in which the minus sign applies if the assumed directions for Q_{oi} and ΔQ_j coincide, and the plus sign applies if these directions are opposed. In place of ΔQ_j as the unknown, the HGL is unknown, and the number of unknowns still equals the number of equations. Should a PRV close that is internal, with two or more corrective discharges circulating through it, then one of these corrective discharges must be expressed in terms of the others, and the HGL of the PRV replaces this ΔQ in the list of unknowns. In our example, if the PRV were in pipe 2 instead of pipe 5, as shown in Fig. 4.28, then $\Delta Q_2 = Q_{o2} + \Delta Q_1$, and it is replaced by this quantity wherever else ΔQ_2 appears, such as in the discharges for pipes 6 and 7.

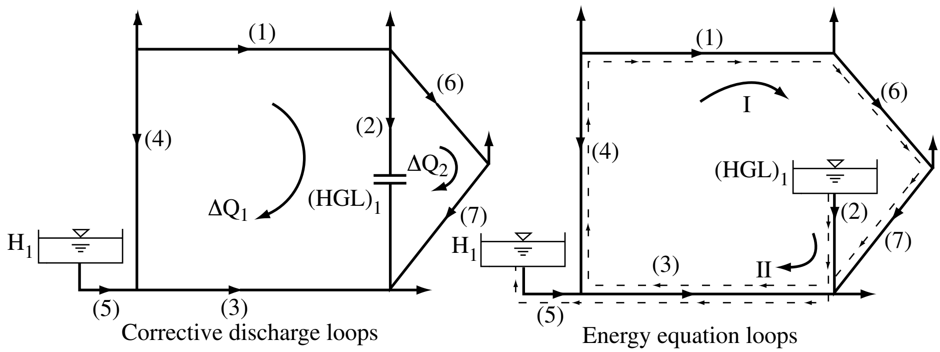


Figure 4.28 The modeling of a PRV in pipe 2 of the 8-pipe network.

To study this problem further, the reader should obtain a listing of FUNCT under the name EPRB4_9.FOR. It can be used to solve this problem. One additional integer variable IDOO has been added to the list of arguments in FUNCT; it is given a value of 0 in the calling program when the equation vector is evaluated and 1 when the Jacobian is evaluated. This variable is needed because we do not want to close the PRV when we evaluate the derivatives. It is instructive to trace the logic that sets Q_5 to zero when the PRV is closed, fixes the value of $\Delta Q_2 = Q_{o5}$ and replaces ΔQ_2 by the HGL as the unknown represented by X(2). These changes are made when Q_5 becomes negative. Subsequent checks might determine whether the HGL becomes less than the PRV setting; if this occurs, the PRV should be reopened. Another modification of this subroutine allows the initial discharges Q_{oi}, i = 1-8, to enter FUNCT through X(i), i = 4-11, thus making it possible to change the demands without changing Q_{oi} within the subroutine.

Example Problem 4.9

Solve the eight-pipe network shown in Fig. 4.27 by using the ΔQ-equations. Obtain this solution first for a demand at node 5 of $QJ_5 = 0.100$ m^3/s and then for $QJ_5 =$ 0.080 m^3/s.

The input data (EPRB4_9.DAT) to solve this problem with a demand of 0.100 m^3/s at node 5 is listed below with the solution:

Input Data				Solution	
DQ1	U	0.00	1	DQ1 =	- 0.00749
DQ2	U	0.00	2	DQ2 =	- 0.00570
DQ3	U	0.00	3	DQ3 =	- 0.01668
Qo1	K	0.12	4	Qo1 =	0.12
Qo2	K	0.00	5	Qo2 =	0.00
Qo3	K	0.11	6	Qo3 =	0.11
Qo4	K	0.07	7	Qo4 =	0.07
Qo5	K	0.04	8	Qo5 =	0.04
Qo6	K	0.06	9	Qo6 =	0.06
Qo7	K	0.08	10	Qo7 =	0.08
Qo8	K	0.18	11	Qo8 =	0.18

Applying these solution values for the ΔQ_i, as appropriate, to the initial discharges gives the final discharges, and then the pipe head losses can be computed, using the proper K and n for each pipe, as listed in the table:

Pipe	1	2	3	4	5	6	7	8
Q, m^3/s	0.1125	- 0.0018	0.1175	0.0792	0.0343	0.0657	0.0633	0.1967
h_L, m	21.40	0.0075	23.20	6.77	2.97	8.03	2.43	12.14

From these discharges the pump heads and local losses are $h_{p1} = 6.18$ m, $h_{p2} = 3.58$ m, $h_{L1} = 0.848$ m, $h_{L2} = 8.18$ m, and $h_{L3} = 1.426$ m. From these the nodal heads can be found as $H_1 = 175.3$ m, $H_2 = 157.5$ m, $H_3 = 155.1$ m, $H_4 = 179.7$ m, $H_5 = 147.0$ m, and $H_{v1} = 156.5$ m. We see that the head upstream from the PRV is 156.5 m which is less than $H_2 = 157.5$ m, so the PRV has not opened fully. The head at node 5 downstream, $H_5 = 147.0$ m, is less than the HGL setting of the PRV (149 m) so the PRV has not closed but operates normally.

When the demand at node 5 is $QJ_5 = 0.080$ m^3/s, then the input data and solution are

Input Data				Solution	
DQ1	U	0.00	1	DQ1 =	- 0.01827
DQ2	U	0.00	2	DQ2 =	149.2
DQ3	U	0.00	3	DQ3 =	- 0.02508
Qo1	K	0.12	4	Qo1 =	0.12
Qo2	K	0.00	5	Qo2 =	0.00
Qo3	K	0.09	6	Qo3 =	0.09
Qo4	K	0.07	7	Qo4 =	0.07
Qo5	K	0.04	8	Qo5 =	0.04
Qo6	K	0.04	9	Qo6 =	0.04
Qo7	K	0.08	10	Qo7 =	0.08
Qo8	K	0.16	11	Qo8 =	0.16

In this input file the initial discharge estimates Qoi have been altered from previous values so that all continuity equations remain satisfied with $QJ_5 = 0.080$ m^3/s. The solution values remind us that ΔQ_2 is actually the HGL at the downstream end of the PRV since it has closed, and FUNCT has set $\Delta Q_2 = - Q_{o5}$ and then used X(2) as the position for HGL. The table lists the discharges and head losses for this situation:

Pipe	1	2	3	4	5	6	7	8
Q, m^3/s	0.1017	0.0217	0.1083	0.0768	0.0	0.0800	0.0549	0.1851
h_L, m	17.83	0.652	19.98	6.40	0.0	11.47	1.887	10.86

The pump heads and local losses are $h_{p1} = 8.02$ m, $h_{p2} = 3.69$ m, $h_{L1} = 0.638$ m, $h_{L2} = 7.25$ m and $h_{L3} = 1.211$ m, with nodal heads of $H_1 = 177.4$ m, $H_2 = 163.2$ m, $H_3 = 160.7$ m, $H_4 = 181.9$ m, and $H_5 = 149.2$ m. Now the PRV has closed entirely so the flow in pipe 5 is zero, and the HGL at its downstream end is above its set point.

* * *

4.4.5. SYSTEMATIC SOLUTION OF THE Q-EQUATIONS

In earlier sections we have developed the three systems of equations that can be used to analyze pipe networks. We have written these equation systems for several small networks, seen how the Newton method can be applied to any system of nonlinear equations and how to solve a problem by using a general purpose program that implements the Newton method for all three equation systems, and finally we have carried out detailed computations by hand to obtain some solutions by iteration. In this section we will see how this knowledge can be used in developing computer programs that will analyze any pipe network by using the Q-equations, and the programs will require only enough information from the user to describe adequately the network and its connectivity. In the next two sections similar programs will be developed for the solution of the H-equations and the ΔQ-equations.

Let us begin by assuming that there are no local losses. If they exist, they can be modeled simply by assigning a larger equivalent sand roughness, or Hazen-Williams C_{HW}, to the pipes containing minor losses. Here we ignore the Manning equation.

In describing any network of pipes, we need two types of information: (1) Pipe information consisting of the diameter, length, roughness coefficient, and end nodes for each pipe. This information can be called pipe-oriented data, since we assemble it by going though a list of the pipes in the network; (2) Junction information, including the demand at the junction, its elevation, and possibly the pipes that join at the junction. This information is called node-oriented data, since it is assembled by moving through the nodes of the network. Actually the connectivity of the network can be defined either by giving the nodes at the ends of each pipe, or by giving the pipe numbers that join at each node. We shall use this duplicative information to check that the user has not erred in defining the network.

Now we shall describe the input data that are required. Details on the form of this input will be provided subsequently. Prior to study of this section the reader should obtain a listing of the program SOLQEQS.FOR (or C if you prefer) from the text CD. The information that is required from the user is the following:

1. A line that gives (a) the number of pipes, (b) the number of nodes, (c) the number of reservoirs that supply the network, (d) the number of pumps, and (e) the options which you wish to change from the default values. (The default options and how these are changed will be described later.)
2. For each pipe, list its (a) number, (b) upstream node, (c) downstream node, (d) length, (e) diameter, and (f) roughness coefficient.
3. For each node, list its (a) number, (b) demand, and (c) elevation, and (d) a list of pipes that join this node.
4. For each reservoir, list (a) the pipe number that connects this reservoir to the network, and (b) the water surface elevation of the reservoir.
5. For each pump, list (a) the number of the pipe that contains the pump, and (b) three (Q, h_p) pairs of discharge vs. pump head that will allow its operating characteristics to be defined.
6. Finally, because the algorithms that could be used to determine the minimum set of independent loops for the energy equations are relatively complex, we require a list of pipe numbers around each loop (with a minus sign before a pipe number if the movement around the loop opposes the assumed direction of flow in that pipe). We require that pseudo loops be provided first, and then the real loops.

The information in item one is used to dimension the arrays that will store the remaining information about the pipe network and to determine how many lines of each information type will be read from the input data file. The information must be provided in the sequence that is listed above.

The program must perform five major tasks:

1. Read the input data that defines the network.
2. Develop from this information the system of Q-equations, i.e., the junction continuity equations and the energy equations around pseudo and real loops of the network. This task defines the equations and also forms each element of the Jacobian matrix.
3. Solve the system of equations. Here we will simply call a standard linear algebra solver. However, a program for larger network problems should have a special

linear algebra solver that takes advantage of the special properties of a sparse Jacobian matrix.

4. Obtain the head H at each node after the pipe discharges have been found.
5. Write the solution results in tables that can be readily understood. We choose to provide these results in two tables: a pipe data table and a node data table.

In reading the pipe numbers that connect at a node and the pipe numbers that define a loop, a pointer is used to separate data for consecutive nodes, and a second pointer separates data for consecutive loops. The pipes that join at nodes are placed consecutively in a one-dimensional array *JN*, with *NN* pointing to the position in this array that separates data for consecutive nodes. A similar one-dimensional array *IK* contains the pipe numbers that form the loops, with *LP* pointing to the first pipe number in each loop. The use of one-dimensional arrays with pointers is a more efficient use of storage than the use of two-dimensional arrays, because the second subscript of a two-dimensional array must then be at least as large as the maximum number of pipes that may exist in a loop.

When solving the Q-equations (or ΔQ-equations), we compute the nodal heads after obtaining the solution for the discharges. These heads are found by starting at all reservoirs and computing each H at the node at the other end of a pipe from (to) the reservoir by subtracting (adding) the pipe head loss from (to) the reservoir water surface elevation. After these heads have been determined, the nodes one pipe away from these can be determined next, and so on. This process continues until the head at every node has been determined.

In program SOLQEQS the computation of heads begins after the PIPE DATA table is written by the `DO 130` loop. This loop finds each head at the other end of a pipe that is connected to a reservoir, and upon computing H the integer array INDX, with its argument equal to this node number, is set to 1 to show that nodal head has been computed. Now loop `DO 160` passes through the nodes, but nothing is done if INDX(I) for node I is zero. Otherwise INDX(I) = 1, and then the pipes that join this node are accessed through array JN; if H at the other end of a pipe is not known, it is computed. Since not all nodal heads will be found from the first pass through the nodes, the integer IJ also accumulates the number of nodes whose head has been found. One way to learn if another pass is needed is to check whether IJ is less than NJ, the total number of nodes. Actually we see whether IJ has increased from the previous pass. If so, we pass through the nodes again. This method may result in passing through the nodes one extra time, but it prevents the creation of an infinite loop if there is an error in the network description so that fewer than NJ heads can be found. After finding every head, the NODE DATA table is written. The program then allows the user to solve another problem whose data is in a different file, or to change the peaking factor for the same network.

Detailed instructions on the preparation of input data to SOLQEQS follows:

Line 1: No. of Pipes (NP), No. of Nodes (NJ), No. of Reservoirs (NRES), No. of Pumps (NPUMP), No. of Options (NOPT), Option Pairs.

The options consist of a letter in quotes followed by a value, as follows:

Letter	Controls What	Value	Default
U or u	ES or SI units	0 = ES, 1 = SI	0
Q or q	Discharge units	0 = ft^3/s, 1 = gal/min, 2 = mgd, 3 = m^3/s, 4 = l/s	0 or 3
D or d	Pipe diameter and roughness units	0 = in, 1 = ft, 2 = m, 3 = cm, 4 = mm	0 if ES 4 if SI
F or f	Peaking factor	Multiplier of demands	1.0
V or v	Kinematic viscosity	ν = value	ES, 1.217E-5 SI, 1.310E-6
G or g	Specific weight	γ = value	ES, 62.4 SI, 9806.0
C or c	Network check	1 = yes, 0 = no	1

Here is an example of specifying options: 2 'U' 1 'F' 1.5. The 2 indicates two options are being changed, to specify SI units and to specify a peaking factor of 1.5. In giving the options, the units (ES or SI) option should appear first if it is to be elected, but otherwise the options can be given in any order.

