École polytechnique fédérale de Lausanne
School of Architecture, Civil and Environmental Engineering
Civil Engineering & Environmental Science and Engineering Sections
CIVIL-312: Hydraulic Engineering and Infrastructures

Introduction to EPANET

Fundamentals and Applications in Pipe Network Modeling

Instructor: Clemente Gotelli
Date: October 10, 2025

Contents

1 Introduction

Water distribution networks are a fundamental component of urban infrastructure. They ensure that clean water is delivered from sources and reservoirs to consumers with sufficient pressure and reliability. Designing and operating such systems is a major engineering challenge, as it requires balancing efficiency, safety, and cost while coping with variable demands, energy consumption, and possible failures. Modern cities often rely on networks composed of hundreds or even thousands of interconnected pipes, pumps, valves, and storage tanks, which interact in complex ways.

Because of this complexity, analytical solutions are not feasible for real-world networks. Engineers must instead rely on numerical models to evaluate how the system behaves under different conditions. These models are used in design (e.g., deciding pipe diameters, pump locations, or tank capacities), in operation (e.g., testing pumping schedules or valve operations), and in planning for resilience (e.g., simulating pipe breaks, peak demands, or contamination events).

Within this context, **EPANET** has become one of the most widely used tools worldwide for simulating water distribution systems. Developed by the U.S. Environmental Protection Agency (EPA), it is open-source, user-friendly, and robust enough to handle a wide range of practical applications. EPANET computes flows, pressures, and water quality parameters in pressurized networks under steady-state and extended-period conditions, making it a standard reference in both academia and practice.

The learning objectives of this module are:

- To understand the mathematical background behind network modeling and the Hardy Cross and Newton–Raphson methods.
- To practice manual solution methods in small networks using simple tools (e.g., Excel).
- To learn how to implement and simulate networks in EPANET.
- To analyze and interpret the results of more complex hydraulic systems.

2 Theoretical Background

The hydraulic behavior of pressurized pipe networks follows two conservation principles that can be written in algebraic form and solved numerically. These principles provide the structure of the nonlinear system whose solution gives nodal heads and pipe flows.

- **Mass conservation at junctions** The algebraic sum of inflows and outflows at each node equals the external demand or supply.
- **Energy conservation along links** The difference in total head between the end nodes of a pipe equals the head loss caused by friction and local components.

In what follows, each relation is stated in general terms and then applied to the small network in Figure 1. Node 1 is a supply node where water is injected into the system. Node 3 is a withdrawal node with a prescribed demand. Nodes 2 and 4 are transit nodes with zero external demand. We adopt the sign convention $D_j > 0$ for withdrawal and $D_j < 0$ for injection, so $D_1 < 0$, $D_3 > 0$, and $D_2 = D_4 = 0$. Arrows in the figure define the positive flow directions used in the formulas.

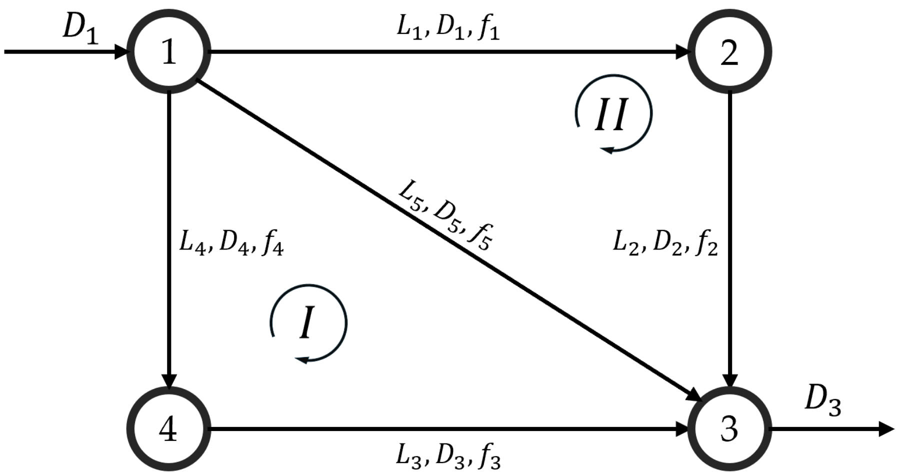


Figure 1: Four-node network. Node 1 injects water into the system and Node 3 withdraws water. Nodes 2 and 4 are transit nodes. Positive flow directions are indicated by arrows.

2.1 Conservation Laws in Pipe Networks

2.1.1 Continuity at Nodes

At every junction the algebraic sum of inflows minus outflows equals the external demand or supply

$$\sum Q_{\text{in}} - \sum Q_{\text{out}} = D,$$

with $D > 0$ for withdrawal and $D < 0$ for injection. This enforces mass conservation at the control volume around the node.

Four-node application Adopt the positive flow directions indicated by the arrows in Figure 1. Let the pipe flows be

$$Q_{12},\ Q_{23},\ Q_{13},\ Q_{14},\ Q_{43}.$$

Node 1 is a supply node ($D_1 < 0$), node 3 is a demand node ($D_3 > 0$), and nodes 2 and 4 are transit nodes ($D_2 = D_4 = 0$). Continuity gives

$$\boxed{Q_{14} + Q_{12} + Q_{13} = D_1}, \qquad \boxed{Q_{12} = D_2 + Q_{23}},$$

$$\boxed{Q_{23} + Q_{13} + Q_{43} = D_3}, \qquad \boxed{Q_{14} = D_4 + Q_{43}}.$$

These four equations, together with the energy relations written along the five pipes $(12,\ 13,\ 14,\ 23,\ 43)$, close the hydraulic problem.

2.1.2 Energy Conservation in Pipes

Between two nodes i and j connected by a pipe, the difference in total head equals the head loss due to friction and local effects

$$H_i - H_j = h_{ij}(Q_{ij}),$$

where H_k is the total head at node k and $h_{ij}(\cdot)$ is a monotone odd function of the signed flow Q_{ij} measured in the chosen positive direction.

Two common formulations used in practice are

- **Darcy–Weisbach**

  $$H_i - H_j = K_{ij}\, Q_{ij}|Q_{ij}|, \qquad K_{ij} = \frac{8 f_{ij} L_{ij}}{g\, \pi^2 D_{ij}^5},$$

  with friction factor f_{ij}, length L_{ij}, and diameter D_{ij}.

- **Hazen–Williams**

  $$H_i - H_j = K_{ij}\, Q_{ij}|Q_{ij}|^{0.852}, \qquad K_{ij} = \frac{10.67\, L_{ij}}{C_{ij}^{1.852}\, D_{ij}^{4.87}},$$

  with roughness coefficient C_{ij}.

Minor losses can be included by adding $K_{ij} V_{ij}^2/(2g)$ or by folding them into effective coefficients.

Four-node application With the pipe set $(12,\ 13,\ 14,\ 23,\ 43)$ and positive directions as in Figure 1, the unknown heads $H_1, \ldots, H_4$ satisfy

$$\boxed{H_1 - H_2 = h_{12}(Q_{12})}, \qquad \boxed{H_1 - H_3 = h_{13}(Q_{13})}, \qquad \boxed{H_3 - H_4 = h_{14}(Q_{14})},$$

$$\boxed{H_4 - H_1 = h_{23}(Q_{23})}, \qquad \boxed{H_2 - H_4 = h_{41}(Q_{41})}.$$

With this sign convention a negative Q_{ij} means the actual flow is opposite to the arrow and the head drop reverses accordingly.

In what follows we apply the Hardy Cross method (sequential loop corrections) and then solve the same network with Newton–Raphson.

2.2 The Hardy Cross Method

The Hardy Cross method is a classical iterative technique for solving pressurized pipe networks. Originally developed in the 1930s, it provides a systematic way to compute steady-state flows by enforcing two conservation principles: (i) *mass conservation* at the nodes, and (ii) *energy conservation* around closed loops.

Its main advantage lies in conceptual simplicity. Starting from any initial set of pipe flows that satisfies node continuity, the method progressively applies loop corrections until the energy balance is achieved within a prescribed tolerance. At each iteration, a single flow correction ΔQ is applied uniformly to all pipes of the loop, which automatically preserves node continuity.

Hardy Cross updates flows sequentially by treating each loop independently. This leads to slow convergence in large or highly interconnected networks, where many iterations may be needed before the system balances. The Newton–Raphson method (next section) linearizes the full system of nonlinear equations and solves all unknowns simultaneously, making it more robust and efficient for large-scale problems.

To illustrate the Hadry Cross method, we use the four–node, two–loop example shown in Figure 1 and derive the flow correction formula step by step.

The loop balance condition requires that the oriented sum of head losses around each closed loop vanish

$$\sum_{(i,j)\in L} s_{ij}\, h_{ij}(Q_{ij}) = 0,$$

where $s_{ij} = +1$ if the pipe's positive direction agrees with the loop orientation and $s_{ij} = -1$ otherwise. A single correction with the same magnitude is applied to all loop members

$$Q_{ij}^{(k+1)} = Q_{ij}^{(k)} + s_{ij}\,\Delta Q,$$

which preserves node continuity at every iteration.

For Darcy–Weisbach with fixed f, the correction takes the form

$$\Delta Q = -\frac{\sum h_f}{2\sum \left| h_f/Q_{ij} \right|},$$

where $\sum h_f$ is the oriented loop sum of head losses. This expression generalizes to other head–loss laws by replacing the factor 2 with the proper exponent m.

Important: If a pipe belongs to more than one loop, it receives both corrections with the appropriate signs. In Figure 1, the diagonal pipe is shared by the two loops, so

$$Q_{\text{diag}}^{\text{new}} = Q_{\text{diag}}^{\text{old}} + \Delta Q_{\text{I}} - \Delta Q_{\text{II}}.$$

2.2.1 Step-by-Step Procedure

1. **Initialization** Select a loop orientation. For each pipe (i, j) in the loop, assign a sign s_{ij}: $s_{ij} = +1$ if the pipe's positive flow direction agrees with the loop orientation, $s_{ij} = -1$ otherwise. Choose initial flows $Q_{ij}^{(0)}$ that satisfy continuity at all junctions. A good initial guess may significantly reduce the number of iterations.

2. **Head losses and loop sum** For the current flows, compute the head loss in each member $h_{ij} = h_{ij}(Q_{ij})$. Typical formulas are given in Table 1. If a pump is present, include its signed head $H_{p,ij}(Q_{ij})$ in the loop sum.

   Form the oriented loop sum

$$\sum h_f = \sum_{(i,j)\in L} s_{ij}\, h_{ij} \quad (+\, s_{ij}\, H_{p,ij} \text{ for pumps}),$$

   and if $|\sum h_f| \leq \text{tol}$, the loop is balanced. Otherwise proceed to the correction step.

