Pipe Network Simulation Analysis Computer Program - NETWK

Roland W. Jeppson

Follow this and additional works at: https://digitalcommons.usu.edu/water_rep

Part of the Civil and Environmental Engineering Commons, and the Water Resource Management Commons

Recommended Citation
https://digitalcommons.usu.edu/water_rep/298
USER'S MANUAL

PIPE NETWORK
SIMULATION ANALYSIS
COMPUTER PROGRAM - NETWK

Civil and Environmental Engineering
Utah Water Research Laboratory
College of Engineering
Utah State University
Logan, Utah 84322

By
Roland W. Jeppson,
Professor
March 1990
USERS MANUAL FOR PIPE NETWORK SIMULATION
ANALYSIS COMPUTER PROGRAM USU-NETWRK

by

Roland W. Jeppson

Civil and Environmental Engineering
and
Utah Water Research Laboratory
Utah State University
Logan, UT 84322

The software described in this document is furnish under a license
and may only be used in accordance with the terms of the license.

Copyright C 1990 by Roland W. Jeppson
Department of Civil Engineering
Utah State University

Previous editions; April 1977, September 1978,
June 1980, April 1982, Jan. 1985

March 1990

Price $25
USU-NETWK ON IBM-PC COMPUTERS

The next couple of pages are applicable if you have obtained a version of USU-NETWK for use on a PC. Read them carefully before installing USU-NETWK.

The diskette that you have received contains the executable element of the network analysis and design program that has been developed at Utah State University. It also contains the executable element of a pre-processor program PIPEINPT to assist in preparing input data files, and a post-processor program PLTNET that will display the network and its solution on the PC monitor provided that the PC contains a graphics card. (All IBM compatibles will not allow the graphics even if they have a graphics card.) The diskette also contains a couple of example data files for small networks. These few pages are intended to get you started and provide instructions to get the program up on your PC.

To effectively use the program you will need to read this manual. This manual is written to also serve the needs of users of USU-NETWK on large computers. Most of the input described in the manual applies to this PC version of the program. The manual indicates which options and commands are acceptable only to the larger program.

Unless you have indicated otherwise, the program you have received requires that the PC contain co-math processor such as an 8087 and will require a PC that has 512 kbytes of RAM. If the PC you plan to use USU-NETWK on does not have a co-math processor, you should request that a diskette be sent that does not require it. Also if you wish to use USU-NETWK on a PC that has only 256 kbytes of memory you should indicate this. The program for a PC with only 256 kbytes will accommodate a pipe network with 300 pipes.

Installing program on your PC

The following instructions assume that your PC has a hard disk. If it does not then you will need to use one diskette for the program and another for the input and output data files (see the next section). It is important that you follow these instructions carefully.

The diskette(s) that contains the USU-NETWK package will have the programs in compressed form on them. In order to get these programs off the diskette and in executable form it is necessary that you properly install and decompress them by completing the following steps (These steps assume your PC is operating under DOS and that the DOS commands are accessible):

1. Place the distribution diskette labelled USU-NETWK in one of your diskette drives. (If the distribution diskette is a 5 1/4 inch, high density, diskette then this must be a 1.2 meg drive.) Make sure that the default drive is where you want the program to go, i.e. when installing USU-NETWK on a hard disk, the default drive should be C:, D: etc.

2. Type A:INST_NET (or B:INST_NET if the diskette is in drive B):
   
   [Note: _ is the underscore not the minus sign -, and is typed by pressing this key while simultaneously holding down the Shift key]

3. Respond to the prompts. You will be asked to provide your license number. This number will be in the license agreement with the distribution diskette. You will be asked to give the drive that the distribution diskette is in. Type either 1 or 2 as appropriate. You will be asked to give a subdirectory that the files will be stored on. A good choice for a subdirectory is NETWK.

   To use USU-NETWK thereafter you should first make the subdirectory into which the programs are installed your default directory by typing CD\NETWK (or whatever the subdirectory name is). This subdirectory should then be used for input and output files as well.

   Ignore any message such as "File not found" that you get during the installation of USU-NETWK. If you get the message that directory NETWK (or the name you gave) cannot be created, it simply means that this directory already exists, and the installation will overwrite any files with the same name in this directory.

   Should you have ordered the PC version of USU-NETWK that also has the capability to do time-dependent problems, and have requested that you also receive the steady state version of USU-NETWK, then this steady state version will be
under a directory \STEADY\ if the distribution diskette is a 1.2 mega 5 1/4 inch (high density) diskette, or on a 1.4 mega 3 1/2 inch diskette. You can determine whether this subdirectory exist by giving the command DIR A: (or DIR B: if the distribution diskette is in drive B:). The steady-state version of USU-NETWK will be under the name NETWKS.EXE. The added S stands for steady state, and to execute it rather than the simulation version type its name. You can install the steady state version of USU-NETWK in the same subdirectory on a hard disk as the simulation version, or a different subdirectory. In either case you need to complete the following steps to also install the steady-state version of USU-NETWK.

1. Make the subdirectory STEADY on the floppy drive where the distribution diskette is the default drive by typing: A: (or B:) then type CD\STEADY.

2. Make the hard disk the default drive again by typing C: (or D: if you which this installed on drive D:, etc.)

3. Type A:INST_NET (or B:INST_NET if the distribution diskette is in drive B)

4. Respond to the prompts as above selecting 5 (to install everything). Generally only the element NETWKS.EXE will be installed. However, another version of PLTNET.COM might also be installed. This latter version of PLTNET with the extension .COM utilizes only CGA graphics capabilities, whereas PLTNET.EXE utilizes EGA, or VGA graphics capabilities. Should PLTNET.COM be installed on the same directory as PLTNET.EXE, and the PC has EGA or VGA graphics, you should either delete PLTNET.COM or rename PLTNET.COM to PLTNETC.COM or some other name or else it will be executed when you give the command PLTNET, rather than PLTNET.EXE. On the other hand if the PC only has CGA graphics, then delete PLTNET.EXE.

INSTALLATION OF TUTOR FOR USU-NETWK

With the distribution of USU-NETWK you will likely have also received a TUTOR program to help you get started. This TUTOR may come on a separate diskette, or be in a subdirectory TUTOR on the distribution diskette if this distribution diskette is 3 1/2 inch in size or is a high density 5 1/4 diskette. If the TUTOR is on a separate diskette then you can use TUTOR directly from the diskette by placing this diskette in a floppy drive, make this the default drive, and type TUTOR. Should TUTOR be on the same distribution diskette with the other USU-NETWK programs, then it is necessary to put TUTOR on your hard disk or another floppy diskette. If you wish to install TUTOR on your hard disk, then place the distribution diskette that contains TUTOR in a floppy drive compatible with the diskette. If a subdirectory TUTOR exists on this diskette, then make this the default directory of the floppy by typing A: (or B: if the distribution diskette is in drive B) followed by CD\TUTOR. Next make the hard disk the default drive again by typing C: (or the drive you want TUTOR installed on). With the hard disk your default drive type A:INSTALL (or B:INSTALL). (Note the name is now INSTALL and not INST_NET as above.)

EXECUTING USU-NETWK

To execute the program after it is installed give the command, NETWK. When the program asks: WHAT FILE CONTAINS INPUT DATA? give the name of the file that you have prepared with an editor or the pre-processor program PIPEINPT, to define the network problem you want solved. If you respond with TTY or REMOTE then USU-NETWK expects to receive input data for the problem directly from the keyboard. Such use is practical only for very small networks. When the program asks: WHAT FILE NAME SHOULD CONTAIN OUTPUT? give the file to which the output is to be written. You can display this output file on your monitor afterward with the command TYPE followed by the same name. If you want the output to be displayed to the monitor directly given TTY or REMOTE for this file name. The length of the file names that USU-NETWK accepts is 12 characters or less including the extension and drive designation if it precedes the name.

NOTE: If you use a word processor such as WordPerfect or WordStar, it is necessary that the file be written to disk as an ASCII (or TEXT files) for NETWK to read them. For example with WordPerfect, you will need to use the test IN/OUT key when writing the data to disk.

EXAMPLE PROBLEMS

Several small example pipe network problems will be included with the distribution of
USU-NETWK. These are in the compressed file INPUT:ZIP, and will be decompressed during the above "install" procedure if you select to have this done when prompted i.e. by selection menu item 5 (to have everything installed). The example problem on the file EXAMPLT.DAT is designed to write files that allow you to use the graphics post processors PLTNET, PROFILM, OFPLOT, etc. The x and y coordinates of the nodes are given in this file, which is necessary to do any graphics. Also to display networks on the monitor of your PC it must contain a graphics adaptor. A good way to get started in using PLTNET is to use the example network, EXAMPLT. The file that PLTNET utilizes is written by NETWK if the option NETPLT=1 is in the $SPECIF list of options. The more general purpose graphics file, that PROFILM and OFPLOT use, is written by NETWK if the option NETPLT=3.