Next group: Pipe data consisting of NP lines, each giving pipe number, node 1, node 2, length, diameter, and roughness. Pipes must be numbered consecutively, starting with 1, but they need not be entered consecutively. The roughness may be either the equivalent sand roughness e (in the same units as the diameter) for use in the Colebrook-White and Darcy-Weisbach equations, or a Hazen-Williams C_{HW}, and these may be mixed. The program decides which equation to use, based on the roughness size.

Next group: Node data consisting of NJ lines, each giving node number, demand, elevation, number of pipes at the node, and a list of these pipe numbers with a minus if the flow is from the junction. This information is used to define the junction continuity equations.

Next group: Reservoir data consisting of NRES lines, each giving the number of the pipe connecting the reservoir to the network and the water surface elevation.

Next group: Pump data consisting of NPUMP lines, each with the number of the pipe containing the pump, followed by three (Q, h_p) pairs to define the pump curve.

Next group: Loop data consisting of NL = NP - NJ lines, one loop on each line with the number of pipes in the loop and a list of these pipes. A negative sign must precede the pipe number if the direction around the loop opposes the assumed direction of flow in this pipe. Pseudo loops connecting reservoirs must appear first in this list, and the real loops follow.

Example Problem 4.10

Use program SOLQEQS to solve the network of Example Problem 4.5. Obtain two solutions: (1) for the given demands and (2) with these demands multiplied by 2.0.

The input data takes the form

```
6 5 1 0 1 'D' 1
1 0 1 1500 0.667 0.000417
2 1 2 1000 0.5 0.000417
3 2 4 1500 0.5 0.000417
4 1 3 1500 0.5 0.000417
5 3 4 1200 0.5 0.000417
6 4 5 1000 0.333 0.000417
1 0.50 350.  3  1  - 2  - 4
2 0.35 350.  2  2  - 3
3 0.50 350.  2  4  - 5
4 0.50 350.  3  3  5  - 6
5 0.25 350.  1  6
1 500
4 2 3 - 5 - 4
```

or, if the diameters and e are given in inches (Inches is the default; either giving 0 options as the last 0 in the first line below, or giving 1 'D' 0, tells SOLQEQS to use inches), then the input would be

```
6 5 1 0 0
1 0 1 1500 8.0 0.005
2 1 2 1000 6.0 0.005
3 2 4 1500 6.0 0.005
4 1 3 1500 6.0 0.005
5 3 4 1200 6.0 0.005
6 4 5 1000 4.0 0.005
1 0.50  350.  3  1  - 2  - 4
2 0.35  350.  2  2  - 3
3 0.50  350.  2  4  - 5
4 0.50  350.  3  3  5  - 6
5 0.25  350.  1  6
1 500
4 2 3 - 5 - 4
```

When prompted after the first solution, we give the peaking factor 2.0. The solution tables follow. In the last NODE DATA table we see that some heads are negative, so the network is unable to supply demands that are double the initial values.

PIPE DATA

PIPE NO.	NODES FROM	TO	L ft.	DIA. ft	e x10⁴ ft	Q ft³/s	VEL. ft/s	HEAD LOSS ft.	HLOSS/ 1000
1	0	1	1500	0.667	4.17	2.10	6.02	23.50	15.67
2	1	2	1000	0.500	4.17	0.82	4.18	11.00	11.00
3	2	4	1500	0.500	4.17	0.47	2.39	5.67	3.78
4	1	3	1500	0.500	4.17	0.78	3.97	14.97	9.98
5	3	4	1200	0.500	4.17	0.28	1.43	1.70	1.42
6	4	5	1000	0.333	4.17	0.25	2.87	8.83	8.83

NODE DATA

NODE	DEMAND Estimate	DEMAND ft³/s	ELEV. ft.	HEAD ft.	PRESSURE lb/in²	HGL ELEV. ft.
1	0.5	0.500	350.0	126.50	54.82	476.50
2	0.3	0.350	350.0	115.50	50.05	465.50
3	0.5	0.500	350.0	111.53	48.33	461.53
4	0.5	0.500	350.0	109.82	47.59	459.82
5	0.3	0.250	350.0	101.00	43.77	451.00

For peaking factor = 2.0:

PIPE DATA

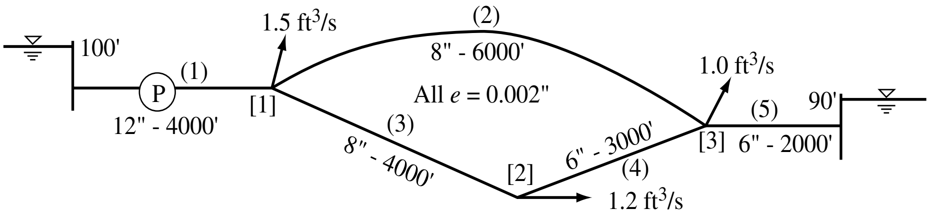
PIPE NO.	NODES FROM	TO	L ft.	DIA. ft	e x10⁴ ft	Q ft³/s	VEL. ft/s	HEAD LOSS ft.	HLOSS/ 1000
1	0	1	1500	0.667	4.17	4.20	12.03	91.53	61.01
2	1	2	1000	0.500	4.17	1.64	8.36	42.48	42.48
3	2	4	1500	0.500	4.17	0.94	4.79	21.53	14.35
4	1	3	1500	0.500	4.17	1.56	7.94	57.69	38.46
5	3	4	1200	0.500	4.17	0.56	2.85	6.32	5.27
6	4	5	1000	0.333	4.17	0.50	5.73	33.66	33.66

NODE DATA

NODE	DEMAND Estimate	DEMAND ft³/s	ELEV. ft.	HEAD ft.	PRESSURE lb/in²	HGL ELEV. ft.
1	1.0	1.000	350.0	58.48	25.34	408.48
2	0.7	0.700	350.0	15.99	6.93	365.99
3	1.0	1.000	350.0	0.79	0.34	350.79
4	1.0	1.000	350.0	- 5.53	- 2.40	344.47
5	0.5	0.500	350.0	- 39.20	- 16.99	310.80

* * *

Example Problem 4.11
Use program SOLQEQS to analyze the 5-pipe, 3-node network in the figure. In pipe 1 is a pump, with the characteristics given in the table, which is connected to a reservoir. Let $\nu = 1.417 \times 10^{-5}$ ft²/sec. The elevation of all nodes is zero.



The pump curve is described by data in the following table:

Q, ft³/s	***H***, ft
4.5	54
4.0	50
3.5	44

The input data this problem are listed first in two columns, followed by the solution tables.

```
5 3 2 1 1 'V' 1.417E-5          2 1.2 0  2  3  - 4
1  0  1  4000  12  0.002        3 1.0 0  3  2  4  - 5
2  1  3  6000  8  0.002         1  100
3  1  2  4000  8  0.002         5  90
4  2  3  3000  6  0.002         1   4.5   54   4.0   50   3.5   44
5  3  0  2000  6  0.002         3   1  2  5
1 1.5 0  3  1  - 2  - 3         3  2  - 4  - 3
```

PIPE DATA

PIPE NO.	NODES FROM	TO	L ft.	DIA. in	e x10³ in	Q ft³/s	VEL. ft/s	HEAD LOSS ft.	HLOSS/ 1000
1	0	1	4000	12.0	2.0	4.13	5.26	26.24	6.56
2	1	3	6000	8.0	2.0	1.21	3.45	29.13	4.85
3	1	2	4000	8.0	2.0	1.42	4.08	26.48	6.62
4	2	3	3000	6.0	2.0	0.22	1.14	2.64	0.88
5	3	0	2000	6.0	2.0	0.43	2.18	5.85	2.93

Pump 1 in pipe 1: Head = 52.21 ft, Q = 4.13 ft³/s

NODE DATA

NODE	DEMAND Estimate	ft³/s	ELEV. ft.	HEAD ft.	PRESSURE lb/in²	HGL ELEV. ft.
1	1.5	1.500	0.0	124.98	54.16	124.98
2	1.2	1.200	0.0	98.50	42.68	98.50
3	1.0	1.000	0.0	95.85	41.54	95.85

*   *   *

Example Problem 4.12

Solve the 7-pipe, 4-node network shown in Fig. 4.6, which contains a PRV in pipe 6, by using program SOLQEQS.

The input data for this problem are listed after this paragraph. Then the two solution tables follow. Several observations should be made here: In the lines of nodal data the information after the nodal demand that lists the pipes that join at a node is used to define the junction continuity equations; therefore the list of pipes that join at node 1 must include pipe 6 with the PRV in it. The input that lists the pipes that define the loops of the network are used to define the energy loop equations; this group should therefore define a loop that starts (or ends) at the artificial reservoir created by the PRV, so for this network there will be two pseudo loops and one real loop. Also, since the downstream part of pipe 6 defines K', its length is 500 ft, and its upstream node is given as 0 (a reservoir).

```
7 4 3 1 1 'C' 0
1 0 1 1000 6 0.02
2 1 2 1000 6 0.02
3 3 2 800 6 0.02
4 0 3 200 6 0.02
5 3 4 2000 6 0.02
6 0 4 500 6 0.02
7 1 3 1500 1 0.02
1 0 50 4 1 - 2 - 6 - 7
2 1 50 2 2 3
3 0 50 4 4 7 -3 -5
4 20 1 2 5 6
1 90
4 100
6 55
1 1.0 60 1.5 55 2.0 48
3 1 7 - 4
3 6 - 5 - 4
3 2 - 3 - 7
```

PIPE DATA

PIPE NO.	NODES FROM	TO	L ft.	DIA. in	e x10^3 in	Q ft^3/s	VEL. ft/s	HEAD LOSS ft.	HLOSS/ 1000
1	0	1	1000	6.0	20.0	1.11	5.65	27.28	27.28
2	1	2	1000	6.0	20.0	1.07	5.43	25.26	25.26
3	3	2	800	6.0	20.0	- 0.07	- 0.34	0.10	0.12
4	0	3	200	6.0	20.0	0.89	4.54	3.53	17.74
5	3	4	2000	6.0	20.0	0.96	4.91	41.47	20.74
6	0	4	500	6.0	20.0	0.04	0.18	0.04	0.04
7	1	3	1500	1.0	20.0	0.01	1.31	25.56	17.04

Pump 1 in pipe 1: Head = 59.09 ft, Q = 1.11 ft^3/s

NODE DATA

NODE	DEMAND Estimate	ft^3/s	ELEV. ft.	HEAD ft.	PRESSURE lb/in^2	HGL ELEV. ft.
1	0.0	0.000	50.0	71.81	31.1	121.81
2	1.0	1.000	50.0	46.55	20.2	96.55
3	0.0	0.000	50.0	46.45	20.1	96.45
4	1.0	1.000	20.0	34.98	15.2	54.98

* * *

4.4.6. SYSTEMATIC SOLUTION OF THE *H*-EQUATIONS

This section is similar to Section 4.4.5, but now the objective is to describe a program that analyzes a network by solving the H-equations. This program will be restricted to the solution of the H-equations for networks that do not contain a PRV or a BPV and in which minor losses can be neglected. (Exercises to include these devices can be found in the end-of-chapter problems.) With these restrictions the Jacobian matrix of the equation system is symmetric. Symmetry occurs because the partial derivatives of terms which describe the discharge in pipe k between nodes i and j, such as $\{(H_i - H_j)/K_k\}^{1/n_k}$, will be the same in the equations where this discharge occurs, so long as neither i nor j are the node for which this junction continuity equation is written. With the sign convention that flow to a junction is positive and flow from a junction is negative, this term will be preceded by a plus sign when j is the junction for which the equation is written. The derivative with respect to H_i will be positive. The derivative with respect to H_j will be negative. If the term describes a pipe whose flow leaves the junction, a negative sign will precede the term and i will be the junction for which the equation is written, and the derivative for the other node j will be positive. Thus the off-diagonal elements of the Jacobian matrix are positive, and the diagonal elements are negative, as we have already seen in an example. We will utilize this symmetry property in developing an algorithm to generate the Jacobian. However, we first examine alternative means for evaluating the derivatives.

The direct way to differentiate $\{(H_i - H_j)/K\}^{1/n}$, in which the pipe number subscript k has been omitted, is to use the power rule of calculus to obtain

$$\pm[\{(H_i - H_j)/K\}^{(1-n)/n}]/(nK) \tag{4.65}$$

in which the minus sign applies when differentiating with respect to H_j. When a pump is present in the pipe, however, it is no longer a straightforward process to differentiate this term, as it now is $\{(H_i + h_p - H_j)/K\}^{1/n}$, in which $h_p = h_p(Q)$ is normally expressed as $h_p = AQ^2 + BQ + C$.

Another way to obtain these derivatives is to start with

$$Q = \left[\frac{H_i - H_j}{K}\right]^{1/n} \tag{4.66}$$

and compute the differential of this formula as

$$dQ = \{H_i - H_j\}^{1/n-1} dH/(nK^{1/n}) = Q^{1-n} dH/(nK) \tag{4.67}$$

The partial derivative with respect to H_i is then

$$\partial Q/\partial H_i = Q^{1-n}/(nK) \tag{4.68}$$

and the partial derivative with respect to H_j is identical, except for a minus sign. So Jacobian matrix elements can be obtained quickly via Eq. 4.68. With this approach we can compute the Jacobian terms for a pipe with a pump in it. Also write $\{(H_i - H_j)/K\}^{1/n}$ as $|(H_i - H_j)/K|^{1/n-1}(H_i - H_j)/K$ to allow a sign change for flows that oppose the assumed direction, which may occur during an intermediate iteration even if the assumed direction is correct for the final solution.

