306
7-19-78

MASTER

44. 269

UCRL-52441

PIPE — A COMPUTER CODE FOR ANALYZING PIPING NETWORKS

G. A. Morris

March 3, 1978

Work performed under the auspices of the U.S. Department of
Energy by the UCLLL under contract number W-7405-ENG-48.

LAWRENCE
LIVERMORE
LABORATORY

University of California / Livermore

NOTICE MN ONLY

PORTIONS OF THIS REPORT ARE ILLEGIBLE. It has been reproduced from the best available copy to permit the broadest possible availability.

Distribution Category
UC-15

LAWRENCE LIVERMORE LABORATORY
University of California/Livermore, California/94550

UCRL-52441

PIPE — A COMPUTER CODE FOR ANALYZING PIPING NETWORKS

G. A. Morris

MS. date: March 3, 1978

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Department of Energy, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

89

PIPE — A COMPUTER CODE FOR ANALYZING PIPING NETWORKS

ABSTRACT

The Lawrence Livermore Laboratory is developing for the Nuclear Regulatory Commission a procedure to assess material control and accounting systems to prevent the possible theft of special nuclear materials (SNM). This paper describes a computer code that evaluates the piping network within a plant and identifies all possible paths for removal of SNM from specified sources. As an example we describe movement of plutonium nitrate solution from storage tanks to various exit points and give simple approximate fluid-flow calculations for each path.

INTRODUCTION

The first step in assessing material control and accounting systems is the identification of potential targets. We define a target as a given type of SNM at a certain location in a plant. We first determine all locations of SNM in the plant and rank them according to possible attractiveness to the adversary. Attractiveness is simply a measure of the effort required to prepare the SNM for malevolent use; we consider the material's chemical species, concentration, radiation hazard, and physical state. For example, plutonium-239 would be more attractive than uranium ore.

After ranking the material, we determine all normal and potential material access points. Potential access points are those into which SNM does not normally flow but into which it could be made to flow by applying pressure or a vacuum, or by changing the plant's configuration. At this point in our procedure, therefore, a plant's vulnerability to diversion can be preliminarily assessed and, in the case of a proposed plant, probably reduced by minor design changes.

This report describes a computer program that evaluates a piping network and determines all possible paths for moving material from one place to another in the network. Each path is identified by showing the pipe sections

and valves through which material passes. A simple, approximate, fluid-flow calculation is done for each path and the flow rate is included as part of the program output for that path. A table showing the number of paths that each valve is part of is also part of the output. The calculated flow rate can help determine which paths may be the most attractive for diversion, while the valve-use table can show which valves should be monitored to detect diversion attempts.

BACKGROUND

Overall assessment of safeguards and material control in any facility usually begins by evaluating small parts and then integrating these into the whole. A computer program was written to help assess the interconnections and material removal paths for any liquid handling facility by providing the following:

- All possible paths from a starting point, usually a storage tank, to an exit point from where SNM can be withdrawn and diverted.
- An approximate calculation of the flow rate for each path, to give an indication of the relative attractiveness for diversion of the different paths.
- A valve-use table showing each valve in the network and the number of paths it is part of to help select valve monitor requirements.

One part of a fuel reprocessing plant that must be assessed is the storage area for plutonium. The test-bed design under consideration at LLL, provides for the storage of plutonium nitrate solution in an array of tanks. A module consists of six tanks with the interconnecting piping, valves, and pump necessary to transfer plutonium solutions into storage, mix them between tanks, add cold chemicals, and transfer them out.

INPUT

The main program input is a description of the piping network to be analyzed. The network is divided into segments by designating arbitrary points on the network as nodes and giving them numbers. Usually between any two nodes there is a valve or a long length of pipe of special interest. No two nodes can have the same number and the numbers 999 and 9999 should not be used because they have a specific meaning and will be described later. Any valve of interest is identified by a number; any pump in the system is identified by a number greater than 2000.

The section of pipe between two nodes (called a link) can have either one-way flow from one node to another or two-way flow. The design use of each pipe can be specified, i.e., plutonium transfer, vent line, or air line.

For each section of pipe, the length in meters and inside diameter in centimeters is input. All the above input comes from a file labeled IN. Some additional input from the teletype is needed and consists of the following:

1. Number of data cards to be read from data file.
2. Number of starting nodes to consider.
3. Starting node numbers.
4. Elevation (m) above arbitrary datum of each starting node.
5. Liquid level in the tank above each starting node (m).
6. Cross section of storage tank above each starting node (m^2).
7. Elevation of exit nodes (m).
8. Fluid density (g/cm^3).
9. Inside diameter (cm) of pipe to be used as reference for fluid flow calculations.

The input format from the file consists of a line of input in the order NOD1, NOD2, NVALV, LINKW, NAMP, DL, and D. They are defined as follows:

- NOD1 and NOD2. Numbers of connected nodes.
- NVALV. Valve number between NOD1 and NOD2, greater than 2000 if a pump.
- LINKW. Flow direction: 0 = two way; 1 = flow only from NOD1 to NOD2; 2 = flow only from NOD2 to NOD1; 3 = no flow possible.
- NAMP. Design use of pipe section between NOD1 and NOD2: 0 = normal plutonium flow; 1 = vent line; 2 = air line.

- DL. Pipe section length (m).
- D. Pipe section diameter (cm).

An exit node must be connected to a node numbered 999. If an examination of the piping diagram determines that a section of pipe cannot be filled with liquid but can only be used as a vent, then LINKW for the section is set = 3 and NAMP = 1. This section must be connected to 999 at one end to show an exit node and it must not contain a valve. The vent acts as a vacuum break to prevent fluid movement and is flagged by the program. A vent pipe on a surge tank with the vent opening above both inlet and outlet is an example.

Figure 1 presents a simple network that has been divided by designating and numbering nodes. The input for each link is shown in detail in Fig. 2 .

Fig. 1. Example network with nodes indicated.

<u>NOD1</u>	<u>NOD2</u>	<u>NVALV</u>	<u>LINKW</u>	<u>NAMP</u>	<u>DL</u>	<u>D</u>
4	10	2002 Pump	0 Two-way	0 Plutonium use normal	10	2
10	12	4 Valve number	0	0	10	2
12	16	0 No valve in line	0	0	10	2
16	999	0 Exit	3 No flow	1 Vent line	0 Exit	0 Connections have no dimensions for fluid flow
16	18	7 Flow only from NOD1 to NOD 2 (16 to 18)	1	0	10	2
18	22	0	0	0	10	2
22	23	0	0	2 Airline	10	2
22	24	6 Flow only from NOD2 to NOD1 (24 to 22)	2	0	10	2
22	999	0 Exit node	0	0	0	0
23	999	0	0	0	0	0

Fig. 2. Input for example network (Fig. 1).

When entering data, NOD1 should always have a smaller number than NOD2, except when connecting a node 999 which is always NOD2. This procedure precludes entering the same path twice i.e., path from node 4 to node 10 is the same as node 10 to node 4 but both should not be entered. The order in which the lines of data are read does not matter; the input listing above line 2 and line 5 could be switched.

COMPUTATION PROCEDURE

General

The calculations done by the program can be divided into different functions that are performed in the following order:

1. Read input from both disk file and from the teletype.
2. Initialize constants and arrays.
3. Beginning with designated starting node, search NOD1 and NOD2 arrays to find a path to an exit point.
4. Do fluid-flow calculations on the path found in step 3.
5. Write output for the path and update counters.
6. Back up one node from the last node used in the present path, reset some flags, and try to find a new path as in step 3.
7. When no new paths can be found because program has backed up to the original starting node either begin at new starting node from input, or terminate program.

Input data read from the disk file are stored in arrays. All seven variables from one line of input have the same index, thereby allowing cross reference of the data. These arrays are filled starting in position 2 of each array. NOD1(1) position is used to store the number 9999 which is a flag used by the program to know when all possible paths from the starting node have been found. NOD2(1) is used to store the starting node for the present calculations and is updated when a new starting node or new calculations are to be done.

Path Finding Search

The method to find a path in step 3 is shown in Fig. 3. The program looks for one link in the path at a time and keeps track of all the links that comprise a path. Given a node number to use as the starting end of a link (NOWND), the program searches the NOD1 array until NOD1(I) and NOWND are the same. At this point it is known that NOD2(I) connects to NOD1(I) - the link starting node NOWND; thus this is one link in the path. To find the next link, NOWND is set equal to NOD2(I) through NODST and the search begins again. The program keeps track of the path by storing connected nodes in the

Fig. 3. Partial program flowsheets.

NODES(I) array. When a link is found that connects to a node number 999, it is known that an exit, and thus a complete path, has been found.

The calculations are performed as follows (the routine is also shown in Fig. 3). At statement 2, NOWND is set = 9999 the first time through or, if not the first time, is equal to the last node in the path being searched and has been set internal to the "DO 100" loop.

