CHAPTER 12
COMPUTER MODELS/EPANET

Lewis A. Rossman
U.S. Environmental Protection Agency,
National Risk Management Research Laboratory
Cincinnati, OH

12.1 INTRODUCTION

Pipe network flow analysis was among the first civil engineering applications programmed for solution on the early commercial mainframe computers in the 1960s. Since that time, advancements in analytical techniques and computing power have enabled us to solve systems with tens of thousands of pipes in seconds using desktop personal computers. This chapter discusses how modern-day computer models are used to analyze the hydraulic and water-quality behavior of distribution systems. It covers how computer models are applied to actual systems, what the internals of the models consist of, and the capabilities and operation of one particular model in the public domain, EPANET. The chapter focuses only on models that analyze successive periods of steady flow through a general arrangement of connected pipes, pumps, valves, and storage facilities. Other, more specialized computer models, such as programs for surge analysis, are not addressed.

12.1.1 Need for Computer Models

The classical pipe-network flow problem asks what the flows and pressures are in a network of pipes subject to a known set of inflows and outflows. Two sets of equations are needed to solve this problem. The first set requires conservation of flow to be satisfied at each pipe junction. The second specifies a nonlinear relation between flow and headloss in each pipe, such as the Hazen-Williams or Darcy-Weisbach equation. Whenever a network contains loops or more than one fixed-head source, these equations form a coupled set of nonlinear equations. Such equations can be solved only by using iterative methods, which for all but the smallest-sized problems require the aid of a computer. Because most distribution systems of interest are looped, computer models have become a necessity for analyzing their behavior.

Computer models also provide other advantages that enhance distribution system modeling. These include the following:
• Systematic organization, editing, and error checking of the input data required by the model
• Aid in viewing model output, such as color-coded maps, time-series plots, histograms, contour plots, and goal-specific queries
• Linkages to other software, such as databases, spreadsheets, computer aided design (CAD) programs, and geographic information systems (GIS)
• Ability to perform other kinds of network analyses, such as optimal pipe sizing, optimal pump scheduling, automated calibration, and water-quality modeling

12.1.2 Uses of Computer Models

Cesario (1991) discusses a number of different ways that network computer models are used in planning, engineering, operations, and management of water utilities. Some examples include the following.

1. Network models are run to analyze what capital improvements will be needed to serve additional customers and maintain existing services in future years. They also can help a utility prepare for planned outages of specific system components, such as reservoirs and pump stations.
2. Network models are used to locate and size specific network components, such as new mains, storage tanks, pumping stations, and regulator valves.
3. Pump scheduling, tank turnover analysis, energy optimization, and operator training are some ways in which network models can be used to improve system operations.
4. Extensions to hydraulic models allow them to analyze a host of questions related to water quality. They can determine how water from different sources blends together throughout a system, how operational changes can reduce the time that water spends in the system, and what steps can be taken to maintain adequate disinfectant residuals without excessive levels of disinfection by-product formation throughout the system.
5. Fire-flow studies are used to determine if adequate flow and pressure are available for fire-fighting purposes, as required for fire insurance ratings.
6. Vulnerability studies are used to test a system's susceptibility to unforeseen occurrences, such as loss of power, major main breaks, extended drought periods, and intrusion of waterborne contamination.

12.1.3 History of Computer Models

The groundwork for computer modeling of distribution systems was laid by the numerical method developed by Hardy Cross in the 1930s for analyzing looped pipe networks (Cross, 1936). The first mainframe programs for pipe-network analysis that appeared in the 1960s were based on this method (Adams, 1961), but these were soon replaced with codes that used the more powerful Newton-Raphson method for solving the nonlinear equations of pipe flow (Dillingham, 1967; Martin and Peters, 1963; Shamir and Howard, 1968).

The 1970s saw a number of new advancements in network solution techniques. New, more powerful solution algorithms were discovered (Epp and Fowler, 1970; Hamam and
Brameller, 1971; Wood and Charles, 1972), techniques for modeling such nonpipe elements as pumps and valves were developed (Chandrashekar, 1980; Jeppson and Davis, 1976). Ways were found to implement the solution algorithms more efficiently (Chandrashekar and Stewart, 1975; Gay et al., 1978). The extension from single-time-period to multitime-(or extended) period analysis was made (Rao and Bree, 1977).

The 1980s were marked by the migration of mainframe codes to personal desktop computers (Charles Howard and Associates; 1984; Wood, 1980). They also saw the addition of water-quality modeling to network analysis packages (Clark et al., 1988; Kroon, 1990). In the 90s, the emphasis has been on graphical user interfaces (Rossman, 1993) and the integration with CAD programs and water utility databases (Haestad Methods, 1998).

### 12.2 USE OF A COMPUTER MODEL

A water distribution system model actually has two parts (Walski, 1983): the computer program that makes the calculations and the data describing the physical components of the water system, customer demands, and operational characteristics. The steps involved in the modeling process can be summarized as follows:

1. Determine the kinds of questions the model will be used to answer.
2. Represent the real-world components of the distribution system in terms that the computer model can work with.
3. Gather the data needed to characterize the components included in the model.
4. Determine water use throughout the modeled network within each time period being analyzed.
5. Characterize how the distribution system is operated over the period of time being analyzed.
6. Calibrate the model against observations made in the field.
7. Run the model to answer the questions identified in step 1 and document the results.