3. **Flow correction** We linearize each pipe loss around the current flows

$$h_{ij}(Q_{ij} + s_{ij}\,\Delta Q) \approx h_{ij}(Q_{ij}) + s_{ij}\,\frac{dh_{ij}}{dQ}(Q_{ij})\,\Delta Q.$$

   Then, summing over the loop gives

$$\Delta Q = -\frac{\displaystyle\sum_{(i,j)\in L} s_{ij}\, h_{ij}(Q_{ij})}{\displaystyle\sum_{(i,j)\in L} \frac{dh_{ij}}{dQ}(Q_{ij})}.$$

   With Darcy–Weisbach, $h_{ij} = K_{ij}\, Q_{ij}\, |Q_{ij}|$ and $\frac{dh_{ij}}{dQ} = 2K_{ij}\, |Q_{ij}| = 2|h_{ij}/Q_{ij}|$. Thus the correction becomes

$$\Delta Q = -\frac{\sum h_f}{2\sum \left| h_{ij}/Q_{ij} \right|}$$

   For a generic power law, $h_{ij}(Q) = K_{ij}\, Q_{ij}\, |Q_{ij}|^{m-1}$, the update reads

$$\Delta Q = -\frac{\displaystyle\sum_{(i,j)\in L} s_{ij}\, h_{ij}}{\displaystyle m \sum_{(i,j)\in L} \left| h_{ij}/Q_{ij} \right|}$$

   which reduces to the Darcy–Weisbach case when $m = 2$, and to Hazen–Williams when $m = 1.852$.

4. **Update flows** Apply the correction to every pipe in the loop

$$Q_{ij}^{\text{new}} = Q_{ij}^{\text{old}} + s_{ij}\,\Delta Q \qquad \forall (i, j) \in L.$$

   If a pipe belongs to two loops, add both corrections with the proper signs.

5. **Repeat** Return to Step 2 until $|\Delta Q|$ and $|\sum h_f|$ are below the prescribed tolerances for all loops.

Table 1: Head loss formulas and coefficient definitions in pipe networks

Method	Head loss expression	Coefficient definition
Darcy–Weisbach	$h_{ij} = K_{ij}\, Q_{ij}\, \lvert Q_{ij}\rvert$	$K_{ij} = \dfrac{8 f_{ij} L_{ij}}{g\,\pi^2 D_{ij}^5}$
Hazen–Williams	$h_{ij} = K_{ij}\, Q_{ij}\, \lvert Q_{ij}\rvert^{0.852}$	$K_{ij} = \dfrac{10.67\, L_{ij}}{C_{ij}^{1.852}\, D_{ij}^{4.87}}$
Generic power law	$h_{ij} = K_{ij}\, Q_{ij}\, \lvert Q_{ij}\rvert^{m-1},\ (m > 1)$	–
Minor loss	$h_{\text{minor}} = \dfrac{K_{L,ij} V_{ij}^2}{2g} = \dfrac{8 K_{L,ij}}{g\,\pi^2 D_{ij}^4}\, Q_{ij}\, \lvert Q_{ij}\rvert$	$K_{L,ij}$ given by fittings or valves

2.2.2 Worked example. Four-node network with two loops

Let's solve the Four-node network example shown before. Figure 2 shows all the necessary details.

Data

- Injection at node 1: $D_1 = -0.6\ \mathrm{m^3/s}$. Withdrawal at node 3: $D_3 = +0.6\ \mathrm{m^3/s}$. Nodes 2 and 4 are transit ($D_2 = D_4 = 0$).
- Pipes and positive directions as in Figure 2. We label $Q_1 = Q_{12}, Q_2 = Q_{23}, Q_3 = Q_{34}, Q_4 = Q_{41}$, $Q_5 = Q_{13}$.
- Darcy–Weisbach with fixed $f = 0.02$. Gravity $g = 9.81\ \mathrm{m/s^2}$. Head loss in pipe i: $h_i(Q_i) = K_i\, Q_i|Q_i|$ with
$$K_i = \frac{8 f L_i}{g\,\pi^2 D_i^5}.$$
- Lengths and diameters
$$(L_1, D_1) = (300\,\mathrm{m}, 0.15\,\mathrm{m}), \quad (L_2, D_2) = (200\,\mathrm{m}, 0.10\,\mathrm{m}), \quad (L_3, D_3) = (300\,\mathrm{m}, 0.10\,\mathrm{m}),$$
$$(L_4, D_4) = (200\,\mathrm{m}, 0.15\,\mathrm{m}), \quad (L_5, D_5) = (360\,\mathrm{m}, 0.10\,\mathrm{m}).$$
- This gives the following K coefficients:
$$K_1 = 6528.542, \quad K_2 = 33050.743, \quad K_3 = 49576.114 \quad K_4 = 4352.361, \quad K_5 = 59491.337.$$

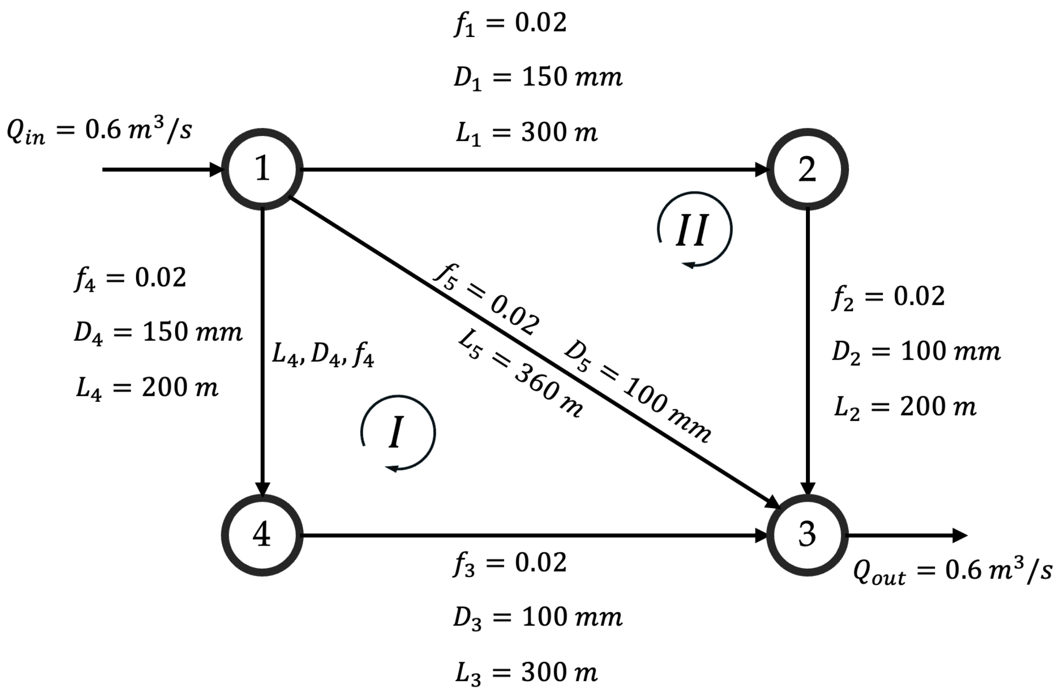


Figure 2: Four-node network with data.

Loop definitions and signs We consider two independent loops:

- Loop I: triangle $1 \to 4 \to 3 \to 1$ with members $(4,3) \equiv 3$, $(1,4) \equiv 4$, $(1,3) \equiv 5$. Signs relative to the pipe directions:
$$s_3 = -1, \quad s_4 = -1, \quad s_5 = +1.$$
- Loop II: triangle $1 \to 2 \to 3 \to 1$ with members $(1,2) \equiv 1$, $(2,3) \equiv 2$, $(1,3) \equiv 5$. Signs:
$$s_1 = +1, \quad s_2 = +1, \quad s_5 = -1.$$

A positive correction increases the flow in the direction of the loop for pipes with $s_i = +1$ and decreases it for $s_i = -1$.

Initialization Choose initial flows that satisfy continuity at all nodes:

$$Q_1^{(0)} = 0.100, \quad Q_2^{(0)} = 0.100, \quad Q_3^{(0)} = 0.400, \quad Q_4^{(0)} = 0.400, \quad Q_5^{(0)} = 0.100 \quad (\mathrm{m}^3/\mathrm{s}).$$

Loop correction (Darcy–Weisbach, $m = 2$) For the current loop flows compute each head loss $h_i = K_i Q_i |Q_i|$. The oriented loop sum is

$$\sum h_f = \sum_{i \in L} s_i \, h_i,$$

and the Hardy Cross correction is

$$\Delta Q = -\frac{\sum h_f}{2 \sum_{i \in L} \left| h_i / Q_i \right|}.$$

Table 2: Hardy Cross iterations for the four-node, two-loop network.

iter	ΔQ_I	ΔQ_{II}	Q_1	Q_2	Q_3	Q_4	Q_5
0	–	–	0.1000	0.1000	0.4000	0.4000	0.1000
1	0.14596	0.01005	0.1100	0.1100	0.2540	0.2540	0.2359
2	0.00306	0.07698	0.1870	0.1870	0.2510	0.2510	0.1620
3	0.03962	0.00518	0.1922	0.1922	0.2114	0.2114	0.1964
4	0.00247	0.02159	0.2138	0.2138	0.2089	0.2089	0.1773
5	0.01108	0.00160	0.2154	0.2154	0.1978	0.1978	0.1868
6	0.00081	0.00608	0.2215	0.2215	0.1970	0.1970	0.1815

$$\Delta Q_I = -\frac{k_3 Q_3|Q_3| + k_4 Q_4|Q_4| + k_5 Q_5|Q_5|}{2k_3|Q_3| + 2k_4|Q_4| + 2k_5|Q_5|}$$

$$\Delta Q_{II} = -\frac{k_1 Q_1|Q_1| + k_2 Q_2|Q_2| + k_5 Q_5|Q_5|}{2k_1|Q_1| + 2k_2|Q_2| + 2k_5|Q_5|}$$

Apply this same ΔQ to all members of the loop:

$$Q_i \leftarrow Q_i + s_i\,\Delta Q \qquad \forall\, i \in L.$$

For the common pipe 5, both loop corrections contribute with opposite signs:

$$Q_5^{\text{new}} = Q_5^{\text{old}} + s_5^{(\text{I})}\,\Delta Q_I + s_5^{(\text{II})}\,\Delta Q_{II} = Q_5^{\text{old}} + \Delta Q_I - \Delta Q_{II}.$$

Numerical iterations Table 2 collects the two-loop corrections per iteration and the updated pipe flows. The converged values are:

Converged flows

$$Q_1 = 0.2215, \quad Q_2 = 0.2215, \quad Q_3 = 0.1970, \quad Q_4 = 0.1970, \quad Q_5 = 0.1815 \quad (\text{m}^3/\text{s}).$$

Loop checks With these flows,

$$K_5 Q_5|Q_5| - K_3 Q_3|Q_3| - K_4 Q_4|Q_4| \approx 0, \qquad K_1 Q_1|Q_1| + K_2 Q_2|Q_2| - K_5 Q_5|Q_5| \approx 0,$$

which confirms that the oriented head-loss sums in both loops are practically zero. Node continuity remains satisfied by construction at every iteration.