The first IF test goes through the NOD1 array until a NOD1(I) is found that matches NOWND. This is then a possible link to a NOD2(I). Then, at 10, LINKW(I) is checked to see if flow is possible from NOD1(I) to NOD2(I) that is LINKW(I) $\neq$ 2. If flow is not possible, LINKG(I) is set = 1 at (2310); if flow is possible, the program does not set LINKG(I) = 1, but goes to 2311 and saves the value of NOD2(I) in NN.

The next IF test checks LINKG(I) to see if the link from NOD1(I) to NOD2(I) is one that should be considered further. LINKG(I) = 1 means the link cannot be used in the path either because it has been used before or because input has indicated that it is always closed to flow in that direction. Thus, the program goes to 2301 through 2302 to set LINKF(I). LINKF(I) is a flag that allows the program to back up correctly from node to node by knowing which links were closed during a forward search and which formed a loop back to a previously used node.

If after 2311 LINKG(I) $\neq$ 1, that link is a possible one for the path; a test is then made to see if NN equals 999. If so, an exit has been found and the path is complete. The valve arrays are then updated and the program goes on to 800 for fluid-flow calculations and output.

If NN is not equal to 999, then a check is made in statements 15 through 20 to determine if the present node NN, is already in the path. If it is, the program goes back to 100, increments I, and starts again. If not, that link is good, and thus from 20 on arrays are updated, the link is included in the path, and the program goes back to find another link.

If no match can be found searching NOD1 array, the NOD2 array is searched until either no match or an exit is found. When an exit is found, a flow calculation (described later in this report) is done on the path and the path is printed with flow rate and other information.

At this point, the program backs up one node in the present path and tries to find a new exit path. If it can, it prints out the path; if not, it backs up one more node and looks for another path. This process continues until the program returns to the starting node and finds no new paths; it then ends or restarts at a new point.

The nodes that comprise a path are kept in array NODES(I). Various other arrays keep track of valves, which links have already been used, possible direction of flow, etc.

Fluid Flow

Fluid-flow calculations in this program help find the relative attractiveness of the various paths for diversion; they are not intended for other use. Because only relative numbers are needed, the most simple type of calculation was chosen. The equation to be solved is the modified Bernoulli equation:

$$h_1 = \frac{v_1^2}{2g} + \frac{p_1}{\gamma} + h_2 + h_f ,$$

where 1 and 2 refer to the fluid interface in the tanks and the exit point, respectively, and f is friction head losses.

The term h_f represents the head loss caused by friction in the pipes and head loss through valves and fittings. The equation to be solved then becomes:

$$h_1 = \frac{v_1^2}{2g} + \frac{p_1}{\gamma} + h_2 + f \frac{L}{D} \frac{v_2^2}{2g} , \quad (1)$$

or

$$h_1 - h_2 - \frac{p_1}{\gamma} = \frac{v_2^2}{2g} + f \frac{L}{D} \frac{v_2^2}{2g} ,$$

where

f is friction factor, h is the height above some datum (m), L is length of piping (m), D is diameter of piping, g is 9.802 m/s^2 , γ is specific weight of liquid, p is pressure above or below pressure datum, and V is fluid velocity (m/s).

As a first step to the solution of Eq. (1), the method of equivalent length is used to change all pipes to the same inside diameter. For example, a pipe of length L_1 and inside diameter D_1 can be converted to an equivalent pipe of diameter D_2 and the length L_2 using the relationship

$$L_2 = L_1 \frac{f_1}{f_2} \left(\frac{D_2}{D_1} \right)^5 ,$$

if f_1/f_2 is known. With this method all pipes in a path are converted to an equivalent length of pipe with an inside diameter equal to the input reference pipe diameter.

From the friction factor relationship for laminar flow we can find f_1/f_2 as follows:

$$f = \frac{64}{\text{Reynolds Number}} = \frac{64 \nu}{VD \rho} ;$$

therefore, $V_1 D_1 f_1 = V_2 D_2 f_2 ,$

and

$$\frac{f_1}{f_2} = \frac{V_2 D_2}{V_1 D_1} .$$

Because total flow rate is constant,

$$Q \text{ total} = V_1 \times \frac{\pi D_1^2}{4} = V_2 \times \frac{\pi D_2^2}{4}$$

$$\frac{V_2}{V_1} = \frac{D_1^2}{D_2^2} ,$$

and

$$\frac{f_1}{f_2} = \frac{D_1}{D_2} .$$

It follows that

$$L_2 = L_1 \left(\frac{D_2}{D_1} \right)^4 . \quad (2)$$

For turbulent flow, a similar calculation of f_1/f_2 is possible. Using the turbulent-flow friction-factor relationship,

$$f = \frac{0.316}{(\text{Reynolds number})^{1/4}} .$$

It follows that

$$L_2 = L_1 \left(\frac{D_1}{D_2} \right)^{0.25} \left(\frac{D_2}{D_1} \right)^5 . \quad (3)$$

For each pipe section in a path, the equivalent pipe length is calculated by first converting the valves to equivalent lengths of pipe using the equation $L(\text{eq}) = \frac{KD}{f}$. This length is then added to the pipe section length and Eq. (2) is used to convert to total equivalent length of pipe of reference diameter D_2 for that link.

A value of $K = 12$ is assumed equal to the value of a globe valve plus a standard tee. A value of $f = 0.05$, typical for flow in pipes, is also assumed to find equivalent lengths for valves.

The above calculations are done for each section of pipe in the path and the sum of these is the total equivalent length of pipe in the path. This information is put into Eq. (1) with f for laminar flow; the equation is then solved for fluid velocity V_2 . A test of the Reynolds number is made and then either a laminar or a turbulent flow calculation is done, depending on whether the Reynolds number is less than or greater than 2100.

To solve Eq. (1) for V_2 , the program calculates the total driving head for fluid flow as follows:

$$\text{DELH} = (\text{HINLET} + \text{HLIQQ}) - \text{HEXIT},$$

where

DELH = head due to difference in fluid level (m),

HINLET = height above some datum plane of the node at the bottom of a tank (m),

HLIQQ = height of fluid level above HINLET (m),

HEXIT = height of exit node above datum plane (m).

Using the head difference calculated above, the total driving head is found by adding the contribution of a vacuum applied at the exit:

$$\text{DELHT} = \text{DELH} + 101. / (9.802 \times \text{Specific gravity of Liquid})$$

$$= h_1 - h_2 - \frac{P_2}{\gamma} \text{ in Eq. (1) ,}$$

where

DELHT = total driving head for fluid flow where the second term is the vacuum contribution.

A fluid-flow time step of one second is assumed, the height of liquid in the tank is adjusted (HDEL) to reflect liquid withdrawn, and a new calculation is performed. The calculation is iterated 60 times and the resulting sum is the flow rate per minute. Two assumptions are made in the calculations: the viscosity of water is used, and in Eq. (1) $\frac{L}{D} > 100$.

Output

The primary output is a table that describes the path found by listing each link in the path by the starting and ending node of each link. The valve number and the design use for that link are also indicated. The flow rate through the path is identified with the added information of equivalent pipe length of path, driving head for flow, height of exit, inlet, and liquid, and the amount the head changed in the last calculation.

If a pump is in the path or if flow with only vacuum is not possible, a statement to show this is printed out. If any node in the path is connected to a direct vent line - i.e., one without a valve that could act as a vacuum break - a note is also printed out.

Example 1

The piping network in Figs. 4a and b shows how to break a path into sections by choosing nodes and numbering valves. The file input for this network would be as follows:

```
3,7,0,0,0,10,3,  
7,10,0,0,0,10,3,  
7,12,1,0,1,10,3,  
6,7,3,0,0,10,3,  
6,400,0,0,0,10,3,  
6,30,0,0,0,10,3,  
4,6,0,0,0,10,3,  
4,30,2001,1,0,10,3.  
30,999,0,0,0,0,0,  
12,999,0,0,0,0,0,  
30,400,0,0,0,10,3,
```

The teletype input for this example would be as follows (lines printed by the program are indicated by arrows):

```
PIPE  
+ PROGRAM TO COMPUTE PATHS IN A FLUID FLOW NETWORK  
+ INPUT IS FROM FILE IN - PIPE DIMENSIONS IN WHOLE NUMBERS  
+ HOW MANY CARDS FOR DATA?  I3  
+ 123  
  11  
+ NNUM AND NL1=    77    11  
+ IS OUTPUT TO FILE ONRC?YES?  
  YES  
+ HOW MANY STARTING NODES?  I3 NOT MORE THAN 15  
+ 123  
  2  
+ LIST STARTING POINTS SEPARATED BY COMMA END WITH COMMA  
  : 10, 3,  
+ LIST ELEVATION OF EACH STARTING POINT IN METERS  
  : 0,0,  
+ LIQUID LEVELS ABOVE STARTING POINTS IN METERS SEPARATE-END WITH COMMA  
  : 10,10,
```

Fig. 4a. Sample problem, No. 1 network.