#### 12.2.1 Network Representation

**Network components.** Computer models require a real-life distribution system to be conceptualized as a collection of links connected together at their end points, which are called nodes. Water flows along links and enters or leaves the system at nodes. The actual physical components of a distribution system must be represented in terms of these constructs. One particular scheme for accomplishing this is shown in Fig. 12.1. In this scheme, links consist of pipes, pumps, or control valves. Pipes convey water from one point to another, pumps raise the hydraulic head of water, and control valves maintain specific pressure or flow conditions. Other types of valves, such as shutoff or check valves, are considered to be properties of pipes. Nodes consist of pipe junctions, reservoirs, and tanks. *Junctions* are nodes where links connect together and where water consumption occurs. *Reservoir nodes* represent fixed-head boundaries, such as lakes, groundwater aquifers, treatment plant clearwells, or connections to parts of a system not being modeled. Tanks are storage facilities, the volume and water level of which can change over an extended period of system operation.
12.2.1.2 Network skeletonization. When building a network model, one must decide which pipes to include in the model. The process of representing only selected pipes within a model is called skeletonization. At one end of the spectrum, a transmission-mains model might include only the major pipelines connecting points of water entry to pump stations, storage tanks, control valves, and major consumers. At the other end, an all-mains or street-level model includes every pipe, short of lateral connections to individual homes. Where a model lies along this spectrum depends in part on the kind of questions the model is being used to answer. For example, a highly skeletonized model may be sufficient for capital improvement planning or for pump-scheduling studies. Such a model might not be suitable for water-quality modeling or for fire-flow analysis, where more localized impacts are of interest.

One usually skeletonizes a network by first deciding on the smallest diameter of pipe to include in the model. To these pipes are added those that connect to large water users, major facilities, and particular points of interest, such as monitoring locations. Additional pipes might be included to close loops that are judged to be important. The advantages of a skeletonized model are reduced data-handling requirements and easier comprehension of model output. Disadvantages include the need to use engineering judgment about which pipes to include and difficulties in aggregating demand from individual water users to the nodes contained in the model. All-mains models provide more accurate depiction of true system behavior at the expense of having to supply more descriptive data and producing output that is more difficult to understand. It has become increasingly easier to develop and use all-mains models as more utilities have used computerized asset management systems and as the user interfaces and data-handling capabilities of network modeling packages have become more sophisticated.

12.2.2 Compilation of Data

Table 12.1 lists the minimum set of properties that must be supplied for the various components included in a network model. Further explanation of some of these items is provided below.
TABLE 12.1 Minimum Set of Properties Needed to Model Network Components

<table>
<thead>
<tr>
<th>Component</th>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Junctions</td>
<td>ID label, Elevation, Demand, Demand pattern</td>
</tr>
<tr>
<td>Reservoirs</td>
<td>ID label, Elevation</td>
</tr>
<tr>
<td>Tanks</td>
<td>ID label, Bottom elevation, Initial water level</td>
</tr>
<tr>
<td></td>
<td>Water level-volume curve</td>
</tr>
<tr>
<td>Pipes</td>
<td>ID label, Start node label, End node label, Diameter, Length, Roughness coefficient</td>
</tr>
<tr>
<td>Pumps</td>
<td>ID label, Start node label, End node label, Head-discharge curve</td>
</tr>
<tr>
<td>Valves</td>
<td>ID label, Start node label, End node label, Type (PRV, PSV, FCV, etc.), Pressure/flow setting</td>
</tr>
</tbody>
</table>

12.2.2.1 *ID labels.* Each node and link must be assigned a unique number or label so that it can be identified during processing. Character labels provide more flexibility than numbers do because it is easier to include useful information, such as pressure zones or place names in the label.

12.2.2.2 *Nodal elevations.* It is important to obtain accurate elevations at system boundary points, such as reservoirs and storage tanks, and at locations where pressure measurements are made for calibration purposes. Each foot of error in elevation will introduce almost half a pound per square inch (psi) of error in pressure estimates.

12.2.2.3 *Pipe diameters.* Historical information on pipe diameters only reflects the size of the pipe at the time of installation. The diameter of unlined iron pipe can be significantly reduced over time because of tuberculation caused by corrosion. The effect of this reduction on computing flows and headlosses is usually lumped together with modifications made to the pipe's roughness coefficient during model calibration. Although this works well enough for hydraulic modeling, use of an incorrect pipe diameter could lead to difficulties when trying to model water quality. The behavior of a water-quality model can be affected by the travel time of water through a pipe, which is a
function of the pipe’s diameter. Thus, a well-calibrated hydraulic model might not perform as well in modeling water-quality behavior.

12.2.2.4 Pipe roughness. A pipe’s roughness coefficient represents the contribution of irregular wall surfaces to headloss caused by friction. The type of coefficient used depends on the headloss formula being used. For the Hazen-Williams formula, the coefficient is a dimensionless quantity known as the C-factor, which decreases in value with increasing surface roughness. The Darcy-Weisbach formula uses a coefficient with units of length that represents the height of roughness elements along the pipe wall. Thus, its value increases with increasing surface roughness. Tables are available that provide typical values for both new and older pipe of different materials (Lamont, 1981, as reproduced in Walski, 1984). Values from these tables can be used as starting points when building a network model. They should ultimately be refined through both field testing and model calibration.