When a pump is present in a pipe, then we can write

$$H_i - H_j + AQ^2 + BQ + C - KQ^n = 0 \tag{4.69}$$

Following the procedure of computing the differential of this equation, we find

$$\partial Q/\partial H = \pm 1/(nKQ^{n-1} - 2AQ - B) \tag{4.70}$$

If the derivative is with respect to H_i, choose the plus sign; otherwise choose the minus sign for H_j. Thus, for a pipe containing a pump, Eq. 4.69 is first solved for Q, and this Q is then used in Eq. 4.70 to evaluate the derivatives for the Jacobian matrix.

Now we can modify SOLQEQS to solve a system of H-equations. Now please obtain a listing from the CD of SOLHEQS and refer to it as you read this section. Here a

NAMELIST (actually an extension of standard Fortran 77, but implemented in many Fortran compilers) will be used to handle options. The NAMELIST will also be used in programs in later chapters. The options that may be in the &OPTIONS list are the following: IU (to set ES or SI units), IQ (to set the discharge units), ID (to give the units for diameters and roughnesses), IC (= 0 to omit checking the dual network connectivity description), VIS (kinematic viscosity), PF (peaking factor), GAMMA (specific weight) and ERR (Newton error criterion). With the *H*-equations there are no loop energy equations, so the input for loops is eliminated, as is the program segment that generates the loop equations. The section that creates the system of equations will include the junction continuity equations, but this section is modified to implement the new method of obtaining the system Jacobian and the *H*-equations. In SOLHEQS the length of array H, which stores the nodal heads, has been augmented to include the reservoir heads, so that H_i and H_j now provide the nodal heads at the ends of each pipe, including those that supply the network from a reservoir. So we can easily detect a pump in a pipe, its upstream node number is changed to a negative value. The function subprogram COMPK_N now supplies n, but $(1 - 1/n)$ is stored in array N for later use.

In this program we must have access to discharge values during any iteration for any pipe containing a pump. We do this by computing the discharge from the heads that exist during any iteration by letting ARG = $[(K/(H_i - H_j)]$ and DD = ARG$^{(1 - 1/n)}$; then we find that Q = ARG/DD. Statements following label 146 in the program listing carry out this step. When a pump exists in the pipe, then the Newton method is used to solve Eq. 4.69 by statements found in the `DO 145` loop.

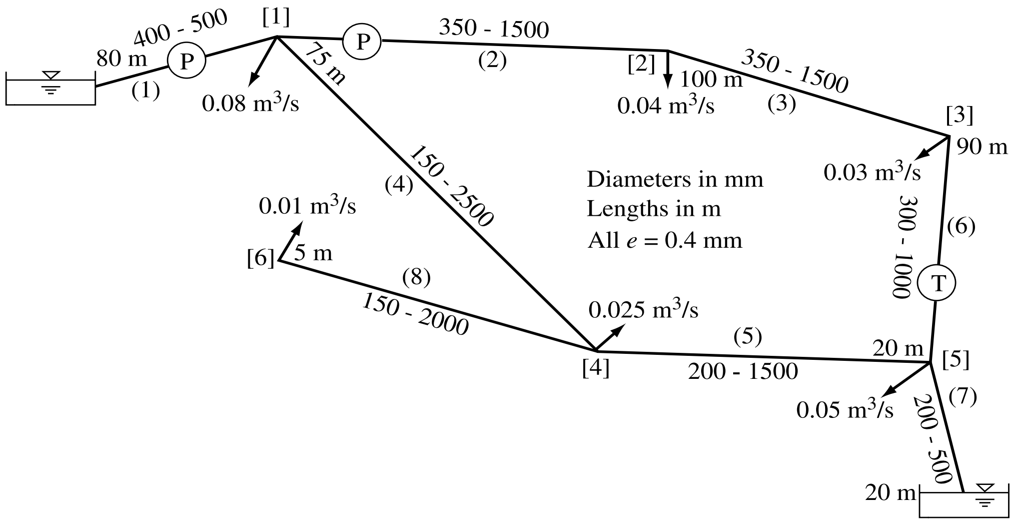
Example Problem 4.13
Prepare suitable input data to analyze the network of Example Problem 4.11 by using program SOLHEQS.

Only minor modifications to the input data in Example Problem 4.11 are needed. First, because the options are entered via a NAMELIST in program SOLHEQS, the first line of the input data now should contain only four values: the numbers of pipes, nodes, reservoirs, and pumps. The second line of input data begins with &OPTIONS, and the next entries contain the namelist variables that differ from the default values, each followed by an equals sign and the value of that variable. This list is ended with a /. Since no loop data are needed, the two lines of loop data are deleted from the input data for the solution to Example Problem 4.11. Since IQ = 0 is the default value, it need not be included in the list after &OPTIONS. With these changes, the input data are now as follows:

```
5 3 2 1
&OPTIONS IQ=0,VIS=1.417E-5/
1 0 1 4000 12 0.002
2 1 3 6000 8 0.002
3 1 2 4000 8 0.002
4 2 3 3000 6 0.002
5 3 0 2000 6 0.002
1 1.5 0 3 1 -2 -3
2 1.2 0 2 3 -4
3 1.0 0 3 2 4 -5
1 100
5 90
1 4.5 54 4.0 50 3.5 44
```

*　　　　*　　　　*

Example Problem 4.14
The network below is supplied by the source pump in pipe 1, and a booster pump is needed to get the water over the hill below nodes 2 and 3. A turbine is placed in pipe 6 to extract the extra head after the water is moved over the hill crest. Analyze this network using program SOLHEQS. Diameters are in mm, and lengths in m. The kinematic viscosity is $\nu = 1.31 \times 10^{-6}$ m^2/s.



Source Pump

Q m^3/s	h_p m
0.20	50
0.30	47
0.50	43

Booster Pump

Q m^3/s	h_p m
0.20	15
0.25	14
0.30	12

Turbine

Q m^3/s	h_t m
0.15	- 30
0.25	- 25
0.35	- 18

The turbine can be modeled as a pump; the heads are recorded as negative values in preparing its operating characteristics. Since this network is described in SI units, the options for units, discharges and diameters must all be specified. The input data file for this problem, listed in two columns, is therefore

```
8 6 2 3
&OPTIONS IU=1,IQ=3,ID=4/
1 0 1 500 400 0.4
2 1 2 1500 350 0.4
3 2 3 1500 350 0.4
4 1 4 2500 150 0.4
5 5 4 1500 200 0.4
6 3 5 1000 300 0.4
7 5 0 500  200 0.4
8 4 6 2000 150 0.4
1 75 0.08 3 1 - 2 - 4
2 100 0.04 2 2 - 3
3  90 0.03 2 3 - 6
4  10 0.05 3 4 5 - 8
5  20 0.025 3 6 - 5 - 7
6   5 0.01  1 8
1 80
7 20
1  0.2  50  0.3  47  0.5  43
2  0.20  15  0.25  14  0.30  12
6  0.15  - 30  0.25  - 25  0.35  - 18
```

The solution tables from SOLHEQS are

PIPE DATA

PIPE NO.	NODES FROM	TO	L m	DIA. mm	e x10^2 mm	Q m^3/s	VEL. m/s	HEAD LOSS m	HLOSS/ 1000
1	0	1	500	400	40.0	0.330	2.63	8.78	17.55
2	- 1	2	1500	350	40.0	0.217	2.26	23.02	15.35
3	2	3	1500	350	40.0	0.177	1.84	15.39	10.26
4	1	4	2500	150	40.0	0.033	1.87	76.55	30.62
5	5	4	1500	200	40.0	0.027	0.86	6.93	4.62
6	- 3	5	1000	300	40.0	0.147	2.08	15.87	15.87
7	5	8	500	200	40.0	0.095	3.03	27.83	55.66
8	4	6	2000	150	40.0	0.010	0.57	5.89	2.95

Pump 1 in pipe 1: Head = 46.22 m, Q = 0.330 m^3/s
Pump 2 in pipe 2: Head = 14.77 m, Q = 0.217 m^3/s
Pump 3 in pipe 6: Head = - 30.11 m, Q = 0.147 m^3/s

NODE DATA

NODE	DEMAND Estimate	m^3/s	ELEV. m	HEAD M	PRESSURE kPa	HGL ELEV. m
1	0.1	0.080	75.0	42.45	416.2	117.45
2	0.0	0.040	100.0	9.19	90.1	109.19
3	0.0	0.030	90.0	3.80	37.3	93.80
4	0.1	0.050	10.0	30.90	303.0	40.90
5	0.0	0.025	20.0	27.83	272.9	47.83
6	0.0	0.010	5.0	30.01	294.3	35.01

* * *

4.4.7. SYSTEMATIC SOLUTION OF THE ΔQ-EQUATIONS

In this section we describe the development of the computer program SOLDQEQS that is based on the ΔQ-equations and analyzes pipe networks. This program requires the user to specify the initial discharges, Q_{oi}, so they satisfy all of the junction continuity equations, because algorithms that do this automatically involve considerable logic. We will also omit the input that provides the dual description of the network connectivity; instead we will generate the pipe numbers that interconnect the network nodes from the data on the nodes at the ends of the pipes. This generated data will be used to verify that the input Q_{oi} do satisfy the junction continuity equations. Finally, this program will not allow a PRV or any similar device in the network. With this restriction the Jacobian matrix will be symmetric and positive definite, thereby allowing a special linear algebra solver that requires only the upper (or lower) triangular and diagonal elements of the Jacobian to be available during the solution process. This approach provides us a solution variant that could also be used in solving the H-equations by the Newton method.

To describe the computer program logic that forms the ΔQ-equations and the derivatives that form the Jacobian elements, it will be convenient to be able to refer to the equations and the nonzero derivatives with respect to ΔQ from an example. At this time obtain a listing of SOLDQEQS from the CD so it can be studied while you read the rest of this section. The network in Example Problem 4.14 will serve the purpose of illustrating the logic of this program. Since this network contains several pumps, one of which produces a negative head as a turbine, this example will help us incorporate pumps correctly into the code. The two ΔQ-equations for this network are

$$\begin{aligned} F_1 = K_1(Q_{o1}+\Delta Q_1)^{n_1} - h_{p1} + K_4(Q_{o4}+\Delta Q_1-\Delta Q_2)^{n_4} \\ - K_5(Q_{o5}-\Delta Q_1+\Delta Q_2)^{n_5} + K_7(Q_{o7}+\Delta Q_1)^{n_7} - 80 + 20 = 0 \end{aligned} \tag{4.71}$$

and

$$\begin{aligned} F_2 = K_2(Q_{o2}+\Delta Q_2)^{n_2} - h_{p2} + K_3(Q_{o3}+\Delta Q_2)^{n_3} + K_6(Q_{o6}+\Delta Q_2)^{n_6} \\ - h_{p3} + K_5(Q_{o5}-\Delta Q_1+\Delta Q_2)^{n_5} - K_4(Q_{o4}+\Delta Q_1-\Delta Q_2)^{n_4} = 0 \end{aligned} \tag{4.72}$$

In these equations the head h_{pj} of pump j is described by $h_{pj} = A_j(Q_{oi} \pm \sum \Delta Q_k)^2 + B_j(Q_{oi} \pm \sum \Delta Q_k) + C_j$, and the coefficients A, B, and C are chosen to fit three pairs of points along the pump curve, as before. These energy equations are written around the same loops in which the corrective discharges ΔQ_1 and ΔQ_2 circulate. Therefore, every term in Eq. 4.71 contains a ΔQ_1, and every term in Eq. 4.72 contains a ΔQ_2.

The Jacobian $[\boldsymbol{D}]$ will have two rows, one for each of the two equations, and two columns corresponding to the two unknowns ΔQ_1 and ΔQ_2, or

$$[D]=\begin{bmatrix}\dfrac{\partial F_1}{\partial \Delta Q_1} & \dfrac{\partial F_1}{\partial \Delta Q_2}\\ \dfrac{\partial F_2}{\partial \Delta Q_1} & \dfrac{\partial F_2}{\partial \Delta Q_2}\end{bmatrix} \tag{4.73}$$

in which the individual elements are

$$\frac{\partial F_1}{\partial \Delta Q_1} = n_4K_4(Q_{o4}+\Delta Q_1-\Delta Q_2)^{n_4-1}+n_5K_5(Q_{o5}-\Delta Q_1+\Delta Q_2)^{n_5-1}$$
$$+n_7K_7(Q_{o7}+\Delta Q_1)^{n_7-1}+n_1K_1(Q_{o1}+\Delta Q_1)^{n_1-1}-2A_1(Q_{o1}+\Delta Q_1)-B_1 \tag{4.74}$$

$$\frac{\partial F_1}{\partial \Delta Q_2} = \frac{\partial F_2}{\partial \Delta Q_1} = -\,n_4K_4(Q_{o4}+\Delta Q_1-\Delta Q_2)^{n_4-1}-n_5K_5(Q_{o5}-\Delta Q_1+\Delta Q_2)^{n_5-1} \tag{4.75}$$

$$\frac{\partial F_2}{\partial \Delta Q_2} = n_2K_2(Q_{o2}+\Delta Q_2)^{n_2-1}+n_3K_3(Q_{o3}+\Delta Q_2)^{n_3-1}+n_6K_6(Q_{o6}+\Delta Q_2)^{n_6-1}$$
$$+n_5K_5(Q_{o5}-\Delta Q_1+\Delta Q_2)^{n_5-1}+n_4K_4(Q_{o4}+\Delta Q_1-\Delta Q_2)^{n_4-1}$$
$$-2A_2(Q_{o2}+\Delta Q_2)-B_2-2A_3(Q_{o6}+\Delta Q_2)-B_3 \tag{4.76}$$

To allow for the possibility that one of more flows might change direction and $Q_{oi} \pm \sum \Delta Q_k$ would become negative, the quantities $K_i(Q_{oi} \pm \sum \Delta Q_k)^{n_i}$ will be rewritten as $K_i\,|(Q_{oi} \pm \sum \Delta Q_k)|^{n_i-1}(Q_{oi} \pm \sum \Delta Q_k)$. Doing this will be convenient since all factors but the last are also needed to evaluate terms in the derivatives.