2.2.3 Note on choosing a head-loss formula

- **Use Hazen–Williams when** Potable water at near-ambient temperature. Fully turbulent regime in most pipes. Quick design or teaching where simplicity is preferred. Roughness is represented by a single coefficient C that you can calibrate from field data. Valid mainly for water and for steady conditions.
- **Prefer Darcy–Weisbach when** Accuracy matters or conditions depart from the typical water-at-room-temperature case. Fluids with different viscosity or temperature. Transitional or possibly laminar segments due to low velocities or large diameters. Scenarios with wide flow variations such as fire flows. Explicit inclusion of minor losses with K coefficients. Works with any fluid and any Reynolds number.

- **Practical tip** If you use Darcy–Weisbach, either keep f fixed per pipe during each iteration or update it from Reynolds number and relative roughness using Colebrook or an explicit approximation such as Swamee–Jain. If results are sensitivity-critical, try both formulas and compare head losses and pressures.

2.2.4 Pros and cons relative to Newton–Raphson

Advantages

- Simple hand calculations on small networks.
- Intuitive loop corrections that build physical insight.
- No matrix assembly.

Limitations

- Slow convergence on large or tightly coupled networks and sensitive to initial flows.
- Loop-by-loop updates do not handle pumps, valves, and pressure-dependent demands as naturally as head-based solvers.
- Not well suited for automation and large systems where Newton–Raphson converges faster and more robustly.

Hardy Cross remains useful for checking small networks by hand. In the next section, we solve the same four-node problem with Newton–Raphson and compare the number of iterations and the ease of setup.

2.3 The Newton–Raphson Method

The Newton–Raphson method is a general iterative technique for solving systems of nonlinear equations. In pipe networks it enforces, simultaneously, the two conservation principles: (i) *mass conservation* at nodes and (ii) *energy conservation* in loops.

Unlike the Hardy Cross method, which balances loops sequentially, Newton–Raphson treats the entire network at once. We adopt a flows-only formulation, so the unknown vector is

$$\mathbf{Q} = \big[\, Q_1,\, Q_2,\, \ldots,\, Q_n \,\big]^T,$$

and the residual vector is

$$\mathbf{F}(\mathbf{Q}) = \big[\, F_1(\mathbf{Q}),\, F_2(\mathbf{Q}),\, \ldots,\, F_m(\mathbf{Q}) \,\big]^T,$$

where components of $\mathbf{F}$ represent nodal continuity equations and loop energy balances. Starting from an initial guess $\mathbf{Q}^{(0)}$, the method updates the solution until all residuals fall below a prescribed tolerance.

At iteration k, we expand $\mathbf{F}$ around $\mathbf{Q}^{(k)}$:

$$\mathbf{F}(\mathbf{Q}^{(k)} + \Delta\mathbf{Q}) \approx \mathbf{F}(\mathbf{Q}^{(k)}) + \mathbf{J}(\mathbf{Q}^{(k)})\,\Delta\mathbf{Q}, \qquad \mathbf{J}(\mathbf{Q}) = \frac{\partial \mathbf{F}}{\partial \mathbf{Q}}.$$

Imposing $\mathbf{F}(\mathbf{Q}^{(k)} + \Delta\mathbf{Q}) = \mathbf{0}$ gives the Newton system

$$\mathbf{J}(\mathbf{Q}^{(k)})\,\Delta\mathbf{Q} = -\,\mathbf{F}(\mathbf{Q}^{(k)}), \qquad \mathbf{Q}^{(k+1)} = \mathbf{Q}^{(k)} + \Delta\mathbf{Q}.$$

2.3.1 Step-by-Step Procedure

The Newton–Raphson iteration for nonlinear systems in pipe networks can be summarized as follows:

1. **Initialization** Select initial guesses for the unknown flows $\mathbf{Q}^{(0)}$.
2. **Form the residuals** At iteration k, evaluate

   $$\mathbf{F}(\mathbf{Q}^{(k)}) = \big[\, F_1(\mathbf{Q}^{(k)}),\, F_2(\mathbf{Q}^{(k)}),\, \ldots,\, F_m(\mathbf{Q}^{(k)}) \,\big]^T,$$

   where each residual F_i represents either a node continuity or a loop energy balance.
3. **Assemble the Jacobian** Construct

   $$\mathbf{J}(\mathbf{Q}^{(k)}) = \begin{bmatrix} \dfrac{\partial F_1}{\partial Q_1} & \cdots & \dfrac{\partial F_1}{\partial Q_n} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial F_m}{\partial Q_1} & \cdots & \dfrac{\partial F_m}{\partial Q_n} \end{bmatrix}_{\mathbf{Q}^{(k)}},$$

   by differentiating each residual with respect to each unknown.
4. **Solve the Newton system** Solve the linear system

   $$\mathbf{J}(\mathbf{Q}^{(k)})\,\Delta\mathbf{Q} = -\,\mathbf{F}(\mathbf{Q}^{(k)}).$$

   Formally, $\Delta\mathbf{Q} = -\,\mathbf{J}(\mathbf{Q}^{(k)})^{-1}\,\mathbf{F}(\mathbf{Q}^{(k)})$, but in practice one does not form the inverse; the system $\mathbf{J}\Delta\mathbf{Q} = -\mathbf{F}$ is solved by Gaussian elimination or by matrix factorization. For small networks the inverse can be shown for pedagogy, but for realistic networks the Jacobian is large and sparse, so direct sparse or iterative solvers are used.

5. **Update the solution** Apply the correction

$$\mathbf{Q}^{(k+1)} = \mathbf{Q}^{(k)} + \Delta\mathbf{Q}.$$

6. **Convergence test** Check whether all residuals satisfy $|F_i(\mathbf{Q}^{(k+1)})| \leq \text{tol}_F$ and all corrections satisfy $|\Delta Q_i| \leq \text{tol}_Q$. If both criteria hold, stop; otherwise return to Step 2.

Each iteration alternates between computing $\mathbf{F}$, assembling $\mathbf{J}$, solving for $\Delta\mathbf{Q}$, and updating the flows. Compared with Hardy Cross, this formulation is computationally more intensive per iteration (matrix assembly and solve) yet it achieves quadratic local convergence and scales well to large networks. This is why Newton–Raphson is the backbone of modern hydraulic solvers such as EPANET.

2.3.2 Worked example. Four-node network with two loops

Data We solve again the system shown in Figure 2 but now by using Newton–Raphson.

Prescribed nodal terms are $D_1 = 0.6\ \mathrm{m}^3/\mathrm{s}$ and $D_3 = 0.6\ \mathrm{m}^3/\mathrm{s}$ (appearing with the signs shown in the residuals). Pipe lengths and diameters are

$$(L_1, d_1) = (300\,\mathrm{m},\ 0.15\,\mathrm{m}), \quad (L_2, d_2) = (200\,\mathrm{m},\ 0.10\,\mathrm{m}), \quad (L_3, d_3) = (300\,\mathrm{m},\ 0.10\,\mathrm{m}),$$

$$(L_4, d_4) = (200\,\mathrm{m},\ 0.15\,\mathrm{m}), \quad (L_5, d_5) = (360\,\mathrm{m},\ 0.10\,\mathrm{m}).$$

With friction factor $f = 0.02$ and $g = 9.81\ \mathrm{m/s^2}$, the quadratic head-loss coefficients are

$$K_1 = 6528.542, \quad K_2 = 33050.743, \quad K_3 = 49576.114, \quad K_4 = 4352.361, \quad K_5 = 59491.337.$$

Residual equations Continuity at the nodes (with pipe orientations as in the figure):

$$\begin{aligned} F_1 &= -Q_1 - Q_4 - Q_5 + D_1, \\ F_2 &= \quad Q_1 - Q_2, \\ F_3 &= \quad Q_2 + Q_3 + Q_5 - D_3. \end{aligned}$$

Loop energy balances:

$$\begin{aligned} F_4 &= K_5\,Q_5|Q_5| \ - \ K_3\,Q_3|Q_3| \ - \ K_4\,Q_4|Q_4|, \\ F_5 &= K_1\,Q_1|Q_1| \ + \ K_2\,Q_2|Q_2| \ - \ K_5\,Q_5|Q_5|. \end{aligned}$$

Note. There is no separate continuity equation for node 4 because it would be linearly dependent on the others. In a network with N nodes and L independent loops, the number of independent continuity equations is $N-1$. Together with the L loop energy equations, this gives exactly $N-1+L$ equations for the same number of unknown pipe flows.

Jacobian The Jacobian is

$$\mathbf{J}(\mathbf{Q}) = \begin{bmatrix} -1 & 0 & 0 & -1 & -1 \\ 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & -2K_3|Q_3| & -2K_4|Q_4| & 2K_5|Q_5| \\ 2K_1|Q_1| & 2K_2|Q_2| & 0 & 0 & -2K_5|Q_5| \end{bmatrix}.$$

Newton system At each iteration,

$$\mathbf{J}(\mathbf{Q}^{(k)})\,\Delta\mathbf{Q} = -\,\mathbf{F}(\mathbf{Q}^{(k)}), \qquad \mathbf{Q}^{(k+1)} = \mathbf{Q}^{(k)} + \Delta\mathbf{Q}.$$

Initialization Choose flows satisfying approximate continuity:

$$Q_1^{(0)} = 0.10, \quad Q_2^{(0)} = 0.10, \quad Q_3^{(0)} = 0.40, \quad Q_4^{(0)} = 0.40, \quad Q_5^{(0)} = 0.10.$$

Iterations The derivatives, residuals, linear solve, and flow corrections are computed at each iteration. Below we show Iteration 1 in detail, followed by a short summary table.