12.2.2.5 Pump curves. Pumps add energy to water to lift it from lower to higher levels. For a fixed rotational speed, a pump has a unique relationship between the head it can deliver and the flow rate it can supply. The curve showing this relationship is known as the head-discharge curve, and it typically has a concave downward shape. The set of curves showing head, efficiency, and power as a function of flow rate are known as a pump’s characteristic curves. They are usually provided by the pump manufacturer. However, since pump characteristics can change with time, pumping tests should be performed periodically to assess the pump’s actual performance.

Another method used to model pump performance when pump curves are not available is to assume that the pump operates at a constant horsepower rating. This approach should be used with caution, particularly with extended-period simulations because the resulting head-discharge combinations produced for the pump can sometimes become totally unrealistic.

Sometimes, it is more convenient to replace a pump connected directly to a supply reservoir with a single node that is assigned a negative demand equal to the pump’s discharge. This can be especially useful in studies where pump performance is not an issue but where it is important to model inflows to the system accurately.

12.2.3 Estimation of Demand

Water demands, or consumption rates, for a distribution system are analogous to the loads placed on a structure. Both play a major role in determining the behavior of their respective systems. Average demands can be estimated and assigned to network junctions in several ways. In order of increasing level of detail and accuracy, they are by category of land use, type and number of dwellings, meter routes, and individual meter billing records. These four methods are illustrated in Fig. 12.2. Special attention must be paid to large water users, such as certain industries, commercial establishments, universities, and hospitals. Unaccounted for water, which can be as much as 10 to 20 percent of total demand, is usually distributed uniformly across all the junctions in the network.

Average rates of water use should be adjusted to reflect the season and time of day for which the model is being run. Seasonal adjustment factors can be based on average system-production rates recorded at different times of the year. Diurnal adjustment factors can be found by performing a system water budget over a 24-h time span. This process computes total system consumption in each hour of the day as the difference between the amount of water entering the system from all external inflows and the net amount of water added to storage. The latter quantity, which can be either positive or negative, can be
determined from changes in water levels in the storage tank. The adjustment factor for any hour of the day equals the consumption in that hour divided by the average daily consumption.

12.2.4 Operating Characteristics

Additional information needed to run a network model includes the status of all pumps and valves, the initial water levels in all storage tanks, and, for water-quality analyses, the initial water quality at all nodes. When making an extended-period analysis, the model also needs to know how pumps and valves are controlled throughout the simulation period. This information might be represented through a fixed time schedule of pump/valve openings and closings or through a set of rules that describe what conditions (e.g., tank water levels or nodal pressures) will cause a pump or valve to change status.

12.2.5 Reaction-Rate Information

When modeling the fate and transport of reactive substances within a distribution system, one must be able to characterize their rate of reaction. Typically, one assumes that reaction rates are proportional to the amount of substance present; in other words, that reaction rates are first-order. Other reaction orders can be used as well (e.g., Vasconcelos et al., 1996). First-order reaction rates can be modeled using a single parameter called a reaction-rate coefficient. Rate coefficients can be different for reactions occurring in the bulk flow, at
the pipe wall, and within storage tanks. Although global values for these coefficients may be convenient to use, localized conditions, such as differences in pipe material and age, may require the use of different coefficients on a pipe-by-pipe basis. The need for such detail can be determined only from field monitoring and model calibration efforts.

12.2.6 Model Calibration

Calibration is the process of making adjustments to model inputs so that the model output reproduces observed measurements to a reasonable degree of accuracy. Adjustable model inputs primarily include pipe-roughness coefficients and nodal demands. For water-quality models, they include initial water-quality conditions and reaction-rate coefficients. Observable model outputs are pressures, flows, tank water levels, and water-quality predictions.

One can perform two levels of calibration. One level serves as a reality check that the model is producing reasonable, but not necessarily highly accurate results. The modeler should check for the following problematic behavior:

- Unreasonably low (e.g., negative) or high pressures.
- Pumps operating outside of their allowable range or being shut down for this reason.
- Pumps cycling on/off in an unreasonable fashion.
- Tanks that continuously keep filling or emptying.
- Nodes disconnected from any source because of closed pipes, pumps, or valves.

Any of these conditions indicates that there was a problem in representing some aspect of the system to the computer.

The second level of calibration involves adjustments to model input parameters that match best with field observations. This requires the collection of field data, preferably under more than one operating condition. When collecting these data, priority should be given to measuring conditions at the system boundaries. This would include flow rates and pressures at supply points or at interzone connections and water levels in storage tanks. For water-quality models, one would want to have constituent concentrations measured at these points as well. Selection of additional sampling points within the system depends on what use is being made of the model. Avoid selecting locations that provide redundant information. If possible, try to include readings from any installed flowmeters because computed flows tend to show more response to changes in input parameters than do pressures. More detailed discussions of model calibration can be found in Chaps. 9 and 14.