For this program we must define the loops around which (1) the energy equations will be written, and (2) each ΔQ circulates. Thus the user must supply the pipe numbers which define each energy loop, with a negative pipe number whenever the direction around the loop opposes the assumed direction of flow in that pipe. This information was also required as input to SOLQEQS. These loop data determine the terms in each equation and the sign of each term. As in SOLQEQS, this data resides in a one-dimensional integer array *IK*, with a pointer *LP* to separate individual loops. The corrective loop discharge data for each pipe is in a similar array *IK1*, with a pointer *LP1* to separate entries between individual pipes. Thus the positions in array *IK1* that will contain information on a corrective loop discharge through pipe *I* will start with subscript (argument of the array) *LP1*(*I*) + 1 and end with subscript *LP1*(*I*+1). Thus *LP1* must have dimension *NP* + 1. In a similar way *LP* must have dimension *NL* + 1 = *NP* - *NJ* + 1.

Let us now examine an algorithm to obtain the corrective loop discharges in each pipe from the loop information. The pipes around the two loops in the example network are

Loop 1: 1 4 - 5 7

Loop 2: 2 3 6 5 - 4

and this data will be stored in *IK* as follows:

IK(1) = 1, *IK*(2) = 4, *IK*(3) = - 5, *IK*(4) = 7,

$IK(5) = 2,\ IK(6) = 3,\ IK(7) = 6,\ IK(8) = 5\ \ IK(9) = -4,$

with $LP(1) = 0,\ LP(2) = 4,$ and $LP(3) = 9.$

Since ΔQ_1 circulates through loop 1 and ΔQ_2 circulates through loop 2, we see that the loop number (the argument of LP) gives the corrective loop discharge through a pipe when the pipe number occurs in the list of IK's for that loop. For example, since pipe 4 is a pipe number in loops 1 and 2, the corrective loop discharges ΔQ_1 and ΔQ_2 both circulate through it, and also ΔQ_1 is in the same direction as the assumed flow in pipe 4 since it is positive in loop 1, whereas ΔQ_2 opposes the assumed flow since it is negative in the list of pipes in loop 2. The number of corrective loop discharges through a pipe is not known in advance, so it is simpler to use a two-dimensional array initially, with the pipe number as the first subscript and the number of corrective loop discharges through that pipe as the second subscript. Hence the second subscript of this array must equal or exceed the maximum number of ΔQ's passing through any pipe, so most of this array space will be unused; once these numbers are known, the information can be transferred into the one-dimensional array $IK1$. Then the two-dimensional array can be deallocated and the memory used for other purposes. An alternative for this array is to EQUIVALENCE it to another array that is not used until later, such as the array for the Jacobian matrix. Figure 4.29 lists Fortran statements that could be used to generate these arrays, with the array $LP1$ zeroed before beginning this algorithm.

A very similar algorithm can be used to generate the pipe numbers that join at each node. The essential difference is that the upstream and downstream nodes in the arrays $L1$ and $L2$ identify the node to which the pipes attach. In program SOLDQEQS this started in the `DO 24` loop. Since we want to verify that the user-supplied initial discharges Q_{oi}

```
      DO 50 I=1,NL
      DO 50 J=LP(I)+1,LP(I+1)
      II=IABS(IK(J))
      NI=LP1(II)+1
      IK2(II,NI)=ISIGN(I,IK(J))
   50 LP1(II)=NI
      NI=0
      NCT=NI
      DO 54 I=1,NP
      DO 53 J=1,LP1(I)
      NI=NI+1
   53 IK1(NI)=IK2(I,J)
      LP1(I)=NCT
   54 NCT=NI
      LP1(NP+1)=NI
```

Figure 4.29 Listing of Fortran code to generate arrays $IK1$ and $LP1$.

do satisfy all of the junction continuity equations, this check is performed immediately after the pipes that join at each node are identified. This information makes it easy to obtain the heads H at the nodes after the discharges and head losses in the pipes are computed by using essentially the algorithm that is in SOLQEQS for this purpose.

Now let's see how to obtain the system of ΔQ-equations and the Jacobian that are needed to implement the Newton method. The symmetry that occurs in the Jacobian, if devices such as PRV's do not exist, will be advantageously used, and a one-dimensional array will store the banded portion of the Jacobian. In SOLDQEQS these tasks are accomplished within the outer `DO 90` loop. The index I in this loop tracks the NL loop equations, and the equation values are generated and stored in the array F. The process begins with F(I)=F(I)+FI* ... The columns of the Jacobian matrix are each related to a ΔQ, and these values are placed in the one-dimensional array IK1. The pipe numbers in each loop, which identify the terms that are needed to evaluate the equations and the Jacobian elements, are stored in the one-dimensional array IK. The array LP is a pointer that separates consecutive equations, e.g. loops, in array IK.

In a banded matrix all elements which are displaced more than the band width from the diagonal are zero. If i is the row number and j is the column number, then the band width NBAND is the maximum difference between a nonzero element in any column and its row number, plus one, or

$$\text{NBAND} = |j - i|_{max} + 1 \tag{4.77}$$

In some literature this definition is the half band width since, if the matrix is not symmetric, as many elements must to be stored to the left of the diagonal as to its right. In any symmetric matrix $[\boldsymbol{A}]$ each element $A_{ij} = A_{ji}$. If a two-dimensional array is used in a computer program to store the elements of a banded matrix, the first subscript (for rows) must be at least as large as the number of equations to be solved, and the second subscript must be as least as large as 2NBAND - 1. A special algorithm that properly accounts for the matrix properties is needed to solve a banded matrix problem. If the banded matrix is symmetric, it is not necessary to store all of the elements above and below the diagonal if the solution algorithm accounts for this symmetry. Either the elements above and on the diagonal, or those below and on the diagonal, are all that must be stored.

Program SOLDQEQS uses a one-dimensional array to store the banded elements of the Jacobian and calls a linear algebra subroutine SYMMAT to return the solution to the linear equation system in the array F. Before calling SYMMAT, the upper triangular portion of a banded symmetric Jacobian matrix is stored in a one-dimensional array DJ. In the declaration portion of SOLDQEQS we will find that DJ is a one-dimensional allocatable array with DJ[ALLOCATABLE](:) and that the number of real positions to store values in DJ is allocated with ALLOCATE(DJ(NL*NBAND-MM)), in which NBAND is the band width and MM = NBAND - 1. Thus a preliminary task is to determine the band width before allocating DJ and storing the nonzero derivative values in it. The listing in Fig. 4.30 determines the required band width.

```
C      FINDS BAND WIDTH
       MM=0
       DO 56 I=1,NL
       DO 56 J=LP(I)+1,LP(I+1)
       IJ=IABS(IK(J))
       DO 56 JJ=LP1(IJ)+1,LP1(IJ+1)
       II=IABS(IK1(JJ))-I
       IF(II.GT.MM) MM=II
    56 CONTINUE
       NBAND=MM+1
```

Figure 4.30 Band width algorithm.

The first position in array DJ is the diagonal element in the first row. The diagonal element of the second row is in position (2 - 1)NBAND + 1, the position of the diagonal element in the third row is (3 - 1)NBAND + 1, and in general the diagonal element in the ith row is in position id = (i - 1)NBAND + 1. An alternative formula for locating the diagonal position is id = i NBAND - MM in which MM = NBAND - 1, the number of elements beyond the diagonal. Thus we see that the storage that is needed by DJ is NL*NBAND - MM (NL is the number of equations), as given in the ALLOCATE statement. The position of off-diagonal elements in this one-dimensional array will be the diagonal position id plus the difference between the column number and the row number for the element. In any equation this position is iu = id + (j - i) = id - i + j. Thus in SOLDQEQS the statement after `DO 90 I=1,NL` that is used to define the NL equations is ID=NBAND*I-MM, which locates the position of the diagonal element for each row, and the statement NI = ID - I is an integer which locates the nonzero off-diagonal positions in DJ when the column position is added. Thus the statements that store the values in the proper locations of DJ are

```
      DJ(NI+JJ1)=DJ(NI+JJ1)+FI*FLOAT(IK1(JJ)/JJ1)*DD1
   87 CONTINUE
   90 DJ(ID)=DJ(ID)+DD1
```

The portion of the code, within the `DO 90` loop in program SOLDQEQS, that generates the system of equations and the values for the Jacobian and then obtains the solution that is used as the Newton correction, consists of the lines listed in Fig. 4.31:

```
      DO 90 I=1,NL
      IB=NBAND*I-MM
      NI=IB-I
      II=LP(I)+1
      II1=LP(I+1)
      DO 90 J=II,II1
      IJ=IABS(IK(J))
      IF(I.GE.NRES.OR.J.GT.II) GO TO 83
      IJ1=IABS(IK(II1))
      DO 80 JJ=1,NRES
      IF(IRES(JJ).EQ.IJ) F(I)=F(I)-ELE(JJ)
      IF(IRES(JJ).EQ.IJ1) F(I)=F(I)+ELE(JJ)
   80 CONTINUE
   83 FI=IK(J)/IJ
      QQ=Q(IJ)
      DO 85 JJ=LP1(IJ)+1,LP1(IJ+1)
      JJ1=IABS(IK1(JJ))
   85 QQ=QQ+FLOAT(IK1(JJ)/JJ1)*DQ(JJ1)
      DD=K(IJ)*ABS(QQ)**N(IJ)
      DD1=DD*(N(IJ)+1.)
      IF(L1(IJ).LT.0) THEN
      JJ=1
      DO 86 WHILE (IPUMP(JJ).NE.IJ)
   86 JJ=JJ+1
      DD1=DD1-2.*AP(JJ)*QQ-BP(JJ)
      F(I)=F(I)+FI*(DD*QQ-(AP(JJ)*QQ+BP(JJ))*QQ-CP(JJ))
      ELSE
      F(I)=F(I)+FI*DD*QQ
      ENDIF
      DO 87 JJ=LP1(IJ)+1,LP1(IJ+1)
      JJ1=IABS(IK1(JJ))
      IF(JJ1.LE.I) GO TO 87
      DJ(NI+JJ1)=DJ(NI+JJ1)+FI*FLOAT(IK1(JJ)/JJ1)*DD1
   87 CONTINUE
   90 DJ(IB)=DJ(IB)+DD1
C     SOLVES LINEAR EQUATIONS
      CALL SYMMAT(NL,NBAND,DJ,F)
```

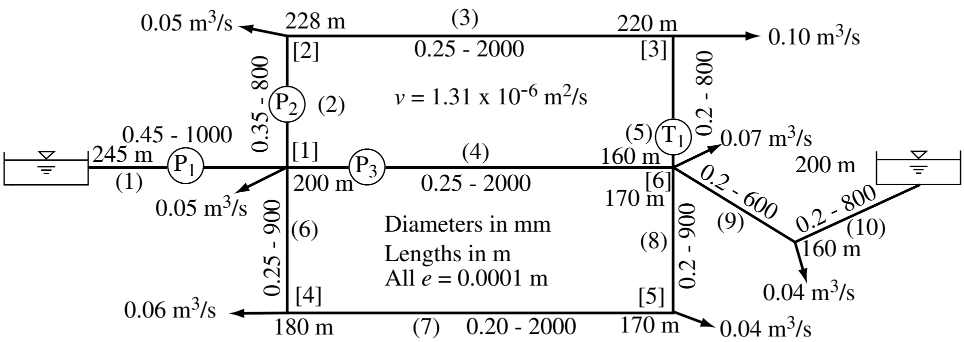
Figure 4.31 The solution algorithm.

To enhance solution efficiency we might try to arrange the equations to reduce the band width as much as possible. Not only will a smaller band width reduce the required amount of computer memory for a solution, but it also reduces the computational effort in solving the linear equation system. As the loop data are developed, the user can reduce the band width of the Jacobian matrix by trying to arrange the ΔQ numbering to be as close as possible to the equation numbering . The band width will equal the maximum difference in any equation between the equation number and the ΔQ number, plus 1.

However, placing this burden on the user is not desirable, especially since a banding algorithm can readily be implemented in computer code that will probably achieve a tighter banding than the user could arrange, even after some attention is given to the order in which equations should be listed and loops formed. One approach to minimizing the band width is described by Jeppson and Davis (1976). This approach is implemented in SOLDQBAN.FOR, which is on the CD. Also on the CD is SOLDQEQ1 that does not use the band width of the Jacobian but instead uses the standard linear algebra solver SOLVEQ, as do SOLQEQS and SOLHEQS, as it solves the ΔQ-equations.

Example Problem 4.15

In the sketch is a network with 10 pipes and 6 nodes which contains three pumps and one turbine. Prepare input data files for SOLQEQS, SOLHEQS and SOLDQEQS so these programs can be used to analyze this network. Use the pairs of (Q, h_p) data in the table to define the pump curves. Then replace the pump curve for pump 1 with the new pump data listed later in the solution, and resolve the problem with all three programs.