Fig. 4b. Sample problem, No. 1 node and valve identification.

```

+ LIST TANK CROSS-SECTIONS ABOVE STARTING POINTS IN SQUARE METERS SEPA
+ RATE AND END WITH COMMAS
  : 1000,1000,
+ AFTER EXIT NODE LIST ITS ELEVATION IN METERS F8.2 FORMAT
+   30
12.
+   12
25.
+ LIST DENSITY OF LIQUID IN GR PER CC F FORMAT
  1.
+ WHAT ID PIPE IS REFERENCE?CENTIMETERS F8.2 FORMAT
  2.54
+ PATH NUMBER=      8 ALL PATHS FOUND PROGRAM ENDED
+ ALL DONE

```

In the above listing, when the program expects an input as an integer or I3 format, it prints out 123 so the input can be correctly spaced when it is entered. The output line with NNUM and NL1 checks to ensure the input file was correctly read. There are seven entries per card and in the example; 11 cards were read so the program shows 77 data points read from 11 cards.

The program outputs results either to the teletype or to a disk file called ONRC, depending on the answer to the question about output. YES gives output to file, NO puts output on teletype.

When the program prints out "After Exit Node List its Elevation in Meters F8.2 Format", the next number printed is an exit number as defined from input. The program waits for the input of elevation then prints out another exit node. Thus, node 30 has an elevation of 12 m and node 12 has an elevation of 25 m.

After the last input (i.d. pipe is reference), the program sets the variable NOWND = NODST = 10 and starts searching the NOD1 array for a node 10. Because there is none, it then searches NOD2 array and finds a node 10 connecting to a node 7. Nodes 10 and 7 are stored in an array called NODES, NOWND

is set to 7, and the search starts again in the NOD2 array. The input shows node 7 connects to 3 but that 3 does not connect to anything else. The program then backs up to node 7 and starts searching the NOD1 array again. Node 7 is found to connect to 12 and 12 to 999, showing an exit node. Thus a complete path has been found and is printed in the following manner:

NUMBER OF PATHS FOUND= 1

REFERENCE PIPE ID== 2.54 CENTIMETERS

FOLLOWING PATH FOUND

NOD1	NOD2	VALVE NUMBER	PIPE USE
9999	10	0	
10	7	0	
7	12	1	VENT
12	999	0	

FLOW THRU PATH= .0E+00 LITERS PER MIN

====FLOW THRU PATH NOT POSSIBLE WITH ONLY VACUUM=====

TOTAL EQUIVALENT LENGTH= 1.4E+01 METERS

DRIVING HEAD FOR FLOW INCLUDING VACUUM= -4.70E+00 METERS

HEXIT, HINLET, HLIQ, HDEL=

2.50E+01	0.E+00	1.00E+01	.0E+00
----------	--------	----------	--------

We can see that no flow is possible along this path with only gravity flow and vacuum.

When the printout is complete the program sets some flags so it will not follow the same path again and also sets NOWND = 12. It finds no new paths, resets NOWND = 7, and starts searching again. The first good new path it finds has NOD2 = 7 and NOD1 = 6. Then, with NOWND = 6, a next link is to NOD1 = 4; NOWND is set to 4 and the next path becomes 4 to 30 and 30 to 999, which is an exit. The following path is printed out:

NUMBER OF PATHS FOUND= 2

REFERENCE PIPE ID== 2.54 CENTIMETERS

FOLLOWING PATH FOUND

NOD1	NOD2	VALVE NUMBER	PIPE USE
9999	10	0	
10	7	0	
7	6	3	
6	4	0	
4	30	2001	
30	999	0	

==THIS PATH HAS PUMP IN IT=====

FLOW THRU PATH= 8.77E+01 LITERS PER MIN TURBULENT

TOTAL EQUIVALENT LENGTH= 2.59E+01 METERS

DRIVING HEAD FOR FLOW INCLUDING VACUUM= 8.30E+00 METERS

HEXIT, HINLET,HLIQ,HDEL=

1.20E+01 .9E+00 1.00E+01 1.46E-06

This searching continues until all paths are found as below, then a table showing the number of times each valve was passed is printed out.

NUMBER OF PATHS FOUND= 3

REFERENCE PIPE ID== 2.54 CENTIMETERS

FOLLOWING PATH FOUND

NOD1	NOD2	VALVE NUMBER	PIPE USE
9999	10	0	
10	7	0	
7	6	3	
6	400	0	
400	30	0	
30	999	0	

FLOW THRU PATH= 9.62E+01 LITERS PER MIN TURBULENT

TOTAL EQUIVALENT LENGTH= 2.20E+01 METERS

DRIVING HEAD FOR FLOW INCLUDING VACUUM= 8.30E+00 METERS

HEXIT, HINLET,HLIQ,HDEL=

1.20E+01 .0E+00 1.00E+01 1.60E-06

NUMBER OF PATHS FOUND= 4

REFERENCE PIPE ID== 2.54 CENTIMETERS

FOLLOWING PATH FOUND

NOD1	NOD2	VALVE NUMBER	PIPE USE
9999	10	0	
10	7	0	
7	6	3	
6	30	0	
30	999	0	

FLOW THRU PATH= 1.10E+02 LITERS PER MIN TURBULENT
TOTAL EQUIVALENT LENGTH= 1.75E+01 METERS
DRIVING HEAD FOR FLOW INCLUDING VACUUM= 8.30E+00 METERS
HEXIT, HINLET,HLIQ,HDEL=
1.20E+01 .0E+01 1.00E+01 1.83E-06

VALVE #	TIMES USED
1	1
3	3
2001	1

Because two starting nodes were input, the program restarts with NODST = 3 and searches for all paths as before.

Example 2

This example uses part of the test bed design of the plutonium nitrate storage area as its network. Figure 5 shows the network with the nodes and valves numbered for input to the computer. Input for the program is shown in Fig. 6 and the output for two of the six possible starting points is included as Appendix B.

The output of the program (Appendix B) shows there are 9 paths in the network that start at node 31 - the bottom of one of the tanks - and go to exit points. These paths can be traced in Fig. 5 using the output tables and node numbers.

For path 1 we can see that the path has a pump, is vented, and that no flow is possible only because of vacuum. The flow rate for this path is zero because the exit is so high relative to the liquid height that the total driving head for flow is negative. This is also shown by the comment of "flow thru path not possible with vacuum only." The total equivalent length of pipe

Fig. 5. Sample problem, No. 2 network and nodes.

```

1,31,0,1,0,1,1
1,7,13,0,0,12,3,
1,8,7,0,0,10,2,
2,32,0,1,0,1,1,
2,7,14,0,0,12,3,
2,11,8,0,0,10,2,
1,999,0,3,1,0,0,
3,33,0,1,0,1,1,
3,7,15,0,0,17,3,
3,17,9,0,0,11,1,
4,7,16,0,0,17,3,
4,34,0,1,0,1,1,
4,16,10,0,0,11,1,
5,35,0,1,0,1,1,
5,7,17,0,0,30,2,
5,23,11,0,0,4,1,
6,36,0,1,0,1,1,
6,7,18,0,0,10,1,
6,26,12,0,0,15,3,
7,20,19,0,0,30,2,
7,14,20,0,0,5,1,
7,15,21,0,0,1,1,
8,9,30,0,0,11,1,
9,10,23,0,2,17,3,
10,999,0,0,0,0,0,
11,12,31,0,0,11,1,
12,13,24,0,0,17,3,
13,999,0,0,0,0,0,
14,29,37,0,0,12,1,
14,30,2045,0,0,15,3,
16,21,33,0,0,12,3,
17,18,32,0,0,10,2,
18,19,25,0,0,12,3,
19,999,0,0,0,0,0,
20,999,0,0,0,0,0,
21,22,26,0,0,10,2,
22,999,0,0,0,0,0,
23,24,34,0,0,15,3,
24,25,27,0,1,10,1,
25,999,0,0,0,0,0,
26,27,35,0,0,4,1,
27,28,28,0,0,30,2,
28,999,0,0,0,0,0,
29,999,0,0,0,0,0,
30,31,1,0,0,10,2,
30,32,2,0,0,1,1,
30,33,3,0,0,1,1,
30,34,4,0,0,1,1,
30,35,5,0,0,1,1,
30,36,6,0,0,1,1,
30,42,22,0,0,15,2,
40,41,29,0,0,30,2,
40,42,36,0,0,1,1,
41,999,0,0,0,0,0,
CCCCCCC
CCCCCCC
CCC THIS FILE IS FOR USE WITH NRCFFL PROGRAMS. IT IS A CLOSE
CCCC DATA FILE FOR THE TEST DESIGN STORAGE FACILITY
CCC BUT THE PIPE DIMENSIONS ARE ASSUMED AND ARE NOT REAL

```

Fig. 6. Input for example 2.

of the reference inside diameter used in the fluid-flow calculation is also printed out with the height of the exit (HEXIT), inlet (HINLET), liquid (HLIQQ), and the amount the liquid level in the tank changed during the last flow timestep (HDEL).

The flowrates for all nine paths show that path 2 gives the highest potential flowrate of 16.2 liters/min; the next highest is path 9.

Following path 9 is a table that shows the number of times valves were used when fluid was assumed to start at node 31. Valve 1 was used in all paths, pump 2045 in eight paths, and valve 20 in seven paths. All other valves were either used only once or not at all. Thus, to detect attempted diversion we would monitor valves 1 and 20 and pump 2045. Monitoring other valves may not be justified for paths originating at node 31.