It is suggested that you complete the following exercises using the example problem EXAMPLT.DAT:

1. Execute USU-NETWK by typing NETWK, and when ask for the input file give the name EXAMPLT.DAT, and give TTY for the output file, so that the solution will be displayed on the screen. Upon completion of the solution USU-NETWK will have written a file PLTNET.DAT for PLTNET to use.

2. Execute the graphics program PLTNET by typing PLTNET. When ask if you wish to change options type an N for no the first time, also when ask WHAT DO YOU WANT ON PLOT? you can select "All setting OK" by typing 10. You will see this network displayed on your PC's monitor. Press the return to activate PLTNET again. Reading the instructions related to the use of PLTNET will inform you about what this processor can do.

3. Use an editor to change the option NETPLT in the file EXAMPLT.DAT to NETPLT=3, save this altered file, and execute NETWK again. Setting option NETPLT=3 cause the general purpose graphics file to be written by USU-NETWK. If you are ask during this execution to give the file name for the graphics data type PLOT10.DAT.

4. Execute the graphics program PROFILM by typing PROFILM. This program is written in PASCAL and it requires strings of three nines to terminate reading of the first lines of input from the graphics file PLOT10.DAT. Programs, such as OFPLOT, written in FORTRAN require that a / terminate reading these lines rather than 999. Should your version of NETWK terminate the first lines of the general graphics file with /s rather than 999 either replace the / with a 999 or attempting to read the graphics file with PROFILM. On the other hand when using OFPLOT either replace the 999's with a / or place a / in front of the 999's with an editor.

The program PROFILM will ask:
SELECT THE NO. FOR TYPE OF PLOT YOU DESIRE.

For the first plot select 2 (for "Pressure bars at nodes of network"). Thereafter you will be ask to GIVE FOLLOWING: 1 - type of bar, 2 - scale factor, 3 - width and 4 - units. The numbers listed after these items are values that PROFILM suggests. For your first display type in these values, i.e. type in 2 .3 10 0 After the plot is displayed take a carefully look at the pipe numbers and the network's layout. To reactivate PROFILM from just displaying the plot press the enter key.

Have PROFILM make a second display, but this time when ask: SELECT THE NO. FOR TYPE OF PLOT YOU DESIRE give 1 for "Head profiles over selected sequences of pipes. Place three such profile on the display by giving 3 when ask, and then give the follow three sequences of pipe numbers:
6 5 4 2 1/
6 5 10 11 16 27 28 31/
30 29 17 27/

Next have PROFILM make a contour plot by responding with 3 ("Contour map of heads or pressures") for the TYPE OF PLOT YOU DESIRE. You will be ask to give 5 values for the following 5 items: (1) No. of Contours, (2) Value of smallest contour, (3) Contour interval, (4) Frequency of heavy line, & (5) maintain x and y scales or fill screen. Appropriate values might be:
32 400 5 4 0
The values give by PROFILM, associated with the request, help you decide what to give for these 5 items.

With this preliminary introduction with PLTNET and PROFILM you might wish to play with the options, changing colors, character styles and sizes until you get what is most pleasing to you.

PC's with two floppy diskette drives but no hard disk

If your PC does not have a hard disk,
then it will be necessary to keep your input and output data files on another diskette, and run NETWK from the program diskette in the other drive. The following description assumes that the program diskette will be in drive A: and the data diskette in drive B:. To install the program initially do the following:

1. After turning the PC on place the DOS diskette in drive A and wait for it to boot.
2. Place an unused diskette in drive B, and give the command FORMAT B: and follow the instructions.
3. Copy the EDLIN editor on the diskette in drive B with the command COPY EDLIN.COM B: (if you want to use another editor, or word processor delete this step.)
4. Give the command B: to make B your default drive.
5. Replace the system diskette with the diskette included herein and give the command A:INSTALL. When asked for a file where the output to the example problem should go give TTY, that indicates the output will go to the monitor directly. (If you give another name then that filename will be established, and you can see the solution output, thereafter, by either giving the command TYPE filename, or TYPE TYPEFILENAME>PRN: to have it printed.

Every time you want to use a new data diskette for input and output data for network problems it is necessary to follow this same procedure. Subsequently you will want to issue the command DEL *.DAT to delete the data files for the example networks, however.

Examples:

In the following example USU-NETWK is used to solve the classical 3 reservoir problem that can be found in most fluid mechanics books.

![Diagram of three reservoir problem](image)

Give the command: NETWK and then enter the data as shown below.

**WHAT FILE CONTAINS INPUT DATA?**
TTY

**WHAT FILE NAME SHOULD CONTAIN OUTPUT?**
TTY

**THREE RESERVOIR PROBLEM**

```
/ *
```

**TITLE GIVEN TO NETWRK**
THREE RESERVOIR PROBLEM
ALL DEMAND FLOWS ARE MULTIPLIED BY 1.000

$SPECIF NPRINT=-1,COEFRO=.005 $END

| PIPES | 1 0 1 1500 8 |
|       | 2 0 1 1000 6 |
|       | 3 1 0 1200   |

| NODES | 1 0 350 |
| RESER | 1 500 |
|       | 2 450 |
|       | 3 300 |

RUN

**Output from NETWK**

| PIPES | 3 |
| NODES | 1 |
| SOURCE PUMPS | 0 |
| BOOSTER PUMPS | 0 |
| RESERVOIRS | 3 |
| MINOR LOSSES | 0 |
| PRVS | 0 |
| NOZZLES | 0 |
| CHECK VALVE | 0 |
| BACK PRES. V. | 0 |
| DIF. HEAD DEV. | 0 |
| RESERVOIR(NOZZLE) PIPES AND THEIR ELEVATIONS ARE |
| 1 500.0 | 2 450.00 | -3 300.00 |
| FLOW FROM PUMPS AND RESERVOIRS EQUALS | .000 |

| ITERATION | 1 | SUM | .125E+01 |
| ITERATION | 2 | SUM | .579E+00 |

---

-vi-
ITERATION 3  SUM .172E+00
ITERATION 4  SUM .166E-01
ITERATION 5  SUM .157E-03

UNITs OF SOLUTION ARE
DIAMETERS - INCH
LENGTH - FEET
ELEVATIONS - FEET
PRESSURES - (PSI)
FLOWRATES -(CFS)

DARCY-WEISBACH FORMULA USED FOR
COMPUTING HEAD LOSS

PIPE DATA

<table>
<thead>
<tr>
<th>NO.</th>
<th>FROM</th>
<th>TO</th>
<th>LENGTH</th>
<th>DIAM</th>
<th>COEF</th>
<th>FLOW RATE</th>
<th>VELOCITY</th>
<th>LOSS /1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>* 1</td>
<td>0</td>
<td>1</td>
<td>1500</td>
<td>8.0</td>
<td>.005000</td>
<td>3.05</td>
<td>8.74</td>
<td>49.17</td>
</tr>
<tr>
<td>* 2</td>
<td>0</td>
<td>1</td>
<td>1000</td>
<td>6.0</td>
<td>.005000</td>
<td>.21</td>
<td>1.07</td>
<td>.06</td>
</tr>
<tr>
<td>* 3</td>
<td>1</td>
<td>0</td>
<td>1200</td>
<td>6.0</td>
<td>.005000</td>
<td>2.84</td>
<td>14.46</td>
<td>150.85</td>
</tr>
</tbody>
</table>

NODE DATA:

<table>
<thead>
<tr>
<th>NO.</th>
<th>DEMAND</th>
<th>(GPM)</th>
<th>ELEV</th>
<th>HEAD</th>
<th>PRESSURE</th>
<th>ELEV</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.000</td>
<td>350</td>
<td>100.66</td>
<td>43.71</td>
<td>450.86</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4650</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>4700</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A more usual mode of solving network
problems is to prepare the input data on a file
first with an editor as illustrated below for the
small network shown in the sketch. In this
example EDLIN will be used to prepare the input
data.

EDLIN EXAMP.DAT

* puts EDLIN in input mode

$SPECIF PCHAR3=0,NODESP=1,
COEFRO=.002 SEND

NODES

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.8</td>
<td>4600</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>4530</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>.9</td>
<td>4520</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1.5</td>
<td>4550</td>
<td></td>
</tr>
</tbody>
</table>

RUN

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>1</td>
<td>4650</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>4700</td>
<td></td>
</tr>
</tbody>
</table>

PIPES

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7</td>
<td>1</td>
<td>1000</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2000</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>3</td>
<td>1700</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>3</td>
<td>1800</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>4</td>
<td>1500</td>
</tr>
</tbody>
</table>

RESER

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>4840</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>242</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td>5</td>
<td>1000</td>
</tr>
</tbody>
</table>

RUN

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A control C is given</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>to exit input mode, and E</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>is given to exit.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Now type NETWK and when ask what the input
file is give

EXAMP.DAT

When ask what file should receive the output give

EXAMP.OUT

You can subsequently display the solution on your
monitor with the command TYPE EXAMP.OUT
or have it printed with the command PRINT
EXAMP.OUT.
A tutor is available to assist you in getting acquainted with the operation and input data requirements for USU-NETWK. This tutor is not intended to replace this manual, but for an individual who would rather sit at a PC terminal than read a manual, the tutor will be helpful in getting started. If you have received the tutor in the distribution of USU-NETWK and wish to use it, then you should read the rest of this page of the manual.