Iteration 1 in detail

- **Initial flows.** We take

$$Q_1^{(0)} = 0.10, \quad Q_2^{(0)} = 0.10, \quad Q_3^{(0)} = 0.40, \quad Q_4^{(0)} = 0.40, \quad Q_5^{(0)} = 0.10.$$

- **Residual vector**

$$\mathbf{F}(\mathbf{Q}^{(0)}) = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -8033.64 \\ -199.12 \end{bmatrix},$$

with

$$F_4 = K_5Q_5|Q_5| - K_3Q_3|Q_3| - K_4Q_4|Q_4|, \qquad F_5 = K_1Q_1|Q_1| + K_2Q_2|Q_2| - K_5Q_5|Q_5|.$$

- **Jacobian at $\mathbf{Q}^{(0)}$**

$$\mathbf{J}(\mathbf{Q}^{(0)}) = \begin{bmatrix} -1 & 0 & 0 & -1 & -1 \\ 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & -39660.89 & -3481.89 & 11898.27 \\ 1305.71 & 6610.15 & 0 & 0 & -11898.27 \end{bmatrix}.$$

- **Inverse-Jacobian action (for illustration only)** For teaching clarity we display the inverse Jacobian numerically at this iterate:

$$\mathbf{J}(\mathbf{Q}^{(0)})^{-1} \approx \begin{bmatrix} -0.04365 & 0.88062 & 0.49724 & 1.25\times10^{-5} & 5.80\times10^{-5} \\ -0.04365 & -0.11938 & 0.49724 & 1.25\times10^{-5} & 5.80\times10^{-5} \\ 0.07270 & 0.08907 & 0.17194 & -2.09\times10^{-5} & -1.25\times10^{-5} \\ -0.92730 & -0.91093 & -0.82806 & -2.09\times10^{-5} & -1.25\times10^{-5} \\ -0.02904 & 0.03032 & 0.33081 & 8.34\times10^{-6} & -4.55\times10^{-5} \end{bmatrix}.$$

  Then

$$\Delta\mathbf{Q} = -\,\mathbf{J}(\mathbf{Q}^{(0)})^{-1}\,\mathbf{F}(\mathbf{Q}^{(0)}) \approx \begin{bmatrix} 0.11227 \\ 0.11227 \\ -0.17023 \\ -0.17023 \\ 0.05796 \end{bmatrix}.$$

  Rather than forming the inverse, one should solve $\mathbf{J}\,\Delta\mathbf{Q} = -\mathbf{F}$ by Gaussian elimination; see Appendix A.

- **Correction and update**

$$\mathbf{Q}^{(0)} = \begin{bmatrix} 0.10 \\ 0.10 \\ 0.40 \\ 0.40 \\ 0.10 \end{bmatrix}, \qquad \Delta\mathbf{Q} \approx \begin{bmatrix} 0.11227 \\ 0.11227 \\ -0.17023 \\ -0.17023 \\ 0.05796 \end{bmatrix}, \qquad \mathbf{Q}^{(1)} = \mathbf{Q}^{(0)} + \Delta\mathbf{Q} \approx \begin{bmatrix} 0.21227 \\ 0.21227 \\ 0.22977 \\ 0.22977 \\ 0.15796 \end{bmatrix}.$$

Table 3: Newton–Raphson iterations for the four-node, two-loop network.

Iteration	Q_1	Q_2	Q_3	Q_4	Q_5
0 (initial)	0.1000	0.1000	0.4000	0.4000	0.1000
1	0.2123	0.2123	0.2298	0.2298	0.1580
2	0.2228	0.2228	0.1940	0.1940	0.1833
3	0.2245	0.2245	0.1923	0.1923	0.1831
4	0.2245	0.2245	0.1923	0.1923	0.1831

The converged values obtained are:

Converged flows

$$Q_1 = 0.2245, \qquad Q_2 = 0.2245, \qquad Q_3 = 0.1923, \qquad Q_4 = 0.1923, \qquad Q_5 = 0.1831 \quad (\mathrm{m}^3/\mathrm{s}).$$

Verification At convergence the residuals F_i are practically zero. The loop energy balances F_4, F_5 confirm that the algebraic sums of head losses vanish, and node continuity is satisfied by construction.

Comparison with Hardy Cross. The Newton–Raphson solution does not exactly match the flows reported with Hardy Cross, although the gaps are small. Both computations use the same data and the same head-loss law, so the difference comes from how the iterations proceed and where they stop, not from different physics. Hardy Cross updates one loop at a time with a uniform correction, which preserves continuity but leaves small residuals in pipes shared by two loops if the process is stopped early. Newton–Raphson solves all unknowns simultaneously by linearizing the full system, so cross-loop residuals are reduced faster and the final numbers reflect the tighter stopping criteria used here. In practice, if both methods are iterated until node continuity and loop energy balances meet the same stringent tolerances, the two solutions coincide within numerical precision.

Why the numbers differ:

- Different stopping criteria and early rounding keep small residuals in Hardy Cross.
- Sequential loop corrections couple through shared pipes, which slows the last digits of convergence.
- Newton–Raphson reduces all residuals at once, so it reaches a tighter joint balance with fewer steps.

3 EPANET

EPANET is a specialized software for the simulation of hydraulic and water quality behavior in pressurized pipe networks. A typical network includes elements such as pipes, junctions, pumps, valves, tanks, and reservoirs. The program computes flows, pressures, and water quality parameters over time, allowing both steady-state and extended-period simulations. It also tracks water age and source contributions, as well as the transport and decay of chemical species.

Originally developed by the U.S. Environmental Protection Agency, EPANET was conceived as a research tool but has become a standard in engineering practice. Its applications range from the calibration of hydraulic models and evaluation of chlorine residuals to the design of operational strategies and assessment of consumer exposure. By simulating scenarios such as pumping schedules, tank operation rules, and targeted pipe rehabilitation, EPANET helps evaluate alternative management strategies for improving system performance.

The software includes a robust hydraulic engine capable of handling networks of any size, with flexibility to model head losses, pump operations, valves, emitters, and variable demands. It also incorporates energy and cost analysis. On the water quality side, it can represent bulk and wall reactions, contaminant propagation, and tank mixing conditions, providing a comprehensive framework to study the physical and chemical processes that occur within water distribution systems.

It should be noted, however, that EPANET focuses on quasi-steady hydraulic simulations and does not account for fast transients such as *water hammer*, surge waves, or other unsteady phenomena. For these advanced problems, specialized software (e.g. Bentley HAMMER, InfoSurge, or custom transient models) must be employed.

3.1 Installing EPANET

To install EPANET on your personal computer:

Supported Operating Systems

- **Windows**: Compatible with all recent versions (Windows 7, 8, 10, 11).
- **macOS / Linux**: No native version available, but it can be run using `Wine`, `PlayOnMac`, or a virtual machine if needed.

Installation Steps

1. Go to the official website: https://www.epa.gov/water-research/epanet
2. Download the **EPANET 2.2** installer for Windows.
3. Run the installer and follow the on-screen instructions.

Alternative: Use EPFL's VDI Platform

If you prefer not to install EPANET or your device is not compatible, you can use the pre-installed version available on EPFL's virtual desktop:

1. Go to https://vdi.epfl.ch and log in with your **Gaspar** credentials.
2. Select the environment **"ENAC-SGC"**.
3. On the virtual desktop, open the folder **"Program GC"** and launch **EPANET** from there.

3.2 Main Elements in EPANET Networks

EPANET represents a distribution system as a set of *nodes* (junctions, reservoirs, tanks) connected by *links* (pipes, pumps, valves). Some nodes can also include emitters.

3.3 Elements of an EPANET Network

Figure 3 illustrates the main components that can be represented in an EPANET model. Each of them plays a specific role in the hydraulic and water quality behavior of the system:

- **Reservoirs**: Boundary nodes that provide a fixed hydraulic head, representing lakes, rivers, or external systems. They can also act as quality source points.
- **Tanks**: Storage nodes with variable volume, characterized by bottom elevation, diameter or shape, and minimum/maximum levels. They regulate pressure, provide storage, and can act as water quality sources.
- **Junctions**: Connection points where two or more links meet. They are defined by elevation, base demand, and initial water quality. Junctions can support multiple demand categories, negative demands, time patterns, or emitters.
- **Pipes**: Links that convey water between nodes. They are defined by diameter, length, roughness, and status (open, closed, or with check valve). Head losses are computed with Hazen–Williams, Darcy–Weisbach, or Chezy–Manning formulas.
- **Pumps**: Links that add energy to the system, defined by their pump curve (head–flow relationship) or as constant energy devices. They can also be variable speed pumps and allow energy cost calculations.
- **Valves**: Links that regulate flow or pressure. EPANET includes several types such as Pressure Reducing Valves (PRV), Pressure Sustaining Valves (PSV), Flow Control Valves (FCV), and General Purpose Valves (GPV).

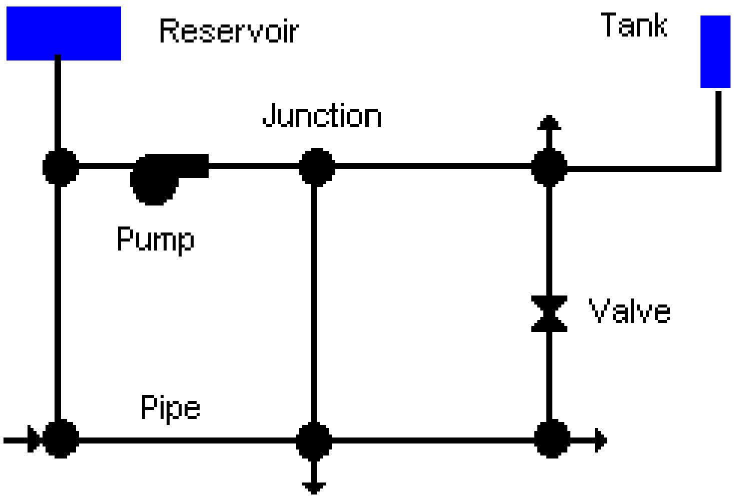


Figure 3: Main components of an EPANET network: reservoirs, tanks, junctions, pipes, pumps, and valves.

3.4 Non-Physical Components in EPANET

In addition to physical elements, EPANET uses informational objects to describe operational rules and system behavior over time. These components do not correspond to physical infrastructure but are essential for defining realistic network operation:

- **Curves** Data pairs representing relationships between two variables. Types include pump curves (head–flow), efficiency curves, tank volume curves, and headloss curves for GPVs. Curves can be shared among multiple objects.
- **Time Patterns** Collections of multipliers applied to quantities such as demands, reservoir heads, pump schedules, or water quality inputs to allow them to vary over time. They are defined by a fixed time interval and can repeat in cycles.
- **Controls** Rules for operating links (pipes, pumps, valves) based on time, tank levels, or nodal pressures. Two types are available: simple controls (if–then statements) and rule-based controls (logical combinations of conditions).