Pump 1		Pump 2		Pump 3		Turbine	
Q m³/s	**H** m	**Q** m³/s	**H** m	**Q** m³/s	**H** m	**Q** m³/s	**H** m
0.40	20.0	0.12	16.0	0.06	8.0	0.09	- 8.0
0.42	18.0	0.15	15.0	0.08	7.5	0.10	- 7.5
0.44	15.0	0.18	13.6	0.10	6.8	0.11	- 6.8

Since SI units are used, options must be changed from the default values. The input data file for each of these three programs are listed next, using two columns for each set:

Input Data For SOLQEQS

```
10 7 2 4 2 'U' 1 'D' 2
1 0 1 1000 0.45 0.0001
2 1 2  800 0.35 0.0001
3 2 3 2000 0.25 0.0001
4 1 6 2000 0.25 0.0001
5 3 6  800 0.20 0.0001
6 1 4  900 0.25 0.0001
7 4 5 2000 0.20 0.0001
8 5 6  900 0.20 0.0001
9 6 7  600 0.20 0.0001
10 7 0 800 0.20 0.0001
1 0.05 200 4 1 -2 -4 -6
2 0.05 228 2 2 -3
3 0.10 220 2 3 -5
4 0.06 180 2 6 -7
5 0.04 170 2 7 -8
6 0.07 160 4 4 5 8 -9
7 0.04 160 2 9 -10
1 245
10 200
1 0.40 20 0.42 18 0.44 15
2 0.12 16 0.15 15 0.18 13.6
4 0.06 8 0.08 7.5 0.1 6.8
5 0.09 -8 0.10 -7.5 0.11 -6.8
4 1 4 9 10
4 2 3 5 -4
4 4 -8 -7 -6
```

Input Data For SOLHEQS

```
10 7 2 4
&OPTIONS IU=1,IQ=3,ID=2/
1 0 1 1000 0.45 0.0001
2 1 2  800 0.35 0.0001
3 2 3 2000 0.25 0.0001
4 1 6 2000 0.25 0.0001
5 3 6  800 0.20 0.0001
6 1 4  900 0.25 0.0001
7 4 5 2000 0.20 0.0001
8 5 6  900 0.20 0.0001
9 6 7  600 0.20 0.0001
10 7 0 800 0.20 0.0001
1 0.05 200 4 1 -2 -4 -6
2 0.05 228 2 2 -3
3 0.10 220 2 3 -5
4 0.06 180 2 6 -7
5 0.04 170 2 7 -8
6 0.07 160 4 4 5 8 -9
7 0.04 160 2 9 -10
1 245
10 200
1 0.40 20 0.42 18 0.44 15
2 0.12 16 0.15 15 0.18 13.6
4 0.06  8 0.08 7.5 0.10 6.8
5 0.09 -8 0.10 -7.5 0.11 -6.8
```

Input Data For SOLDQEQS

```
10 7 2 4
&OPTIONS IU=1,IQ=3,ID=2/
1 0 1 1000 0.45 0.0001  0.44
2 1 2 0800 0.35 0.0001  0.20
3 2 3 2000 0.25 0.0001  0.15
4 1 6 2000 0.25 0.0001  0.12
5 3 6 0800 0.20 0.0001  0.05
6 1 4 0900 0.25 0.0001  0.07
7 4 5 2000 0.20 0.0001  0.01
8 5 6 0900 0.20 0.0001 -0.03
9 6 7 0600 0.20 0.0001  0.07
10 7 0 800 0.20 0.0001  0.03
1 0.05 200
2 0.05 228
3 0.10 220
4 0.06 180
5 0.04 170
6 0.07 160
7 0.04 160
1 245
10 200
1 0.4 20 0.42 18 0.44 15
2 0.12 16 0.15 15 0.18 13.6
4 0.06 8.0 0.08 7.5 0.10 6.8
5 0.09 -8.0 0.10 -7.5 0.11 -6.8
4 1 4 9 10
4 2 3 5 -4
4 4 -8 -7 -6
```

The solution tables from SOLQEQS and SOLDQEQS are identical, as shown below. SOLHEQS failed to converge. The failure was caused by the relative inaccuracy of the initial values of the heads that were provided to the Newton method by the automated estimator in the code; the values were too crude in relation to the sensitivity of the code to the way that the three pairs of points for pump 1 define its operating characteristics. If

PIPE DATA

PIPE NO.	NODES FROM	TO	L m	DIA. m	e x10^4 m	Q m^3/s	VEL. m/s	HEAD LOSS m	HLOSS/ 1000
1	0	1	1000	0.450	1.0	0.436	2.74	12.61	12.61
2	1	2	800	0.350	1.0	0.163	1.70	5.39	6.73
3	2	3	2000	0.250	1.0	0.113	2.31	36.76	18.38
4	1	6	2000	0.250	1.0	0.118	2.40	39.64	19.82
5	3	6	800	0.200	1.0	0.013	0.42	0.74	0.92
6	1	4	900	0.250	1.0	0.105	2.14	14.29	15.88
7	4	5	2000	0.200	1.0	0.045	1.43	19.20	9.60
8	5	6	900	0.200	1.0	0.005	0.16	0.14	0.15
9	6	7	600	0.200	1.0	0.066	2.10	11.92	19.87
10	7	0	800	0.200	1.0	0.026	0.82	2.56	3.20

Pump 1 in pipe 1: Head = 15.71 m, Q = 0.436 m^3/s
Pump 2 in pipe 2: Head = 14.44 m, Q = 0.163 m^3/s
Pump 3 in pipe 4: Head = 6.02 m, Q = 0.118 m^3/s
Pump 4 in pipe 5: Head = - 5.17 m, Q = 0.013 m^3/s

NODE DATA

NODE	DEMAND Estimate	DEMAND m^3/s	ELEV. m	HEAD m	PRESSURE kPa	HGL ELEV. M
1	0.1	0.050	200.0	48.10	471.7	248.10
2	0.1	0.050	228.0	29.15	285.8	257.15
3	0.1	0.100	220.0	0.39	3.8	220.39
4	0.1	0.060	180.0	53.81	527.7	233.81
5	0.0	0.040	170.0	44.61	437.5	214.61
6	0.1	0.070	160.0	48.46	475.2	208.46
7	0.0	0.040	160.0	42.56	417.3	202.56

this pump curve is plotted, we see immediately how the curve turns steeply downward outside each end of the given data. Using pump curves of this nature should be avoided. To obtain a solution from SOLHEQS, either the points that define the pump curve must be adjusted, or the code must be modified so the user can supply the initial estimates of the heads for the Newton method. If the pump curve for pump 1 is modified so the three discharge-head data pairs are (0.40, 16.0), (0.43, 15.8), and (0.46, 15.5), then SOLHEQS can solve the modified problem.

* * *

4.5 CONCLUDING REMARKS

This chapter concentrated on the analysis of pipeline networks. The first area of emphasis was on the development of the three kinds of systems of equations to describe mathematically the flow in a pipe network, first for simpler networks and then for networks which contain pumps or turbines and loss-producing devices such as a pressure reducing valve or a back pressure valve. The Newton method for the solution of these equation systems was introduced and later included in computer programs to solve the equation systems. Later sections of the chapter developed solution routines and implemented them for each of the three types of equation systems.

There are features that a production network program would usually include that these programs do not have. For example, rather than requiring the user to provide a set of estimated initial discharges Q_{oi} that satisfy all of the junction continuity equations, the program should develop these values. One way to create these values is to reduce the network to a branched system by deleting some pipes with smaller diameters and then using the methods in this chapter to obtain a solution for the branched network.

Another burden that would not be placed on the user of a production program is the need to supply the pipe numbers that define the loops around which the energy equations are written and the corrective loop discharges circulate. An algorithm for this task must satisfy two criteria: (a) the pipes that define any loop should be minimum in number, and also (b) these loops must lead to the creation of energy equations that are independent so that none of the equations are a combination of any group of the other equations. The first criterion can be achieved by using a "minimum path algorithm," and the second criterion requires each new loop to contain at least one pipe that does not exist in any of the previous loops.

Production programs will also take full advantage of the sparsity of the Jacobian matrix in computing network solutions in an efficient manner.

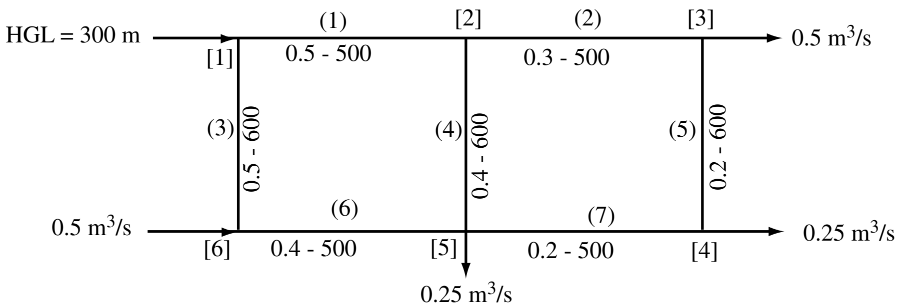
Network solvers can also allow the user to obtain time-dependent solutions. Such solutions, which do not account for the forces that are required to accelerate or decelerate the fluid columns, have become known as "extended time simulations." To develop an extended time simulation, additional information of several kinds is needed, such as demand functions which describe how one or several external nodal demands QJ vary with time, rules based on pressures at nodes or on water surface elevations in tanks or reservoirs can determine how many pumps should operate in series or parallel, and storage-elevation-capacity curves can be used to describe the behavior of tanks, and so on.

The use of programs for network analysis can also allow designers to obtain answers for the many questions that naturally occur during the design process. For example, what head and capacity should a pump produce to maintain a prescribed pressure and/or discharge at the far end of the network? What is the discharge from a junction if the pressure is known from a measurement there? How much head should a PRV dissipate so the pressure does not exceed a set value? How much flow can be obtained from a fire hydrant if its discharge characteristics are known? What are the flows from sprinkler heads if their sizes are known? How does a contaminant spread through a pipe network if it is accidentally introduced at a point?

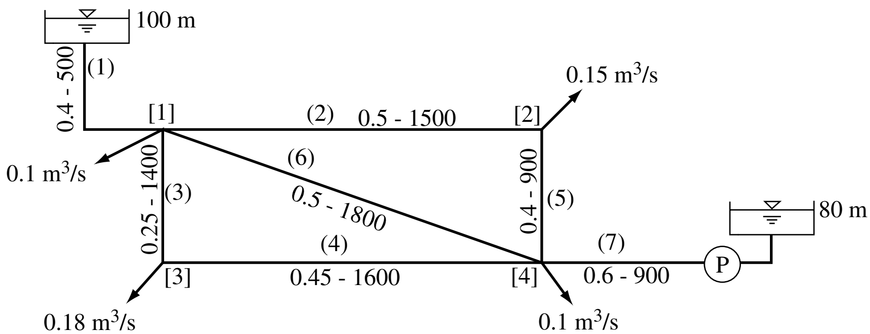
Chapter 5 will explore the design of these pipe networks, and Chapter 6 will examine further several topics, including extended-time simulations.

4.6 PROBLEMS

4.1 For the two pipe networks shown below, write the system of Q-equations. In writing these equations, use K and n with subscripts that correspond to the pipe number.



Diameters in mm
Lengths in m
All e's for both networks = 0.00002 m

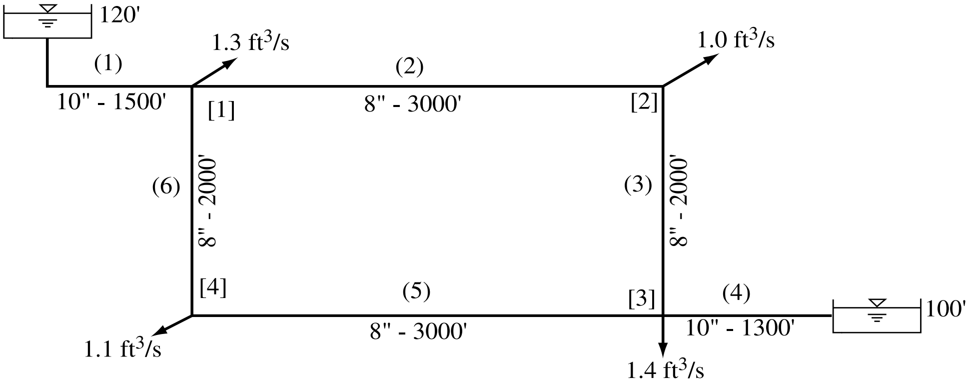


Pump

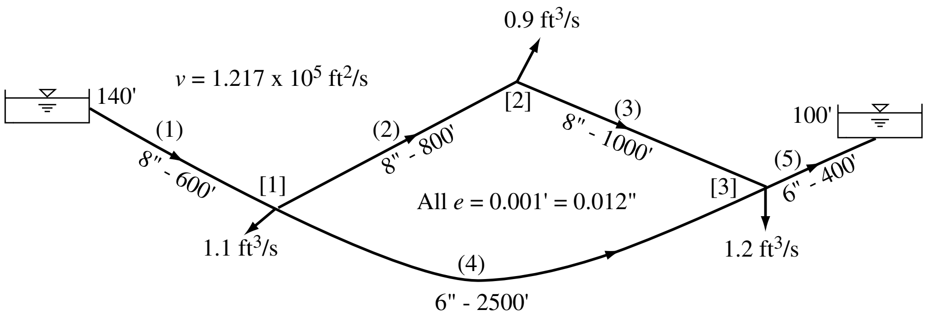
Q m³/s	h_p m
0.2	30
0.4	27
0.7	21

4.2 Write the system of Q-equations for the network shown. It is not necessary to substitute the values of K and n from the table into the equations; instead use K_i and n_i where i is the pipe number. If the discharge in pipe 1 is $Q_1 = 3.1$ ft^3/s, then what is the friction factor f for this pipe?