At this point the program restarts beginning with node 32. Paths 10 through 18 are found and the information is printed out. The last part of the output is a summary of input data used for this run of the program. Appendix C is a listing of the program.

Limitations and Comments

This program as shown in Appendix C contains some programming statements that are probably unique to LLL. Most of these statements are concerned with input-output and can easily be reprogrammed for use elsewhere. The specific statements are:

- Program Pipe sets up some of the input output conventions used at LLL can be replaced with needed cards at new computers when input-output statements are changed.

- CALL FFINC() calls a subroutine that allows field free input to be used. It can be replaced by a read statement in a DO loop when the input is formatted.

- CALL DEVICE() sets up a file on disk to be used as an output file. This file is labeled ONRC and can be replaced by any means of selecting output.

In READ or WRITE statements, when the device indicator is 59, teletypewriter is the output or input device. Other statements used are either FORTRAN IV compatible or can be easily replaced by such statements.

The number of nodes - size of the network - that the program can handle is determined mainly by the size of arrays set in dimension statements. These sizes can be increased to augment the size of the network, and, if needed, more efficient use of space could be made by some reprogramming effort.

The fluid-flow calculations are for relative flows only and are not good for any type of process flow use. This is because of the assumptions needed to make the calculations simple and very general.

APPENDIX A

DEFINITIONS OF DIMENSIONED VARIABLES

AREA(K)	Cross sectional area of the tank above starting node INODST(K).
D(I)	Diameter of pipe section between NOD1(I) and NOD2(I) in centimeters. (INPUT)
DL(I)	Length of pipe section between NOD1(I) and NOD2(I) in meters. (INPUT)
FLOWT(K)	Contains labels for type of flow: Laminar or Turbulent.
HIN(K)	The height of the starting nodes (INODST(K)) above or below some arbitrary datum.
HLIQ(K)	Height of liquid above starting node for that liquid volume. Starting node is usually bottom of a tank and the liquid level is measured from that.
HOUT(I)	Height of the exit nodes above or below datum plane.
INODST(K)	The node to be used as starting points in the calculations.
IX(K)	A dummy variable used only because the method of input to this program could be eliminated by use of a new input routine for the program. The size of this array must be seven times as large as the arrays for input of NOD1 or NOD2.
LINKC(I)	A flag that indicates whether a link from NOD1(I) to NOD2(I) is not open for flow. This link cannot be in a fluid path but it can be a vent line. LINKC(I) = 1 link is closed for entire run of program.
LINKF(I)	A flag that, when set, tells the program why a certain LINKG(I) is not useable and whether it can be reset to be used as the program backs up to find a new path, after all paths from one node have been found. This flag prevents looping but allows links that could not be used along a certain path to be used on the next path. LINKF(I) = 1 means the link was used going from NOD1 to NOD2, LINKF(I) = 2 means the link was used going from NOD2 to NOD1.

LINKG(I) A flag that shows whether a link is useful in the particular path being considered. This flag is changed in the program when a new path is being computed. **LINKG(I)** = 1 means link is not to be used at that point of the search, but it may have been used in the same path before or **LINKG(I)** = 1.

LINKW(I) A variable that sets possible direction of flow in the link between **NOD1(I)** and **NOD2(I)**. If = 0 two-way flow; if = 1 only from **NOD1(I)** to **NOD2(I)**; if = 2 only from **NOD2(I)** to **NOD1(I)**; if = 3 means no flow possible and is used for all lines that are only vent lines to atmosphere. (INPUT)

NAMP(I) Sets the normal intended use of the link from **NOD1(I)** to **NOD2(I)**. If = 0 is a plutonium line, if = 1 is VENT line if = 2 is an AIR line. (INPUT)

NNVAL(I) Array that keeps track of the valves which are in the path denoted by the **NODES** array. Used for printout purposes.

NOD1(I) Define points within network that are somehow connected,

NOD2(I) usually by a valve, pump, or pipe length. (INPUT)

NODES(K) This array keeps the node numbers in a path in the sequence in which they were found; also keeps track of which are to be printed out as a path, and which is the last node used in a path.

NPIPEN(I) An array used to store the pipe use, VENT or AIR for print-out with the links.

NPNAM(K) Contains the names VENT and AIR for pipe names.

NVALV(I) The number assigned to a valve to keep track of it. If number >2000 it is a pump. The valve carried in location **I** = **K** is the valve between **NOD1(K)** and **NOD2(K)**. (INPUT)

NVALVC(I) Array keeps track of how many different paths go through each valve or gives the number of times a valve is used in all possible paths found.

The following variables should have equal array sizes, and at least one location larger than the number of data lines in Input file: **NOD1**, **NOD2**, **NVALV**, **LINKW**, **NAMP**, **DL**, **D**, **HOUT**, **LINKG**, **LINKC**, **LINKF**, **NVALVC**, and **NODES**. The following variables array should have the same size. This size is also needed as a limit in some DO LOOPS: **INODST**, **HLIQ**, **AREA**, and **HIN**.

DEFINITION OF NONDIMENSIONED VARIABLES

ANS	User response to program question YES or NO.
AREA1	Flow area of pipe being used as reference pipe for flow calculations (m^2).
AREA2	Cross sectional area of tank above the node being used as starting node for a calculation. Is set equal to AREA(I) for calculations using the I'th starting node (m^2).
DELH	Head caused by difference in fluid level for fluid-flow calculation (m).
DELHT	Total head for fluid flow includes DELH and a vacuum contribution (m).
DEN	Density of fluid for flow calculations (g/cm^3).
DIA	Ratio of reference pipe inside diameter to inside diameter of pipe being converted to equivalent length from input: $RPID/D(I)$ used in turbulent flow calculations.
HDEL	The incremental change in the fluid level in a tank caused by fluid withdrawn the previous second. Used to adjust total driving head.
HEXIT	Height of the exit point (above datum) in a path being used for fluid-flow calculation. Is equal to HOUT(I) for calculations using the I'th exit point.
HINLET	Height of the starting node (above datum plane) being used for fluid-flow calculations. Is set equal to HIN(I) for calculations using the I'th starting node.
HLIQ	Height of liquid level above starting node for fluid flow calculations. Is set equal to HLIQ(I) for calculations using I'th starting node.
IAN	Number of starting nodes to consider in a calculation.
ICOUNT	Flag to prevent program from getting into a loop. Terminates program when equal to 1000.
IER, NNUM, NLI, INUM	Flags used by the CALL PFINC input subroutine.
IERR	Flag used by CALL DEVICE subroutine.
IFLW	Flag used to print out correct label for type of fluid flow.

IIN, IOT	Flags that determine whether input and output from the program go into teletype or disk file.
IN	Name of disk file that contains the input describing the piping network.
IREAD	Dummy used to set up the input limits from the file IN.
ISSET	Dummy used to set the correct name for pipe use into memory.
ITIME	Flag keeps track of how many starting nodes have already been used for path determination.
ITTIME	Flag that has the number of starting nodes to be considered in a particular run.
LIMIT	Dummy used as a DO LOOP limit to run through all nodes is equal to NCARD + 1 because first card of input goes into location 2 of an array.
LINK1, LINK2	Flags that keep track of last link to be used during searches for a path. Allows correct resetting of some variables after a path has been found and the program is backing up to find a new path.
NCARD	Number of cards to be read from data input.
NN	Dummy that is the last node number to have been found in a path. Changes as each link is found.
NODE1	Flag that tells if the last node in a good link was found in NODE1 or NODE2 array.
NODST	Node that is starting point for finding the next link in a path.
NOWND	Node that is currently the last value of NODST. Change only when good link is found.
NTRIAL	Number of paths found during a run up to that time.
ONRC	Name of disk file to be used for output from program.
Q	Flow during the one-second interval being calculated (m^3).
QLIT	Total flow for one-minute interval or until tank is empty (liters/min).
QTOT	Total flow to time increment being considered (m^3).
RPID	Inside diameter to be used as a reference pipe for computing equivalent lengths.
TL, TLT	Length of equivalent pipe for a link including valve (laminar, turbulent flow).

TOTL, TOTLT	Length of path up to point being considered (laminar, turbulent flow).
TPL, TPLT	Length of pipe in a link including valve (laminar, turbulent flow).
TV	Length of pipe to account for valve in link.
RE	Reynolds number
V	Flow velocity (m/s).