To use the USU-NETWK tutor you should do the following:
1. Place the TUTOR diskette in a floppy disk drive, i.e. drive A:
2. If this is not your default drive, make this drive the default drive by typing A:
3. Type the command TUTOR to run the program.

Most of the instructions given by TUTOR are in text form. Sketch of small networks are provided for which input data is subsequently prepared. The PC that you are using must have an IBM graphics card, and be fully IBM compatible for these displays to work properly. If your PC has a color monitor, numbers that denote pipes, and nodes on these sketches will be in different colors so that these values can be easily distinguished. If your PC does not have a graphics card, or has a graphics card that is not compatible with what TUTOR requires, you can still benefit by use of this tutor.

In making copies of this tutor diskette it is necessary that the DOS DISKCOPY command be used rather than the DOS COPY command. The files on this diskette that have the extension .DAT are input files for 8 small problems that are covered by TUTOR. You might want to print these input data files, and have the output that USU-NETWK produces from them go to a file with the same name but with the extension .OUT. If you print these output files, you can see if your program of USU-NETWK produces the same solutions as given by the TUTOR. The two solutions should be identical.

Figure 1. Small example network used in first 7 problems by TUTOR.
This manual describes the data that a versatile computer program USU-NETWK requires to complete a time dependent solution of flow rates and pressures throughout a pipe network. This program can be used for steady-state analysis and design, as well as simulation solutions through some period of time. In fact, unless specified otherwise, a steady-state solution is assumed.

A smaller version of the program is not capable of the time-dependent solution, but is capable of doing most of the types of steady-state analyses described in this manual. If you have the version of USU-NETWK for use on 16 bit IBM compatible PC's it may only do steady-state analyses.

You, as a user, may have acquired this smaller version of the program, because you believe this capability would meet your needs, your computer limitations, or for other reasons. This manual is still applicable for your program. All basic information which describes the physical features of the pipe network are identical for the simulation and the steady-state versions of USU-NETWK. The difference is that the simulation version accepts additional data describing the time dependent operation and characteristics of the network. In reading the manual you will be able to identify those data items which are "additional time dependent data." Therefore, if your version handles only steady-state solutions this manual is applicable to your program, if you ignore those parts of the manual devoted to input data for simulations. Also a few seldom used options are not implemented for the steady-state version of the program. These are denoted by #.

In addition to a description of data requirements for USU-NETWK, a pipe network analysis and design program, this manual contains descriptions of auxiliary program that can be used in connection with USU-NETWK. These programs consist of: 1. A preprocessor, PIPEINPT that assists in the preparation of steady-state input data, 2. A post processor, PLTNET that displays the network and selected solutions items to the monitor of a PC, and 3. A post processor program PROFILM that (a) will display profiles of pressure, pressure head, or HGL elevations through selected sequences of pipes, (b) display the pressure, pressure head, or HGL elevation as bars at the nodes of the network, or (c) draw a contour type map of the pressure, pressure head, or HGL elevation. 4. A program OFPLOT that is contains the same calcomp graphics plotting capability in the full program but is intended for off line use. 5. A program PRINT that is capable of efficiently supplying requested specific information from a direct access file written by the simulation version of USU-NETWK, related to the solution at any time. This last auxiliary program is described at in Appendix E of this manual. 6. A program PLTTIM that will display variables from the solution plotted against time, from time-dependent solutions.

Programs PIPEINPT and PLTNET are described near the first of the manual after a preview of the input to USU-NETWK is provided. A TUTOR program is also available with PC's to help users get acquainted with USU-NETWK and some of the flexibility it allows in input. Also a documentation program for larger computers is available that can be used in place of this manual for help in using USU-NETWK. Help in using this TUTOR are given in a preliminary page of this manual.

The full version of USU-NETWK has the ability to drive a calcomp plotter to produce a graph of the network and its solution for inclusion in reports. If your installation subscribes to the DISSPLA graphics routines, then USU-NETWK is also capable of producing a three-dimensional perspective drawing of the network and its solution, as well as generating contour type maps of pressure or HGL elevation lines. These can be sent to a variety of graphics devices. Other auxiliary graphics programs are also available.

The complete program allows several forms for the input data. The standard form is discussed in the body of this user's manual with other allowable forms described in Appendixes A and B. If you decide on a given form of input and wish to use it exclusively, the size of the program can be reduced. For example, if only the standard form of input is desired, then subroutine AINPUT and the references to it and its entry points may
be removed. A reduced version of the simulation program does not contain subroutine AINPUT and other subroutines. If the command PIPE-nodes is not to be utilized then the subroutine PNREAD may be removed. The size of the program can be further reduced by discretion. For example three different solution algorithms exist in the original program. The elimination of the subroutine SOLSYM removes the solution algorithms which establishes a symmetric positive definite Jacobian matrix for the Newton Method of solution, which is ordinarily called upon if no pressure reduction valves, back pressure valves, or other devices exist that destroy symmetry in this matrix. Solution subroutines SOLNYM and BAND handle any problem. Also subroutines SPARSE, SPARSA, SPARSB and EQUATS, which utilize special sparse matrix solution methods, handle any problem. Therefore, a solution to any network could still be obtained with SOLSYM removed; and/or with the sparse matrix methods or SOLNYM and BAND removed. The PC steady-state version of USU-NETWK contains only the sparse matrix subroutines, as does a smaller version of the simulation program. The program was initially developed using pre-77 FORTRAN that did not allow list directed read statements. To allow input to be free of any format requirements, USU-NETWK has its own free format input routines. List of options are handled likewise by USU own equivalent NAMELIST \O capabilities, or they may be replaced by NAMELIST \O if this is allowed for in the FORTRAN being used.

The number of network components that can be accommodated are established at the time the program is compiled and linked. When using a computer such as a VAX, MicroVAX, IBM mainframe, or UNIX Engineering work stations the maximum number of pipes, and nodes, etc that may be accommodated is large such as several thousand. A VAX version of USU-NETWK dimensioned for up to 3,000 pipes, 2,580 nodes, 80 pumps, etc. is kept on file ready for distribution. However, for use on 16 bit or 32 bit PC computers under DOS the amount of memory that can be addresses is 640 Kbytes and this limits the size of networks that can be accommodated. Currently for steady-state PC version of USU-NETWK is limited to 1000 pipes, 860 nodes, 600 loops, 40 pumps, etc. The version of USU-NETWRK designed for PC under DOS that will also perform time-dependent analysis is dimensioned for 600 pipes, etc. A version for a PC that is dimensioned for 500 pipes, and 430 nodes will run slightly faster than the version that allows up to 1000 pipes. Since most all networks can be skeletonized to 500 pipes or less this smaller program is often the one distributed.

In this user's manual there is no attempt to describe the methods used in USU-NETWK to provide solutions to either analysis or design problems. Rather this user's manual is devoted exclusive to describing the input data, and the use of USU-NETWK in solving various types of network problems.
# TABLE OF CONTENTS

**USU-NETWK ON IBM-PC COMPUTERS** iii

**TUTOR** viii

**PREFACE** x

**TABLE OF CONTENTS** xii

**PREVIEW** xiv

**FLUID MECHANICS PROBLEMS** xvii

**UNIT AND CONVERSION FACTORS** xx

**SUMMARY MOST FREQUENTLY USED OPTIONS** xxi

**SUMMARY OF COMMANDS THAT ENTER DATA** xxiii

**INTRODUCTION** 1

**VERSATILITY OF COMPUTER PROGRAM** 3

**USU-NETWKS'S PROCESSOR: PIPEINPT** 6

**GENERAL SPECIFICATIONS (OPTIONS)** 11

**DETAILED STEADY-STATE DATA REQUIREMENTS** 47

**CHANGE DATA** 67

**COST INPUT DATA** 71

**DETAILED DYNAMIC SIMULATION DATA REQUIREMENTS**

**ILLUSTRATIVE EXAMPLES OF NETWORKS AND INPUT DATA USING THE STANDARD FORM OF INPUT** 111

**APPENDIX A (1st alternate input to USU-NETWK)** A-1

**APPENDIX B (2nd alternate input to USU-NETWK)** B-1

**APPENDIX C - GRAPHICS ASSOCIATED WITH USU-NETWK** C-1

**APPENDIX D - IMPROVING DESIGN** D-1

**APPENDIX E - AUXILIARY PROGRAM PRINT** E-1

**APPENDIX F - PROGRAM PMPCUR for display of pump curves** F-1
The program USU-NETWK is designed to make the input as simple as possible, while simultaneously allowing the user considerable flexibility in describing a network problem for a computer solution. At the first rapid reading of this manual, the new user may get the impression that a lot must be understood before preparing the input data for USU-NETWK. To dispel this possible false impression this preview section is given at the beginning of the manual. You should read this section if you are a new user, and prepare data for some simple networks such as those illustrated below, before reading the remainder of the manual. That is you should get solutions to some problems before you concern yourself with all the possibilities that are allowed, and are described in the rest of this manual.