3.5 Tutorial Example

In this tutorial we analyze the simple distribution network shown in Fig. 4 below. It consists of a source reservoir (e.g., a treatment plant clearwell) from which water is pumped into a two-loop pipe network. There is also a pipe leading to a storage tank that floats on the system. The ID labels for the various components are shown in the figure. The nodes in the network have the characteristics shown in Table 4. Pipe properties are listed in Table 5. In addition, the pump (Link 9) delivers a head of $45.72\,\mathrm{m}$ at a flow of $37.85\,\mathrm{L/s}$. The tank (Node 8) has diameter $18.29\,\mathrm{m}$, initial level $1.07\,\mathrm{m}$, and maximum level $6.10\,\mathrm{m}$.

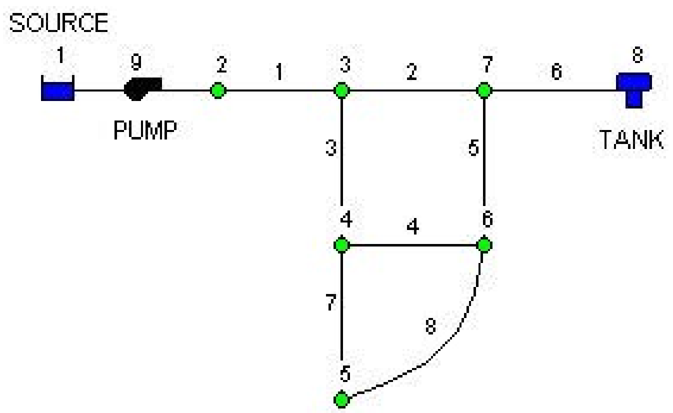


Figure 4: Example pipe network.

3.5.1 Project Setup

The first task is to create a new project in EPANET and make sure that certain default options are selected. To begin, launch EPANET, or if it is already running select `File » New` (from the menu bar) to create a new project. Then select `Project » Defaults` to open the dialog form shown in Fig. 5. We will use this dialog to have EPANET automatically label new objects with consecutive numbers starting from 1 as they are added to the network. On the ID Labels page of the dialog, clear all of the ID Prefix fields and set the ID Increment to 1. Then select the Hydraulics page of the dialog and set the choice of Flow Units to LPS (liters per second). This implies that International Customary units will be used for all other quantities as well. Also select Hazen - Williams (H-W) as the headloss formula. If you want to save these choices for all future new projects you could check the Save box at the bottom of the form before accepting it by clicking the OK button.

Table 4: Node properties

Node	Elevation (m)	Demand (L/s)
1	213.36	0
2	213.36	0
3	216.41	9.46
4	213.36	9.46
5	198.12	12.62
6	213.36	9.46
7	213.36	0
8	252.98	0

Table 5: Pipe properties

Pipe	Length (m)	Diameter (m)	C-Factor
1	914.40	0.3556	100
2	1524.00	0.3048	100
3	1524.00	0.2032	100
4	1524.00	0.2032	100
5	1524.00	0.2032	100
6	2133.60	0.2540	100
7	1524.00	0.1524	100
8	2133.60	0.1524	100

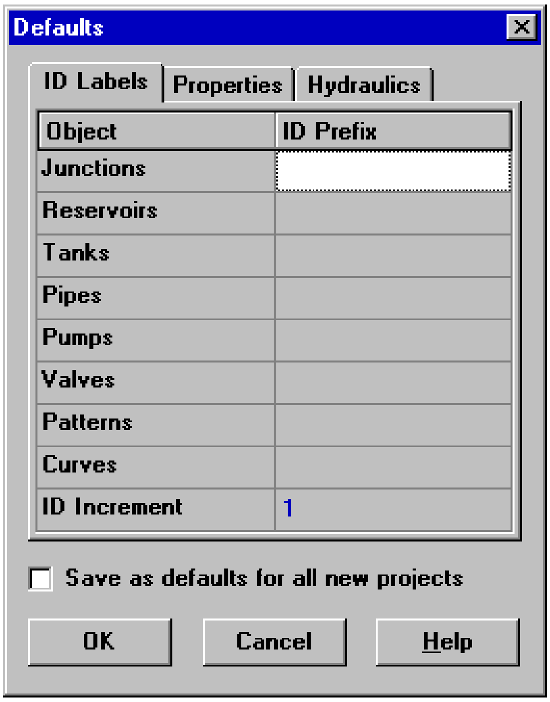


Figure 5: Project defaults dialog.

Next we will select some map display options so that as we add objects to the map, we will see their ID labels and symbols displayed. Select `View » Options` to bring up the Map Options dialog form. Select the Notation page on this form and check the settings shown in Fig. 6 below. Then switch to the Symbols page and check all of the boxes. Click the OK button to accept these choices and close the dialog.

Finally, before drawing our network we should insure that our map scale settings are acceptable. Select `View » Dimensions` to bring up the Map Dimensions dialog. Note the default dimensions assigned for a new project. These settings will suffice for this example, so click the OK button.

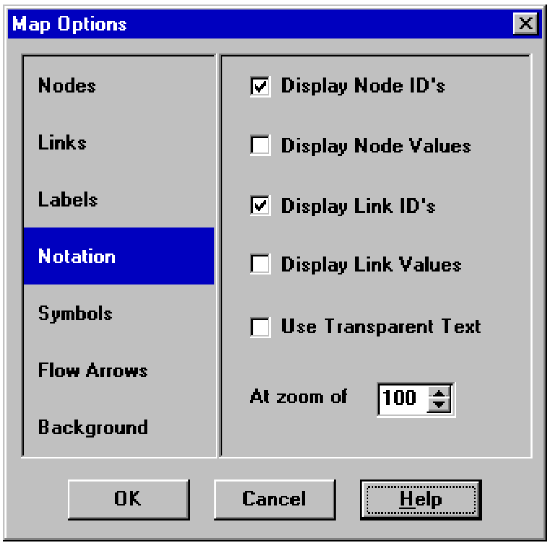


Figure 6: Map options dialog.

3.5.2 Drawing the Network

We are now ready to begin drawing our network by making use of our mouse and the buttons contained on the Map Toolbar shown below. (If the toolbar is not visible then select `View » Toolbars » Map`).

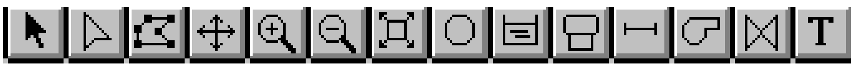


Figure 7: Map toolbar.

First we will add the reservoir. Click the Reservoir button. Then click the mouse on the map at the location of the reservoir (somewhere to the left of the map).

Next we will add the junction nodes. Click the Junction button and then click on the map at the locations of nodes 2 through 7.

Finally add the tank by clicking the Tank button and clicking the map where the tank is located. At this point the Network Map should look something like the drawing in Fig. 8.

Next we will add the pipes. Let's begin with pipe 1 connecting node 2 to node 3. First click the Pipe button on the Toolbar. Then click the mouse on node 2 on the map and then on node 3. Note how an outline of the pipe is drawn as you move the mouse from node 2 to 3. Repeat this procedure for pipes 2 through 7.

Pipe 8 is curved. To draw it, click the mouse first on Node 5. Then, as you move the mouse towards Node 6, click at those points where a change of direction is needed to maintain the desired shape. Complete the process by clicking on Node 6.

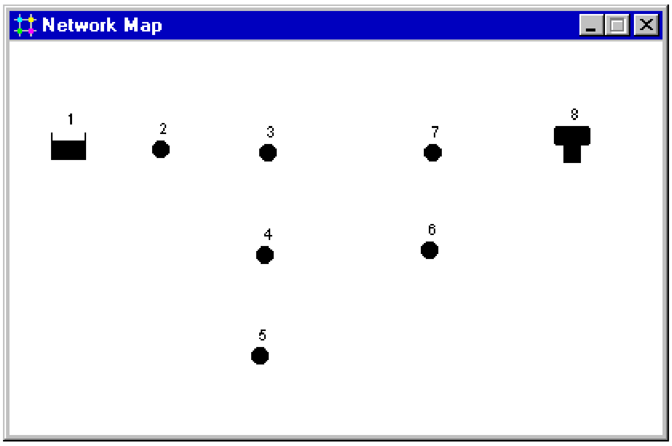


Figure 8: Network map after adding nodes.

Finally we will add the pump. Click the Pump button, click on node 1 and then on node 2.

Next we will label the reservoir, pump and tank. Select the Text button on the Map Toolbar and click somewhere close to the reservoir (Node 1). An edit box will appear. Type in the word SOURCE and then hit the Enter key. Click next to the pump and enter its label, then do the same for the tank. Then click the Selection button on the Toolbar to put the map into Object Selection mode rather than Text Insertion mode.

At this point, we have completed drawing the example network. Your Network Map should look like the map in Fig. 4. If the nodes are out of position, you can move them around by clicking the node to select it, and then dragging it with the left mouse button held down to its new position. Note how pipes connected to the node are moved along with the node. The labels can be repositioned in a similar fashion. To re-shape the curved Pipe 8:

1. First click on Pipe 8 to select it and then click the button on the Map Toolbar to put the map into Vertex Selection mode.
2. Select a vertex point on the pipe by clicking on it and then drag it to a new position with the left mouse button held down.
3. If required, vertices can be added or deleted from the pipe by right-clicking the mouse and selecting the appropriate option from the popup menu that appears.
4. When finished, click the black pointer button to return to Object Selection mode.

3.5.3 Setting Object Properties

As objects are added to a project they are assigned a default set of properties. To change the value of a specific property for an object one must select the object into the Property Editor (Fig. 9). There are several different ways to do this. If the Editor is already visible then you can simply click on the object or select it from the Data page of the Browser. If the Editor is not visible then you can make it appear by one of the following actions:

- Double-click the object on the map,
- Right-click on the object and select Properties from the pop-up menu that appears,
- Select the object from the Data page of the Browser window and then click the Browser's Edit button.