12.3 COMPUTER MODEL INTERNALS

The computer code that solves a network model typically includes the following functions:

- input processing,
- topological processing,
- hydraulic solution algorithm,
- linear equation solver,
- extended period algorithm,
- water-quality algorithm, and
- output processing.
12.3.1 Input Processing

The input processor takes a description of a network model that is meaningful to a human analyst and converts it into an internal representation that can be used by the computer. The network description typically is encoded into a text file with some specified format for storing the type of information shown in Table 12.1. The computer model’s input processor parses and interprets the contents of this file and assigns each piece of data to the correct internal data structure. The better computer codes obey the principle of not requiring the user to supply information that the computer can determine for itself. Thus, there should be no need for the user to specify how many nodes or links are contained in the network (the computer can count for itself as the data for each object is read). Nor should the user have to provide information on network connectivity, such as identifying closed loops or determining which nodes are linked to a given node through a link (it is enough to know which two nodes make up the end points of a link).

12.3.2 Topological Processing

The computer needs to determine certain topological relations between the nodes and links in a model so that its solution algorithms can be implemented. First among these is a determination of which nodes are directly connected to other nodes via links. This can be determined by creating an adjacency list for each node. The adjacency list is a linked list of data structures. Each element of the list contains three items: the index of the link that connects to the node in question, the index of the node on the other end of the link, and a pointer to the next item in the list. An array is used to store the address of the first element of the list for each node.

A second type of topological processing is required for hydraulic solution methods that use Kirchoff’s law, which requires that the sum of headlosses and gains around each closed loop in a network be zero. An efficient means is needed to determine a set of basic or fundamental loops. The number of such loops in a network with \( NN \) nodes and \( NL \) links is \( NL - NN + 1 \). Osiadacz (1987) described a number of ways available to generate a set of loops, all of them based on finding a spanning tree for the network. A spanning tree is a collection of links that connects every junction node in the network back to a reservoir or tank node without forming any loops. The spanning tree also is useful for determining an initial set of flows that satisfies nodal continuity.

12.3.3 Hydraulic Solution Algorithms

The basic equations required to solve for flows and hydraulic heads (from which pressures can be obtained) in a network are

\[
\sum_i Q_{ij} - \sum_k Q_{kj} = D_j \quad \text{for each node } j \tag{12.1}
\]

and

\[
H_i - H_j = aQ_{ij}Q_{ij}^{-*} \quad \text{for each link connecting nodes } i \text{ and } j \tag{12.2}
\]

where \( Q_{ij} \) = flow in link connecting nodes \( i \) and \( j \) (positive if flow is from \( i \) to \( j \), otherwise negative), \( H_j \) = head at node \( j \), and \( D_j \) = demand at node \( j \). Equation (12.1) is the nodal continuity relation and Eq. (12.2) is the flow-headloss relationship, where \( a \) and \( b \) are coefficients. For the Hazen-Williams formula,
\[ a = \frac{10.69L}{C^{1.85}d^{1.87}} \]  
(12.3)

and

\[ b = 1.85 \]  
(12.4)

where \( L \) = pipe length (m), \( d \) = pipe diameter (m), and \( C \) = a roughness coefficient.

Computer models can reduce these equations into a simpler system to be solved in four different ways: (1) the node method (\( H \) equations), (2) the flow method (\( Q \) equations), (3) the loop method (\( \Delta Q \) equations), and (4) the gradient or node-loop method (\( H-Q \) equations). The Newton-Raphson technique is used with each method to solve the resulting system of nonlinear equations by means of an iterative solution of a system of linear equations. Well-behaved systems typically will converge in under four to six iterations. More information about the computational details of these methods can be found in Jeppson (1976), Osiadacz (1987), and Salgado et al. (1988). Although the original Hardy Cross method cannot compete with the numerical efficiency of these procedures, it still appears in sample computer programs for pipe flow analysis found in many textbooks (e.g., Clark et al., 1977).

Table 12.2 compares some pertinent characteristics of the four different methods. The flow method results in the largest number of equations and the loop method has the smallest. The node method solves for heads, and the flow and loop methods solve for flows. Once either heads or flows are known, it is a straightforward matter to determine the complementary set of unknowns using the flow-headloss relations. The gradient and node-loop methods are unique in that both heads and flows are determined in a recursive fashion during the Newton-Raphson iterations, with each new set of flows serving as a feedback signal used to update the next new set of heads.

Both the flow and loop methods require that a set of fundamental loops must be identified. The loop method also requires that an initial estimate of link flows must be found to satisfy continuity. The node method is known to have convergence problems on some networks that have low resistance links connected to high-resistance ones and

| TABLE 12.2 Characteristics of Different Hydraulic Solution Methods |
|-----------------|-----------------|-----------------|-----------------|
|                 | Node Method     | Flow Method     | Loop Method     |
| Number of equations | \( NJ \)        | \( NJ + NL \)  | \( NL - NJ \)  |
| Variable solved for | Head            | Flow            | Flow adjustment |
| Requires loop generation | No              | Yes             | Yes             |
| Requires initial flow solution | No              | No              | Yes             |
| Convergence properties | Poor to good    | Good            | Good            |
| Symmetric coefficient matrix | Yes             | No              | Yes             |
| Relative degree of sparsity | High            | Medium          | Low             |

<table>
<thead>
<tr>
<th></th>
<th>Gradient/Node Loop Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>NJ</td>
<td></td>
</tr>
<tr>
<td>Head and flow</td>
<td></td>
</tr>
</tbody>
</table>

Abbreviations: \( NJ \) = number of junctions; \( NP \) = number of links.
pumps with steep head-discharge curves (Salgado et al., 1988). The node, loop, and gradient/node-loop methods all produce symmetric coefficient matrices for the system of linear equations resulting from the Newton-Raphson procedure. Symmetric matrices require less computer memory for storage and permit the use of more efficient solution techniques. The node and gradient/node-loop methods have an easier time accommodating nonpipe elements—particularly check valves, regulating valves, and closed pipes—than do the loop-based methods. Special techniques are required by the loop-based methods to handle situations in which a pipe can be closed and its flow becomes zero.