From experiences in teaching undergraduate Fluid Mechanics classes in the College of Engineering at USU, the writer has found that a 15-30 minute lecture on the use of USU-NETWK, accompanied by a two page handout similar to this preview equips students with sufficient understanding to use the program in solving assigned steady state analysis and design problems. The input data needed to define a network analysis problem is described below and illustrated by a small example that follows.

**MINIMUM INPUT DATA REQUIREMENTS**

**Line No. 1** - Provides the title for the network. This title is followed by a /* in columns 1 and 2 in a new line after the title is complete.

**Subsequent Lines** - The remaining lines of input contain command names followed by data of the type dictated by that name. Command names are words such as PIPES, NODES, PUMPS, BOOSTer, RESERvoirs, starting in column 1 and spelled to 5 upper case characters correctly. Each command is followed by lines of data which provide information of the type denoted by the command. These data are free format, i.e. separated by one or more blanks, or a comma, and may be terminated with a / if the data given previously for this item is to be used. The word RUN or END denotes end of input data. The items contained in lines of input after the command names are as given below.

**PIPES**
Pipe no., Upstream node no., Downstream node no., Length, Dia., Coef.

**NODES**
Node no., demand, elev.

**PUMPS**
Pipe No., \( Q_1', H_1 \), \( Q_2', H_2 \), \( Q_3', H_3 \), Sump elev.

**BOOST**
Pipe No., \( Q_1', H_1 \), \( Q_2', H_2 \), \( Q_3', H_3 \)

**RESER**
Pipe No., Elev. of water surface in reservoir

The network on the following page gives an example of how this is actually done. If the data given below under "Input Data" were stored on a computer disk file, USU-NETWK executed, and this file name given for the input file when prompted for it, then the output given under "SOLUTION" would shows the output produced by USU-NETWK for this network.

**Input data**
THIS IS A SIMPLE EXAMPLE /*

**PIPES**
1 0 1 500 8 .0102
2 1 2 800 6/
3 4 2 1200/
4 5 3/
5 2 3 1000/
6 4 3 800 8/
7 0 4 500/

**NODES**
1 0 50
2 1/
3 1.5/
4 0/

**PUMPS**
7 1 55 1.2 54 1.4 5 50

**RESER**
1 100

**RUN**

When USU-NETWK is executed it will ask you what the name of the input data file is. You then give the name of the disk file that contains the input data for the network.
give TTY or REMOTE in response to this prompt, then USU-NETWK expects you to enter the data line after line directly from the keyboard. Next you will be ask (prompted) to supply a name for the output file. A file with the name you give will be opened and the solution written to it for later printing or examination by you with an editor. If in response to this second prompt you give TTY or REMOTE, then the solution output will come directly to your terminal, or monitor if you are working on a PC.

SOLUTION

TITLE GIVEN TO NETWORK

THIS IS A SIMPLE EXAMPLE

ALL DEMAND FLOWS ARE MULTIPLIED BY 1.0000

PIES 7
NODES 4
SOURCE PUMPS 1
BOOSTER PUMPS 0
RESERVOIRS 1
MINOR LOSSES 0
FRES 0
NOZZLES 0
CHECK VALVE 0
BACK PRES V 0
DIF. HEAD DEV 0
SPECIFIED PRES 0

PIPE 2ND ORDER COEF LINEAR COEF SHUT-OFF HEAD SUMP ELEV
RES. (NOZZLE) PIPES & THEIR ELEV. ARE

<table>
<thead>
<tr>
<th></th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>800.00</td>
<td>1315.00</td>
<td>-660.00</td>
<td>50.00</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

JUNCTION EXT. FLOW PIPES AT JUNCTION

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>1</td>
<td>0.000</td>
<td>1.000</td>
<td>0.500</td>
<td>1.500</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1.000</td>
<td>-2</td>
<td>-3</td>
<td>-4</td>
<td>-5</td>
<td>-6</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>1.000</td>
<td>-5</td>
<td>-4</td>
<td>-3</td>
<td>-2</td>
<td>-1</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>0.000</td>
<td>3</td>
<td>6</td>
<td>9</td>
<td>12</td>
<td>15</td>
</tr>
</tbody>
</table>

FLOW FROM PUMPS AND RESERVOIRS EQUALS 2.500

ITERATION= 1 SUM= .538E+00
ITERATION= 2 SUM= .547E-01
ITERATION= 3 SUM= .499E-02

PUMPS:

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>45.62</td>
<td>1.25</td>
<td>6.49</td>
<td>4.84</td>
<td>116.14</td>
<td></td>
</tr>
</tbody>
</table>

UNITS OF SOLUTION ARE

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>50</td>
<td>100</td>
<td>inch</td>
<td>foot</td>
</tr>
<tr>
<td>6</td>
<td>30</td>
<td>100</td>
<td>foot</td>
<td>psi</td>
</tr>
<tr>
<td>5</td>
<td>100</td>
<td>100</td>
<td>cfs</td>
<td></td>
</tr>
</tbody>
</table>

DARCY-MEISBACH FORMULA USED FOR COMPUTING HEAD LOSS

PIPE DATA

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>500.0</td>
<td>8.0</td>
<td>.010200</td>
<td>1.25</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>800.6</td>
<td>6.0</td>
<td>.010200</td>
<td>1.25</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>3</td>
<td>1200.0</td>
<td>6.0</td>
<td>.010200</td>
<td>1.25</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>4</td>
<td>500.0</td>
<td>8.0</td>
<td>.010200</td>
<td>1.25</td>
</tr>
</tbody>
</table>

NODE DATA:

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.000</td>
<td>.00</td>
<td>50</td>
<td>46.72</td>
<td>20.25</td>
</tr>
<tr>
<td>2</td>
<td>1.000</td>
<td>448.83</td>
<td>50</td>
<td>39.48</td>
<td>17.11</td>
</tr>
<tr>
<td>3</td>
<td>1.000</td>
<td>673.24</td>
<td>50</td>
<td>39.48</td>
<td>17.11</td>
</tr>
<tr>
<td>4</td>
<td>.000</td>
<td>.00</td>
<td>50</td>
<td>42.31</td>
<td>18.33</td>
</tr>
</tbody>
</table>
ANOTHER POSSIBILITY

An alternative to providing pipe lengths is to give the x and y coordinates at the nodes of the network. These coordinates are necessary if a plot of the network and its solution is requested. To inform USU-NETWK that pipe lengths will not be given, but rather these should be computed from x and y coordinates, the option LENGON=0 must be added to a list of options between $SPECIF$ and $END$ i.e. a line such as,

$SPECIF$ LENGON=0 $END$

is added after the */. With the option LENGON=0 included in the list of options, lengths are not given as the 4th item after the PIPES command, but rather x and y coordinates are given as items 4 and 5 after the NODES command. In order to give the locations, e.g. the coordinates, of the sources of supply (the reservoir and the source pump) it will also be necessary to number sources of supply as nodes. The reservoir that supplies the previous "simple network" will be numbered as node 5, and the source pump will be given the node number 6. If the x and y coordinates are as given in the small table below (these coordinates don't give the same length to all pipes as used previously), then an alternative "Input Data File" would contain the following:

<table>
<thead>
<tr>
<th>Node</th>
<th>x-cord.</th>
<th>y-cord.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>500</td>
<td>500</td>
</tr>
<tr>
<td>2</td>
<td>1000</td>
<td>1125</td>
</tr>
<tr>
<td>3</td>
<td>1591</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>2091</td>
<td>625</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>500</td>
</tr>
<tr>
<td>6</td>
<td>2591</td>
<td>625</td>
</tr>
</tbody>
</table>

In addition to the differences discussed above, please note the following differences occur in this input data and that used earlier because NODESP=1 is given in the options list: 1. there are 6 lines of data after the NODES command instead of 4; two extra lines for the two source nodes, 2. the demands at these source nodes are 0, e.g. the amounts of flow supplied by the sources are not demands, but will be flowrates determined as part of the solution, 3. source PUMPS and RESERVOIRS are identified by their node numbers now rather than the pipes that connect them to the network, and 4. pipes 1 and 7 now list the source nodes 5 and 6 respectively, as their upstream nodes rather than 0. In the previous "input data" 0's for node numbers denoted "non-existent" nodes, but are needed for if there were omitted then the pipe lengths would be taken as the downstream node numbers.

Why don't you write down the input data that would be correct if the option LENGON=0 were not included in the $SPECIF$ list, but NODESP=1 remains in the list.

Lengths for pipes, that will be used in the solution for computing headlosses, etc. can be given under the PIPES command and x and y coordinates, that will be used for plotting of the network, can also be given under the NODES command by leaving LENGON=0 out of the options list but adding NETPLT=1 to this list that indicates that a plot of the network is to be created.

