CE 3372 WATER SYSTEMS DESIGN

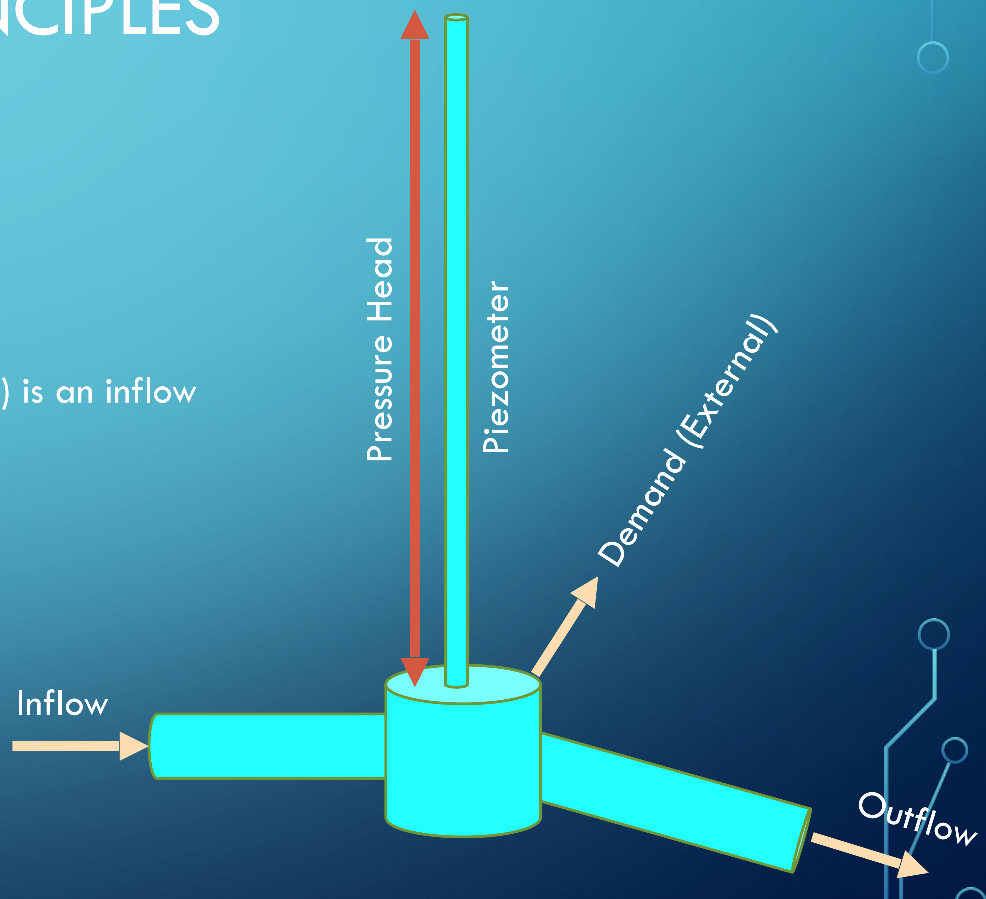
LESSON 8: HYDRAULICS OF NETWORKS - FALL 2020

OUTLINE

- Network Hydraulic Principles
- Constructing the Network System of Equations
  - Link-Node Continuity Partition
  - Link-Node Energy Partition
- Adding Pumps

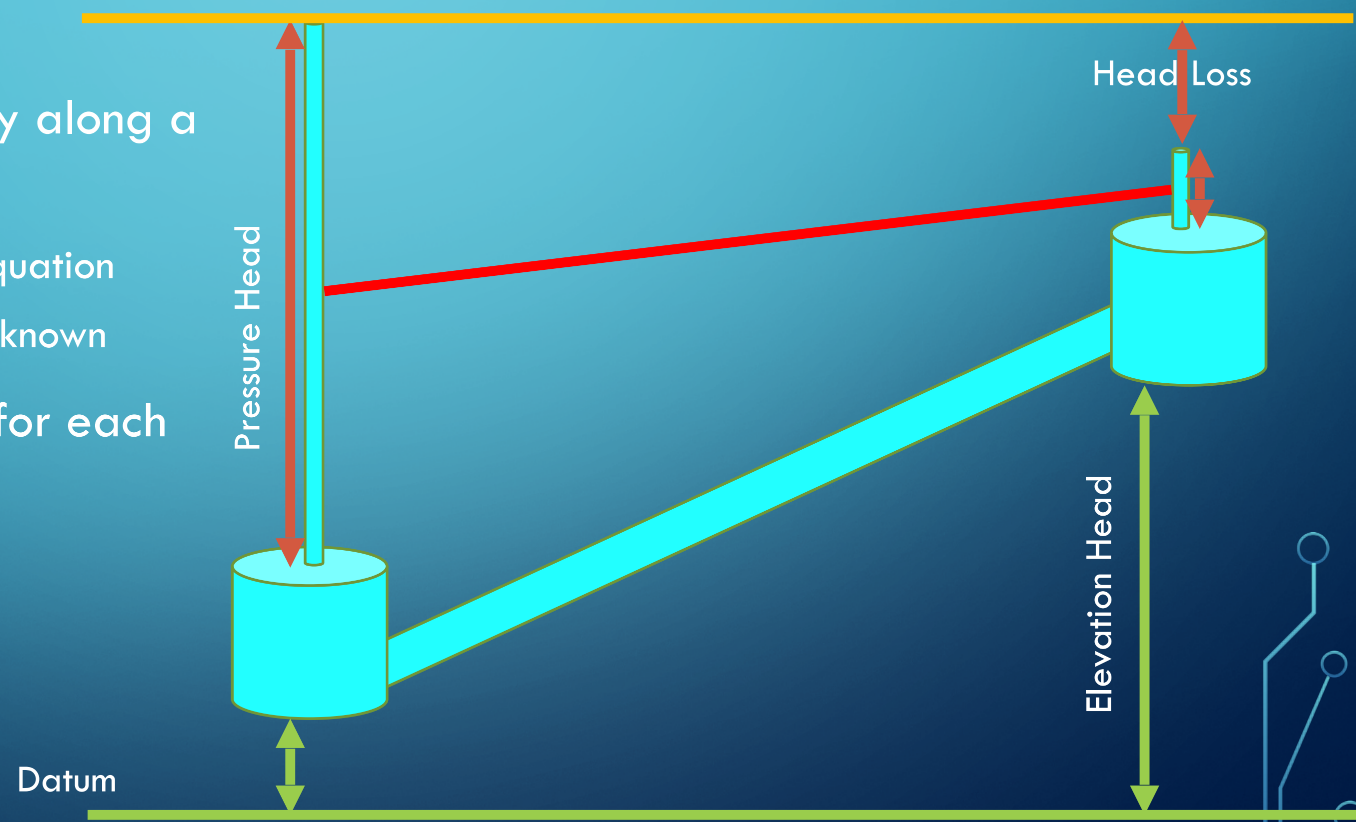
NETWORK HYDRAULIC PRINCIPLES

- Conservation of mass at a junction
  - ΣInflow – ΣOutflow = 0 (at a node)
    - Positive demand at a node is an outflow
    - Negative demand (injection into the network) is an inflow
    - One balance equation for each node
  - Head has a single value
    - Usually expressed as a pressure
    - Visualize as height of column of water in a piezometer at the junction



NETWORK HYDRAULIC PRINCIPLES

- Conservation of energy along a conduit
  - Modified Bernoulli's equation
  - Include fitting losses if known
- One energy equation for each conduit (link)



BRANCHED SYSTEM

- Distribution networks are multi-path pipelines

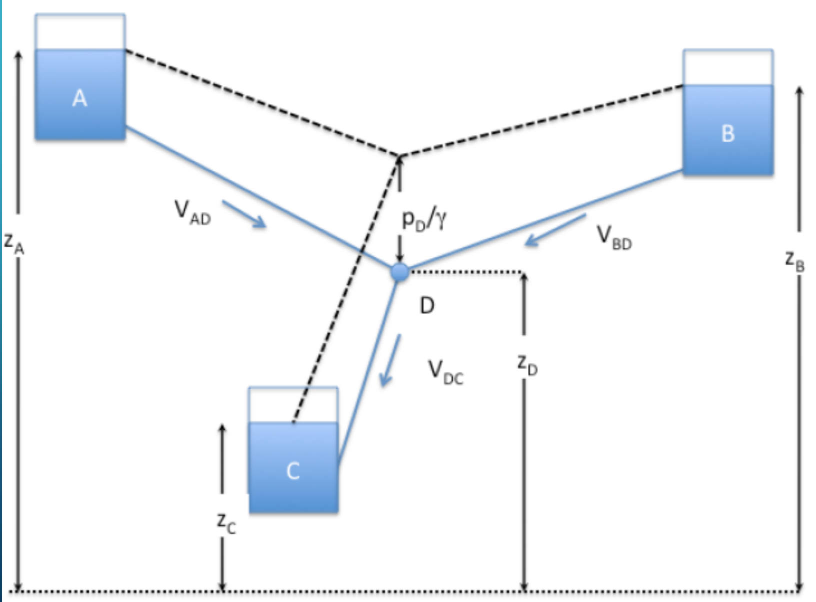


Figure 9: Branched pipe system

- One topological structure is branching
- One node (continuity)
- Three links (energy)
- Four unknowns: $V_{AD}, V_{BD}, V_{DC}, H_D$