12.3.4 Linear-Equation Solver

Most of the work in computing network hydraulics involves solving the system of simultaneous linear equations that results at each iteration of the Newton-Raphson method. Indeed, improvements in this aspect of the computer model are likely to have more impact on the size and speed at which network problems can be solved than will the choice of solution algorithm. The coefficient matrices associated with the various solution methods are moderately to highly sparse, meaning that most entries are zeroes. For example, the number of nonzero coefficients in a row of the matrix derived from either the node or the gradient/node-loop methods represents the number of links connected to a particular node. Because pipe networks rarely have more than five links connecting to a node, a system with 1000 nodes will have at most only 5000 nonzero entries in the 1,000,000 elements of its coefficient matrix. Having to store and carry all the zero-value elements through the process solution can impose a significant computational penalty.

Direct methods for solving sets of linear equations use a series of elementary operations (e.g., addition and multiplication) on the elements of the coefficient matrix to transform it into a triangular form so that the equations can be solved via a sequence of simple substitutions. This process, known as factorization, adds additional nonzero entries to the transformed coefficient matrix, called fill-ins, which in turn increases the computational burden. Simply by rearranging the rows and columns of the matrix, the number of fill-ins can often be reduced significantly. For the node and gradient/node-loop methods, such a rearrangement is equivalent to simply reordering (or renumbering) the nodes in the network. The techniques used to minimize fill-in and to store and operate on only the nonzero elements of the coefficient matrix are known as sparse matrix methods (see George and Liu, 1981, and Pissanetzky, 1984, for further discussions). Needless to say, without the use of such methods, it would be impossible to analyze large pipe networks in reasonable amounts of time using personal computers.

12.3.5 Extended-Period Solver

Simulating the behavior of a distribution system over an extended period of time is known as extended-period simulation (EPS), which allows the modeler to capture the effects that changes in customer demands and tank water levels have on system performance. EPS also is a prerequisite for performing meaningful water-quality analysis. The method used to perform EPS is to integrate the differential equation that represents the change in head at storage tanks with respect to time, using steady-state network analysis to compute network flows as a function of tank heads. The equations to be solved for each storage tank \( s \) can be expressed as

\[
\frac{dV_s}{dt} = Q_s, \quad (12.5)
\]
and

\[ H_s = E_s + h(V_s) \]  

(12.6)

where \( V_s \) = volume of water in storage tank \( s \), \( t \) = time, \( Q_s \) = net flow into (+) or out of (-) tank \( s \), \( H_s \) = head (water surface elevation) in tank \( s \), \( E_s \) = elevation of bottom of tank \( s \), and \( h(V_s) \) = water level as a function of water volume in tank \( s \).

A number of different methods are available to perform this integration. The simplest method is known as the Euler method, which replaces the \( dV_s/dt \) term in Eq. (12.5) with its forward difference approximation so that

\[ V_s(t + \Delta t) = V_s(t) + Q_s(t)\Delta t \]  

(12.7)

and

\[ H_s(t + \Delta t) = E_s + h(V_s(t + \Delta t)) \]  

(12.8)

where \( X(t) \) denotes the value of \( X \) at time \( t \). Under this scheme, the tank levels \( H_s(t) \) and nodal demands existing at time \( t \) are used to solve a network flow analysis that produces a set of net flows \( Q_s(t) \) into the tanks at time \( t \). Equations (12.7) and (12.8) are used to determine new tank levels after a period of time \( \Delta t \). Then a new steady state analysis is run for time \( t + \Delta t \), using the new tank levels as well as new demands and operating conditions that apply to this new time period. The simulation proceeds in this fashion from one time period to the next.

Under most conditions, the Euler method produces acceptable results because over a typical time period of say, 1 h, demands and tank levels do not change dramatically. Other integration schemes, such as the predictor-corrector method (Rao and Bree, 1977), are available to accommodate more rapid changes in conditions. These methods require additional flow solutions to be made at intermediate time steps. As mentioned above, EPS uses a succession of steady-state flow solutions that does not account for either inertial or compressibility effects in pipe flow. Hence, it cannot be called a truly dynamic simulation approach. However, for the kinds of model uses described in Sec. 12.1.2, it has proved to work well enough in practice.

### 12.3.6 Water-Quality Algorithms

The numerical methods for tracking the propagation and fate of water-quality constituents in distribution systems were reviewed in Chap. 9. All of these methods require as input the link flows computed over each time step of an extended-period hydraulic analysis. The time steps used in a water-quality analysis typically are much shorter than those used for extended-period hydraulic analysis (e.g., 5 min. instead of 1 h). Because all the methods use some form of subdiscretization of the network's links, a substantial amount of computer memory can be required for water-quality analysis. Thus, most models compute water-quality conditions after a hydraulic analysis has already been performed rather than attempt to implement the two procedures simultaneously.