Whenever the Property Editor has the focus you can press the F1 key to obtain fuller descriptions of the properties listed

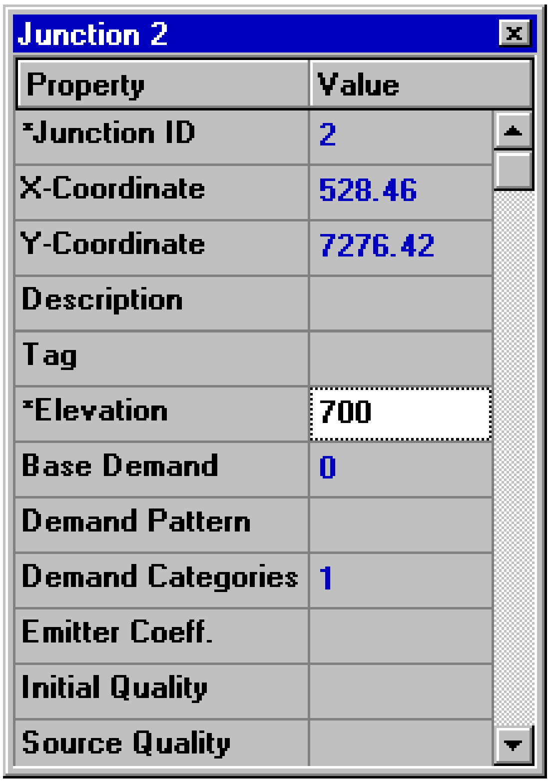


Figure 9: Property Editor.

Let us begin editing by selecting Node 2 into the Property Editor as shown above. We would now enter the elevation and demand for this node in the appropriate fields. You can use the Up and Down arrows on the keyboard or the mouse to move between fields. We need only click on another object (node or link) to have its properties appear next in the Property Editor. (We could also press the Page Down or Page Up key to move to the next or previous object of the same type in the database.) Thus we can simply move from object to object and fill in elevation and demand for nodes, and length, diameter, and roughness (C-factor) for links.

For the reservoir you would enter its elevation (213 m) in the Total Head field. For the tank, enter 253 m for its elevation, 1.1 m for its initial level, 6 m for its maximum level, and 18 m for its diameter. For the pump, we need to assign it a pump curve (head versus flow relationship). Enter the ID label 1 in the Pump Curve field.

Next we will create Pump Curve 1. From the Data page of the Browser window, select Curves from the dropdown list box and then click the Add button. A new Curve 1 will be added to the database and the Curve Editor dialog form will appear (see Fig. 10). Enter the pump's design flow (38 L/s) and head (490) into this form. EPANET automatically creates a complete pump curve from this single point. The curve's equation is shown along with its shape. Click OK to close the Editor.

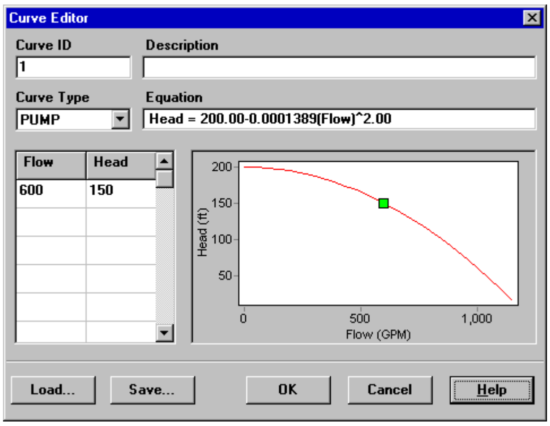


Figure 10: Curve editor.

3.5.4 Saving and Opening Projects

Save the project to a `.net` file. Export network text if desired. Use `File` » `Open` to reopen later.

Having completed the initial design of our network it is a good idea to save our work to a file at this point.

1. From the File menu select the Save As option.
2. In the Save As dialog that appears, select a folder and file name under which to save this project. We suggest naming the file tutorial.net. (An extension of .net will be added to the file name if one is not supplied.).
3. Click OK to save the project to file.

The project data is saved to the file in a special binary format. If you want to save the network data to the file as readable text, use the `File` » `Export` » `Network` command instead.

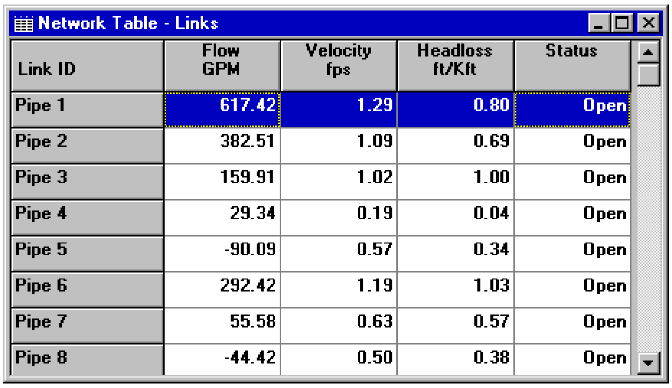
To open our project at some later time, we would select the 'Open' command from the File menu.

3.5.5 Running a Single Period Analysis

We now have enough information to run a single period (or snapshot) hydraulic analysis on our example network. To run the analysis select `Project` » `Run Analysis` or click the Run button on the Standard Toolbar. (If the toolbar is not visible select `View` » `Toolbars` » `Standard` from the menu bar).

If the run was unsuccessful, then a Status Report window will appear indicating what the problem was. If it ran successfully, you can view the computed results in a variety of ways. Try some of the following:

- Select Node Pressure from the Browser's Map page and observe how pressure values at the nodes become color-coded. To view the legend for the color-coding, select `View` » `Legends` » `Node` (or right-click on an empty portion of the map and select Node Legend from the popup menu). To change the legend intervals and colors, right-click on the legend to make the Legend Editor appear.
- Bring up the Property Editor (double-click on any node or link) and note how the computed results are displayed at the end of the property list.
- Create a tabular listing of results by selecting `Report` » `Table` (or by clicking the Table button on the Standard Toolbar). Fig. 11 displays such a table for the link results of this run. Note that flows with negative signs mean that the flow is in the opposite direction to the direction in which the pipe was drawn initially.



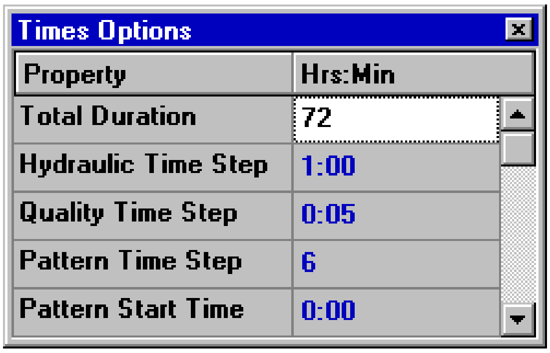
Network Table - Links

Link ID	Flow GPM	Velocity fps	Headloss ft/Kft	Status
Pipe 1	617.42	1.29	0.80	Open
Pipe 2	382.51	1.09	0.69	Open
Pipe 3	159.91	1.02	1.00	Open
Pipe 4	29.34	0.19	0.04	Open
Pipe 5	-90.09	0.57	0.34	Open
Pipe 6	292.42	1.19	1.03	Open
Pipe 7	55.58	0.63	0.57	Open
Pipe 8	-44.42	0.50	0.38	Open

Figure 11: Example of link results table.

3.5.6 Running an Extended Period Analysis

To make our network more realistic for analyzing an extended period of operation, we will create a Time Pattern that makes demands at the nodes vary in a periodic way over the course of a day. For this simple example, we will use a pattern time step of 6 hours, thus making demands change at four different times of the day. (A 1-hour pattern time step is a more typical number and is the default assigned to new projects.) We set the pattern time step by selecting Options-Times from the Data Browser, clicking the Browser's Edit button to make the Property Editor appear (if it's not already visible), and entering 6 for the value of the Pattern Time Step (as shown in Fig. 12 below). While we have the Time Options available, we can also set the duration for which we want the extended period to run. Let's use a 3-day period of time (enter 72 hours for the Duration property).



Times Options

Property	Hrs:Min
Total Duration	72
Hydraulic Time Step	1:00
Quality Time Step	0:05
Pattern Time Step	6
Pattern Start Time	0:00

Figure 12: Time options.

To create the pattern, select the Patterns category in the Browser and then click the Add button. A new Pattern 1 will be created and the Pattern Editor dialog should appear (see Fig. 13). Enter the multiplier values 0.5, 1.3, 1.0, 1.2 for the time periods 1 to 4 that will give our pattern a duration of 24 hours. The multipliers are used to modify the demand from its base level in each time period. Since we are making a run of 72 hours, the pattern will wrap around to the start after each 24-hour interval of time.

We now need to assign Pattern 1 to the Demand Pattern property of all of the junctions in our network. We can utilize one of EPANET's Hydraulic Options to avoid having to edit each junction individually. If you bring up the Hydraulic Options in the Property Editor you will see that there is an item called Default Pattern. Setting its value equal to 1 will make the Demand Pattern at each junction equal Pattern 1, as long as no other pattern is assigned to the junction.

Next run the analysis (select `Project » Run Analysis` or click the lightning button on the Standard Toolbar). For extended period analysis you have several more ways in which to view results:

- The scrollbar in the Browser's Time controls is used to display the network map at different

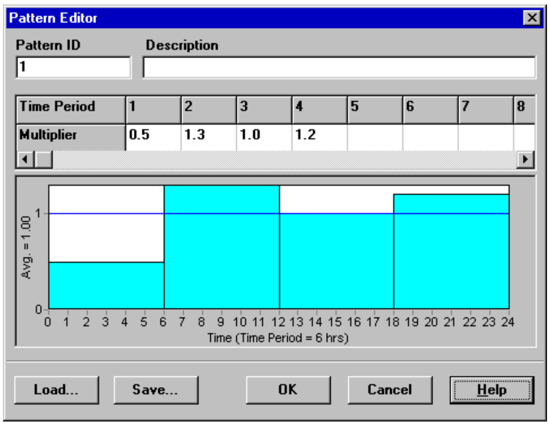


Figure 13: Pattern editor.

points in time. Try doing this with Pressure selected as the node parameter and Flow as the link parameter.

- The buttons in the Browser can animate the map through time. Click the Forward button image25 to start the animation and the Stop button image26 to stop it.
- Add flow direction arrows to the map (select View » Options, select the Flow Arrows page from the Map Options dialog, and check a style of arrow that you wish to use). Then begin the animation again and note the change in flow direction through the pipe connected to the tank as the tank fills and empties over time.
- Create a time series plot for any node or link. For example, to see how the water elevation in the tank changes with time:
  1. Click on the tank.
  2. Select `Report » Graph` (or click the Graph button on the Standard Toolbar) which will display a Graph Selection dialog box.
  3. Select the Time Series button on the dialog.
  4. Select Head as the parameter to plot.
  5. Click OK to accept your choice of graph.

Note the periodic behavior of the water elevation in the tank over time (Fig. 14).