APPENDIX B

SAMPLE PROBLEM 2 OUTPUT

```

NUMBER OF PATHS FOUND= 1
REFERENCE PIPE ID== 2.00 CENTIMETERS
FOLLOWING PATH FOUND
  NOD1  NOD2  VALVE  NUMBER      PIPE USE
  9999  31    0
  31    30    1
  30    14    2045
  14    7     20
  7     1     13
  1     8     7
  8     9     30
  9     10    23      AIR
  10    999    0
=====
==THIS PATH HAS PUMP IN IT=====
FLOW THRU PATH= .0E+00 LITERS PER MIN
=====THIS PATH VENTED AT NODE= 1
=====FLOW THRU PATH NOT POSSIBLE WITH ONLY VACUUM=====
TOTAL EQUIVALENT LENGTH= 3.39E+02 METERS
DRIVING HEAD FOR FLOW INCLUDING VACUUM= -2.41E+00 METERS
HFXIT,HINLET,HLTD,HDEL=
  1.50E+01  2.00E+00  2.00E+00  .0E+00

```

```

NUMBER OF PATHS FOUND= 2
REFERENCE PIPE ID== 2.00 CENTIMETERS
FOLLOWING PATH FOUND
  NOD1  NOD2  VALVE  NUMBER      PIPE USE
  9999  31    0
  31    30    1
  30    14    2045
  14    7     20
  7     20    19
  20    999    0
=====
==THIS PATH HAS PUMP IN IT=====
FLOW THRU PATH= 1.91E+01 LITERS PER MIN      TURBULENT
TOTAL EQUIVALENT LENGTH= 2.20E+02 METERS
DRIVING HEAD FOR FLOW INCLUDING VACUUM= 1.45E+01 METERS
HFXIT,HINLET,HLTD,HDEL=
 -2.00E+00  2.00E+00  2.00E+00  9.62E-04

```

```

NUMBER OF PATHS FOUND= 3
REFERENCE PIPE ID== 2.00 CENTIMETERS
FOLLOWING PATH FOUND
  NOD1  NOD2  VALVE  NUMBER      PIPE USE
  9999  31    0
  31    30    1
  30    14    2045
  14    7     20
  7     7     14
  2     11    8
  11    12    31
  12    13    24
  13    999    0
=====
==THIS PATH HAS PUMP IN IT=====
FLOW THRU PATH= 7.99E+00 LITERS PER MIN      TURBULENT
TOTAL EQUIVALENT LENGTH= 5.36E+02 METERS
DRIVING HEAD FOR FLOW INCLUDING VACUUM= 7.56E+00 METERS

```

HEXIT,HTNLET,HLIQ,HDEL= 5.00E+00 2.00E+00 2.00E+00 3.99E-04

NUMBER OF PATHS FOUND= 4
REFERENCE PIPE ID== 2.00 CENTIMETERS
FOLLOWING PATH FOUND

NOD1	NOD2	VALVE	NUMBER	PIPE USE
9999	31		0	
31	30		1	
30	14		2045	
14	7		20	
7	3		15	
3	17		9	
17	18		32	
18	19		25	
19	999		0	

==THIS PATH HAS PUMP IN IT=====

FLOW THRU PATH=	1.06E+01 LITERS PER MIN	TURBULENT
TOTAL EQUIVALENT LENGTH=	5.35E+02 METERS	
DRIVING HEAD FOR FLOW INCLUDING VACUUM=	1.26E+01 METERS	

HEXIT,HTNLET,HLIQ,HDEL= 5.00E+00 2.00E+00 2.00E+00 5.33E-04

NUMBER OF PATHS FOUND= 5
REFERENCE PIPE ID== 2.00 CENTIMETERS
FOLLOWING PATH FOUND

NOD1	NOD2	VALVE	NUMBER	PIPE USE
9999	31		0	
31	30		1	
30	14		2045	
14	7		20	
7	4		16	
4	15		10	
16	21		33	
21	22		26	
22	999		0	

==THIS PATH HAS PUMP IN IT=====

FLOW THRU PATH=	1.28E+01 LITERS PER MIN	TURBULENT
TOTAL EQUIVALENT LENGTH=	5.36E+02 METERS	
DRIVING HEAD FOR FLOW INCLUDING VACUUM=	1.75E+01 METERS	

HEXIT,HTNLET,HLIQ,HDEL= 5.00E+00 2.00E+00 2.00E+00 5.45E-04

NUMBER OF PATHS FOUND= 6
REFERENCE PIPE ID== 2.00 CENTIMETERS
FOLLOWING PATH FOUND

NOD1	NOD2	VALVE	NUMBER	PIPE USE
9999	31		0	
31	30		1	
30	14		2045	
14	7		20	
7	5		17	
5	23		11	
23	24		34	
24	25		27	VENT
25	999		0	

==THIS PATH HAS PUMP IN IT=====

FLOW THRU PATH= 1.01E+01 LITERS PER MIN TURBULENT
 TOTAL EQUIVALENT LENGTH= 6.65E+02 METERS
 DRIVING HEAD FOR FLOW INCLUDING VACUUM= 1.46E+01 METERS
 HEXIT, MINLET, HLTD, HDL= -2.00E+00 2.00E+00 2.00E+00 5.12E-04

NUMBER OF PATHS FOUND= 7
 REFERENCE PIPE ID= 2.00 CENTIMETERS

FOLLOWING PATH FOUND			
NOD1	NOD2	VALVE NUMBER	PIPE USE
9999	31	0	
31	30	1	
30	14	2045	
14	7	20	
7	6	19	
6	26	12	
26	27	35	
27	29	29	
29	999	0	

===THIS PATH HAS PUMP IN IT=====

FLOW THRU PATH= 1.01E+01 LITERS PER MIN TURBULENT
 TOTAL EQUIVALENT LENGTH= 6.65E+02 METERS
 DRIVING HEAD FOR FLOW INCLUDING VACUUM= 1.46E+01 METERS
 HEXIT, MINLET, HLTD, HDL= -2.00E+00 2.00E+00 2.00E+00 5.12E-04

NUMBER OF PATHS FOUND= 8
 REFERENCE PIPE ID= 2.00 CENTIMETERS

FOLLOWING PATH FOUND			
NOD1	NOD2	VALVE NUMBER	PIPE USE
9999	31	0	
31	30	1	
30	14	2045	
14	29	37	
29	999	0	

===THIS PATH HAS PUMP IN IT=====

FLOW THRU PATH= 1.11E+01 LITERS PER MIN TURBULENT
 TOTAL EQUIVALENT LENGTH= 3.74E+02 METERS
 DRIVING HEAD FOR FLOW INCLUDING VACUUM= 9.55E+00 METERS
 HEXIT, MINLET, HLTD, HDL= 3.00E+00 2.00E+00 2.00E+00 5.60E-04

NUMBER OF PATHS FOUND= 9
 REFERENCE PIPE ID= 2.00 CENTIMETERS

FOLLOWING PATH FOUND			
NOD1	NOD2	VALVE NUMBER	PIPE USE
9999	31	0	
31	30	1	
30	42	22	
42	41	36	
40	41	29	
41	999	0	

FLOW THRU PATH= 1.79E+01 LITERS PER MIN TURBULENT
 TOTAL EQUIVALENT LENGTH= 1.29E+02 METERS
 DRIVING HEAD FOR FLOW INCLUDING VACUUM= 7.57E+00 METERS
 HEXIT, MINLET, HLTD, HDL= 5.00E+00 2.00E+00 2.00E+00 6.99E-04

VALVE #	TIMES USED
13	1
7	1
14	1
8	1
15	1
9	1
16	1
10	1
17	1
11	1
18	1
12	1
19	1
20	7
21	0
20	1
23	1
31	1
24	1
37	1
2045	8
33	1
32	1
25	1
26	1
34	1
27	1
35	1
28	1
1	0
2	0
3	0
4	0
5	0
6	0
22	1
29	1
36	1

NUMBER OF PATHS FOUND= 10

REFERENCE PIPE ID== 2.00 CENTIMETERS

FOLLOWING PATH FOUND

NODE1	NODE2	VALVE NUMBER	PIPE USE
9999	32	0	
32	30	2	
30	14	2045	
14	7	20	
7	1	13	
1	8	7	
8	9	30	
9	10	23	ATD
10	000	0	

==THIS PATH HAS PUMP IN IT=====

FLOW THRU PATH= .0E+00 LITERS PER MIN

==THIS PATH VENTED AT NODE= 1

==FLOW THRU PATH NOT POSSIBLE WITH ONLY VACUUM=====

TOTAL EQUIVALENT LENGTH= 3.59E+02 METERS

DRIVING HEAD FOR FLOW INCLUDING VACUUM= -2.41E+00 METERS

HEXIT,HINLET,HLIO,HDEL=

1.50E+01 2.00E+00 2.00E+00 .0E+00

NUMBER OF PATHS FOUND= 11

REFERENCE PIPE ID= 2.00 CENTIMETERS

FOLLOWING PATH FOUND

NOD1	NOD2	VALVE	NUMBER	PIPE USE
9999	32		0	
32	30		2	
30	14		2045	
14	7		20	
7	20		19	
20	999		0	

==THIS PATH WAS PUMP IN IT=====

FLOW THRU PATH= 1.77E+01 LITERS PER MIN TURBULENT

TOTAL EQUIVALENT LENGTH= 2.66E+02 METERS

DRIVING HEAD FOR FLOW INCLUDING VACUUM= 1.46E+01 METERS

HEXIT,HINLET,HLIO,HDEL=

2.00E+00 2.00E+00 2.00E+00 2.86E-04

NUMBER OF PATHS FOUND= 12

REFERENCE PIPE ID= 2.00 CENTIMETERS

FOLLOWING PATH FOUND

NOD1	NOD2	VALVE	NUMBER	PIPE USE
9999	32		0	
32	30		2	
30	14		2045	
14	7		20	
7	2		14	
2	11		8	
11	12		11	
12	13		24	
13	999		0	