BRANCHED SYSTEM - ANALYSIS

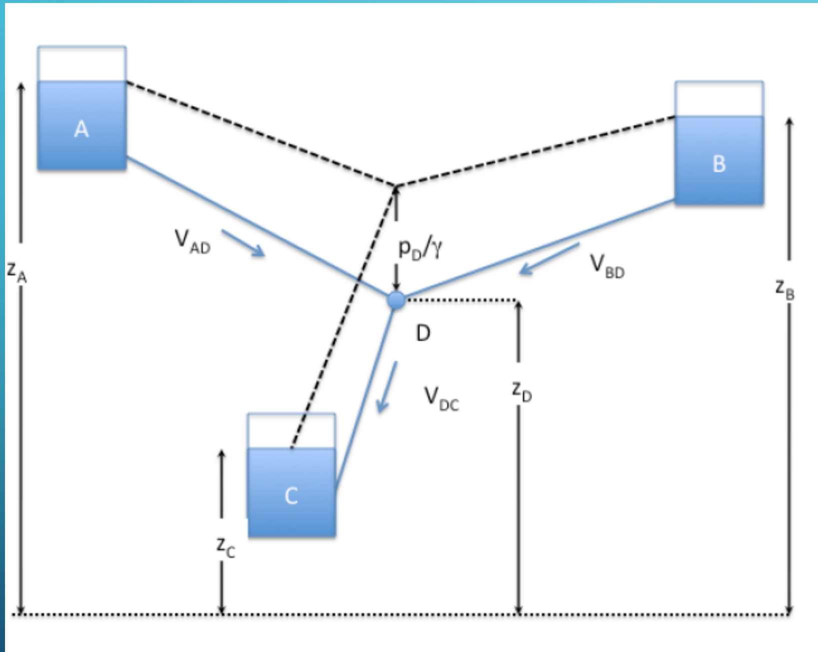


Figure 9: Branched pipe system

Head loss in each pipe

$$z_A = z_D + \frac{p_D}{\gamma} + f_{AD}\frac{L_{AD}}{D_{AD}}\frac{V_{AD}^2}{2g}$$

$$z_B = z_D + \frac{p_D}{\gamma} + f_{BD}\frac{L_{BD}}{D_{BD}}\frac{V_{BD}^2}{2g}$$

$$z_D + \frac{p_D}{\gamma} = z_C + f_{DC}\frac{L_{DC}}{D_{DC}}\frac{V_{DC}^2}{2g}$$

Continuity at the node

$$A_{AD}V_{AD} + A_{BD}V_{BD} = A_{DC}V_{DC}$$

BRANCHED SYSTEM

- 4 Equations, 4 unknowns
- Non-linear so solve by
  - Newton-Raphson
  - Quasi-Linearization
- Quadratic in unknown, so usually can find solution in just a few iterations

$$z_A = z_D + \frac{p_D}{\gamma} + f_{AD}\frac{L_{AD}}{D_{AD}}\frac{V_{AD}^2}{2g}$$

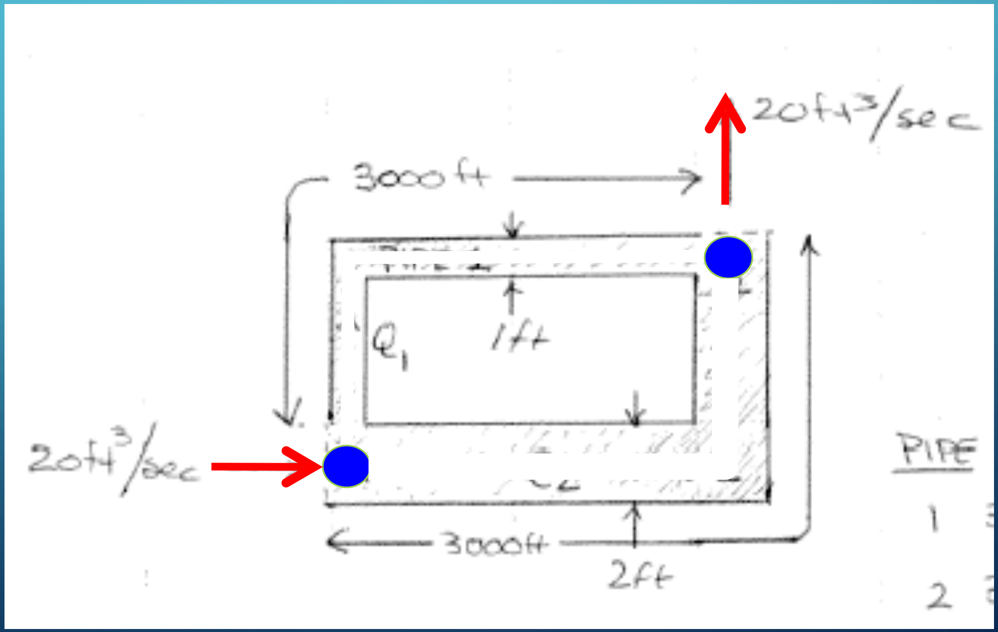
$$z_B = z_D + \frac{p_D}{\gamma} + f_{BD}\frac{L_{BD}}{D_{BD}}\frac{V_{BD}^2}{2g}$$

$$z_D + \frac{p_D}{\gamma} = z_C + f_{DC}\frac{L_{DC}}{D_{DC}}\frac{V_{DC}^2}{2g}$$

$$A_{AD}V_{AD} + A_{BD}V_{BD} = A_{DC}V_{DC}$$

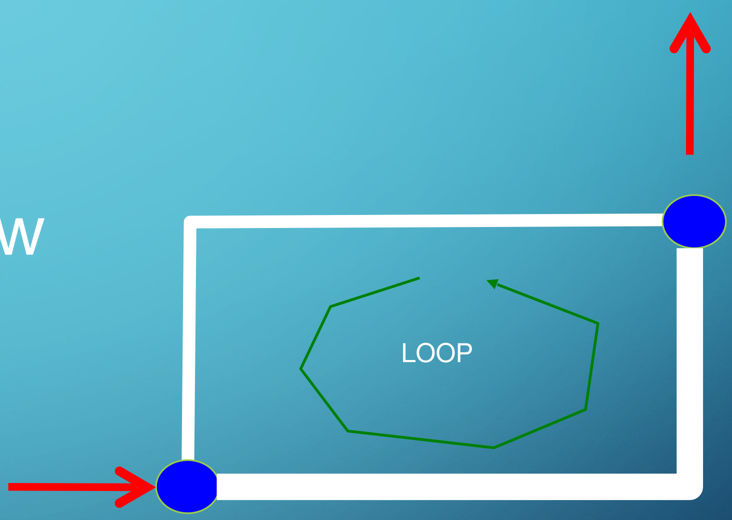
LOOPED SYSTEM

- Looped system is where one or more pipes rejoin at a different node.



LOOPED SYSTEM

- Nodes:
    - Inflow = Outflow
    - Energy Unique
- Links
    - Head loss along pipe
    - Head loss in any loop is zero



EXAMPLE – BRANCHED SYSTEM

Consider the branched system shown in Figure 19. In this example the friction factor is assumed constant for simplicity, but in practice would vary during the solution computations.

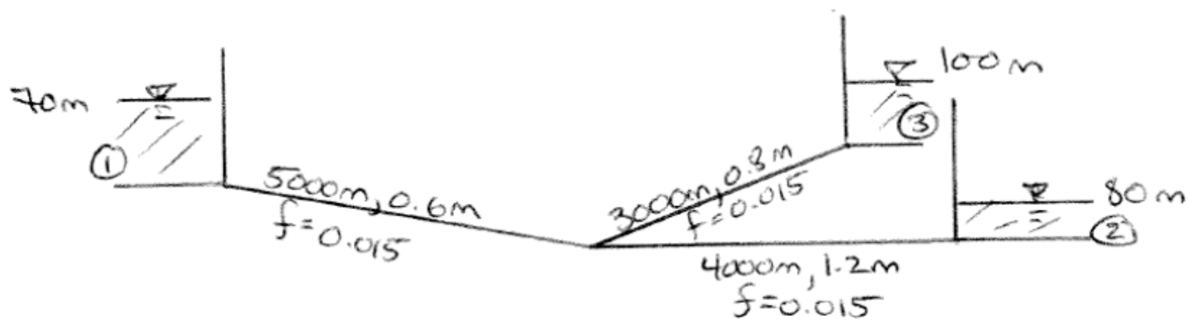


Figure 19: Branched pipe system

The hydraulics question is what is the discharge in each pipe and what is the total head at the junction (notice we don't know the junction elevation in this example — if the elevation were specified, we could also find the pressure head).