Pipe	K	n
1	1.841	1.928
2	11.47	1.871
3	7.47	1.839
4	1.615	1.914
5	11.08	1.828
6	7.69	1.884

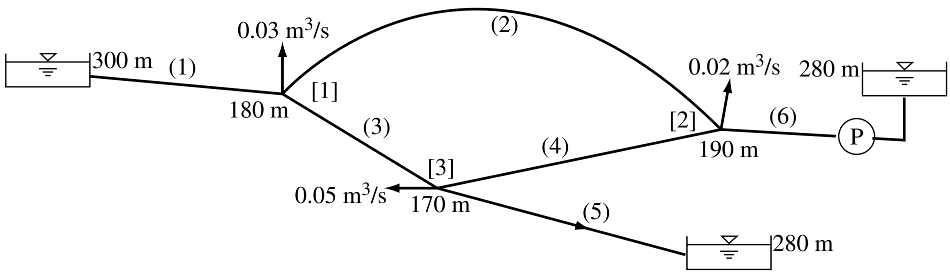


4.3 A 5-pipe, 3-node network appears below. On this diagram the first number along each line is the pipe diameter in inches, and the second number is the pipe length in feet. All pipes have an equivalent sand roughness $e = 0.001$ ft $= 0.012$ inches. Do the following: (a) compute the values of K and n in $h_f = KQ^n$ for pipe 1, based on the Darcy-Weisbach equation, and (b) write the system of Q-equations for this network. (Use subscripts on K, n and Q corresponding to the pipe number.)



Pipe	K	n
1		
2	3.53	1.961
3	4.44	1.929
4	48.6	1.934
5	6.40	1.817

4.4 In the sketch the network consists of 6 pipes and 3 nodes. A source pump and one reservoir supply the network, and the lower reservoir receives water. Do the following tasks: (a) write the system of Q-equations; (b) write the system of H-equations; (c) write the system of ΔQ-equations; (d) if the discharge in pipe 5 is $Q_5 = 0.026$ m^3/s into the reservoir, what is the elevation of the HGL at node 3; (e) if the discharge in pipe 6 is $Q_6 = 0.112$ m^3/s, what are the HGL and pressure at node 2?

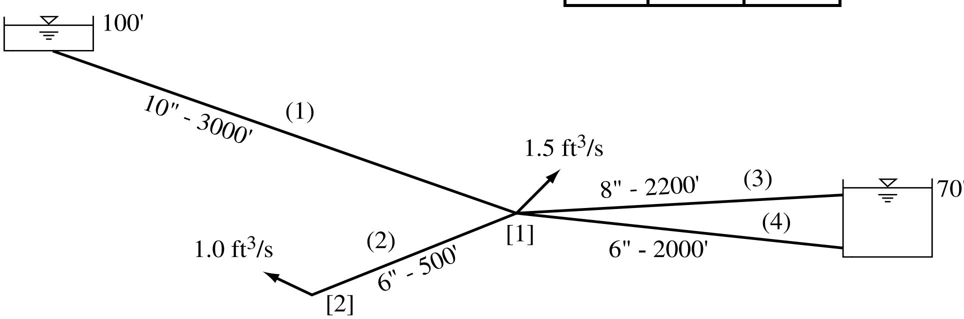


Pipe	Dia. m	Length m	K	n
1	0.30	1000	543	1.886
2	0.20	2500	13700	1.946
3	0.20	1000	3270	1.839
4	0.30	1500	1077	1.965
5	0.15	1000	27400	1.974
6	0.35	800	260	1.968

Pt.	Q m^3/s	h_p m
1	0.05	35
2	0.10	31
3	0.15	24

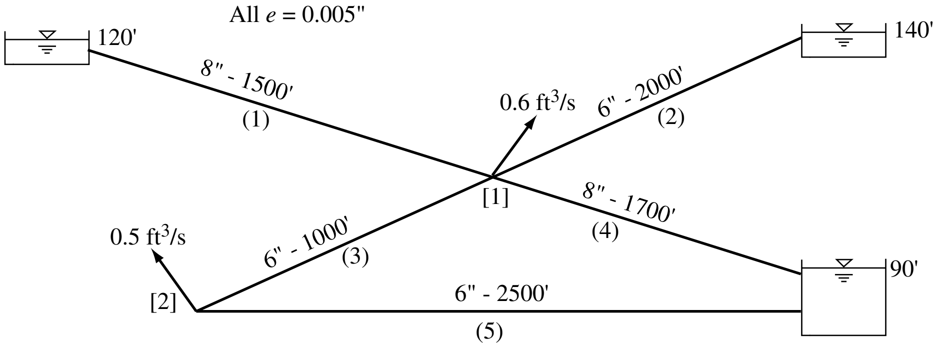
4.5 For the network below: (a) write the Q-equations; (b) write the H-equations; and (c) write the ΔQ-equations. (Use the symbols K and n with correct subscripts for the pipes in the equations.)

Pipe	K	n
1	3.59	1.922
2	7.97	1.917
3	7.94	1.821
4	28.80	1.809



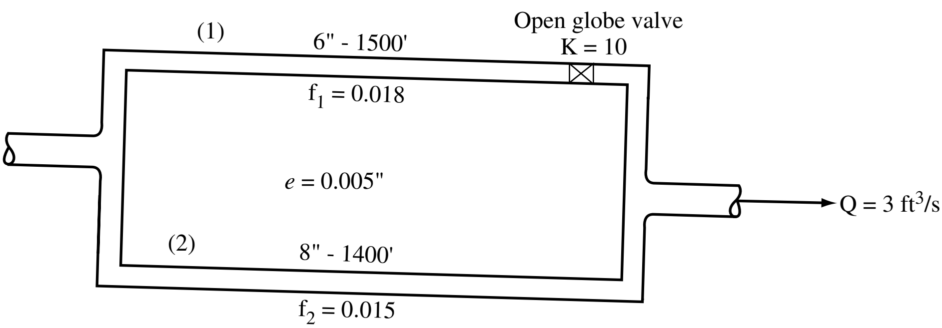
4.6 For the network shown, write (a) the Q-equations, (b) the H-equations, and (c) the ΔQ-equations. The K and n values for the pipes in this network are given in the table which follows. (Your equations should contain only numerical values and unknowns.)

Pipe	D in.	L ft.	K	n
1	8	1500	5.72	1.930
2	6	2000	33.00	1.931
3	6	1000	16.30	1.889
4	8	1700	6.53	1.913
5	6	2500	40.70	1.890

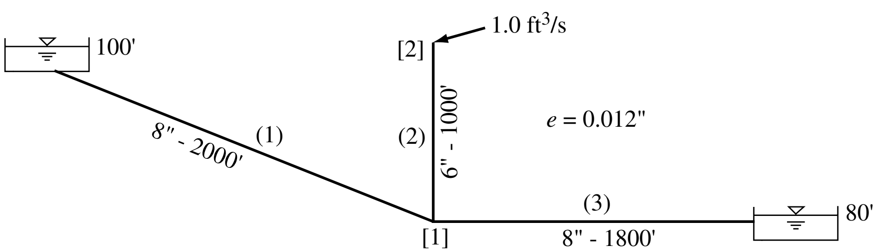


4.7 Prepare the input data for, and obtain the solution from, NETWK for the network described in Problem 4.6.

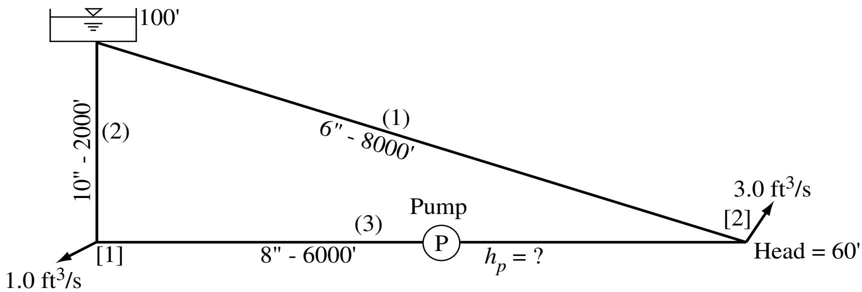
4.8 A pipe branches into a 6-in diameter, 1500-ft long pipe and a 8-in diameter, 1400-ft long pipe and then rejoins, so the two pipes are in parallel. Pipe 1 contains an open globe valve with a local loss coefficient $K = 10$. If the total discharge is $Q = 3$ ft^3/s, determine the discharges Q_1 and Q_2 in the individual pipes. For simplicity, we shall assume $f_1 = 0.018$ and $f_2 = 0.015$.



4.9 This sketch of a small water system shows two reservoirs, with a pipe connected to the center node with an inflow of 1.0 ft^3/s at the other end. Set up the three equation systems that could be used to solve this problem, and then obtain a solution by using one of them.

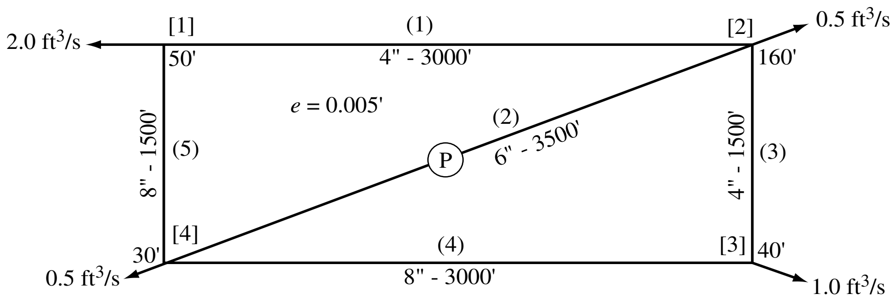


4.10 In the diagram three pipes that form a triangle are supplied by a reservoir at one vertex of the triangle, and demands of 1.0 and 3.0 ft^3/s are found at the other two vertices. A booster pump exists in pipe 3. What head should the pump in pipe 3 produce so the pressure at node 2 causes an HGL of 60 ft there. The ground elevation is everywhere 0 ft. You may solve this problem by using any, or all, of the equation systems that are available. Assume $e = 0.012$ in and $\nu = 1.41\text{x}10^{-5}$ ft^2/s.



4.11 In the network of Problem 4.10 the diameters of pipes 2 and 3 have been enlarged to 12 inches. In place of the pump a pressure reduction valve is now needed in pipe 3 to create a pressure head at node 2 of 60 ft. Determine the required setting, i.e. HGL, of the pressure reduction valve. You can use any of the equation systems.

4.12 A network is shown in the diagram. Write the system of ΔQ-equations for this network and complete one Newton iteration toward a solution of the problem. Verify this result by using an application software package such as MathCAD or TK-Solver.



Pump Characteristics

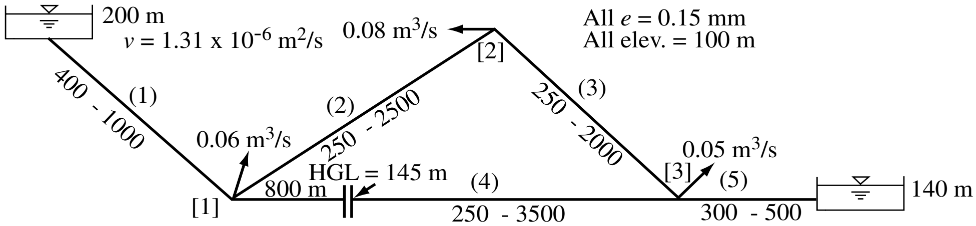
Q ft^3/s	h_p ft
0.3	40
0.5	35
0.8	28

Pipe	K	n
1	375.0	1.860
2	57.4	1.902
3	190.0	1.898
4	11.5	1.880
5	5.72	1.930

4.13 Write the system of H-equations for the two networks in Problem 4.1.

4.14 Write the system of ΔQ-equations for the two networks in Problem 4.1.

4.15 The following network contains a pressure reducing valve (PRV) that is set so it will produce a HGL of 145 m on its downstream side. This valve is 800 m downstream from node 1. Do the following: (a) write the system of Q-equations; (b) write the system of H-equations; (c) write the system of ΔQ-equations; (d) using the Newton method, solve the system of ΔQ-equations; (e) and what is the HGL on the upstream side of the PRV?



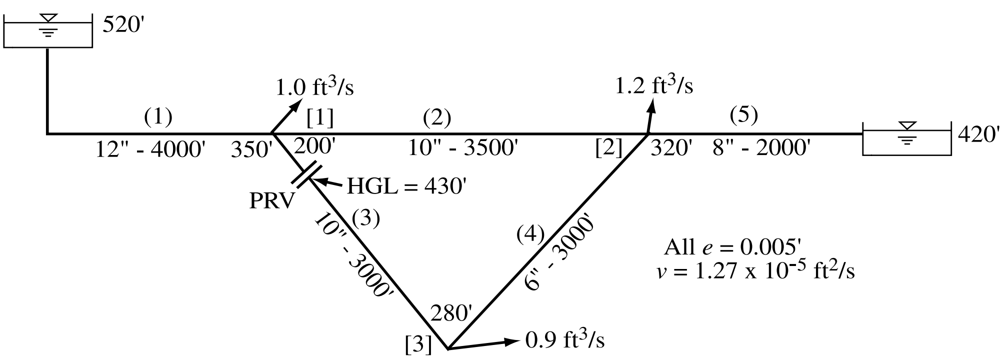
Pipe	1	2	3	4	5
K	196	3520	2380	4130	192
n	1.819	1.955	1.895	1.892	1.834

4.16 The reservoir water surface elevation at the beginning of pipe 1 in Problem 4.15 is lowered by 50 m so it is WS_1 = 150 m, and a pump with the characteristics given below is installed in pipe 1. Write the three equation systems and solve one of them, also finding the HGL elevation on the upstream side of the PRV.