==THIS PATH WAS PUMP IN IT=====

FLOW THRU PATH= 7.55E+00 LITERS PER MIN TURBULENT

TOTAL EQUIVALENT LENGTH= 5.80E+02 METERS

DRIVING HEAD FOR FLOW INCLUDING VACUUM= 7.58E+00 METERS

HEXIT,HINLET,HLIO,HDEL=

5.00E+00 2.00E+00 2.00E+00 1.26E-04

NUMBER OF PATHS FOUND= 13

REFERENCE PIPE ID= 2.00 CENTIMETERS

FOLLOWING PATH FOUND

NOD1	NOD2	VALVE	NUMBER	PIPE USE
9999	32		0	
32	30		2	
30	14		2045	
14	7		20	
7	3		15	
3	17		9	
17	18		12	
18	19		25	
19	999		0	

==THIS PATH WAS PUMP IN IT=====

FLOW THRU PATH= 1.01E+01 LITERS PER MIN TURBULENT

TOTAL EQUIVALENT LENGTH= 5.80E+02 METERS

DRIVING HEAD FOR FLOW INCLUDING VACUUM= 1.26E+01 METERS

HEXTT,HTNLEF,HLIO,HDEL=

-5.00E+00 -2.00E+00 -2.00E+00 -1.60E-04

NUMBER OF PATHS FOUND= 14

REFERENCE PIPE ID= 2.00 CENTIMETERS

FOLLOWING PATH FOUND

NOD1 NOD2 VALVE NUMBER PIPE USE

9999	32	0
32	30	2
30	14	2045
14	7	20
7	4	16
4	15	10
15	21	13
21	22	26
22	999	0

==THIS PATH HAS PUMP IN IT=====

FLOW THRU PATH= 1.22E+01 LITERS PER MIN TURBULENT

TOTAL EQUIVALENT LENGTH= 5.80E+02 METERS

DRIVING HEAD FOR FLOW INCLUDING VACUUM= 1.76E+01 METERS

HEXTT,HTNLEF,HLIO,HDEL=

-5.00E+00 -2.00E+00 -2.00E+00 -2.04E-04

NUMBER OF PATHS FOUND= 15

REFERENCE PIPE ID= 2.00 CENTIMETERS

FOLLOWING PATH FOUND

NOD1 NOD2 VALVE NUMBER PIPE USE

9999	32	0
32	30	2
30	14	2045
14	7	20
7	5	17
5	23	11
23	24	14
24	25	27
25	999	0

VENT

==THIS PATH HAS PUMP IN IT=====

FLOW THRU PATH= 9.79E+00 LITERS PER MIN TURBULENT

TOTAL EQUIVALENT LENGTH= 7.10E+02 METERS

DRIVING HEAD FOR FLOW INCLUDING VACUUM= 1.46E+01 METERS

HEXTT,HTNLEF,HLIO,HDEL=

-2.00E+00 -2.00E+00 -2.00E+00 -1.63E-04

NUMBER OF PATHS FOUND= 16

REFERENCE PIPE ID= 2.00 CENTIMETERS

FOLLOWING PATH FOUND

NOD1 NOD2 VALVE NUMBER PIPE USE

9999	32	0
32	30	2
30	14	2045
14	7	20
7	6	19
6	26	12
26	27	15
27	28	28
28	999	0

==THIS PATH WAS PUMP IN IT=====

FLOW THRU PATH= 9.79E+00 LITERS PER MIN TURBULENT

TOTAL EQUIVALENT LENGTH= 7.11E+02 METERS

DRIVING HEAD FOR FLOW INCLUDING VACUUM= 1.46E+01 METERS

HEXIT,HINLET,HLTO,HDEL=

-2.00E+00 2.00E+00 2.00E+00 1.63E-04

NUMBER OF PATHS FOUND= 17

REFERENCE PIPE ID= 2.00 CENTIMETERS

FOLLOWING PATH FOUND

NO01	NO02	VALVE NUMBER	PIPE USE
0000	32	0	
32	30	2	
30	14	2045	
14	20	37	
20	000	0	

==THIS PATH WAS PUMP IN IT=====

FLOW THRU PATH= 1.04E+01 LITERS PER MIN TURBULENT

TOTAL EQUIVALENT LENGTH= 4.18E+02 METERS

DRIVING HEAD FOR FLOW INCLUDING VACUUM= 9.58E+00 METERS

HEXIT,HINLET,HLTO,HDEL=

2.00E+00 2.00E+00 2.00E+00 1.73E-04

NUMBER OF PATHS FOUND= 19

REFERENCE PIPE ID= 2.00 CENTIMETERS

FOLLOWING PATH FOUND

NO01	NO02	VALVE NUMBER	PIPE USE
0000	32	0	
32	30	2	
30	42	22	
42	40	36	
40	41	29	
41	000	0	

FLOW THRU PATH= 1.51E+01 LITERS PER MIN TURBULENT

TOTAL EQUIVALENT LENGTH= 1.73E+02 METERS

DRIVING HEAD FOR FLOW INCLUDING VACUUM= 7.57E+00 METERS

HEXIT,HINLET,HLTO,HDEL=

5.00E+00 2.00E+00 2.00E+00 2.51E-04

VALVE #	TIMES USED
13	1
7	1
14	1
8	1
15	1
9	1
16	1
10	1
17	1
11	1
18	1
12	1
19	1
20	7
21	0
30	1
23	1
31	1

24	1								
37	1								
2045	1								
33	1								
32	1								
25	1								
26	1								
34	1								
27	1								
15	1								
28	1								
1	0								
2	0								
3	0								
4	0								
5	0								
6	0								
22	1								
29	1								
36	1								
NOD1	NOD2	VALV	LINK STATUS	FLOWWAY	PIPE USE	DLFNG	DIA	WTEXT	
9999	22	0	1=CLOSED	0=2WAY		.0	.0	.0	
1	31	0	0	1		1.00	1.00	.0	
1	7	13	0	0		12.00	3.00	.0	
1	8	7	0	0		10.00	2.00	.0	
2	32	0	1	1		1.00	1.00	.0	
2	7	14	0	0		12.00	3.00	.0	
2	11	8	0	0		18.00	2.00	.0	
1	999	0	1	3	VENT	.0	.0	.0	
3	33	0	0	1		1.00	1.00	.0	
3	7	15	0	0		17.00	3.00	.3	
3	17	9	0	0		11.00	1.00	.0	
4	7	16	0	0		17.00	3.00	.0	
4	34	0	0	1		1.00	1.00	.0	
4	16	10	0	0		11.00	1.00	.0	
5	25	0	0	1		1.00	1.00	.0	
5	7	17	0	0		30.00	2.00	.0	
5	23	11	0	0		4.00	1.00	.0	
6	36	0	0	1		1.00	1.00	.0	
6	7	18	0	0		10.00	1.00	.0	
6	26	12	0	0		15.00	3.00	.3	
7	20	19	0	0		30.00	2.00	.0	
7	14	20	0	0		5.00	1.00	.1	
7	15	21	0	0		1.00	1.00	.3	
8	9	30	0	0		11.00	1.00	.0	
9	10	23	0	0	ATC	17.00	3.00	.0	
10	999	0	0	0		.0	.0	15.00	
11	12	31	0	0		11.00	1.00	.0	
12	13	24	0	0		17.00	3.00	.0	
13	999	0	0	0		.0	.0	5.00	
14	20	37	0	0		12.00	1.00	.0	
14	30	2045	0	0		15.00	3.00	.0	
16	21	33	0	0		12.00	3.00	.0	
17	18	32	0	0		10.00	2.00	.0	
18	19	25	0	0		12.00	3.00	.0	
19	999	0	0	0		.0	.0	.0	
20	999	0	0	0		.0	.0	-2.00	
21	22	26	0	0		10.00	2.00	.0	
22	999	0	0	0		.0	.0	-5.00	

23	24	34	0	0	VENT	15.00	7.00	.0
24	25	27	0	0		19.00	1.00	.0
25	26	35	0	0		.0	.0	-2.00
26	27	35	0	0		4.00	1.00	.0
27	28	28	0	0		30.00	2.00	.0
28	29	1	0	0		.0	.0	-2.00
29	30	0	0	0		.0	.0	7.00
30	31	1	0	0		10.00	2.00	.0
31	32	2	1	0		1.00	1.00	.0
32	33	3	0	0		1.00	1.00	.0
33	34	4	0	0		1.00	1.00	.0
34	35	5	0	0		1.00	1.00	.0
35	36	6	0	0		1.00	1.00	.0
36	37	7	0	0		15.00	2.00	.0
37	38	8	0	0		30.00	2.00	.0
38	39	9	0	0		1.00	1.00	.0
39	40	10	0	0		.0	.0	5.00
40	41	11	0	0				
41	42	12	0	0				
42	43	13	0	0				