EXAMPLE – BRANCHED SYSTEM

Solution First populate the four equations with the appropriate numerical values.

$$70 = h_D + (0.015)\frac{5000}{0.6}\frac{V_{AD}|V_{AD}|}{2(9.8)} \tag{35}$$

$$100 = h_D + (0.015)\frac{3000}{0.8}\frac{V_{BD}|V_{BD}|}{2(9.8)} \tag{36}$$

$$-80 = -h_D + (0.015)\frac{4000}{1.2}\frac{V_{DC}|V_{DC}|}{2(9.8)} \tag{37}$$

$$\frac{\pi(0.6)^2}{4}V_{AD} + \frac{\pi(0.8)^2}{4}V_{BD} = \frac{\pi(1.2)^2}{4}V_{DC} \tag{38}$$

EXAMPLE – BRANCHED SYSTEM

Next, compute all the constants, and organize the 4 equations into a system of simultaneous equations

$$\begin{array}{ccccc} h_D & 6.377V_{AD}|V_{AD}| & 0 & 0 & = 70 \\ h_D & 0 & 2.869V_{BD}|V_{BD}| & 0 & = 100 \\ h_D & 0 & 0 & -2.551V_{DC}|V_{DC}| & = 80 \\ 0 & 0.2827V_{AD} & 0.5026V_{BD} & -1.1309V_{DC} & = 0 \end{array} \quad (39)$$

Because the system is non-linear we need to employ some tricks. First, it is a quadratic system so if a solution exists, Excel solver should be able to find one.

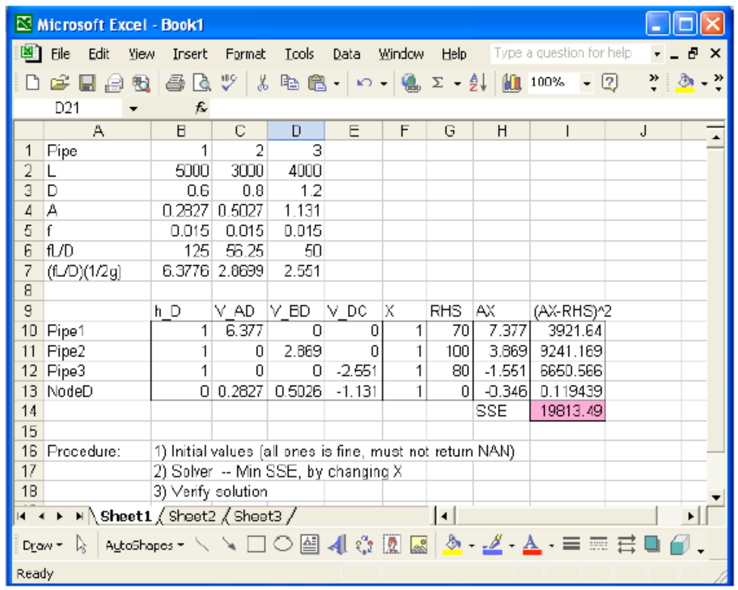
First a more conventional vector-matrix structure

$$\begin{pmatrix} 1 & 6.377|V_{AD}| & 0 & 0 \\ 1 & 0 & 2.869|V_{BD}| & 0 \\ 1 & 0 & 0 & -2.551|V_{DC}| \\ 0 & 0.2827 & 0.5026 & -1.1309 \end{pmatrix} \bullet \begin{pmatrix} h_D \\ V_{AD} \\ V_{BD} \\ V_{DC} \end{pmatrix} - \begin{pmatrix} 70 \\ 100 \\ 80 \\ 0 \end{pmatrix} = \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \end{pmatrix} \quad (40)$$

Now we can use solver to find a solution. We notice that the right-hand-side is a vector of "errors". If we square these values and sum them ($\vec{\epsilon}^T\vec{\epsilon}$), then minimize this sum of squared errors by changing h_D, V_{AD}, V_{BD}, and V_{DC}, the result should be the solution (if the error is small enough).

EXAMPLE – BRANCHED SYSTEM

Figure 20 is a screen capture of an Excel worksheet before starting the solution algorithm. Notice the column labeled **X** contains an initial guess for the unknowns.



	A	B	C	D	E	F	G	H	I	J
1	Pipe	1	2	3						
2	L	5000	3000	4000						
3	D	0.6	0.8	1.2						
4	A	0.2827	0.5027	1.131						
5	f	0.015	0.015	0.015						
6	fL/D	125	56.25	50						
7	(fL/D)(1/2g)	6.3776	2.8699	2.551						
8										
9		h_D	V_AD	V_BD	V_DC	X	RHS	AX	(AX-RHS)^2	
10	Pipe1	1	6.377	0	0	1	70	7.377	3921.64	
11	Pipe2	1	0	2.869	0	1	100	3.869	9241.169	
12	Pipe3	1	0	0	-2.551	1	80	-1.551	6650.566	
13	NodeD	0	0.2827	0.5026	-1.131	1	0	-0.346	0.119439	
14								SSE	19813.49	
15										
16	Procedure:	1) Initial values (all ones is fine, must not return NAN)								
17		2) Solver -- Min SSE, by changing X								
18		3) Verify solution								

Figure 20: Branched pipe system — Excel solution set-up

EXAMPLE – BRANCHED SYSTEM

After building the spreadsheet and preparing the columns, then the engineer must select solver to minimize the error vector (the last column in the worksheet). The **SOLVER** add-in does the job nicely, **GOAL SEEK** might work, but provides less control of precision and stopping criterion.

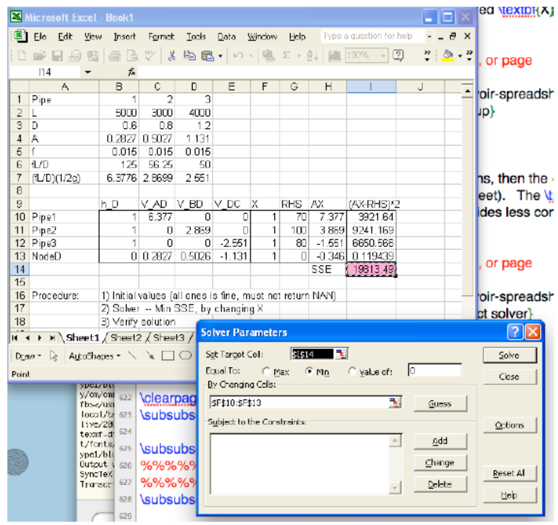
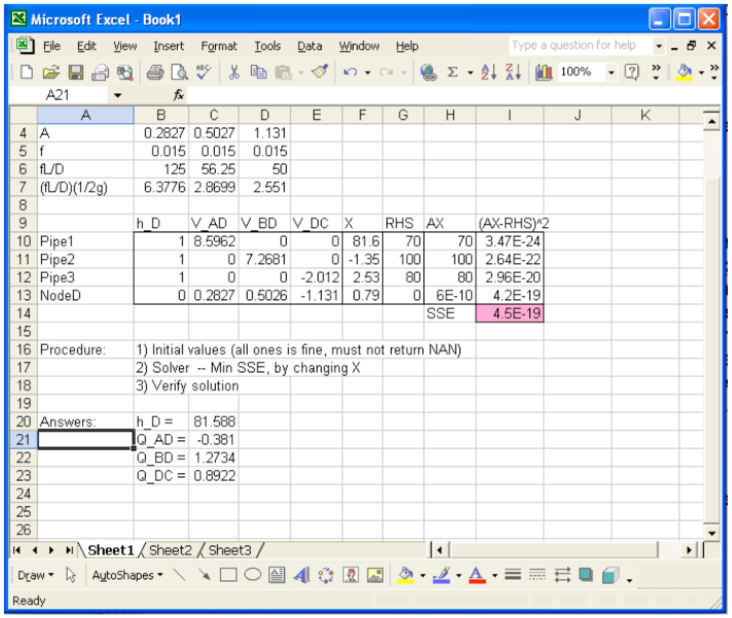


Figure 21: Branched pipe system — Excel solution select solver

Figure 21 is the solver set-up for this example. When the engineer selects **SOLVE** then the program will attempt to minimize the error by changing values of the **X** vector. The example does NOT use program defaults, but instead reduces the tolerance and precision values (in the Options section) and chooses the conjugate-gradient direction finding method and centered differences.

EXAMPLE – BRANCHED SYSTEM

Finally, when the solver is completed, we recover the solution by multiplying the results by pipe areas to obtain the flows. Figure 22 is the solution for this example (the arithmetic was added at the bottom after the program converged to the near-zero "sum-squared-error").