Example Simple Network

![Diagram of network](example_simple_network.png)
FLUID MECHANICS PROBLEMS

While USU-NETWK is designed to provide solutions to large looped systems of pipe it can also be used to solve small problems involving one or two pipes. Below are brief statements and sketches for simple fluid mechanics problems similar to those that may be given to college students as homework assignments to be worked using a pocket calculator, after applying the appropriate principles from fluid mechanics. Most of these problems would require considerable effort if done by hand. You may find it interesting to sharpen your fluid mechanics skills by solving the problems by hand. Then use USU-NETWK to obtain a solution to each problem. In solving these problems you will be using a number of options, etc. that USU-NETWK allows. If you don't have a lot of time to do computations you might set up the equations that need to be solved, and use the solutions from USU-NETWK to verify that you have used the correct equations to describe the problems. The input data for each problem follows, but you should see if you can make up the input data without looking first.

1. What head is required to produce the flow rate of 500 gpm in the pipeline below?

2. What flow rate occurs between the reservoirs?

3. What is the flow rate in each pipe?

4. Assume the head produced by the pumps are known and equal to Hp(1)=50 ft and Hp(2)=40 ft. What is the flow rate in each pipe?

5. The heads of the pumps are Hp(1)=50 ft, Hp(2)=40 ft, Hp(3)=35 ft, and Hp(4)=60 ft. Find the flow rate in each pipe.

6. The pumps in Problem 5 have characteristics given in the table below. Find the flow rate in each pipe.

<table>
<thead>
<tr>
<th>Pump</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>0.3</td>
<td>0.3</td>
<td>0.1</td>
<td>0.15 cfs</td>
</tr>
<tr>
<td>H1</td>
<td>60</td>
<td>60</td>
<td>60</td>
<td>75 ft</td>
</tr>
<tr>
<td>Q2</td>
<td>1.0</td>
<td>1.0</td>
<td>0.6</td>
<td>8.5 cfs</td>
</tr>
<tr>
<td>H2</td>
<td>55</td>
<td>45</td>
<td>35</td>
<td>60 ft</td>
</tr>
<tr>
<td>Q3</td>
<td>1.8</td>
<td>1.8</td>
<td>1.0</td>
<td>1.0 cfs</td>
</tr>
<tr>
<td>H3</td>
<td>30</td>
<td>25</td>
<td>25</td>
<td>40 ft</td>
</tr>
</tbody>
</table>

7. What head and capacity should the pump produce in order to maintain a pressure of 40 psi at node 3?

8. What is the least cost combination of pump and pipe size to use for the problem below? The following are basic economic data: Life of pipe = 80 yrs, life of pump = 20 yrs, capital cost of pumping station, etc. = $40,000, cost of pipe is as program defaults to, costs of energy = 0.12/kilowatt-hour, interest rate = 10%. A single size pipe is to be used.

9. What diameters are needed to have 40 psi at all nodes?

10. Pressures were measured for the pipe flow below as shown on the sketch. Determine both the appropriate Darcy-Weisbach equivalent sand roughness coefficient, and the Hazen Williams coefficient for the two pipe.
FLUID MECHANICS PROBLEM 1
/*
$SPECIF NFLOW=1 $END
PIPES
1 1 2 2500 6 .005
NODES
1 -500 0
2 500 0
RUN
2 92.3
------------------

SIMPLE FLUID MECHANICS PROBLEM 2
/*
$SPECIF NUNIT=3,NFLOW=3,OUTPU1=2 $END
PIPES
1 0 1 1500 15 .02
2 1 0 1500 15 .02
NODES
1 0 0
RESER
1 100
2 40
RUN
------------------

SIMPLE FLUID MECHANICS PROBLEM # 3
/*
$SPECIF OUTPU1=2 $END
PIPE-
1 8. 1500. .004 1 0. 360.
2 6. 1200. .004 1
3 6. 1500. .004 1
RESER
1 400
2 380
3 350
RUN
------------------

SIMPLE FLUID MECHANICS PROBLEM # 4
/*
$SPECIF OUTPU1=2 $END
PIPE-
1 8. 1500. .003 1 0. 50.
2 6. 2500. .003 1 0. 50.
3 6. 2200. .003 1 2
4 8. 1000. .003 2
RESER
1 100
4 90
DHEAD
2 50
3 40
RUN
------------------

SIMPLE FLUID MECHANICS PROBLEM # 5
/*
$SPECIF COEFRO=150,OUTPU1=2 $END
PIPE-
1 6. 3500. 1 -3. 0. 2 3. 0.
2 6. 3000. 1 2
3 4. 2800. 1 2
4 4. 4000. 1 2
DHEAD
1 50.
2 40.
3 35.
4 60.
RUN
2 92.3
------------------

SIMPLE FLUID MECHANICS PROBLEM # 6
/*
$SPECIF COEFRO=.005,OUTPU1=2 $END
PIPE-
1 6. 3500. 1 -3. 0. 2 3. 0.
2 6. 3000. 1 2
3 4. 2800. 1 2
4 4. 4000. 1 2
BOOSTER
1 .3 60 1 55 1.8 30/
1 .3 50 1 45 1.8 25/
3 .3 40 .6 35 1.25/
4 .15 75 .5 60 1.40/
RUN
2 92.3
------------------

SIMPLE FLUID MECHANICS PROBLEM # 7
/*
$SPECIF COEFRO=.005,OUTPU1=2 $END
PIPE-
1 10. 1500. 1 1. 40.
2 8. 1200. 1 2. 8 50.
3 6. 900. 2 3. 6 50.
4 6. 1000. 2
RESER
1 100
4 90
DHEAD
1 50. -3 4 40
RUN
------------------

SIMPLE FLUID MECHANICS PROBLEM 8
/*
$SPECIF ICOST=1,NFLOW=1,NPGPM=1 $END
PIPE-
1 18. 1800. 1 -2050. 0. 2
2 18. 3000. 2 50. 0.
RESER
2 132.31
RUN
INTEREST=.10
LIFE=20
PUMPS
CAPI=40000.
UNIT=.12
LIFE=80
END
------------------

SIMPLE FLUID MECHANICS PROBLEM 9
/*
$SPECIF DESIGN=1,NOMSOL=1,IRGL=0 $END
PIPES
1 0 1 1000 0 .003
2 1 2 1800. 1
3 1 5 1900/
4 2 3 1600/
5 5 6 1800/
6 3 4 1900/
7 6 7 1500/
NODES
1 2 45 40.
2 5 38
3 .5 30
4 1. 20
5 4 40
6 .6 25
7 .9 14
RESER
1 150.
RUN
------------------

SIMPLE FLUID MECHANICS PROBLEM 10
/*
$SPECIF DESIGN=3,IRGL=0 $END
PIPES
1 1 2 2500. 10 0
2 2 3 3000. 8. 0
NODES
1 1 2 0 52.2
2 1.5 20 36.4
3 1 0 40.
RUN
3 92.31
------------------
Problem from Fluid Mechanics Text

Below is a pipeline problem taken from a Fluid Mechanics Text Book. Solve the problem using USU-NETWK.

"At a section of a pipe where it divides into two pipes the pressure is 300 kPa, \( z = 40 \) m and \( V = 3 \) m/s. One of the parallel pipes, which is 0.2 m dia and 400 m long, contains a globe valve, two closed return bends, and a turbine which extracts 98 Nm/kg of energy from the flow. The other pipe, which is 400 m long, contains a pump which supplies 50 Nm/kg of energy to the flow. The pipes both exist into a reservoir of surface elevation 50 m. The pipes are commercial steel. The temperature of the water is 15°C. Find the diameter of the pipe containing the pump if the \( Q \) entering the parallel pipes is 0.0877 \( \text{m}^3/\text{s} \)."

The following could be done in defining the problem for USU-NETWK: (1) This could be interpreted as a 3 pipe network; pipe 1 joins into the 2 parallel pipes and its other end is attached to a reservoir with the proper surface elevation just slightly different from the specified head of, \( z+p/\gamma+V^2/2g = 40+300/9.8+9/19.62 = 71.07\) m specified at the junction of the parallel pipes in the problem. This pipe 1 will be small in diameter and long so it will carry a negligible amount of flow, especially since the HGL at its two ends will be about equal. (2) The two parallel pipes of the problem will feed reservoirs each with a water surface elevation of 50 m, as stated in the problem. (3) The turbine will be handled as a pump with a negative head. The information given for the pump and turbine indicate units of Nm/kg of energy. If these units are divided by \( g \) (the acceleration of gravity), then the more conventional meters of head are obtained. Thus the extraction of 98 Nm/kg is equivalent to 9.9898 m of head, and the input by the pump of 50 Nm/kg is equivalent to 5.0968 m of head.