APPENDIX C

PROGRAM LISTING

```

PROGRAM PIPE(ITN,TAPE2=IN,CNRC,TAPE3=ONRC,TAPE59)
CCCC THIS IS PROGRAM FOR TURBULENT FLOW
CCCC THIS PROGRAM USES FIELD FREE INPUT FROM FILE IN.
CCCC THIS IS NRC PROGRAM WITH ONLY ONE TIME THRU POSSIBILITY
CCCC NO DATA CAN BE CHANGED AFTER A RUN STARTS
      DIMENSION NOD1(100),NOD2(100),NVALV(100),NDOES(100),
      1 LINKK(100),NVAL(100),NVALVC(100),LINKC(100),LINKF(100)
      2,LINKW(100),NAMP(100),NPNAM(3),NPIPEN(100),IX(700),INDOST(15)
      1,CL(100),DL(100),HIN(15),HCUT(100),HLIO(15),ARFA(15),FLOWI(6)
CCCC DATA INPUT IS IN ORDER NCD1 NCD2 VALVENUMBER FLOW DIRECTION
CCC PIPE USE PIPE LENGTH BETWEEN NOD1 AND NOD2 PIPE DIAMETER
CCCC
      DATA NPNAM /4H ,4HVENT,3HAI2/
      DATA FLOWI /6H LAMINA,1HR,6HTURBUL,3HENT,1H ,1H /
CCCC THIS PROGRAM ASSUMED THAT IF A VALVE WITH NUMBER GREATER
CCC THAN 2000 IS ENTERED IT REPRESENTS A PUMP AND IT
CCC WILL PRINT OUT A WARNING TO THAT EFFECT.....
CCCC IF LINKW=0 IT MEANS TWO WAY FLOW
CCC IF LINKW=1 THERE IS FLOW ONLY FROM NOD1 TO NCD2
CCCC IF LINKW=2 FLOW IS ONLY FROM NOD2 TO NOD1
CCCC IF LINKW = 3 THERE IS NO FLOW AND THAT LINK IS
CCC CLOSED BY SETTING LINKC =1
CCC
CCCC IN INPUT FOR PIPE USE IF NAMP=0 THE PIPE IS ASSUMED
CCCC USED FOR NORMAL PLUTONIUM USE IF NAMP=1 USE IS A
CCCCVENT LINE AND IF NAMP=2 IT IS AN AIR LINE
      IFILE=0
      1 WRITE(59,715)
      NRTAL=0
      ITIME=0
      715 FORMAT(* PROGRAM TO COMPUTE PATHS IN A FLUID FLOW NETWORK*)
      7 WRITE(59,714)
      714 FORMAT(* INPUT IS FROM FILE IN-- PIPE DIMENSIONS IN WHOLE NUMBERS*)
      16 ITN=2
      IREAD=1
      18 WRITE(59,709)
      READ(59,701)NCARD
      NCARD=NCARD+1
      NCD1(1)=9999
      709 IREAD=NCARD-1
      INUM=7*IREAD
      CALL FFINC(ITN,IX,INUM,IREAD,3,NNUM,NLT,IER)
      WRITE(59,4009)NNUM,NLT
      4009 FORMAT(*NNUM AND NLT=*,1X,T4,3X,I4)
      J=1
      DO 786 I=1,NCARD
      NOD1(I+1)=IX(I)
      NOD2(I+1)=IX(J+1)
      NVALV(I+1)=IX(J+2)
      LINKW(I+1)=IX(J+3)
      NAMP(I+1)=IX(J+4)
      DL(I+1)=IX(J+5)
      DL(I+1)=IX(J+5)
      J=J+7
      786 CONTINUE
      DO 9 J=1,NCARD
      IF(LINKK(I).EQ.3)1004,9
      1004 LINKC(I)=1
      LINKC(I)=1
      9 CONTINUE

```



```

      DO 5000 I=1,NCARP
      ISET=NAMP(I)+1
      NAME(I)=NCHAN(ISET)
5000 CONTINUE
      GO TO 1002
1002 WRITE(59,730)
730 FORMAT(*TS OUTPUT TO FILE ONRCYES<*)
      READ(59,710)ANS
      IF(ANS.EQ.34445)731,732
731 ICT=3
      N=10000
      IF(IFTLE.EQ.0)763,741
763 CALL DEVIC(FAHCRATE,4HONRC,N,IERR)
      IFTLE=1
      IF(IERR.NE.0)740,741
740 WRITE(59,751)
750 FORMAT(*ERROR IN CPATE CALL*)
      CALL EXIT
741 GO TO 737
732 ICT=59
733 IF(IFTIME.EQ.0)735,736
735 WRITE(59,3000)
3000 FORMAT(*HOW MANY STARTING NOOFS< IS NOT MORE THAN 15*/123*)
      READ(59,705)IANS
      WRITE(59,3001)
      ITIME=IANS
3001 FORMAT(*LIST STARTING POINTS SEPARATED BY COMMA-END WITH COMMA*)
      CALL FFNC(59,INOCST(I),IANS,0,1,NNUM,NLI,IER)
      DO 7001 J=1,IANS
      DO A I=1,NCARP
      IF(NOD1(I).OR.NOD2(I).EQ.INOCST(J))7001,8
      A CONTINUE
      GO TO 701
7001 CONTINUE
      WRITE(59,3004)
3004 FORMAT(*LIST ELEVATION OF EACH STARTING PCINT IN METERS*)
      CALL FFNC(59,HIN(I),IANS,0,1,NNUM,NLI,IER)
      WRITE(59,3005)
3005 FORMAT(*LIQUID LEVELS ABOVE STARTING POINTS IN METERS SEPARATE-END
      1 WITH COMMA*)
      CALL FFNC(59,HLOT(I),IANS,0,1,NNUM,NLI,IER)
      WRITE(59,3012)
      CALL FFNC(59,AREA(I),IANS,0,1,NNUM,NHLL,TER)
      WRITE(59,3006)
3006 FORMAT(*FACTO EXTT NOOE LIST ITS ELEVATION IN METERS FA.2 FORMAT*)
      DO 3009 I=1,NCARP
      IF(NOD2(I).EQ.999.AND.LINKW(I).NE.3)3008,3009
3008 WRITE(59,3010)NOD1(I)
3010 FORMAT(I8)
      READ(59,3011)HOUT(I)
3011 FORMAT(F8,2)
3009 CONTINUE
3012 FORMAT(*LIST TANK CROSS-SECTIONS ABOVE STARTING POINTS IN SQUARE
      1 PETERS SEPARATE AND END WITH COMMAS*)
      WRITE(59,3013)
3013 FORMAT(*LIST DENSITY OF LIQUID IN GR PER CC F FORMAT*)
      READ(59,3014)DEN
      WRITE(59,3051)
9051 FORMAT(*WHAT IN PIPE IS REFERENCE<CENTIMETERS F8.2 FORMAT*)

```

```

      READ(59,7914)RPID
      ARE81=3.1415*(RPTD/200.)**2
3014  ECRBAT(IE,2)
      TTIME=TTIME+1
      NCOST=INCOST(TTIME)
      AFA2=APE4(TTIME)
      HTNLE=HIN(TTIME)
      HLTQ=HLTQ(TTIME)
      NCQ2(I)=NCOST
      NCOST=9999
      L=1
      LIMIT=NCARD+1
      DO 7 I=1,LIMIT
3       NODES(I)=0
      DO 5 I=1,NCARD
      NNVAL(I)=0
      NVALVC(I)=0
      IF(LINKC(I).EQ.1)1020,1021
1021  LINKG(I)=0
1020  CONTINUE
      LINKF(I)=0
5       CONTINUE
      ICOUNT=0
      NCQES(L)=NCOST
CCCCC
CCCCC
---CCC HERE TO 200 SET UP AND FIND PATH FROM NODE TO NCQES
2       NCWNO=NCOST
      IF(ICOUNT.EQ.1000)760,761
761    CONTINUE
      DO 100 I=1,NCARD
      IF(NON1(I).EQ.NCWNO) 10,100
10     IF(LINKW(I).EQ.2)2310,2311
2310  LINKG(I)=1
2311  NN=NON2(I)
      IF(LINKG(I).EQ.1)2301,14
2301  DO 2302 J=1,L
      IF(NOD2(I).EQ.NODES(J))2300,2302
2300  LINKF(I)=1
2302  CONTINUE
      GO TO 100
14     IF(NN.EQ.999)12,15
C INPUT OF 999 FOR A NODE MEANS THAT IT IS AN EXIT POINT
15     DO 20 J=1,LIMIT
      IF(NODES(J).EQ.NN)793,20
793    LINKG(I)=1
      LINK1=I
      GO TO 100
20     CONTINUE
      L=L+1
      NCQES(L)=NN
      NCOST=NN
      ICOUNT=ICOUNT+1
      LINKG(I)=1
      LINKF(I)=1
      NNVAL(L-1)=NNVAL(L-1)
      GO TO 2
12     LINKG(I)=1
      L=L+1
      NNVAL(L-1)=NNVAL(L-1)