	A	B	C	D	E	F	G	H	I
4	A	0.2827	0.5027	1.131					
5	f	0.015	0.015	0.015					
6	fL/D	125	56.25	50					
7	(fL/D)(1/2g)	6.3776	2.8699	2.551					
8									
9		h_D	V_AD	V_BD	V_DC	X	RHS	AX	(AX-RHS)^2
10	Pipe1	1	8.5962	0	0	81.6	70	70	3.47E-24
11	Pipe2	1	0	7.2681	0	-1.35	100	100	2.64E-22
12	Pipe3	1	0	0	-2.012	2.53	80	80	2.96E-20
13	NodeD	0	0.2827	0.5026	-1.131	0.79	0	6E-10	4.2E-19
14								SSE	4.5E-19
15									
16	Procedure:	1) Initial values (all ones is fine, must not return NAN)							
17		2) Solver -- Min SSE, by changing X							
18		3) Verify solution							
19									
20	Answers:	h_D =	81.588						
21		Q_AD =	-0.381						
22		Q_BD =	1.2734						
23		Q_DC =	0.8922						

Figure 22: Branched pipe system — Excel solution

EXAMPLE – BRANCHED SYSTEM

The results are:

1. $Q_{AD} = -0.381 m^3/s$, from Node D to Reservior A (opposite the original assumed direction).
2. $Q_{BD} = 1.273 m^3/s$, from Reservoir B to the node (the assumed flow direction in this pipe).
3. $Q_{DC} = 0.89 m^3/s$ from the Node D to reservoir C (the assumed direction in this pipe).
4. $h_D = 81.6 m$.

An alternate solution approach is to guess values of h_D and use the energy equations to find resulting flows in each pipe. Then adjust the guess until continuity is satisfied. Lastly, the reader might (correctly) conclude that we could have built the spreadsheet to directly solve by the Newton-Raphson method where the Jacobian is computed from the coefficient matrix (and some calculus) and an update formula is directly built into the spreadsheet. This approach (Newton-Raphson) is outlined in the section on networks and is the method employed in the `EPA-NET` computer program.

EXAMPLE – LOOPED SYSTEM

Example
Figure 24 is a parallel pipe system with two pipes that together convey 20 cubic-feet-per-second of water.

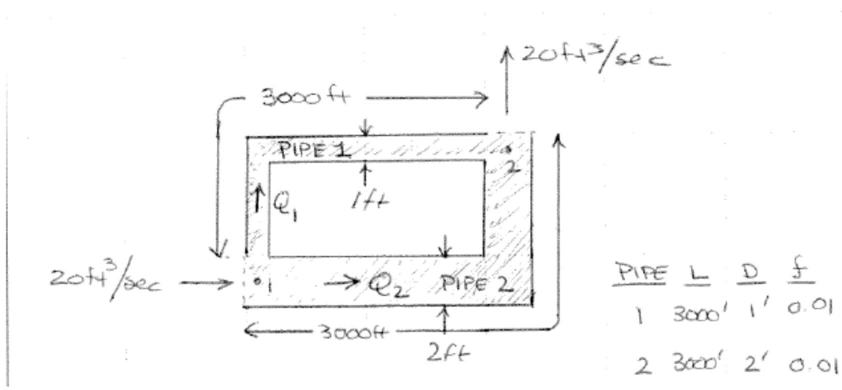


Figure 24: Parallel pipe system. Find Q_1 and Q_2.

Pipe 1 is a 3000-foot long, 12-inch diameter pipe with constant friction factor of 0.01. Pipe 2 is a 3000-foot long, 24-inch diameter pipe, also with friction factor 0.01. Find the discharge in each pipe.

EXAMPLE – LOOPED SYSTEM

Solution

This example is really a simple network. First we assume flow directions (shown in the sketch). Next we will write the node equations.

$$\begin{aligned} P_1: &\quad +20 \quad -Q_1 \quad -Q_2 \quad -Q_f \quad = 0 \\ P_2: &\quad -20 \quad +Q_1 \quad +Q_2 \quad +0 \quad = 0 \end{aligned} \tag{41}$$

The fictitous pipe (Q_f) will be explained shortly — at the solution its flow value will be forced to zero.

Next we write a head loss equation around the loop, the sign convention is that if the loop traverse direction (CCW or CW) is the same as the assumed flow direction, then the head loss is positive. If the loop traverse direction and the flow direction are opposite, then the head loss is negative. Figure 25 illustrates this convention (and uses CCW as the traverse direction).

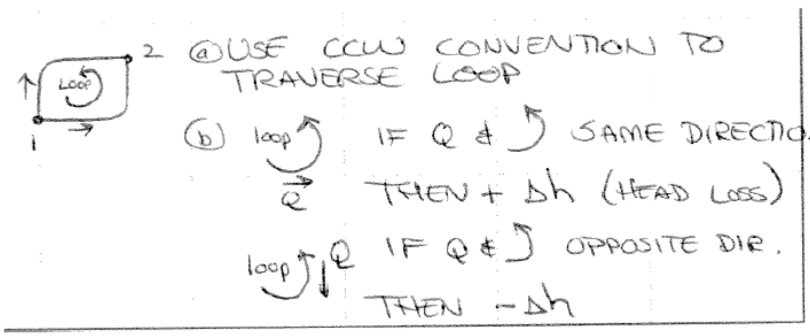


Figure 25: Sign convention in loop traverse

EXAMPLE – LOOPED SYSTEM

Using this convention we then write the loop equation(s).

$$L_1: \quad +0 \quad -\frac{8fL_1}{\pi^2 g D_1^5} Q_1|Q_1| \quad +\frac{8fL_2}{\pi^2 g D_2^5} Q_2|Q_2| \quad +0 \quad = 0 \tag{42}$$

Next we combine the node and loop equations into a system of equations (as before) and build a solution algorithm.

$$\begin{pmatrix} -1 & -1 & 1 \\ 1 & 1 & 0 \\ -\frac{8fL_1}{\pi^2 g D_1^5}|Q_1| & \frac{8fL_2}{\pi^2 g D_2^5}|Q_2| & 0 \end{pmatrix} \bullet \begin{pmatrix} Q_1 \\ Q_2 \\ Q_f \end{pmatrix} = \begin{pmatrix} -20 \\ 20 \\ 0 \end{pmatrix} \tag{43}$$

EXAMPLE – LOOPED SYSTEM

We now have a matrix-vector form of three non-linear simultaneous equations in three unknowns (with the added knowledge that one of the unknowns must be darned close to zero at the answer!). We could solve the same way as before, but instead, will directly build a Newton-Raphson solution.

To do that, we need to recall some calculus. Recall Newton's method for non-linear root finding:

$$x^{k+1} = x^k - \frac{f(x^k)}{\frac{df}{dx}\big|_{x^k}} \tag{44}$$

This update equation solves $f(x) = 0$ if the initial guess is adequate and the derivative is non-zero. Our problem is the same except we need a multi-dimensional extension of Newton's method.

EXAMPLE – LOOPED SYSTEM

Our function is $\mathbf{G}(\mathbf{Q}) = \mathbf{0}$ and the multi-dimensional "derivative" is the Jacobian of $\mathbf{G}$ evaluated at the current (^{k}) guess.

Constructing $\mathbf{G}$ is straightforward, we just subtract the right hand side (the demand vector) from the model.

$$\mathbf{G}(\mathbf{Q}) = \begin{pmatrix} -1 & -1 & 1 \\ 1 & 1 & 0 \\ -\frac{8fL_1}{\pi^2 g D_1^5}|Q_1| & \frac{8fL_2}{\pi^2 g D_2^5}|Q_2| & 0 \end{pmatrix} \bullet \begin{pmatrix} Q_1 \\ Q_2 \\ Q_f \end{pmatrix} - \begin{pmatrix} -20 \\ 20 \\ 0 \end{pmatrix} \tag{45}$$

The Jacobian is trickier. The Jacobian is the matrix of partial derivatives of $\mathbf{G}$, evaluated at the current guess.

$$\mathbf{J}(\mathbf{G})^k = \begin{pmatrix} \frac{\partial G_1}{\partial Q_1} & \frac{\partial G_1}{\partial Q_2} & \frac{\partial G_1}{\partial Q_f} \\ \frac{\partial G_2}{\partial Q_1} & \frac{\partial G_2}{\partial Q_2} & \frac{\partial G_2}{\partial Q_f} \\ \frac{\partial G_3}{\partial Q_1} & \frac{\partial G_3}{\partial Q_2} & \frac{\partial G_1}{\partial Q_f} \end{pmatrix} = \begin{pmatrix} -1 & -1 & 1 \\ 1 & 1 & 0 \\ -2\frac{8fL_1}{\pi^2 g D_1^5}|Q_1| & 2\frac{8fL_2}{\pi^2 g D_2^5}|Q_2| & 0 \end{pmatrix} \tag{46}$$

EXAMPLE – LOOPED SYSTEM

Now the update formula by analogy. The single variable derivative term is replaced by the vector matrix operation using $\mathbf{G}$ and its Jacobian

$$-\frac{f(x^k)}{\frac{df}{dx}\big|_{x^k}} \approx -\mathbf{J}[\mathbf{G}(\mathbf{Q^k})]^{-1} \bullet \mathbf{G}(\mathbf{Q^k}) \quad (47)$$