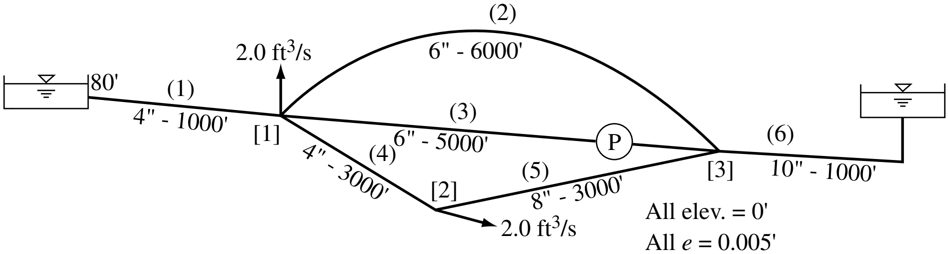
Q m³/s	h_p m
0.18	55
0.22	51
0.26	44

4.17 For the network shown below write (a) the Q-equations, (b) the H-equations, and (c) the ΔQ-equations. Pipe 3 contains a pressure reduction value 200 ft downstream from node 1 that is set to maintain an HGL = 430 ft on its discharge side. Use the notation K_i and n_i in these equations.

Pipe	K	n
1	1.93	1.935
2	4.44	1.940
3	3.50	1.840
4	47.90	1.866
5	7.67	1.917



4.18 For the network below: (a) write the Q-equations; (b) write the H-equations; (c) write the ΔQ-equations; and (d) solve the system of ΔQ-equations. The water surface elevation of the right reservoir is 300 ft, and the following three (Q, h_p) pairs define the pump characteristic curve: (1.0, 26), (1.5, 24), (2.2, 20).



4.19 Solution tables from NETWK follow, with four values omitted. Fill in the missing values. What head drop occurs across the PRV? What horsepower does this loss represent?

PIPE DATA

PIPE NO.	NODES FROM	TO	L ft.	DIA. in	e x10^3 in	Q ft^3/s	VEL. ft/s	HEAD LOSS ft.	HLOSS/1000
1	0	1	4000	12.0	5.0	4.72	6.01		9.69
2	1	2	3500	10.0	5.0	3.26	5.97	41.87	11.96
3	1	3	3000	10.0	5.0	0.46	0.85	0.91	0.30
4	2	3	3000	6.0	5.0	0.44	2.22	10.21	3.40
5	2	0	2000	8.0	5.0	1.62	4.64	19.36	9.68

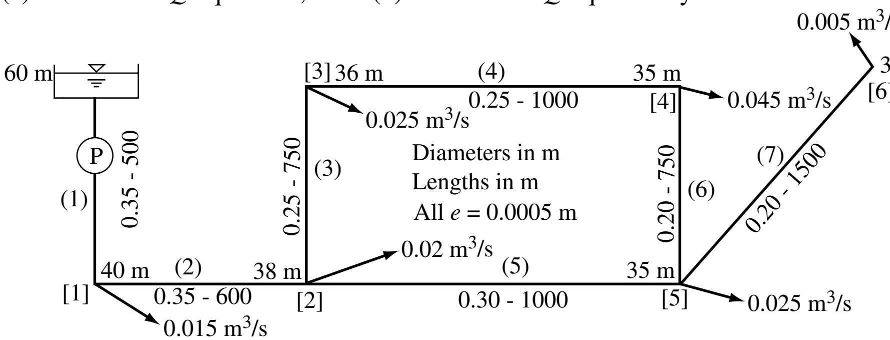
AVE. VEL. = 3.94 ft/s, AVE. HL/1000 = 7.01, MAX. VEL. = 6.01 ft/s, MIN. VEL. = 0.85 ft/s

NODE DATA

NODE	DEMAND ft^3/s	DEMAND gal/min	ELEV. ft.	HEAD ft.	PRESSURE lb/in^2	HGL ELEV. ft.
1	1.0	449	350	131.2	56.87	481.2
2	1.2	539	320	119.4	51.72	439.4
3	0.9	404	280			

AVE. HEAD = 133.3 ft, AVE. HGL = 450.0 ft
MAX. HEAD = 149.2 ft, MIN. HEAD = 119.4 ft

4.20 For the network shown: (a) write the Q-equations; (b) write the H-equations; (c) write the ΔQ-equations; and (d) solve the ΔQ-equation system.



Q m^3/s	h_p m
0.120	40
0.140	38
0.165	35

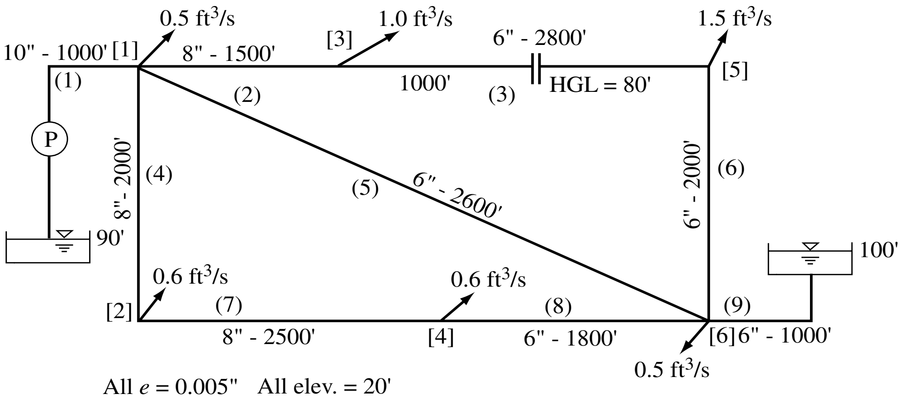
4.21 For the two networks in Problem 4.1, solve the Q-equation system using the Newton method.

4.22 For the two networks in Problem 4.1, solve the H-equation system using the Newton method.

4.23 For the two networks in Problem 4.1, solve the ΔQ-equation system using the Newton method.

4.24 Determine the pressures in lb/in^2 at the six nodes of Problem 4.1a.

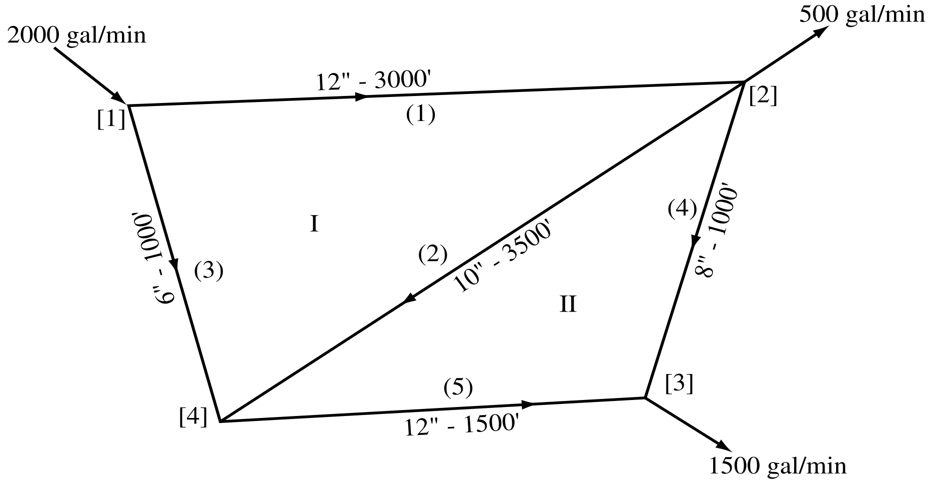
4.25 For the network below, write the ΔQ-equation system and solve them, and verify your solution by obtaining a computer solution by using NETWK.



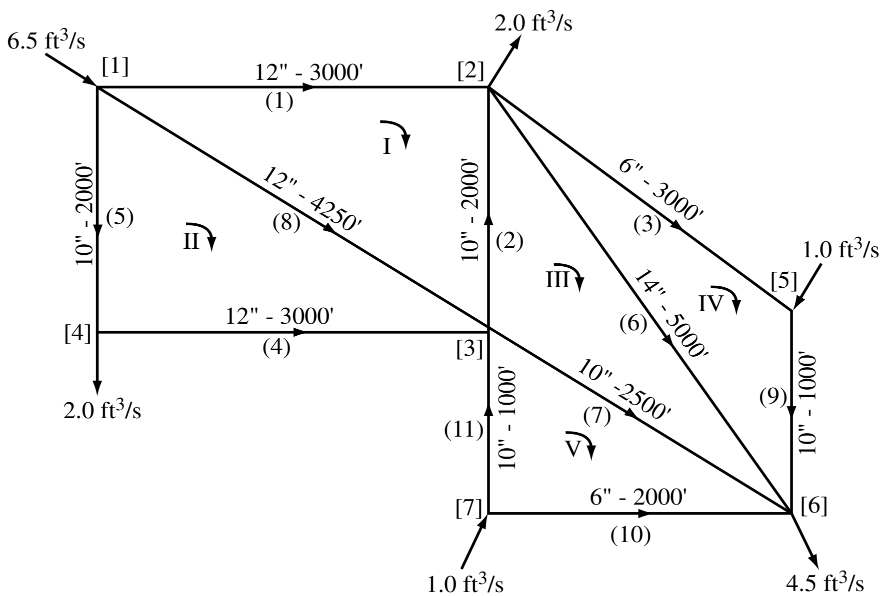
Q ft^3/s	h_p ft
1.0	50
2.0	48
3.0	45

4.26 Determine the discharge and head loss in each pipe of the networks shown on the following pages by first determining a set of values for the coefficients K and n; then setting up and solving the equations without using a computer, except perhaps to solve each linear system of equations that is formed as a part of the Newton method.

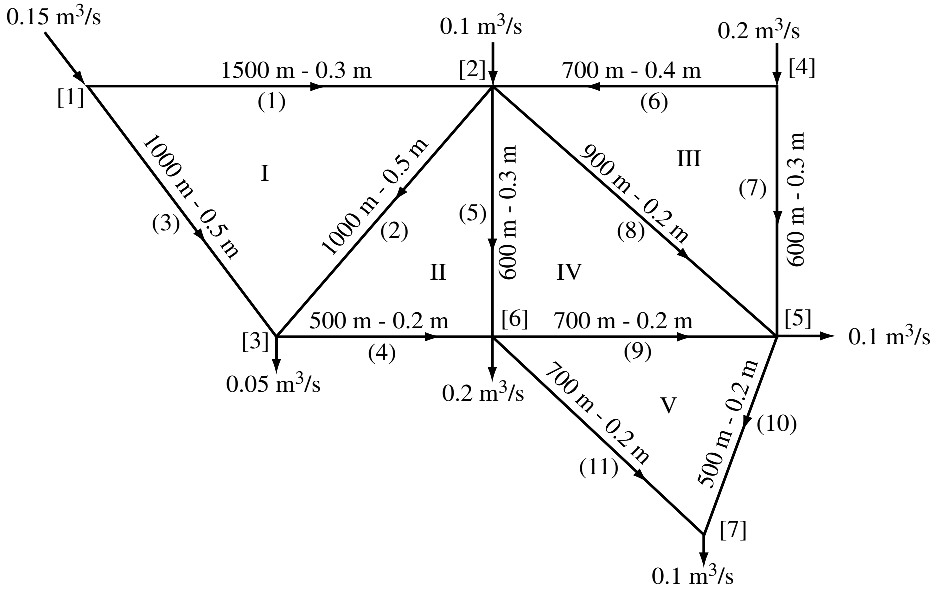
(a) Analyze this network with the Hazen-Williams equation and $C_{HW} = 120$ for all pipes.



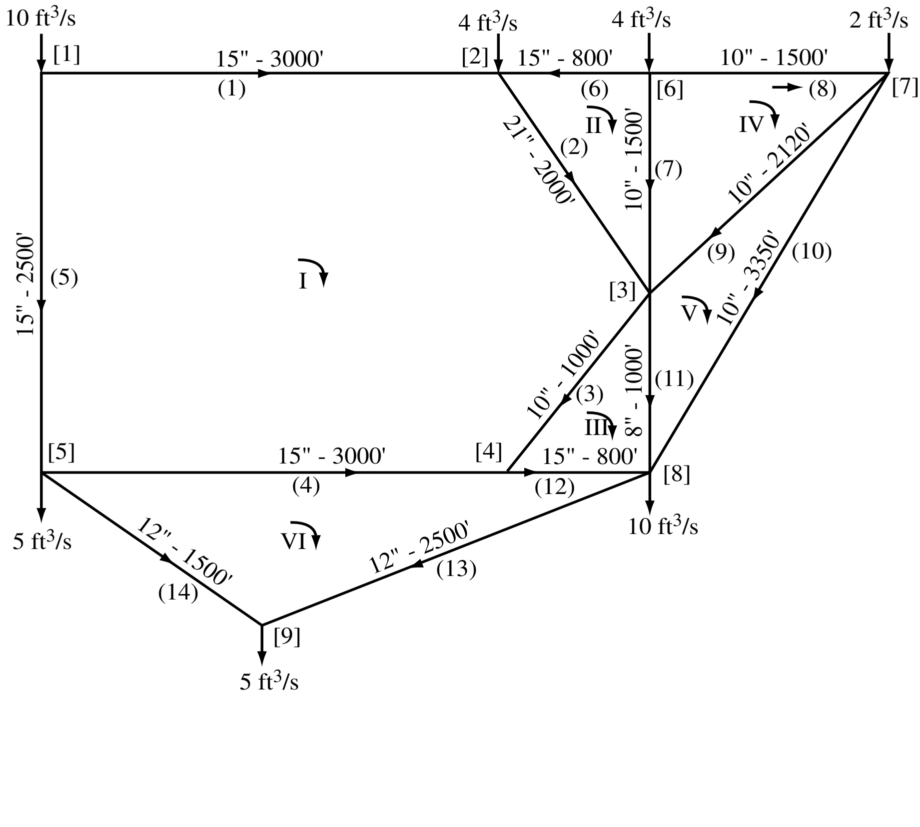
(b) Analyze this network by using the Hazen-Williams equation; for pipes 1 through 5 use $C_{HW} = 120$, and for pipes 6 through 11 use $C_{HW} = 100$.