### 12.3.7 Output Processing

Output processing conveys the results computed by the models in a format that is useful and informative to the model's users. The amount of information generated by these models can be enormous—flows, pressures, heads, and water quality at thousands of
nodes and links for dozens or even hundreds of time periods. Such huge amounts of data require that selective output reporting be used or that disk files be used to archive it. Several issues and options arise in designing effective output processors. These include the following:

1. **Time steps.** Extended-period hydraulic simulations usually proceed at some fixed time step set by the model’s user. However, new solutions can be generated at intermediate times, such as when tanks are taken off-line because they become empty or full or when pumps open and close because certain tank levels are reached. A decision must be made regarding whether such intermediate results are saved and made available to the model’s user. A compromise strategy is to keep a separate log of changes in system status that records when intermediate solutions occur and what system components changed state but does not provide complete reporting of such solutions to the user.

2. **Reporting options.** Input options should be available to allow the user to state which nodes and links should be reported and in which time periods. A common technique used in water-quality analyses to establish repeating periodic conditions over a prescribed operating period is to run the model in repeating fashion for dozens or even hundreds of such periods until this state is achieved. Clearly, there is no need to save results generated during the transient start-up phase of such simulations. Another type of reporting option allows the user to request that reports be made only when a certain condition is encountered, such as the pressure at a node being below a set level or the head loss in a pipe exceeding a certain level.

3. **Binary output files.** To warehouse the potentially enormous amount of output data that can be generated by a network model, the use of binary files is a must. These provide much quicker data access and more compact data storage than do formatted text files. Creation of such files provides great flexibility in using postprocessing software to examine the data in a multitude of different ways. As alternatives to simple binary files, output results also can be saved in formats used by commercial spreadsheet and database programs, which also are usually binary in nature.

4. **Error reporting.** Output processing should report all error and warning conditions encountered in processing a network’s input data and in running an analysis. Noteworthy conditions that warrant such reporting include failure of a hydraulic analysis to converge to a solution, pumps operating outside the limits of their head-discharge curves, occurrence of negative pressures, and nodes disconnected from any source of water as a result of link closures.

---

**12.4 EPANET PROGRAM**

**12.4.1 Background**

EPANET is a public-domain, water-distribution-system modeling package developed by the U.S. Environmental Protection Agency’s Water Supply and Water Resources Division. It performs extended-period simulation of hydraulic and water-quality behavior within pressurized pipe networks and is designed to be a research tool that improves our understanding of the movement and fate of drinking-water constituents within distribution systems. EPANET first appeared in 1993 (Rossman, 1993), and a new version is slated for release in 1999. The program can be downloaded from the World Wide Web at [http://www.epa.gov/ORD/NRMRL/wswrd/epanet.html](http://www.epa.gov/ORD/NRMRL/wswrd/epanet.html).
12.4.2 Program Features

EPANET models a distribution system using the same objects as described in Sec. 12.2.1.1. These objects include the following:

- Junctions (where pipes connect and water consumption occurs)
- Reservoirs (which represent fixed-head boundaries)
- Tanks (which are variable-volume storage facilities)
- Pipes (which can contain either shutoff valves or check valves)
- Pumps (which can include fixed-speed, variable-speed, and constant-horsepower pumps)
- Control valves (which can include pressure-reducing valves, pressure-sustaining valves, flow control valves, and throttle-control valves)

In addition to these physical objects, the following informational objects also can be used to represent a distribution system:

- Time patterns (sets of multipliers used to model diurnal water demands)
- Curves (x-y data used to represent head-discharge curves for pumps and water level-volume curves for tanks)
- Operational controls (rules that change link status depending on such conditions as tank levels, nodal pressures, and time)
- Hydraulic analysis options (choice of headloss formula, flow units, viscosity, and specific gravity)
- Water-quality options (choice of type of water-quality analysis, type of reaction mechanism, and global reaction rate coefficients)
- Time parameters (simulation duration, time steps for hydraulic and water-quality analyses, and time interval at which output results are reported)

In addition to steady-state or extended-period hydraulic analysis, EPANET can be used to run the following kinds of water-quality analyses: (1) tracking the propagation of a nonreactive constituent, such as one that would be used in a tracer study or for reconstruction of a contamination event, (2) determining what percentage of water from a particular source is received by each location in a network, (3) estimating the age of water received at various locations in the network, (4) modeling the fate of chlorine and chloramines, which decay with time and can react both in the water phase and at the pipe wall, and (5) modeling the growth of certain disinfection by-products, such as trihalomethanes, which grow with time up to a limiting value.

EPANET consists of two modules. One is a network solver that performs the hydraulic and water-quality simulation; the other is a graphical user interface that serves as a front-and-back end for the network solver. In the program’s normal mode of operation, the user interacts directly with the graphical interface with the solver’s presence being transparent. The solver also can be run as a stand-alone executable, receiving its input from a text file and writing its results to a formatted text-report file or an unformatted, binary-output file. The solver also exists as a library of callable functions that third-party developers can use in custom applications.
12.4.3 User Interface

EPANET’s graphical user interface is responsible for constructing the layout of the network to be simulated, editing the properties of the network’s components and its simulation options, calling on the solver module to simulate the behavior of the network, and accessing results from the solver to display to the user in a variety of formats. It was written specifically for the Windows 95/98/NT platform using Inprise’s Delphi language (an object-oriented version of Pascal).