One might use the type #1 DHEAD command for these heads, or alternatively specify pumps, as suggested above. If pumps are used then either a pump curve must be generated, or a guess about the flow rate made, the power computed, and if this guess is not very close adjust the horsepower of the pumps. Notice for the turbine the heads associated with the flow rates will be negative, and the power will be negative if this option is selected. (4) The minor losses can be combined into a single minor loss coefficient equal to about 11. With this approach the for USU-NETWK to solve this problem might be:

Fluid Mechanics problem from text book
/*
 SPECIFIC DESIGN=1,NUNIT=2,NFLOW=3,NPGPM=3 SEND
 PIPES
 1 0 1 1000 .05 .00046
 2 1 0 400 .2 /
 3 1 0 400 ./
 NODES
 1 -.0877 40 71.07
 RESER
 1 71.08
 2 50
 3 50
 MINOR
 2 11
 BOOSTER
 3 .02 5.2 .04 .5 .06 4.9 0 ./
 2 .02 -9.5 .05 -10 .11 -14 0 ./
 RUN

 You might adjust the pump curves to get a more precise answer.
## UNITS and CONVERSION FACTORS

### BASIC CONVERSION FACTORS -- ES TO SI SYSTEM

| LENGTH | 1 meter (m) = 3.280 839 9 feet (ft) | 1 ft = 0.3048 m |
| FORCE | 1 Newton (N) = 0.224 809 lb | 1 lb = 4.448 23 N |
| MASS | 1 kilogram (kg) = 0.068 521 78 slugs | 1 slug = 14.593 9 kg |
| TEMPERATURE | 1 °Kelvin = 9/5 °Rankine | °F = 32 + (9/5)°C |

| LENGTH | 1 ft = 0.3048 m |
| | 1 inch = 0.0254 m |
| | 1 statute mile = 1.609 km |
| | 1 nautical mile = 1.852 km |
| PRESSURE | 1 psi = 6894.24 N/m² |
| | 1 lb/ft² = 47.877 N/m² |
| | 1 in Hg = 0.491 psi |
| | 1 standard atmosphere = 14.699 psi |
| SPECIFIC WEIGHT | 1 lb/ft³ = 15.707 N/m³ |
| | 1 gm/cm³ = 62.43 lb/ft³ |
| KINEMATIC VISCOSITY | 1 ft²/s = 0.0929 m²/s |
| | 1 stoke = 1 cm²/s = 0.001 08 ft²/s |
| FLOW RATES | 1 cfs = 0.283 m³/s (cms) |
| | 1 slug/s = 14.594 kg/s |
| | 1 lb/s = 4.448 N/s |
| POWER | 1 ft.lbm/s = 1.356 N.m/s |
| | 1 hp = 33,000 ft.lbm/min = 550 ft.lbm/s = 746 N.m/s |
| | 1 kwatt = 1.341 hp = 738 ft.lbm/s = 0.947 Btu/s |

### KINEMATIC VISCOSITY of WATER

<table>
<thead>
<tr>
<th>°F</th>
<th>ft²/s</th>
<th>°C</th>
<th>m²/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>1.931 x 10⁻⁵</td>
<td>0</td>
<td>1.785 x 10⁻⁶</td>
</tr>
<tr>
<td>40</td>
<td>1.664 &quot;</td>
<td>5</td>
<td>1.519 &quot;</td>
</tr>
<tr>
<td>50</td>
<td>1.410 &quot;</td>
<td>10</td>
<td>1.306 &quot;</td>
</tr>
<tr>
<td>60</td>
<td>1.217 &quot;</td>
<td>15</td>
<td>1.139 &quot;</td>
</tr>
<tr>
<td>70</td>
<td>1.059 &quot;</td>
<td>20</td>
<td>1.003 &quot;</td>
</tr>
<tr>
<td>80</td>
<td>0.930 &quot;</td>
<td>25</td>
<td>0.893 &quot;</td>
</tr>
<tr>
<td>90</td>
<td>0.826 &quot;</td>
<td>30</td>
<td>0.800 &quot;</td>
</tr>
</tbody>
</table>
SUMMARY OF MOST FREQUENTLY USED OPTIONS

(These are included in the $SPECIF$ list. A * denotes default values)

**COEFRO** default roughness coef. \((0.02^2)\)

**DESIGN** analysis or special design sol.
0* analysis solution
1 special design sol.
3 special solution to solve for roughness coefficient

**GAMMA** specific weight of fluid 62.4 (for water)

**HGL** type of pressure specification at nodes (No meaning if \(DESIGN=0\))
0* psi (or kPa for SI units)
1 head at node in ft (m for SI units)
2 \(HGL\) at node in ft (m for SI units)
(in addition 3 & negative values indicate special input will be given)

**INPUTA** form of input used
0* standard in body of manual
1 Appendix A input
2 Appendix B input #1
3 Appendix B input #2

**ISIML** steady-state or time dep. sol.
0* steady state analyses
1 time varying simulation

**LENGON**
0 length of pipe not given but other x-y coordinate
1* length of pipe given

**NETCHK** single or dual definition of network
0* single input of network data
1,2,-1 or -2, duplicate input of network layout

**NETPLT**
0 no plot
1 calcomp plot; write PLTNET plot file
2 or 3 dim. perspective plot; write special plot file
13 or 14 formatted input from PIPEINPT, etc

**NPRINT** level of intermediate output -3 to 10 allowed, 0 is default level
0* pipes & nodes identified by 8 char. strings
11 pipes & nodes identified by 8 char. strings
12 only nodes identified by 8 char. strings
1000+,etc control of format in solution tables

**NUMPIP** numbering of pipes
0* pipe numbers given in input data
1 numbers of pipes not included, (numbered according to order of input data)

**NSYMMT** method of solution to be used
0 nonsymmetric Jacobian, tighter banding
5* sparse matrix methods
2 or 7 smoothing to laminar flow for D.W. eq.

**OUTPUT** form of solution tables
0 single table, 130 cols.
1* two tables, 80 cols.
11 pipes & nodes identified by 8 char. strings
12 only pipes identified by 8 char. strings
13 only nodes identified by 8 char. strings
1000+,etc control of format in solution tables

**PEAKF** \((FLOWFC)\) multiplier of given demands (1.0 default)

**PRESMA** max. pressure that should not be flagged (default--does not flag)

**PRESMI** min. pressure that should not be flagged (default--does not flag)

**VELMAX** max. velocity that should not be flagged (default--does not flag)

**VELMIN** min. velocity that should not be flagged (default--does not flag)

**VISC** kinematic viscosity of fluid \((1.41 \times 10^{-5} \text{ ft}^2/\text{sec})\)--ES units,
SUMMARY OF COMMANDS THAT ENTER DATA
and default order of data after command

PIPECES
No., upstream node, downstream node, length, dia., coef.

NODES
No., demand, elevation (pressure, x & y coord. and check data may also follow)

RESERVOIRS
designation(pipe--NODEESP=0,node--NODEESP=1), water surface elevation

PUMPS
designation, Q, H, Q, H, Q, H, elev. of water surface
or
designation, power, Q, elev. of water surface/

BOOSTER (or BPUMPS)
designation, Q, H, Q, H, Q, H, 0
or
designation, power, Q, 0/
or
designation, coefficient of 2nd degree polyn.

ROTATIONAL speed
designation, ratio of speed to that of data

SERIES
designation, no. of pumps in series

PARAL
designation, no. of pumps in parallel

DHEAD (differential head)
pipe, ΔH, node, source of supply, HGL

VALVE (pressure reduction valves, PRVs)
pipe, dist. to valve, HGL setting

VALVC (check valves)
list of pipe numbers/

BPVAL (back pressure valve, BPVs)
pipe, dist. from valve to end of pipe, HGL setting

NOZZLES (see manual if NODESP=1)
pipe, dia., discharge coef., elev., est. of HGL

PIPE-(items below may be omitted, pipe and node number are integers; all other values must contain a decimal point)
pipe, dia., length, coef., upstream node, demand, elev. (pressure, x-cord, y-cord), downstream node, demand elev. (pressure, x-cord, y-cord)

MINOR
pipe, loss coef,

QGIVE #
pipe, flowrate

DFRAC
mult. of following demands

ITERA
No. of iterations allowed for solution

UNITS
designator of type of units

ERROR
convergence criteria

VISCOSITY
kinematic viscosity of fluid

WEIGHT
specific weight of fluid

PEAKF #
peaking factor, list (range) of nodes

PRINT
designator of level of intermediate output

DESIGN
specifies that a special design solution is wanted (same as DESIGN in $SPECIF list) SETPR & SETHG pressure or HGL elev. known at some node but demand is unknown elsewhere, etc.

SETPR & SETHG (pressure or HGL elevation known at some node but demand is unknown)
node of known pressure (HGL), node of unknown source pipe no., known pressure (HGL)

LPIPE, NLPIPE, LNODE, NLNODE (The 1st L or NL refer to list, or not list, respectively, and the PIPE and NODE part of the name indicate lines in the PIPE DATA or NODE DATA table, respectively.)
List (range) of pipe or node numbers that should, or should not be listed in output solution tables

FORMAT (This command allows USU-NETWK to be told that formatted data follows, and gives the FORTRAN FORMAT of this data.)
FORMAT specification of input data that follows including the beginning and ending parentheses

CHECK (no data follows)
check data for errors but does not obtain a solution

END (no data follows)
end of detailed input data

RUN (no data follows)
same meaning as END, if command CHECK is included in input END must be used

# implement only on simulation version of USU-NETWK
INTRODUCTION

Analysis of flow in pipe networks is required to determine the ability of water distribution systems to supply water to residential, commercial, and industrial users at acceptable pressures and to meet emergency demands, such as those needed to fight fires. The digital computer has made analyses of larger pipe networks feasible. However, a good computer program should use computer resources wisely, accurately simulate hydraulics, and minimize the effort required by a user to obtain solutions. Program USU-NETWK was developed to meet these requirements.