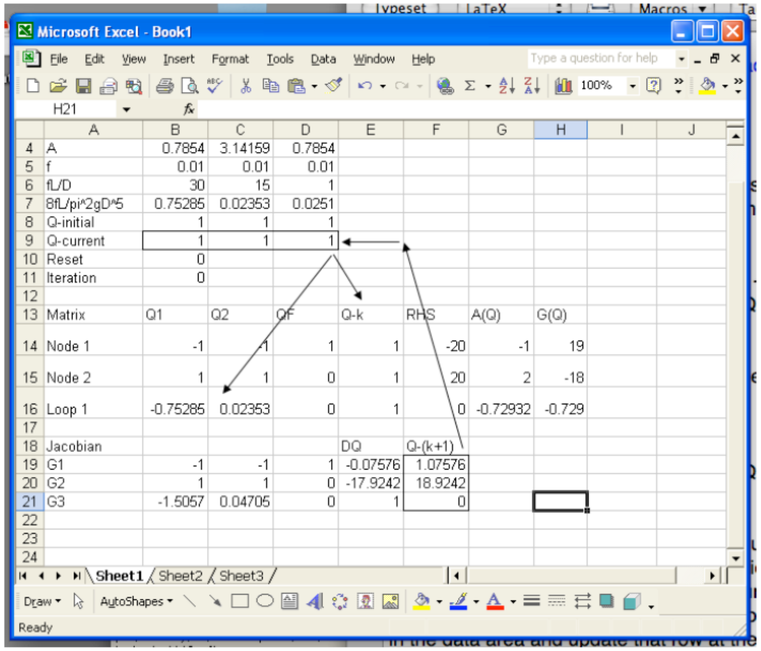
Thus the update formula (updates each element of the Q vector) becomes

$$\mathbf{Q^{k+1}} = \mathbf{Q^k} - \mathbf{J}[\mathbf{G}(\mathbf{Q^k})]^{-1} \bullet \mathbf{G}(\mathbf{Q^k}) \quad (48)$$

In this case we stop when the function actually gets to zero or when the solution stops changing. The trick in setting up a spreadsheet is to be sure the update occurs after the

EXAMPLE – LOOPED SYSTEM

Figure 26 is a spreadsheet implementation of this example, before the updating begins.

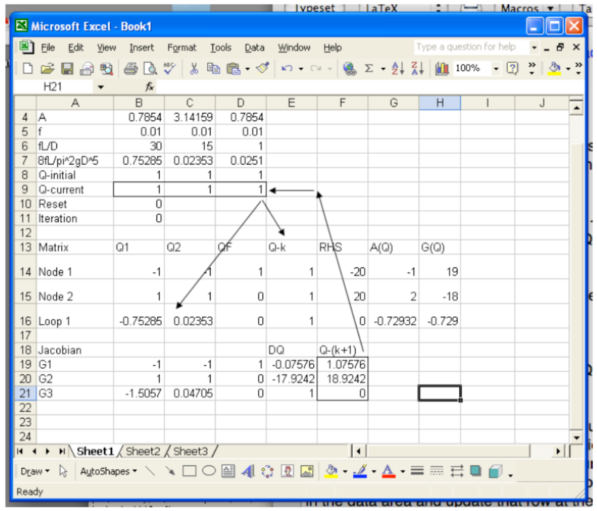


	A	B	C	D	E	F	G	H
4	A	0.7854	3.14159	0.7854				
5	f	0.01	0.01	0.01				
6	fL/D	30	15	1				
7	8fL/pi^2gD^5	0.75285	0.02353	0.0251				
8	Q-initial	1	1	1				
9	Q-current	1	1	1				
10	Reset	0						
11	Iteration	0						
12								
13	Matrix	Q1	Q2	QF	Q-k	RHS	A(Q)	G(Q)
14	Node 1	-1	-1	1	1	-20	-1	19
15	Node 2	1	1	0	1	20	2	-18
16	Loop 1	-0.75285	0.02353	0	1	0	-0.72932	-0.729
17								
18	Jacobian				DQ	Q-(k+1)		
19	G1	-1	-1	1	-0.07576	1.07576		
20	G2	1	1	0	-17.9242	18.9242		
21	G3	-1.5057	0.04705	0	1	0		

Figure 26: Simple network model using Newton-Raphson updates before iterations. Arrows indicate update path — notice the loopback is at bottom and right, so all rows are done before update. Also notice reset and iteration counter. These help control computations.

EXAMPLE – LO

Figure 26 is a spreadsheet implementation of this example, before the updating begins.



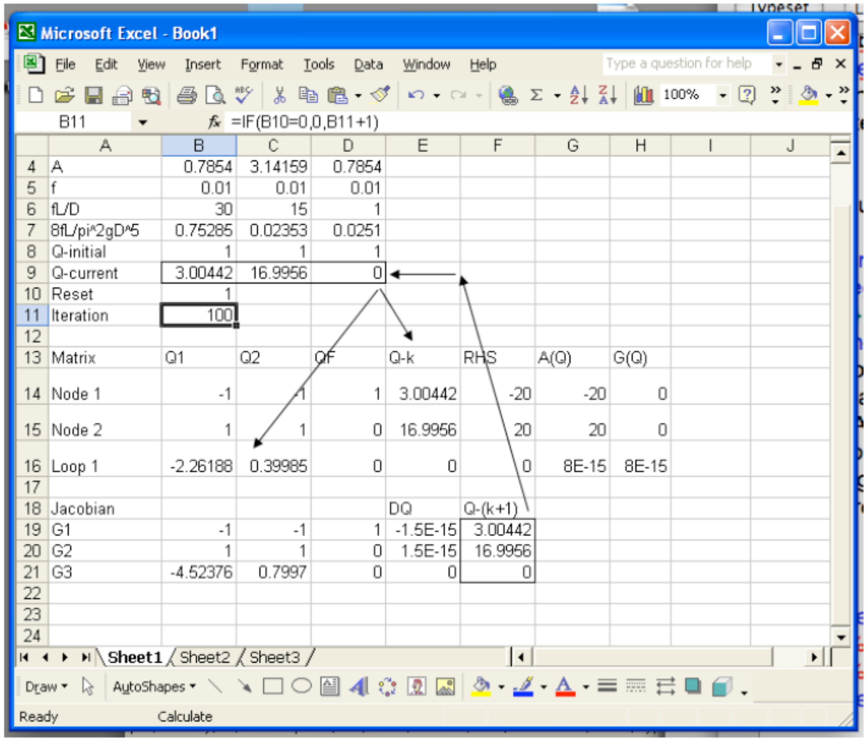
	A	B	C	D	E	F	G	H	I	J
4	A	0.7854	3.14159	0.7854						
5	f	0.01	0.01	0.01						
6	fL/D	30	15	1						
7	8fL/pi^2gD^5	0.75285	0.02353	0.0251						
8	Q-initial	1	1	1						
9	Q-current	1	1	1						
10	Reset	0								
11	Iteration	0								
12										
13	Matrix	Q1	Q2	QF	Q-k	RHS	A(Q)	G(Q)		
14	Node 1	-1	-1	1	1	-20	-1	19		
15	Node 2	1	1	0	1	20	2	-18		
16	Loop 1	-0.75285	0.02353	0	1	0	-0.72932	-0.729		
17										
18	Jacobian				DQ	Q-(k+1)				
19	G1	-1	-1	1	-0.07576	1.07576				
20	G2	1	1	0	-17.9242	18.9242				
21	G3	-1.5057	0.04705	0	1	0				
22										
23										
24										

Figure 26: Simple network model using Newton-Raphson updates before iterations. Arrows indicate update path — notice the loopback is at bottom and right, so all rows are done before update. Also notice reset and iteration counter. These help control computations.

The function matrix is shown and the **G(Q)** is computed in column H. The **A(Q)** is simply a matrix multiplication from which the **RHS** is subtracted.

The Jacobian matrix is built from the function matrix. The column **DQ** is the ratio of the function and derivative at the current guess, and the lower right column **Q-(k+1)** is the update following the multi-dimensional analog to Newton's method.

EXAMPLE – LOOPED SYSTEM



	A	B	C	D	E	F	G	H
4	A	0.7854	3.14159	0.7854				
5	f	0.01	0.01	0.01				
6	fL/D	30	15	1				
7	8fL/pi^2gD^5	0.75285	0.02353	0.0251				
8	Q-initial	1	1	1				
9	Q-current	3.00442	16.9956	0				
10	Reset	1						
11	Iteration	100						
12								
13	Matrix	Q1	Q2	QF	Q-k	RHS	A(Q)	G(Q)
14	Node 1	-1	-1	1	3.00442	-20	-20	0
15	Node 2	1	1	0	16.9956	20	20	0
16	Loop 1	-2.26188	0.39985	0	0	0	8E-15	8E-15
17								
18	Jacobian				DQ	Q-(k+1)		
19	G1	-1	-1	1	-1.5E-15	3.00442		
20	G2	1	1	0	1.5E-15	16.9956		
21	G3	-4.52376	0.7997	0	0	0		
22								
23								
24								

Figure 27: Simple network model using Newton-Raphson after 100 iterations. $\mathbf{G(Q)}$ is for all practical purposes $\mathbf{0}$, hence the problem is completed.

EXAMPLE – LOOPED SYSTEM

Figure 27 is the result of 100 iterations and represents the solution. In this example, the flow distribution is

1. $Q_1 = 3.00 m^3/s$ in the assumed direction.
2. $Q_2 = 17.00 m^3/s$ in the assumed direction.
3. $Q_f = 0$, which it should at the solution!
4. $\Delta h_{1->2} = 6.79 m$, regardless of path. The head loss around a loop must be zero, but just as important is the head loss between the two nodes.