Figure 12.3 depicts the user interface as it might appear when editing a network. The Network Map provides a schematic layout of the distribution system that gives the user a visual sense of where components are located and how they connect together. Because the nodes and links can be colored according to the value of a particular variable, such as pressure or flow, the Network Map also offers a holistic view of how quantities vary spatially across the network. The toolbar that appears above and to the right of the map allows the user to add new components visually using point-and-click with the mouse. Existing items on the map also can be selected and then moved, edited, or deleted. The toolbar also includes icons for panning and zooming in or out on the map as well.

The Browser window is the central control panel for EPANET. It is used to (1) select specific network objects, (2) add, delete, or edit network objects including such nonvisual objects as time patterns, operating rules, and simulation options, (3) select which variable

FIGURE 12.3 EPANET's user interface.
is viewed via color-coding on the map, and (4) select which time period is viewed on the map for extended-period simulations.

The third, smaller window shown in Fig. 12.3 is the Property Editor, which is used to change the properties of the item currently selected on the Network Map and in the Browser. Other, more specialized property editors exist for editing the network's nonvisual data objects, such as time patterns, curves, and operating rules.

The Network Map, the Browser, and the Property Editor are all connected to one another in the sense that any selection or change made in one is carried over to the others. For example, if the user clicks on a specific node on the map, that node becomes the current object shown in the Browser and in the Property Editor. If the user changes the diameter of a pipe in the Property Editor and diameter is the current variable selected in the Browser for viewing, the pipe will be redrawn on the map with its color changed.

EPANET's user interface keeps all the data that describes a network in its own object-oriented internal database. When the user wants to run an analysis, the program writes these data to a text file, which is then passed on to the solver module for processing. The solver makes it computations and writes its results to an unformatted binary file. The user interface then accesses this file to display selected results back to the user on request.

Figure 12.4 provides some examples of the kinds of output views that EPANET's user interface can generate after a successful simulation has been made. The top left window shows the results of a query made to the Network Map, asking it to identify all nodes where the pressure was below 50 psi. The top right window is a tabular display of link results at hour 6 of the simulation, where a filter was applied to list only those links where the headloss per 1000 ft was above 1.0. The bottom left window depicts a time-series plot for pressure at two different locations in the network. Finally, the bottom right window displays a contour plot of pressure throughout the network at hour 6.

![FIGURE 12.4 Examples of EPANET's output views.](image-url)
EPANET also provides tools to help aid in calibration of the network. These tools are illustrated in Fig. 12.5 for the case where a 54-h. fluoride tracer study was made in a particular network. The top left window shows a time series plot for fluoride at a specific node where both the simulated and measured values are shown for comparison. The window to its right depicts a calibration report. It compares errors between observed and computed fluoride values at all measuring locations. The two lower plots are different views of the same data. One compares measured and computed results for all samples at each location; the other compares the mean values of the computed and observed samples for each location.

12.4.4 Solver Module

EPANET's solver program is written in American National Standards Institute standard C with separate code modules for input processing, hydraulic analysis, water-quality analysis, sparse matrix/linear-equation analysis, and report generation. This modular approach facilitates making modifications to the program's features and computational procedures. The data-flow diagram for the solver is shown in Fig. 12.6. The processing steps depicted in the diagram can be summarized as follows:

1. The input processor module of the solver receives a description of the network being simulated from an external input file (.INP). This description is written in an easily understood Problem Description Language that will be discussed in more detail below. The file's contents are parsed, interpreted, and stored in a shared memory area.
2. The hydraulics module carries out a complete, extended-period hydraulic simulation, with the results obtained at every time step written to an external, unformatted (binary) hydraulics file (.HYD). Some of these time steps might represent intermediate points in time when system conditions change because tanks become full or empty or pumps turn on or off because of level controls or timed operation.

3. If a water-quality simulation was requested, the water-quality module accesses the flow data from the hydraulics file as it computes substance transport and reaction throughout the network over each hydraulic time step. During this process, it writes both the formerly computed hydraulic results as well as its water-quality results for each pre-set reporting interval to an unformatted (binary) output file (.OUT). If no water-quality analysis was called for, then the hydraulic results stored in the .HYD file are simply written out to the binary output file at uniform reporting intervals.

4. If requested by the input file, a report writer module reads back the results of the computed simulation from the binary output file (.OUT) for each reporting period and writes out selected values (as instructed by the user) to a formatted report file (.RPT). Any error or warning messages generated during the run also are written to this file.

When called by the Windows user interface, the solver skips Step 4 because the interface itself is used to generate output reports. The input file fed to the solver is written using a Problem Description Language (PDL) that makes it easily readable and self-documenting. Excerpts from such a file are shown in Fig. 12.7. Each category of input data is placed in a separate section identified by a keyword in brackets. Comment lines, beginning with a semicolon, can be placed throughout the file. The properties for multiple network objects of the same type, such as junctions and pipes, are entered in a columnar
Sample Pipe Network

[JUNCTIONS]
; ID Elev. Demand
; ID ft. gpm
---
1 1090
101 1090
2 1122
3 1138
4 1157 500
5 1180
<etc.>

[TANKS]
; ID ft. Level Level Level ft.
---
17 910
18 810 10 0 20 50

[PIPES]
; ID From To Length Diam.
; ID Node Node ft. In. C-factor
---
1 1 2 1500 12 130
2 2 3 1000 8 130
3 3 4 1200 10 120
4 4 5 2000 10 120
<etc.>

[PUMPS]
; ID From To Head Flow
; ID Node Node ft. gpm
---
171 17 101 456 2700

[VALVES]
; ID From To Setting
; ID Node Node Type psi
---
25 21 23 PRV 75

[OPTIONS]
UNITS GPM
HEADLOSS H-W

FIGURE 12.7 Excerpts from an EPANET input data file.
format that saves space and enhances readability. Singular properties, such as analysis options, are entered in keyword-value format. When working with the Windows user interface, the existence of the PDL input file is invisible to the user, although the user can generate such a file so that a stand-alone, human-readable version of a network’s data can be made available.