This manual provides the basic information needed in preparing data needed to obtain analyses and design solutions, including simulations through time, with USU-NETWK. The manual does not describe the methods used by the program to obtain these solutions. Methods for obtaining such solutions can be found in the book "Analysis of Flow in Pipe Networks" which is devoted to methods for solving large flow distribution systems, as well as the other technical literature on the subject. The material herein is intended to acquaint the user with the input data requirements of USU. In brief, however, the program solves the corrective flowrate system of equations using the Newton method.

USU-NETWK has been developed to allow the user much versatility in obtaining computer solutions to any steady-state flow problems dealing with flow in a piping system. The program has been designed to make it as easy as possible to use. The amount of data need to describe a network is minimal, and free of any format requirements, e.g. items do not need to be in any specified columns. The more general program allows alternate forms of input. The form which is considered most basic is described in the main portion of this manual. In this standard form of input, names such as PIPES, NODES, PUMPS etc. are used to enter data of a given type. Appendices A and B describe alternative forms of input. While there are several variations of input data allowed within each category, it is useful to consider the input data as being in: 1. the category of pipe oriented input, or 2. node oriented input. With pipe oriented input the network's layout is defined by the end nodes associated with each pipe. These end nodes define the network's connectivity and topology. In this form of input it is convenient to also supply pipe diameters, lengths, and wall roughness coefficients on the same line as the pipe number and its upstream and downstream nodes. Elevations of nodes and the demands at the nodes are additional data, that are needed for an analysis but which are not needed to define the network's layout. In the standard form of input these additional node data are given on separate lines.

A command PIPE-nodes, however, allows this additional data to be given after any node number, as does the form of input described in Appendix A. In preparing the data under the pipe oriented input, category 1 above, the user goes from one pipe to the next of the network, until he includes every pipe in the network; therefore it is pipe oriented.

In the node oriented form of input, category 2 above, the layout, or topology, of the network is defined by giving the pipe numbers which join at each node of the network. In this form of input the user moves from node to node of the network and jots down the pipe numbers that join at each node. In this form of input the demand at each node, as well as the elevation of the node are logically given on the same line as that node. However, such necessary auxiliary data as pipe diameters, lengths and wall roughness coefficients are given on separate lines. This latter node oriented form of input is described in Appendix B.

There are a couple of reasons for discussing the different type of input here. One, to let you know why the Appendixes are included in the manual, but more importantly to alert you to the fact that an effective way of discovering errors that you make in input is to use the other type of data in checking your network. USU-NETWK prints out the node oriented form of data that describes the network whether you use the pipe or node oriented input under the default level of extra output. That is the pipes numbers are listed that join at each node. Likewise, with the next level of extra output, a special table is printed that gives the node numbers at the ends of each pipe. When you prepared data on the basis
of "pipe oriented" input a most effective way of checking the network that you defined is to examine the tables giving pipes at nodes while looking at the map of the network, and vice versa.

By giving USU-NETWK both the node as well as the pipe oriented input you can have the program check that the two definitions are identical. Any discrepancies will be identified, and processing will be terminated. Often giving such a dual definition of a network is preferred over checking by hand, but it does require more input data. (See the option NETCHK for a description of how to define the layout of a network twice.)

The input form described in the main portion of this manual uses a description name, or command name, to enter data of a given type. These names are associated with the type of data they enter. Some of these names are: PIPES, NODES, PUMPS, RESERvoirs, VALVES. There is a list of data items under each name, but the order of items in the list can be changed with an option so that often input files that were developed for other network solvers, can be read with minor changes by USU-NETWK. In the alternative forms of input described in Appendix A and B, no such command is used. Rather the different data must be ordered in accordance with the prescribed sequence.

Some users wish to identify pipes and/or nodes by characters, and not just numbers. If desired strings of any 8 characters, including numbers and blanks, may be used to identify pipes and/or nodes. When using this option, however, the free format input is lost for this portion of the data because field widths of 8 characters must be used for these identifies. (See the option OUTPU1 for a description of how this is done.)

For any of the forms of input, loops of the network may be given if desired. Generally users prefer to let USU-NETWK find the loops in the network because giving the loops requires additional data, and puts more burden on the user, with more possibility for error. Therefore, giving data for loops in the network is not a recommended practice. When the user opts to give loop data, then rather than determining the loops internally, USU-NETWK will use the loops that are given. This option may be used if the program is unable to generate the loops properly for a given network, which is highly unlikely. This option might also be used if a large number of separate analyses are anticipated that do not change the network. Then the process of defining the loops can be eliminated but taking the loops defined by USU-NETWK from the output file and editing them into the input file.

Extensive internal checking of the data for consistency and correctness in describing a valid network occurs, and meaningful error messages are provided to assist the user in locating errors should a problem be detected by USU-NETWK. USU-NETWK has an engineering economic analysis package attached to the hydraulic analysis packages. Therefore, if desired, you can request as complete a cost analysis be done as you are willing to provide cost data for. Having an engineering economic analysis done as part of the computer "run" provides you information related to costs of alternative designs or configurations.

The general features and capabilities of USU-NETWK, are described in the next section. This description should assist you in understanding the types of networks that can be analyzed, and the relatively large number of options that are available. This description will help you decide what components of your distribution system to include in the analysis, but you may wish to read it later, after you have had some experience in using USU-NETWK. You may skip to the following section, "Description of Input Data."
VERSATILITY OF COMPUTER PROGRAM

Features of the program are:

1. A solution will be provided for any system of connected pipes with (or without) supply or source pumps, booster pumps, supply reservoirs or tanks, minor loss devices, pressure regulators (or pressure reducing valves, PRV), check valves and orifices at ends of pipes. The network generally will contain loops, but may consist only of pipes supplied by reservoirs and/or source pump but without any natural loops, or may consist only of pipes (with or without loops) with all inflows and outflows known. Branching networks without real or pseudo loops will also be accommodated. Special input allows p consist of pipes in some branched system to be described by just a few kilines and the solution for such will size all pipes to give the specified sof the hydraulic grade line.

2. Either English (ES) or International (SI) units may be used. If ES units are used, pipe diameters may be given in inches or feet, pipe lengths in feet of 1,000 feet, and demands (or consumptions) may be in cubic feet per second (cfs), fallons per minute (gpm) or million gallons per day (mgd). If SI units are used, pipe diameters may be given in meters or centimeters, and lengths in meters or 1,000 meters and demands (or consumptions) in cubic meters per second, or liters per second. In addition to the above volumetric flowrates, mass or weight flowrates may be used. When using ES units weight flowrates are in Ib/sec and mass flowrates in slugs/second. When using SI units weight flowrates are in newtons/second and mass flowrates in kg/second.

3. Frictional head losses may be based on the Darcy-Weisbach formula, the Hazen-Williams formula or the Mannings formula. The program will determine internally whether the Darcy-Weisbach or Hazen-Williams formula is to be used by the magnitude of the first roughness coefficient. When the Darcy-Weisbach formula is used, no approximations are used. Rather, the proper equation is determined and the friction factor computed exactly dependent upon both the Reynolds number and the relative roughness c/D. Since the Darcy-Weisbach equation is more fundamentally sound from a hydraulic viewpoint and its use results in zero (or insignificant) amount of additional computer time for a network analysis its use is recommended.

4. Loops in the network are determined internally. If PRV's are present and a path of connected pipes does not exist between consecutively listed reservoirs and/or source pumps, this is noted, and all alternative reservoirs, and source pumps are examined to find a path. If no path is available, it is noted that the PRV's isolated the network into two or more pressure zones as the solution continues. If desired, the end points for such pseudo loops between reservoirs and/or source pumps can be specified. This option permits the user to assist the program in minimizing the amount of computations by assisting the program in producing a narrower banded Jacobian matrix. An option can be turned on the informs USU-NETWK that the pairs of sources of supply selected between which to form pseudo loops should be optimized to result in the least number of terms in these energy equations.

5. Reservoirs (or tanks) and source pumps may supply the network, and booster pumps may exist in any pipeline. Operating characteristics of pumps may be defined by (a) supplying three pair (or up to 10 pairs if selected by an option) of flow rate versus head produced. These values are taken from a pump's characteristic curve, (b) by giving the power (horsepower if ES units or kilowatts if SI units) that the pump supplies the flow and its normal capacity (flow rates may be cfs, gpm or mgd if ES units are used, and m³/s or liters per second if SI units are used), or (c) by providing the coefficients for a 2nd degree polynomial that becomes the mathematical description of how much head the pump produces as a function of the flow passing through it. Any number of pumps may operate in series or parallel at a given station. If not specified, one is assumed. The pump characteristics can be given for different rotational speeds than the pumps are operating at, and the rotational speeds of pumps can be changed during time-dependent solutions, or series of solutions for the same basis network with components changed.

6. Input data are free format, i.e., can be punched in any columns with a blank or a comma as the delimiter between individual data items. The exception to this rule is that commas separate
option items in the $SPECIF$ and $DATA$ lists. Any time the last items are identical to those given previously these need not be included in the line of data. The line is terminated by a /.