```

```

      NCDEF(L)=NN
      NTRIAL=NTRIAL+1
      GO TO 200
100 CONTINUE
      NCDEF=1
202 NCWNO=NCOST
      DO 201 I=1,NCARD
      IF(NOD2(I).EQ.NCWNO)210,201
240 IF(4*LNKWH(I).EQ.1)2410,2411
2410 LINKG(I)=1
2411 NN=NOD1(I)
      IF(LINKG(I).EQ.1) 2401,212
2401 DO 2402 J=1,L
      IF(NOD1(I).EQ.NODS(J))2400,2402
2400 LINKG(I)=2
2402 CONTINUE
      GO TO 200
212 DO 220 J=1,LIMIT
      IF(NODS(J).EQ.NN)792,220
792 LINKG(I)=1
      LINK2=I
      GO TO 200
220 CONTINUE
      LINKG(I)=1
      L=L+1
      NCDEF(L)=NN
      LINKG(I)=2
      NNVAL(L-1)=NNVAL(I)
      NCDEF=0
      NCOST=NN
      TCOUNT=TCOUNT+1
      GO TO 202
200 CONTINUE
      IF(L.EQ.0)801,300
300 IF(NOD1.EQ.1)401,2
401 L=L+1
400 K=0
      IF(TCOUNT.EQ.1000)760,762
CCCCHERE TO 914 CHECK TO SEE IF ANOTHER PATH POSSIBLE
CCCCFROM THE LAST NONE IF YES GO TO 2 AND START AGAIN
762 NCDEF(L)=0
      NNVAL(L-1)=0
      L=L-1
      DO 900 I=1,NCARD
      IF(NOD1(I).EQ.NCDEF(L))911,900
911 IF(LINKG(I).EQ.1)913,912
912 NCDEF=L
      TCOUNT=TCOUNT+1
      GO TO 2
913 CONTINUE
900 CONTINUE
      DO 915 I=1,NCARD
      IF(NOD2(I).EQ.NCDEF(L))916,915
916 IF(LINKG(I).EQ.1)920,917
917 DO 921 J=1,L
      IF(NOD1(I).EQ.NODS(J))915,921
921 CONTINUE
      TCOUNT=TCOUNT+1
      NCDEF=NCDEF(L)
      GO TO 2

```

```

920 CONTINUE
915 CONTINUE
      IF (L.EQ.1) 302,918
CC HERE TO ADD BACKUP 1 NODE AND RESET ALL LINKSTO
CC TRY NEW PATH
918 L=L+1
      IF (NODES(L).EQ.9999) 802,919
919 DO 511 I=1,NCARD
      IF (NOD1(I).EQ.NODES(L+1)) 508,511
508 DO 510 J=1,L
      IF (NOD2(I).EQ.NODES(J)) 2100,510
2100 IF (NODES(J).EQ.NODES(L)) 511,2107
2107 IF (LINKF(I).EQ.1) 2108,2101
2101 LINKF(I)=0
2108 GO TO 511
510 CONTINUE
      IF (LINKC(I).EQ.1) 611,1022
1022 LINKG(I)=0
      LINKG(LINK1)=0
511 CONTINUE
      DO 611 I=1,NCARD
      IF (NOD2(I).EQ.NODES(L+1)) 608,611
608 DO 610 J=1,L
      IF (NOD1(I).EQ.NODES(J)) 2200,610
2200 IF (NODES(J).EQ.NODES(L)) 611,2007
2007 IF (LINKF(I).EQ.2) 2006,2201
2201 LINKG(I)=0
2006 GO TO 611
610 CONTINUE
      IF (LINKC(I).EQ.1) 611,1023
1023 LINKG(I)=0
      LINKG(LINK2)=0
611 CONTINUE
      J=L+1
      DO 6 I=J,LIMIT
      NODES(I)=0
6 CONTINUE
      DO 7000 I=J,NCARD
      NVAL(I-1)=0
7000 CONTINUE
      K=L+1
      DO 2601 I=1,NCARD
      IF (LINKF(I).NE.0) 2603,2601
2603 IF (NOD1(I).OR.NOD2(I).EQ.NODES(L)) 2600,2601
2600 DO 2604 J=1,K
      IF (NOD2(I).OR.NOD1(I).EQ.NODES(K)) 2604,2602
2602 LINKF(I)=0
2604 CONTINUE
2601 CONTINUE
      ICOUNT=ICOUNT+1
      LINK1=0
      LINK2=0
      NCOST=NCOST(L)
      GO TO 2
CCCCC AFTER THIS POINT IS THE PART OF THE PROGRAM THAT
CCC WRITES OUT DATA AND COMPUTES THE FLOW THRU PATHS IT
CCC IS ENTERED ONLY AFTER A COMPLETE PATH HAS BEEN IDENTIFIED
CCCCC
CCCCC
      DOB WRITE(I*Y,706)NTRIA*

```



```

13 DC 4 K=2,J
   IF(NNODES(K).EQ.NNODE1(I))791,4
791 WRITE(IOT,712)NNODES(K)
712 FORMAT(=====THIS PATH VENTED AT NODE=*,I4)
4 CONTINUE
11 CONTINUE
   IF(OLIT.EC.0.0)9005,9006
9005 WRITE(ICT,9007)
9007 FORMAT(=====FLOW THRU PATH NOT POSSIBLE WITH CALY VACUUM=====)
9006 WRITE(ICT,9000)TOTL
9000 FORMAT(*TOTAL EQUIVALENT LENGTH=*,E12.2,* METERS*)
   WRITE(ICT,716)DELMT
716 FORMAT(*DRIVING HEAD FOR FLOW INCLUDING VACUUM=*,E12.2,* METERS*)
   WRITE(IOT,9002)HEXIT,HINLET,HLIQ,HDEL
9002 FORMAT(*HEXIT,HINLET,HLIQ,HDEL=*/,4(E12.2)
      OLIT=Q*TOT=DFLH=HDEL=0.0
      DC 803 I=1,J
      IF(NVAL(I).EQ.0)807,807
807 DC 804 M=1,NCARD
      IF(NVAL(I).EQ.NVALV(M))805,804
805 NVALV(M)=NVALV(M)+1
804 CONTINUE
803 CONTINUE
      GO TO 400
801 WRITE(59,711)INCDST(J)
711 FORMAT(*NO SUCH NODE IN DATA AS NODE=*,I5)
      ITIME=0
      GO TO 733
802 WRITE(IOT,723)
723 FORMAT(*VALVE #      TIMES USED *)
      DC 725 I=1,NCARD
      IF(NVALV(I).EQ.0)725,724
724 WRITE(ICT,726)NVALV(I),NVALV(I)
725 CONTINUE
726 FORMAT(2X,I4,5X,I4)
      IF(ITIME.EQ.0)ITIME=734,733
734 WRITE(59,709)NTOTAL
1026 WRITE(ICT,730)
      DC 1025 I=1,NCARD
      WRITE(IOT,781)NOD1(I),NOD2(I),NVALV(I),LINKG(I),LINKW(I),NAMP(I)
      1,PL(I),C(I),HOUT(I)
1025 CONTINUE
      GO TO 1000
760 WRITE(59,764)
764 FORMAT(*COUNT=1000 POSSIBLE LOOP*)
1000 CALL EXIT
4003 C2=(.001/(RPT*DEN))**.25
      C3=(.316*TOTL*100.)/(RPT*.2*.9.892)
      V1=LOGF(DELMT/(C2*C3))/1.75
      V=EXP(V1)
      RE=(RPT*V*DEN*.1/.001)
      IF(RE.EC.2100.1)4004,9008
4004 DC 4005 I=1,60
      DELMT=DELMT*(101./(9.802*DEN))
      IF(DELMT.LE.0.)9003,4006
4006 V1=LOGF(DELMT/(C2*C3))/1.75
      V=EXP(V1)
      Q=V*AREA1
      QTOT=Q*TOT+Q
      HDEL=Q/AREA2

```

DELH=DELH-HDEL
4005 CONTINUE

IFLW=1

TCTL=TCITL

GC TO 4010

700 FORMAT('HOW MANY CARDS FOR DATA? I3'/'123')

780 FORMAT('NOD? VALV LINK STATUS FLOWWAY PIPE USE'/'
1 1=CLOSED 0=PHAY DLENG DIA

2 *****)

781 FORMAT(I4,3X,I4,3X,I4,3X,I4,6X,I4,10X,A6,4X,3(F4.2,2X))

701 FORMAT(I3)

705 FORMAT(I3)

706 FORMAT('***NUMBER OF PATHS FOUND=*,I4)

707 FORMAT('FOLLOWING PATH FOUND'/' NOD? VALV NUMBER
PIPE USE')

709 FORMAT('PATH NUMBER=*,I6,* ALL PATHS FOUND PROGRAM ENDED')

710 FORMAT(A3)

708 FORMAT(I3,I4,3X,I4,6X,I4,12X,A6)

END