The solver uses the gradient method with extensions, as described in Salgado et al. (1988), to solve the network hydraulic equations that result at each time step of the simulation. A special procedure is implemented on top of this to keep track of changes in status that can occur in pumps and valves as the iterations of the gradient method unfold. The minimum-degree node reordering method of George and Liu (1981) is used to minimize fill-ins in the system of linear equations solved via Cholesky factorization. The coefficient matrix for these equations is stored in a sparse row-wise matrix format (Pissanetzky, 1984). Simple Euler integration is used to update water levels in storage tanks between hydraulic time steps.

The water-quality solution method used by the solver is based on a time-driven, lagrangian transport scheme described in Rossman and Boulos (1997). Reactions in both the bulk fluid and at the pipe wall can be modeled using the approach described in Rossman et al. (1994). General nth-order reactions with a limiting growth or decay potential can be modeled in the bulk phase, whereas either zero or first-order, mass-transfer-limited reactions can be modeled at the pipe wall. Storage tanks can be modeled as completely mixed, plug-flow, or two-compartment reactors.

12.4.5 Programmer’s Toolkit

The functions in EPANET’s network solver have been compiled into a library of routines that can be called from other applications. The toolkit functions permit a programmer to (1) open an EPANET input file, read its contents, and initialize all necessary data structures, (2) modify the value of selected network objects, such as nodal demands, pipe diameters, and roughness coefficients, (3) run repeated hydraulic and water-quality simulations using sets of modified parameters, (4) retrieve the value of selected simulation results, and (5) generate a custom report on simulation results.

The toolkit allows network modelers to incorporate state-of-the-art hydraulic and water-quality analysis routines into their own custom applications without having to worry about the details of programming these capabilities on their own. The toolkit should prove useful for developing specialized applications, such as optimization models or parameter estimation models, that require running many network analyses with modified input parameters. It also can simplify adding analysis capabilities to integrated network-modeling environments based on CAD, GIS, and database packages.

12.5 CONCLUSION

Computer models of distribution systems have achieved a remarkable degree of power and sophistication over the past 30 years. Indeed, it is probably unlikely that any more major advances can be made in solving the basic network-flow problem. This is not to say that network modeling is a dead issue. Several key themes now pervade the water industry’s view of network modeling and will continue to challenge developers of computer models in the years to come.

Foremost among these themes is systems integration. Network models do not exist in a vacuum. They rely on many different sources of data to supply the information needed
to run them, and their results can be used by a variety of different groups within a water utility. The issue involves being able to link network models efficiently and productively to other corporate information systems, such as engineering CAD systems, GIS, customer billing records, customer complaint records, water-quality monitoring records, pipe replacement and repair records, and real-time Supervisory Control and Data Acquisition (SCADA) systems. The topic continues to be one of high interest within the water industry, as evidenced by the recent papers of Kroon (1997) and Lerner and DiSera (1997).

A second ongoing theme is the embedding of network simulation models in other types of computer models aimed primarily at questions of system optimization and control. Examples include optimization models for pipe sizing (Dandy et al., 1996), optimal calibration programs (Lingireddy and Ormsbee, 1998), neural network models of system operation (Swiercz, 1994), and models for optimally locating and operating satellite chlorination stations (Boccelli et al., 1998). Developing libraries of network simulation functions using an open architecture, such as the EPANET Programmer’s Toolkit, will help to simplify the task of embedding simulation capabilities in other codes and perhaps even encourage the creation of new and more innovative applications.

A third piece of unfinished business for network computer models involves enhancements to their water-quality modeling capabilities. Advancements continue to be made as our understanding of the chemical and biological behavior of treated water within the pipe environment improves. Whereas the first generation of models could address only nonreactive substances or substances following first-order decay reactions, recent improvements have added capabilities to model pipe-wall reactions (Rossman et al., 1994), production of trihalomethane (Vasconcelos et al., 1996), and bacterial growth (Piriou et al., 1996). Further advances are needed to model the transport and fate of alternative disinfectants, such as chloramines; to account for the effect that blending of water from different sources has on reaction kinetics; and to track the movement and fate of particulates.

Computer modeling of distribution systems has become a mainstay of the water industry. It is a classic example of how research and development carried out mainly at universities and government laboratories has been transferred into practical, useful tools for everyday practitioners. Distribution system modeling has reached a level of maturity and reliability that make it a valuable asset to any water utility.

REFERENCES


Cross, H., *Analysis of Flow in Networks of Conduits or Conductors*, University of Illinois Engineering Experiment Station Bulletin 286, Urbana, IL, 1936.