NETWORKS

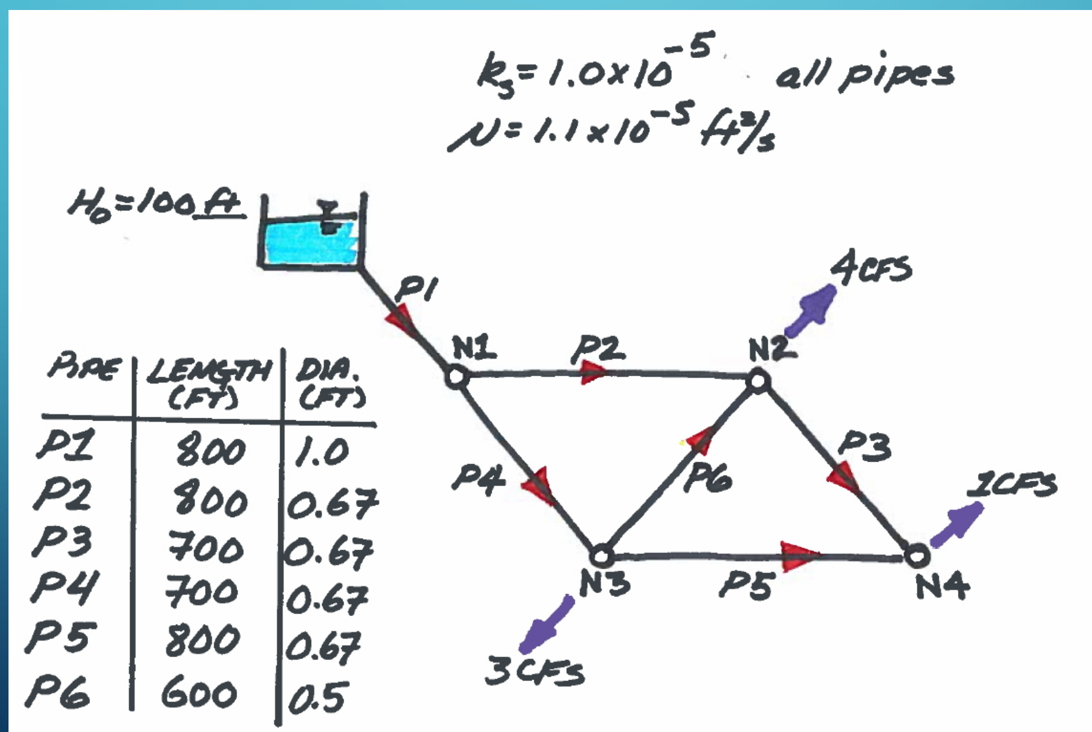
- Branched and looped systems are called networks
  - Multiple paths for water to flow in the system
    - External “demands” are applied at nodes
    - External “supply” is treated as negative demand

NETWORKS

- Analysis techniques
  - Hardy-Cross (Loop Only Equations)
  - Newton-Raphson (Node+Loop Equations)
  - Hybrid Method (Haman and Brameller (1971))(Node+Pipe Equations)
    - Build node and pipe equations (LIKE THE BRANCH SYSTEM)
    - Then solve the SIMULTANEOUS non-linear system for flows and heads

NETWORKS

- Consider the system below



PIPE	LENGTH (FT)	DIA. (FT)
P1	800	1.0
P2	800	0.67
P3	700	0.67
P4	700	0.67
P5	800	0.67
P6	600	0.5

NETWORKS

- Six pipes
- 4 nodes + 1 RESERVOIR (Fixed grade node)
- Demands shown at nodes
- Flow arrows for sign convention

NETWORKS

- Continuity is written at nodes (node equations).
- Energy loss (gain) is written along links (pipe equations).
- The entire set of equations is solved simultaneously.

NETWORKS

- Define an unknown vector (X)

Define the flows in each pipe and the total head at each node as Q_i and H_i where the subscript indicates the particular component identification. Expressed as a vector, these unknowns are:

$$[Q_1,\quad Q_2,\quad Q_3,\quad Q_4,\quad Q_5,\quad Q_6,\quad H_1,\quad H_2,\quad H_3,\quad H_4] \;=\; \mathbf{x}$$

NETWORKS

- Build the node and pipe equations

If we analyze continuity for each node we will have 4 equations (corresponding to each node) for continunity, for instance for Node N2 the equation is

$$Q_2 \quad -Q_3 \qquad Q_6 \qquad = \; 4$$

Similarily if we define head loss in any pipe as $\Delta H_i = f\frac{8L_i}{\pi^2 g D_i^5}|Q_i|Q_i$ or $\Delta H_i = L_i Q_i$, where $L_i = f\frac{8L_i}{\pi^2 g D_i^5}|Q_i|$, then we have 6 equations (corresponding to each pipe) for energy, for instance for Pipe (P2) the equation is[37]

$$-L_2 Q_2 \qquad H_1 \quad -H_2 \qquad = \; 0$$

NETWORKS

- Build the node and pipe equations

If we now write all the node equations then all the pipe equations we could construct the following coefficient matrix below:[38]

$$
\begin{array}{cccccccccc}
\hline
1 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
\hline
-L_1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & -L_2 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\
0 & 0 & -L_3 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \\
0 & 0 & 0 & -L_4 & 0 & 0 & 1 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & -L_5 & 0 & 0 & 0 & 1 & -1 \\
0 & 0 & 0 & 0 & 0 & -L_6 & 0 & -1 & 1 & 0 \\
\hline
\end{array}
$$

Declare the name of this matrix $\mathbf{A}(\mathbf{x})$, where $\mathbf{x}$ denotes the unknown vector of Q augmented by H as above. Next consider the right-hand-side at the correct solution (as of yet still unknown!) as

NETWORKS

- Build the node and pipe equations

Declare the name of this matrix $\mathbf{A(x)}$, where $\mathbf{x}$ denotes the unknown vector of Q augmented by H as above. Next consider the right-hand-side at the correct solution (as of yet still unknown!) as

$$[0, \quad 4, \quad 3, \quad 1, \quad -100, \quad 0, \quad 0, \quad 0, \quad 0, \quad 0] = \mathbf{b}$$

So if the coefficient matrix is correct then the following system would result:

$$\mathbf{A}(\mathbf{x}) \cdot \mathbf{x} = \mathbf{b}$$

which would look like

NETWORKS

- Build the node and pipe equations

$$
\left(\begin{array}{cccccccccc}
1 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
\hline
-L_1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & -L_2 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\
0 & 0 & -L_3 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \\
0 & 0 & 0 & -L_4 & 0 & 0 & 1 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & -L_5 & 0 & 0 & 0 & 1 & -1 \\
0 & 0 & 0 & 0 & 0 & -L_6 & 0 & -1 & 1 & 0
\end{array}\right)
\begin{pmatrix} Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \\ Q_5 \\ Q_6 \\ H_1 \\ H_2 \\ H_3 \\ H_4 \end{pmatrix}
=
\left(\begin{array}{c} 0 \\ 4 \\ 3 \\ 1 \\ \hline -100 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{array}\right)
\tag{114}
$$

Observe, the system is non-linear because the coefficient matrix depends on the current values of Q_i for the L_i terms. However, the system is full-rank (rows == columns) so it is a candidate for Newton-Raphson.

NETWORKS (HYBRID METHOD)

- Continuity is written at nodes (node equations).
- Energy loss (gain) is written along links (pipe equations).
- The entire set of equations is solved simultaneously.