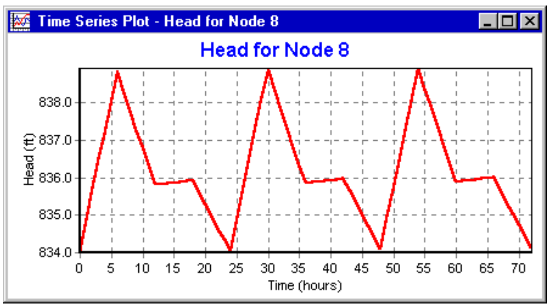
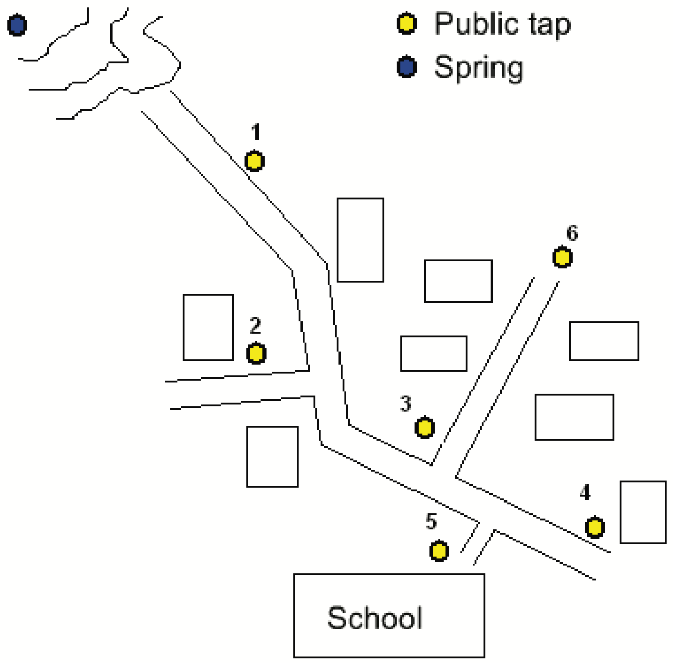


Figure 14: Example of time series plot.

4 Exercise

The small town of **Villars-le-Terroir** had been in need of a water supply system for some time. Traditionally, water was transported by donkey from a stream **6 km** away, but funds are now available to use a **spring** on the hill at an elevation of **36 m**. The flow is estimated at **3 l/s**.

A system is planned to supply **6 public fountains**, all of them at an elevation of **17 m**, except for fountain **6 (22 m)** and fountain **1 (25 m)**, according to the following sketch:



Distances

- Spring → Fountain 1: 800 m
- 1 → 2: 400 m
- 2 → 3: 300 m
- 3 → 4: 250 m
- 3 → 6: 500 m
- 5 → Tee: 200 m

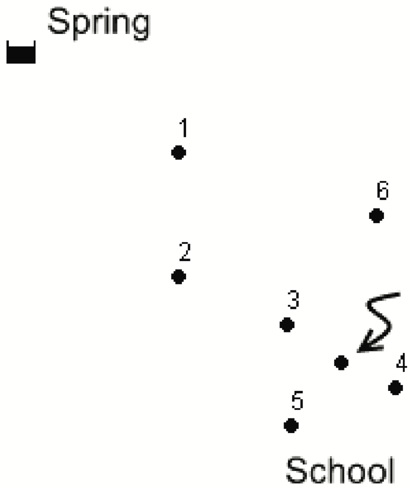
The system will use **PVC pipes** to supply **0.2 l/s** to each fountain and **1.0 l/s** to the school, maintaining a **minimum pressure of 10 m** at all points. Find the minimum diameter possible that keeps pressure over 10 m.

Table 6: List of available diameters for PVC and HDPE. ND: nominal diameter. We will use the "ID: internal diameter" for EPANET.

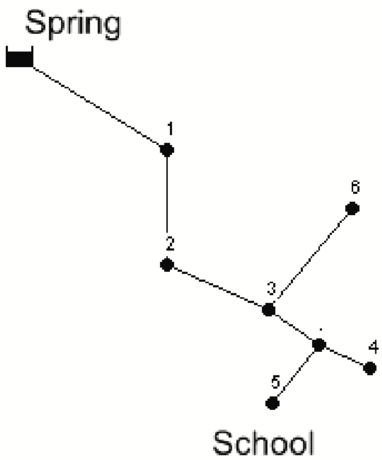
ND	25	32	40	50	63	75	90	110	125	140	160	180	200	250	315	400	450	500
ID HDPE	20	26	35	44	55	66	79	97	110	123	141	159	176	220	277	353	397	462
ID PVC	21	29	36	45	57	68	81	102	115	129	148	159	185	231	291	369	–	462

Instructions

1. Configure the default values in a way that saves you the most work.
2. Draw the nodes of the network.
3. Node **5** will be connected to an intermediate point on the pipe that runs between **3** and **4**. To represent this, you should draw a node without demand that corresponds to a **T-junction**. In the figure below, this T is shown unearthed and as an extra point in the Epanet plan:



4. Connect the nodes with pipes in the way that seems most logical and uses the least material. Here, it seems logical to follow the main road. The length of section **3-4** should be divided approximately in proportion to the diagram or real distances. For example: **150 m** for the left-hand side pipe and **100 m** for the right-hand one, totaling **250 m**.

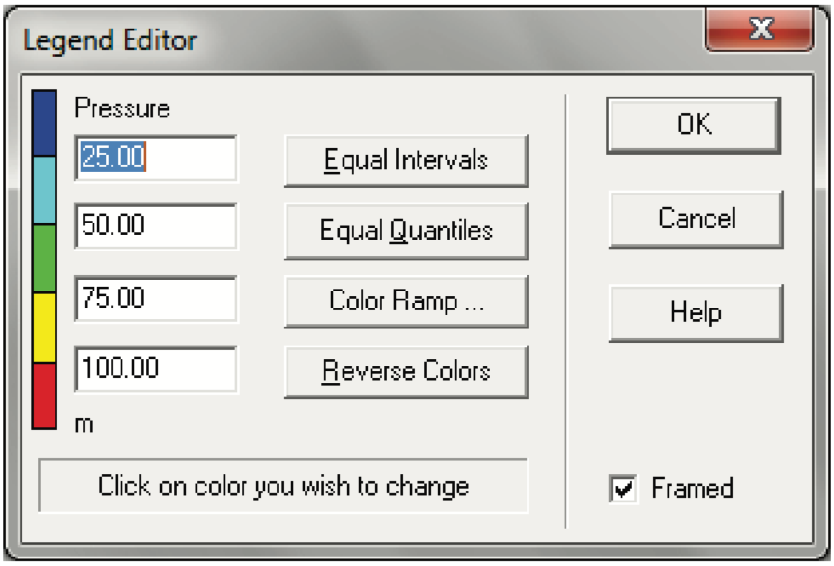


5. Enter the data for **elevations, lengths, demands, and frictions** where there are no default values. PVC roughness ranges between **140** and **150**.

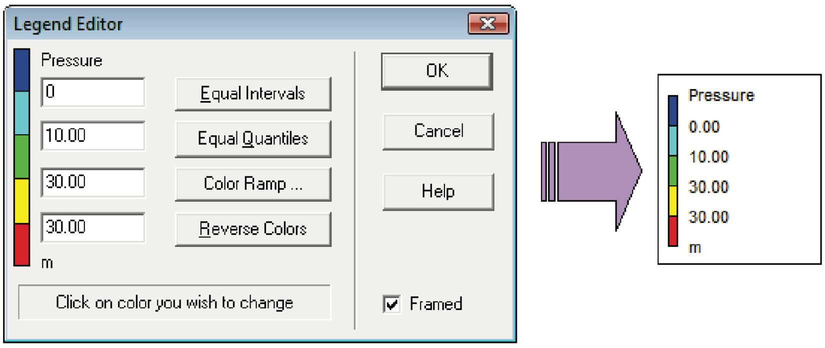
6. Calculate the network by clicking the **lightning icon**. Most likely, you will see a message saying `Valid Simulation`, meaning that the pipes are sufficiently large. But be careful: **"sufficiently large"** can mean any diameter from the smallest one that works up to the size of the Solar System!

7. Display the pressure results by selecting **Pressure** in the Node drop-down menu in the **Map** tab of the browser.

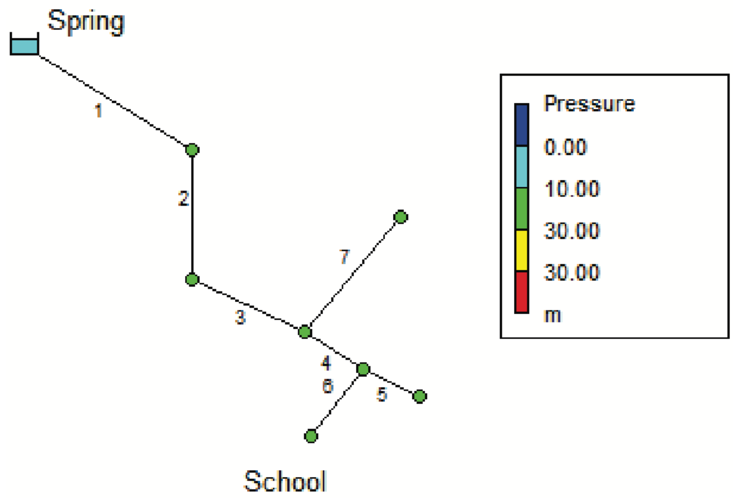
8. Change the scale of the legend to see the results more clearly. To open the dialogue that allows you to do this, right-click on the legend. Double left-clicking makes the legend disappear. To see it again: `View > Legends > Node`.



In this legend, for example, the additional yellow scale has been ignored:



When we have changed the legend the diagram of the screen is brought up to date:

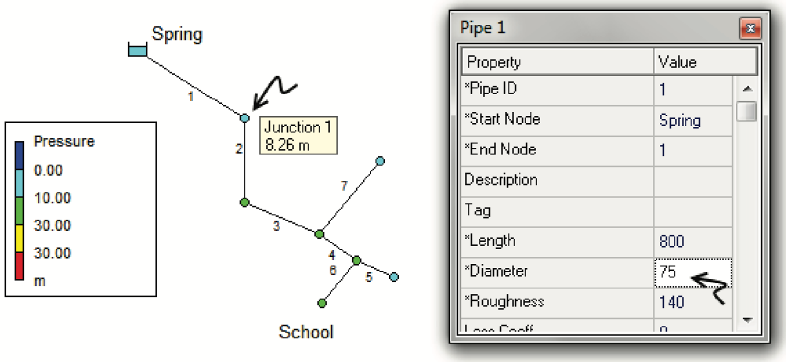


This system gives you all the correct values of pressure. But remember that the work is not finished yet; first you must carry out an optimization. Verify that if you change the pipe that runs from the spring to fountain 1 to one with a diameter of 1 km (1.000.000 mm), the system continues to come out correct in spite of the fact that it is complete nonsense. Imagine how much a pipe of 1 km of diameter would cost? Would it even be buildable? Do not forget that the notice "Valid Simulation" is only an invitation to optimize the system and not a green light on the part of Epanet for design.