(c) Use the Hazen-Williams equation to analyze this network; all pipes are cast iron.



(d) Analyze this network with the Darcy-Weisbach equation, assuming $e = 0.001$ ft for the 8-in and 10-in pipes and $e = 0.0005$ ft for all other pipes.

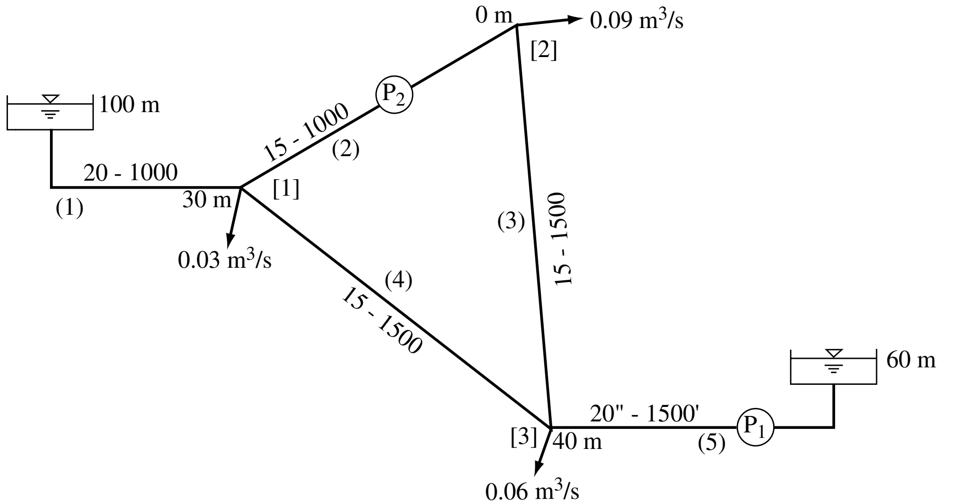


(e) Analyze the network of part (a) by using the Darcy-Weisbach equation with a roughness of $e = 0.0005$ ft for all pipes.

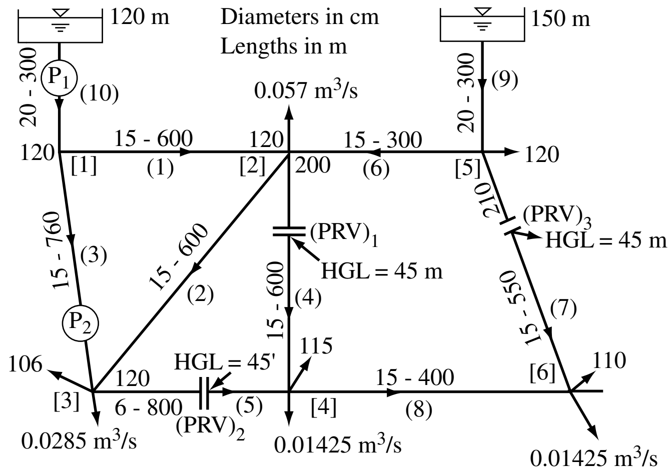
(f) Analyze the network of part (b) by using the Darcy-Weisbach equation; for pipes 1 through 5 use $e = 0.005$ ft, and for pipes 6 through 11 use $e = 0.006$ ft.

(g) Use the Hazen-Williams equation to analyze this network; all pipes are made of cast iron. The diameters are given in centimeters; lengths and elevations are in meters. Pump performance data are listed in the table.

Pump 1		Pump 2	
Q m^3/s	h_p m	Q m^3/s	h_p m
0.10	50	0.05	10
0.15	48	0.10	8
0.20	44	0.15	5



(h) Use the Hazen-Williams equation to analyze this network that is diagrammed atop the next page; all pipes are made of cast iron. The diameters are given in centimeters; lengths and elevations are in meters. Pump performance data are listed in the table.



Pump 1		Pump 2	
Q m^3/s	h_p m	Q m^3/s	h_p m
0.0425	36.6	0.0283	12.2
0.0708	30.5	0.0425	10.7
0.0991	22.9	0.0566	8.5

4.27 Only the Colebrook-White equation is used in subroutine COMPK_N that determines the values of K and n in the exponential formula for programs SOLQEQS, SOLQEQS and SOLDQEQS. Modify this subroutine so it will allow laminar flow in the pipe. Also modify the subroutine so the discharge can become zero; e.g., this might commonly occur for initial discharges Q_{oi} for use with the ΔQ-equations.

4.28 Assume that the flow in all pipes will always be turbulent; however, a user might select initial values for Q_{oi} that are zero when solving the ΔQ-equations. Then the subroutine COMPK_N would fail, as it is now written in SOLQEQS, SOLHEQS and SOLDQEQS. Modify COMPK_N so it can accept a value of zero for the discharge.

4.29 Modify SOLQEQS so an option allows the user to supply starting values for Q that will be used in the Newton method rather than generating these values internally.

4.30 Modify SOLHEQS so it has an option that allows the user to supply initial values for H that will be used in the Newton method rather than generating these values internally. The additional input could be supplied from another read statement, or these heads could be listed after the nodal elevations in the node data.

4.31 Modify SOLDQEQS so it will allow PRV's to exist in the network. This change will require two sets of loop data to be read as input data (unless you wish to obtain these loops internally), one around which the ΔQ's circulate, and one around which the energy equations are written. Since the two sets of loops will not be identical, the Jacobian will not be symmetric.

4.32 Modify SOLHEQS and/or SOLDQEQS so it calls a symmetric matrix solver such as SYMMAT.

4.33 SOLQEQS, SOLHEQS and SOLDQEQS can all analyze networks that contain local losses if the user will provide the actual length of the pipe and the additional length of pipe that would cause a frictional pipe loss that is equivalent to what the local loss device would cause. Modify one, or all, of these programs so each equivalent pipe length is computed internally within the program and then added to the actual length before the problem is solved.

4.34 Rather than compute an equivalent length of pipe for a local loss, as in Problem 4.33, modify SOLQEQS so that local losses, where they occur, are treated by adding a head loss term of the form $h_L = KQ^2/(2gA^2)$ to the energy loop equation.

4.35 Repeat Problem 4.34, but modify SOLDQEQS.

4.36 Use SOLQEQS, SOLHEQS, and SOLDQEQS to analyze the network depicted in Problem 4.20.

4.37 Use SOLQEQS to analyze the network in Problem 4.25. This network contains a PRV in pipe 3 that is located 1000 ft downstream from the beginning of this pipe.

4.38 SOLQEQS, SOLHEQS and SOLDQEQS all represent pump performance by fitting three (Q, h_p) pairs of pump characteristic curve data with a second-order polynomial. Modify one or all of these programs so they accept the normal capacity (discharge at best efficiency) and head at this discharge as input, and then the relation between h_p and Q is obtained from the power equation $P = \gamma Q h_p$ under the assumption that the power P remains constant.

4.39 Modify the program that was developed in Problem 4.38 so the efficiency of the pump is a linear function of the difference of the discharge from its normal capacity.

4.40 Place a PRV in pipe 2 of the network in Example Problem 4.5 with a pressure setting of HGL = 445 ft. Obtain a solution for this network using SOLQEQS. Verify this solution using NETWK.

4.41 SOLQEQS contains a code segment that cross checks the connectivity of the network by looking at the two node numbers at the ends of a pipe and at the pipe numbers that join at a junction. It also checks that upstream node numbers are negative and that downstream node numbers are positive. But the algorithm currently can not determine whether an extra pipe might be connected to a node. Modify the code so a check can identify any extra pipe(s) that might be specified in the data that lists the pipes that are connected to nodes.

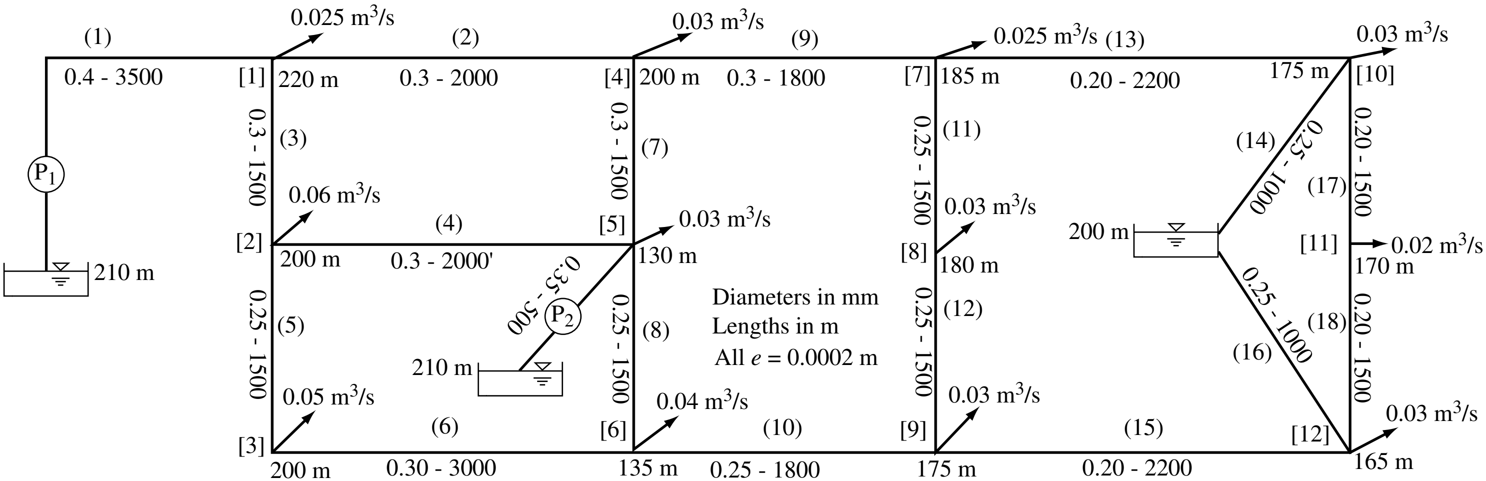
4.42 Modify SOLDQEQS so PRV's can exist in the network. Now two separate kinds of loops will exist, those around which the corrective loop discharges circulate and those around which the energy equations are written. Therefore two sets of loop data must be included in the input data file.

4.43 Use the resulting computer program from Problem 4.42 to obtain a solution to the network in Example Problem 4.5 with a PRV in pipe 2 having a pressure setting that causes the downstream head to be HGL = 445 ft. Verify this solution by (1) using NETWK and by (2) changing subroutine FUNCT in program EQUSOL1.

4.44 The network diagram below lists average demands on it. The storage tank that is connected to the network by pipes 14 and 16 has a 20-m diameter; at 6 a.m. its water surface elevation should be 200 m. The demands at all nodes change according to the peaking factors in the table. The pump characteristics represent two pumps in parallel at each location. Obtain a series of solutions for the times at which the peaking factors are given. For each solution of this series determine the new water level in the tank and the electrical energy consumed by each pump during the latest time increment. Suggest when one pump at each station should be shut off. What might be done to improve the design and thereby the operation of the system?

Time	6 a.m.	9 a.m.	12 Noon	4 p.m.	7 p.m.	10 p.m.	12 Mid.	3 a.m.
PF	1.0	1.8	1.3	1.3	1.7	1.5	0.6	0.2

Pump 1		**Pump 2**	
Q m^3/s	h_p m	Q m^3/s	h_p m
0.15	50	0.20	30
0.25	47	0.25	28
0.35	42	0.30	25



(1)
0.4 - 3500
P1
210 m
0.025 m3/s
(2)
0.3 - 2000
[1]
220 m
0.3 - 1500
(3)
0.06 m3/s
[2]
200 m
(4)
0.3 - 2000'
0.25 - 1500
(5)
0.05 m3/s
[3]
200 m
(6)
0.30 - 3000
0.35 - 500
P2
210 m
0.03 m3/s
[4]
200 m
0.3 - 1500
(7)
[5]
0.03 m3/s
130 m
0.25 - 1500
(8)
[6]
135 m
0.04 m3/s
(9)
0.3 - 1800
(10)
0.25 - 1800
Diameters in mm
Lengths in m
All e = 0.0002 m
0.025 m3/s
[7]
185 m
0.25 - 1500
(11)
0.03 m3/s
[8]
180 m
0.25 - 1500
(12)
0.03 m3/s
[9]
175 m
(13)
0.20 - 2200
(15)
0.20 - 2200
200 m
175 m
(14)
0.25 - 1000
(16)
0.25 - 1000
0.03 m3/s
[10]
0.20 - 1500
(17)
[11]
0.02 m3/s
170 m
0.20 - 1500
(18)
[12]
0.03 m3/s
165 m