ADDING PUMPS

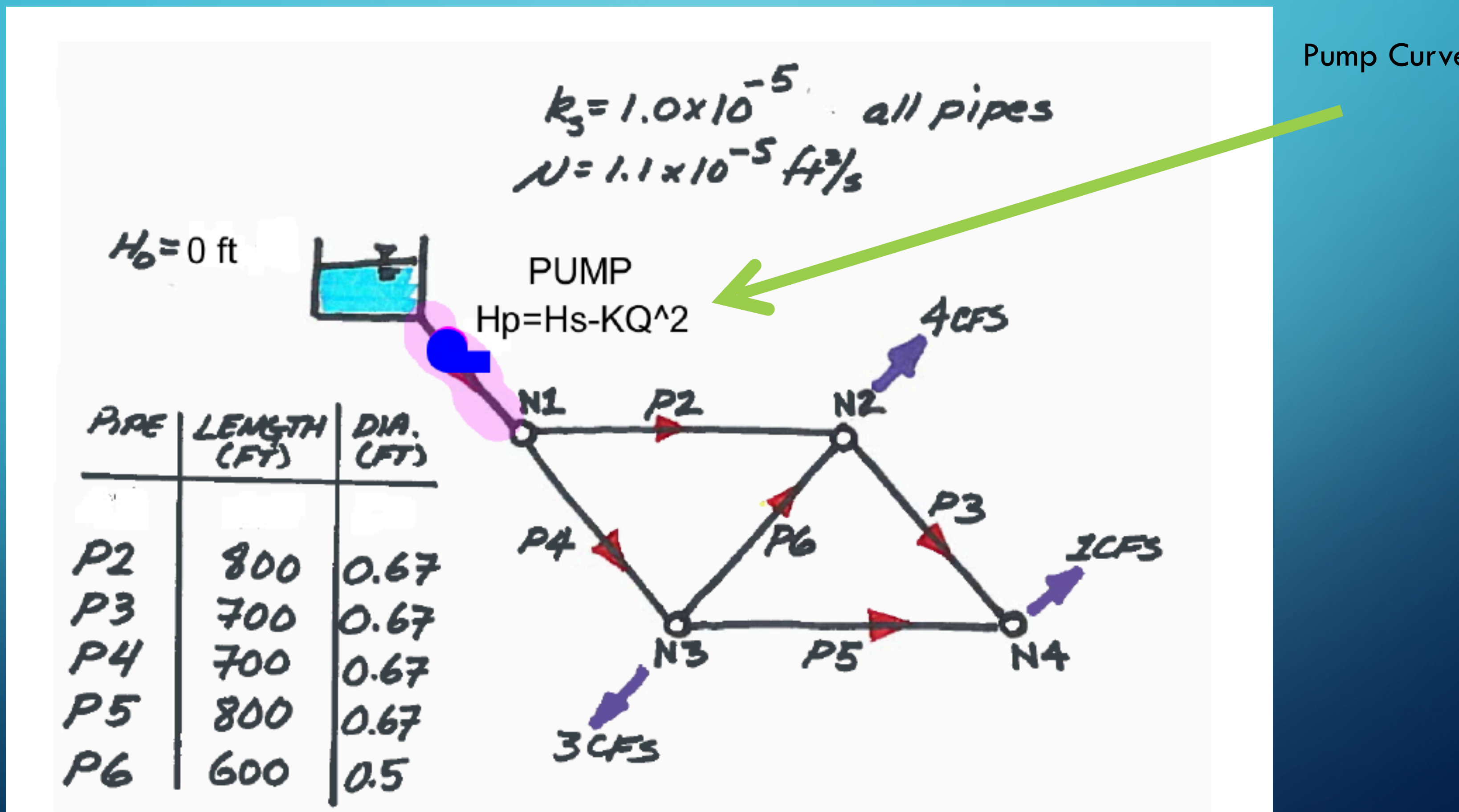
- A pump is treated as a link that adds head in a predetermined flow direction rather than removing head
- So it will replace a link equation somewhere in the network model
- Typical equation structure is

$$h_p = H_{\text{shutoff}} - K_{\text{pump}} Q^{\text{exponent}}$$

NETWORK ANALYSIS WITH A PUMP

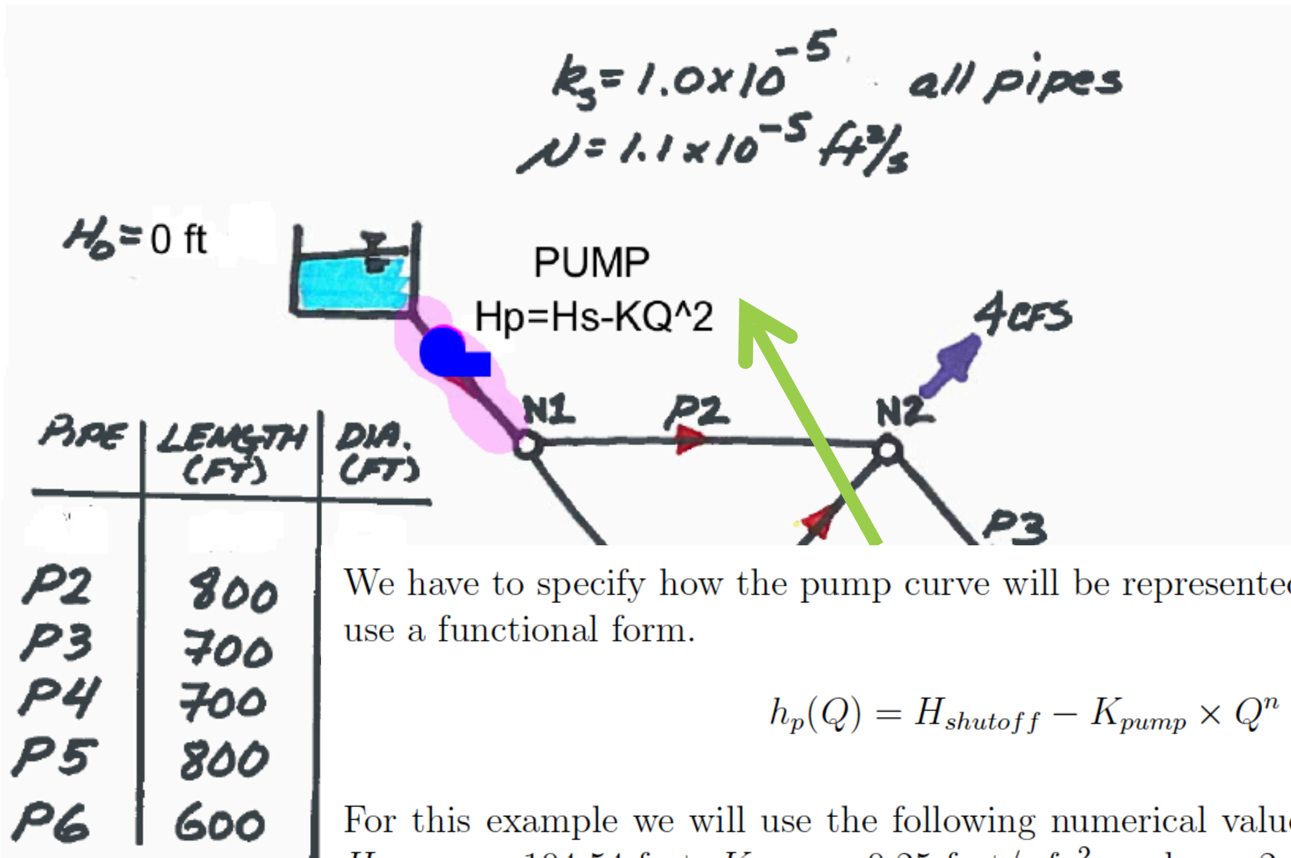
- Use the prior example as the network, but will replace the supply pipe with a pump and see how that changes the system of equations that need to be solved.

NETWORK ANALYSIS WITH A PUMP



PIPE	LENGTH (FT)	DIA. (FT)
P2	800	0.67
P3	700	0.67
P4	700	0.67
P5	800	0.67
P6	600	0.5

NETWORK ANALYSIS WITH A PUMP



We have to specify how the pump curve will be represented. In this example we will use a functional form.

$$h_p(Q) = H_{shutoff} - K_{pump} \times Q^n \tag{105}$$

For this example we will use the following numerical values for the pump function: $H_{shutoff} = 104.54$ feet, $K_{pump} = 0.25$ feet$/cfs^2$, and $n = 2$.

NETWORK ANALYSIS WITH A PUMP

- The actual pump curve function is built in a similar fashion as the loss terms from the previous lesson.

$$h_p(Q) = [H_{shutoff}/|Q| - K_{pump} \times |Q|]Q$$

- Again sign conveys direction
- We should test for the implicit divide-by-zero that can occur as Q=0, and just use the shutoff head at that value

NETWORK ANALYSIS WITH A PUMP

- Here is how the "equations" change : Original Problem

$$
\left(\begin{array}{cccccccccc}
1 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
\hline
-L_1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & -L_2 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\
0 & 0 & -L_3 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \\
0 & 0 & 0 & -L_4 & 0 & 0 & 1 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & -L_5 & 0 & 0 & 0 & 1 & -1 \\
0 & 0 & 0 & 0 & 0 & -L_6 & 0 & -1 & 1 & 0
\end{array}\right)
\left(\begin{array}{c}
Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \\ Q_5 \\ Q_6 \\ H_1 \\ H_2 \\ H_3 \\ H_4
\end{array}\right)
=
\left(\begin{array}{c}
0 \\ 4 \\ 3 \\ 1 \\ \hline -100 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0
\end{array}\right)
$$

NETWORK ANALYSIS WITH A PUMP

- Here is how the “equations” change : Change Supply Head

$$
\left(\begin{array}{cccccccccc}
1 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
\hline
-L_1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & -L_2 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\
0 & 0 & -L_3 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \\
0 & 0 & 0 & -L_4 & 0 & 0 & 1 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & -L_5 & 0 & 0 & 0 & 1 & -1 \\
0 & 0 & 0 & 0 & 0 & -L_6 & 0 & -1 & 1 & 0
\end{array}\right)
\left(\begin{array}{c}
Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \\ Q_5 \\ Q_6 \\ H_1 \\ H_2 \\ H_3 \\ H_4
\end{array}\right)
=
\left(\begin{array}{c}
0 \\ 4 \\ 3 \\ 1 \\
\hline
\cancel{100} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0
\end{array}\right)
$$