9. You must therefore reduce the diameter of the pipes to the minimum that maintains the pressure in all the points above **10 m**. The first changes, by way of example, are described in the following points, but before this there is an important notice on the philosophy:

   The most logical is to start with the zones closest to the source of water. If you begin with those farthest away, you will see that the changes that you later make to the pipe diameter near the source will alter the results of those you had so carefully optimized previously. This will result in a never ending spiral of changes.

10. Start by changing the diameter of the pipe from the spring to fountain 1 to **75 mm** and run the calculation. If fountain 1 shows a pressure of **8.26 m**, it does not reach the minimum.

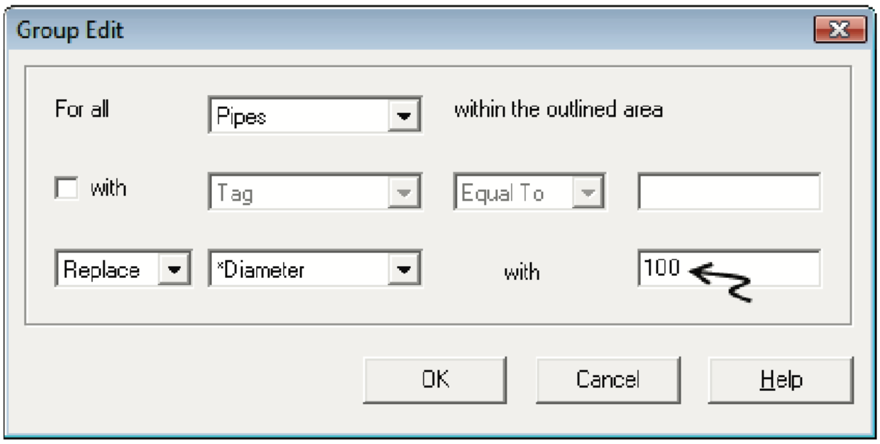


11. Change the diameter to **100 mm**. With a pressure of **10.33 m**, this result can be accepted.

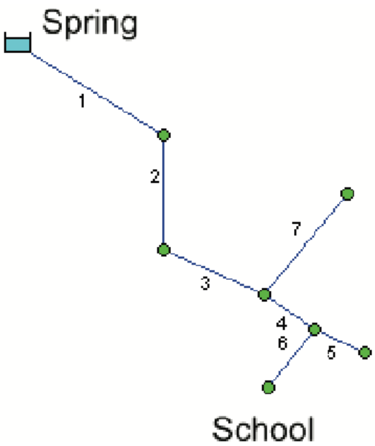
    Do not use arbitrary diameters like **92.319 mm**. Use only commercially available pipe diameters: `25, 40, 63, 75, 100, 125, 150, 200, 250, 300...`

12. If the pipe from the spring is now **100 mm**, it is very likely that all the other pipes will also be **100 mm** or smaller, otherwise you would create a bottleneck at the source, which is only done in special cases.

    Edit all the pipes at once to set their diameter to **100 mm**. The fastest way to do this: `Edit > Select All` and then `Edit > Group Edit`.



13. Continue reducing the pipe diameters until you obtain the **optimal system**. There is no single solution, but several possibilities. Remember to recalculate after each set of changes, otherwise Epanet and you could end up working on different networks!
14. One possible solution might look like this:



Link ID	Length m	Diameter mm
Pipe 1	800	100
Pipe 2	400	75
Pipe 3	300	75
Pipe 4	150	50
Pipe 5	100	18
Pipe 6	200	50
Pipe 7	500	50

For simplicity, we have assumed that the best design is the one that uses the smallest pipe diameters and therefore the lowest cost. However, there are many other considerations to keep in mind, for example: Could the pipes become blocked? Does the design allow for future expansion?

Appendix

A. Gaussian elimination method: step by step

This appendix illustrates the Gaussian elimination procedure for solving the linear system $\mathbf{J}\,\Delta\mathbf{Q} = \mathbf{b}$. The process involves three main stages: (1) forward elimination to form an upper triangular matrix, (2) forward substitution to solve for an intermediate vector $\mathbf{y}$, and (3) back substitution to find $\Delta\mathbf{Q}$.

—

Step 1. Pivot in column 1

For numerical stability, we swap $R_1 \leftrightarrow R_5$:

$$\left[\begin{array}{ccccc|c} 1305.71 & 6610.15 & 0 & 0 & -11898.27 & 199.12 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -39660.89 & -3481.89 & 11898.27 & 8033.64 \\ -1 & 0 & 0 & -1 & -1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \end{array}\right].$$

We eliminate entries below the pivot using the multipliers $m_{21} = \dfrac{1}{1305.71} = 7.6587 \times 10^{-4}$ and $m_{41} = \dfrac{-1}{1305.71} = -7.6587 \times 10^{-4}$:

$$\begin{aligned} R_2 &\leftarrow R_2 - m_{21}R_1 = \big[0,\ -6.0624947,\ 0,\ 0,\ 9.1124905\big], \\ R_4 &\leftarrow R_4 - m_{41}R_1 = \big[0,\ 5.0624947,\ 0,\ -1,\ -10.112491\big]. \end{aligned}$$

—

Step 2. Pivot in column 2

Using R_2 as the pivot row, we eliminate the entries below it with:

$$m_{42} = \frac{5.0624947}{-6.0624947} = -0.8350514, \qquad m_{52} = \frac{1}{-6.0624947} = -0.1649486.$$

Thus:

$$\begin{aligned} R_4 &\leftarrow R_4 - m_{42}R_2 = \big[0,\ 0,\ 0,\ -1,\ -2.5030925\big], \\ R_5 &\leftarrow R_5 - m_{52}R_2 = \big[0,\ 0,\ 1,\ 0,\ 2.5030925\big]. \end{aligned}$$

—

Step 3. Pivot in column 3

Using R_3 as pivot, eliminate the entry in row 5:

$$\begin{aligned} m_{53} &= \frac{1}{-39660.89} = -2.5214 \times 10^{-5}, \\ R_5 &\leftarrow R_5 - m_{53}R_3 = \big[0,\ 0,\ 0,\ -0.08779152,\ 2.8030926\big]. \end{aligned}$$

—

Step 4. Pivot in column 4

Finally, using R_4 as pivot, we eliminate the entry in row 5:

$$m_{54} = \frac{-0.08779152}{-1} = 0.08779152,$$
$$R_5 \leftarrow R_5 - m_{54}R_4 = \big[0,\ 0,\ 0,\ 0,\ 3.02284291\big].$$

—

Step 5. Upper triangular form

After elimination, the upper triangular matrix is:

$$\mathbf{U} = \begin{bmatrix} 1305.71 & 6610.15 & 0 & 0 & -11898.27 \\ 0 & -6.0624947 & 0 & 0 & 9.1124905 \\ 0 & 0 & -39660.89 & -3481.89 & 11898.27 \\ 0 & 0 & 0 & -1 & -2.5030925 \\ 0 & 0 & 0 & 0 & 3.02284291 \end{bmatrix}.$$

The multipliers define the **unit lower-triangular matrix**:

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0.00076587 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ -0.00076587 & -0.8350514 & 0 & 1 & 0 \\ 0 & -0.1649486 & -2.5214\times10^{-5} & 0.08779152 & 1 \end{bmatrix}.$$

We also define the permutation matrix $\mathbf{P}$ accounting for the initial swap $(R_1 \leftrightarrow R_5)$. The permuted right-hand side is:

$$\mathbf{Pb} = \begin{bmatrix} 199.12 \\ 0 \\ 8033.64 \\ 0 \\ 0 \end{bmatrix}.$$

—

Step 6. Forward substitution

Solve $\mathbf{Ly} = \mathbf{Pb}$:

$$y_1 = 199.12, \qquad y_2 = -0.152499, \qquad y_3 = 8033.64, \qquad y_4 = 0.0251546, \qquad y_5 = 0.175195.$$

—

Step 7. Back substitution

Finally, solve $\mathbf{U}\,\Delta\mathbf{Q} = \mathbf{y}$:

$$\Delta Q_5 = \frac{y_5}{U_{55}} = \frac{0.175195}{3.02284291} = 0.057957,$$

$$\Delta Q_4 = \frac{y_4 - U_{45}\Delta Q_5}{U_{44}} = \frac{0.0251546 + 2.5030925 \cdot 0.057957}{-1} = -0.170227,$$

$$\Delta Q_3 = \frac{y_3 - U_{34}\Delta Q_4 - U_{35}\Delta Q_5}{U_{33}} = -0.170227,$$

$$\Delta Q_2 = \frac{y_2 - U_{25}\Delta Q_5}{U_{22}} = 0.112270,$$

$$\Delta Q_1 = \frac{y_1 - U_{12}\Delta Q_2 - U_{15}\Delta Q_5}{U_{11}} = 0.112270.$$

Thus, the solution is:

$$\boxed{\Delta\mathbf{Q} = \begin{bmatrix} 0.11227 \\ 0.11227 \\ -0.17023 \\ -0.17023 \\ 0.05796 \end{bmatrix}}$$

B. Typical Friction Factor Values for Pipes

The following table provides updated values of friction coefficients used in hydraulic calculations. The Hazen–Williams coefficient C is commonly used in water distribution systems, the Darcy–Weisbach absolute roughness ϵ is used to calculate friction factors, and the Manning's n is applied in open channel and pipe flow analysis.

Table 7: Updated friction factor ranges for common pipe materials using Hazen–Williams C, Darcy–Weisbach roughness ϵ, and Manning's n.

Pipe Material	**Hazen–Williams C**	**ϵ (ft × 10^{-3})**	**Manning's n**
Cast Iron	130 − 140	0.85	0.012 − 0.015
Concrete / Concrete Lined	120 − 140	1.0 − 10	0.012 − 0.017
Galvanized Iron	120	0.5	0.015 − 0.017
Plastic (PVC, PE, etc.)	140 − 150	0.005	0.011 − 0.015
Steel	140 − 150	0.15	0.015 − 0.017
Vitrified Clay	110	–	0.013 − 0.015

These values are indicative and should be adjusted according to manufacturer data or field measurements when available.