-0

NETWORK ANALYSIS WITH A PUMP

- Here is how the "equations" change : Change Supply Pipe to Pump

Insert Head Function

$$h/Q = [H_{shutoff}/|Q| - K_{pump} \times |Q|]$$

$$\begin{pmatrix} 1 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & & & & & & & & & \\ 0 & & & & & & & & & \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \hline -L_1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & -L_2 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & -L_3 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & -L_4 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -L_5 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & -L_6 & 0 & -1 & 1 & 0 \end{pmatrix} \begin{pmatrix} \\ \\ \\ Q_4 \\ Q_5 \\ Q_6 \\ H_1 \\ H_2 \\ H_3 \\ H_4 \end{pmatrix} = \begin{pmatrix} 0 \\ \\ \\ 1 \\ \hline - \; 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

NETWORK ANALYSIS WITH A PUMP

- Here is how the “equations” change : New Model Equations

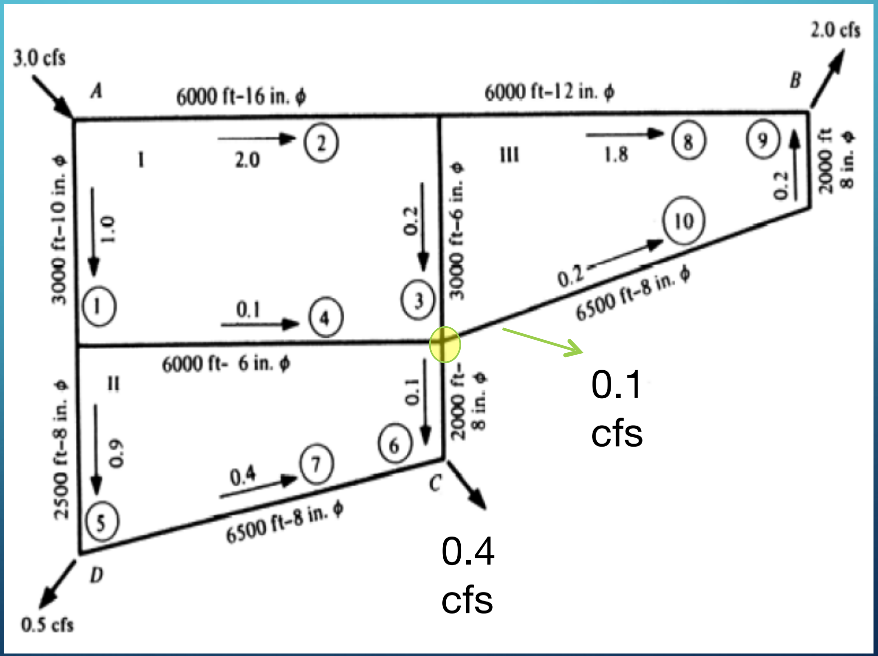
$$
\begin{pmatrix}
1 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
\hline
\boldsymbol{h/Q} & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & -L_2 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\
0 & 0 & -L_3 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \\
0 & 0 & 0 & -L_4 & 0 & 0 & 1 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & -L_5 & 0 & 0 & 0 & 1 & -1 \\
0 & 0 & 0 & 0 & 0 & -L_6 & 0 & -1 & 1 & 0
\end{pmatrix}
\begin{pmatrix}
Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \\ Q_5 \\ Q_6 \\ H_1 \\ H_2 \\ H_3 \\ H_4
\end{pmatrix}
=
\begin{pmatrix}
0 \\ 4 \\ 3 \\ 1 \\
\hline
- \quad 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0
\end{pmatrix}
$$

NETWORK ANALYSIS WITH A PUMP

- Now, encode the new model into a solution tool and proceede.

NETWORKS – WHY MODEL?

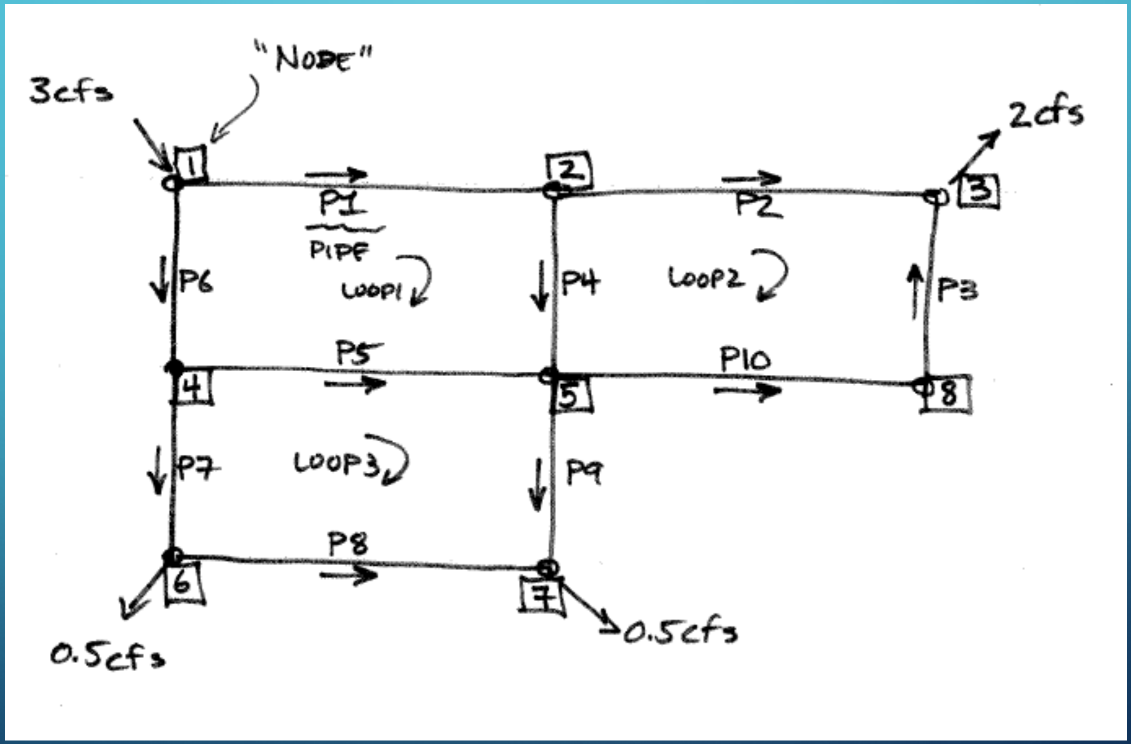
- What is the flow distribution if the demands are changed as shown?



NETWORK MODELING PROTOCOL

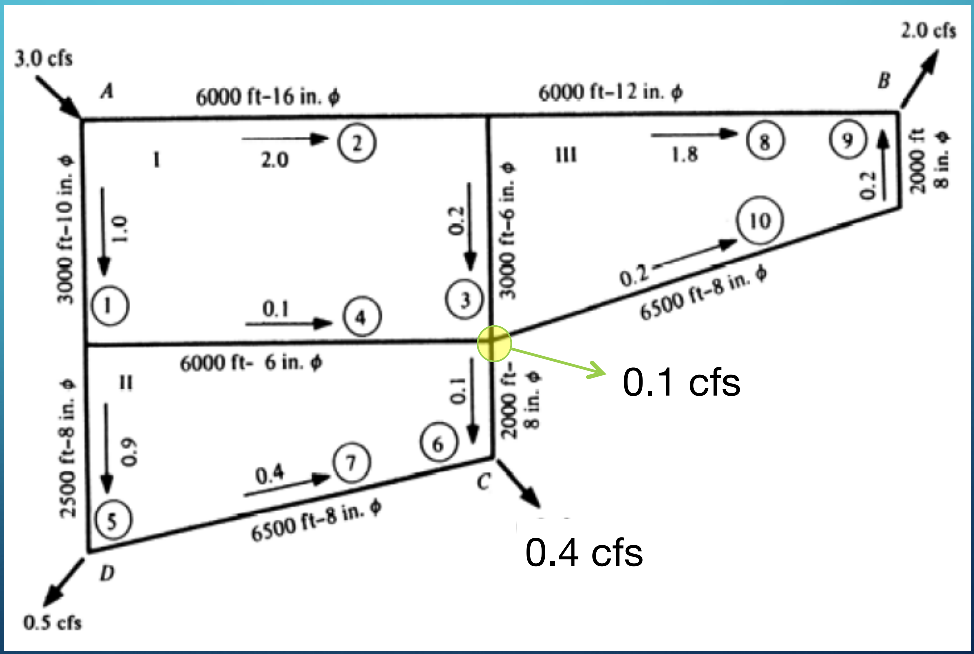
- Sketch the network
    - Label the nodes, links, supply, and demands
- Build the node-arc incidence equations
    - Continuity at each node
- Build the pipe head-loss equations
    - Head at start node, minus head loss equals head at end node
- Build and run a computation tool (spreadsheet)
    - Make an initial guess of flow rates
    - Execute the computation tool, update guesses, continue until have flow balance and zero error term

NETWORK MODELING PROTOCOL



WHAT IF?

- Once the tool is built, then "what-if" questions can be posed.
  - What is the flow distribution if the demands are changed as shown?



- Once the tool is built, then "what-if" questions can be posed.
  - As the network component count increases, the system of equations get big fast – A really complex problem would be beyond the practical limits of spreadsheet computing (the spreadsheet could do it, it would be hard to construct and maintain)