Termination of some lists, which may otherwise be very long is with a /. When using the PIPE-NODES command items are arbitrarily omitted within the record but not terminated with a /.

Generally pipe lengths are given. However, in place of giving lengths, the x- and y-coordinates at nodes may be used to define the physical layout of the network.

7. Pressure reducing valves and/or back pressure valves may be inserted in any pipe of the network provided they don't exist in consecutive pipes with no other pipes at their joint junction or the pipe supplied by reservoirs or source pumps, i.e., PRV's don't exist in pipes which are supply lines for then they become the effective reservoir. The PRV's act as check valves if flow reverses in pipes containing them during the solution. If the upstream head (or pressure) is less than the valve's setting, the program will note this and replace the PRV with a minor loss device. Thus, the 3 ways that PRV's operate are simulated, i.e., (a) They may operate in maintaining the downstream pressure and head constant. (b) They may shut off preventing reverse flow but allowing the downstream pressure to drop below the valve's pressure setting. (c) They act as check values shutting off the flow in that pipe should the flow attempt to be in the direction opposite to the direction of the PRV.

The same modes of operation exist of back pressure valves, BPV's.

8. A string of separate analyses of the same basic network is allowed in which selected components of the network are changed. Any of the following changes are allowed: (a) multiplication of all previous demands by a specified factor, (b) the diameter, length or roughness of selected pipes, (c) the demands at selected modes, (d) the water surface elevation of selected reservoirs of water surface from which source pumps obtain their supply, (e) the number of pumps operating in parallel or series may be changed, (f) the rotational speed of the pump can be changes, (g) the elevation of any nodes, (h) all pipe roughness to a new value, and (i) coefficients for minor loss devices.

9. Minor loss devices can be inserted in any pipe of the network.

10. The printed output is self-explanatory, and the amount of intermediate output is easily controlled. Two alternatives are available for the final solution results (not available when doing time dependent simulations). One such alternative produces two tables: one with an entry for each pipe and one with an entry for each node. (These tables are limited to 80 columns for teletype output. The other alternate gives the node data as part of the entry for each pipe. (This output uses 131 columns.) If checking of problem specifications is desired before completing a solution, this can be accomplished by examining the printed output if the "check" option is specified giving pipes joining at junctions, internally generated loops and other such data.

11. Extensive checking of input data occurs, which will identify the majority of the commonly made mistakes. To further verify the correctness of the network the user can select an option in the program that allows him to define the layout of the network twice, once using the pipe oriented form of input and once from the node oriented form of input. USU-NETWK will identify any discrepencies between these two definitions of the network.

12. Check valves are allowed in any pipe. Should check valves shut-off, the system of equations being solved is altered to reflect this rather than setting the resistance of this pipe high as is commonly done. Therefore, pipes containing shut-off check valve, or PRV's have exactly zero flow through them.

13. Reservoirs and source pumps may be given a node number or left unnumbered, i.e., given a zero when providing the input data. Generally, they are left unnumbered; thus, reducing the amount of data required. If such source nodes are numbered, then reservoirs and source pumps are identified in the input data by this node number rather than the usualy pipe number which connects this source to the network. Furthermore, and option can be turned on that allows pipes, nodes, or both pipes and nodes to be identified by 8 character strings. Also remarks can be placed at the end of input lines, and these remarks can be printed in the solution tables. Also lines in the input may be "comment out" of active use by placing a % in front of them.

14. The pipes and nodes may be numbered arbitrarily (integers less than 5000), and
do not need to be entered in the sequence of ascending numbers. Furthermore, pipes do not need to be numbered. If not numbered, then they are identified in the output by the sequence number of the input data as well as the nodes at their two ends.

15. Nozzles (or orifices) are allowed at the ends of any pipe. The diameters of these nozzles can be specified differently than the pipe diameters and discharge coefficients may be given. Otherwise, default discharge coefficients are taken.

16. The program is designed for batch or time-shared use. (Use of the program on a PC is essentially identical to time-shared use in that the user responds to prompts for the input and output files, etc.) If use occurs from a teletype under time-shared use, the program goes into an interactive mode allowing input and output files to be designated dynamically by the user.

17. The program can be instructed to go into simulation mode. If this vastly expanded capability of the program is utilized, the performance of the network over any designated period of time is analyzed.

If operated in simulation mode, additional data are needed. These data give: (1) dimensionless demand functions for various uses and designates which nodes are associated with each such demand function, (2) elevation-storage capacity curves for reservoirs and tanks and (3) operating rules or schedule for pumping stations, and many other commands related to controlling and specifying what changes with time or as a function of conditions within the system.

In simulation mode considerable flexibility of output is allowed. (1) Separate tables such as those for a single steady stage analysis may be printed for each time step, or a specified multiple of time steps. (2) Special tables giving pressures are designated nodes as a function of time can be selected. (4) A direct (random) access file containing the simulation solution can be created. If number (4) is selected, any item of information can be printed subsequently as the user, in time-shared mode or batch mode, requests; that is by means of simple instructions any desired information at any time step can be obtained at near zero computer costs.

18. USU-NETWK assumes water at standard temperature is the fluid. Other fluids can be accommodated by altering the fluid properties such as the specific weight and kinematic viscosity to that of any desired incompressible fluid.

19. If in using the two table form of output, remarks are to be printed to the right of lines for certain pipes or node in these tables, these remarks can be part of the input.

20. Three separate forms of input are allowed: 1) That described in the body of this users manual, which will be referred to as the standard form of input, 2) that described in Appendix A, and 3) that described in Appendix B. There are variations allowed within these different forms as well. The standard form of input enters data of a given type through a command name. The other forms of input have a prescribed order and do not use names. The first two forms are pipe oriented input data, i.e., the layout or topology of the network is defined by the pipe and its end nodes. The Appendix B input is node oriented input that defines the layout or topology of the network from the pipe numbers that join at each node.

In addition to the different forms of input, options permit changes in the order of items lists on the more important input records (or cards). Consequently, the program can read data directly, with few if any changes, that have been prepared for another program.

21. By specifying both pressure (or HGL) at nodes in addition to the demand, network components can be sized. For each such dual specification a component such as a pump pipe diameter, or pressure reduction valve is sized. In the limit it is possible to size as many components as there are junctions in the network. In this limit an efficient algorithm can be selected that does not require an iterative solution. With wise use of this design capability all pipes can be sized by first reducing the network to a branched system that contains as many junctions as pipes, and after these pipe sizes are determined add the remaining pipes and have the program size them.

22. A versatile engineering economic analysis of the network can be performed. One time capital costs and/or unit costs can be given for various system components. The results of these cost analyses give the present worth, and equivalent annual reoccurring costs for each such component identified.
USU-NETWK'S PREPROCESSOR

In this section of the manual the use of an auxiliary program PIPEINPT is described that assists in preparing input data files for use by USU-NETWK. Because this program may be used prior to using USU-NETWK for obtaining a solution of a network problem it is referred to as USU-NETWK's preprocessor PIPEINPT. The use of this preprocessor program, is not required. Some people will prefer to get acquainted with the input data allowed by USU-NETWK from reading this manual, and then using an editor, or even word processor that they are acquainted with in preparing the input data. Others will find the use of PIPEINPT helpful in preparing input data files. However, PIPEINPT was never designed to do all the editing, etc. that is generally done in modifying a given data file for the various solutions that are typically obtained in a complete study of a water system. There is an full screen editor with a help file that describes how to use it that can be acquired as part of PIPEINPT, but this editor is generally not included in the PIPEINPT program since most individual will not want to learn another editor, but use the editor (or word processor) they are familiar with to edit USU-NETWK files. PIPEINPT was designed specifically for users that have the PC version of USU-NETWK. It only runs on IBM compatible PC's, but can be used to prepare input data files for USU-NETWK that is running on a larger system. If this later mode of use occurs a convenient method of operation is to let PIPEINPT help prepare input data files on the PC and then transfer these files to the disk of the larger computer system with a software communications package such as KERMIT, or CROSSTALK.

The description of the use of PIPEINPT given below assumes that you have some familiarity with the input data requirements of USU-NETWK. As a minimum familiarity with this input you should read the PREVIEW section in this manual and obtain solutions to the small networks given there by executing USU-NETWK in solving them before reading this section, and using PIPEINPT.

The preprocessor program PIPEINPT is designed to help prepare input data for the network analysis and design program USU-NETWK. This preprocessor program prompts the user so he is constantly aware of what is needed next. The output file written by PIPEINPT consists of input data that can be used for the steady-state version of USU-NETWK that runs on PC's directly, but if desired this file can also be used for the larger version of the program that also will performance a solution through time. If the later solution through time is desired the output file from PIPEINPT can be used for the majority of the input, and the additional data that describes the time varying characteristics of the problem can be added thereafter using an editor, or word processor.

The program PIPEINPT prompts its user relative to what it wants next, and therefore the best method for getting acquainted with its use is to run the program as you read this three page description.

Since program PIPEINPT writes a formatted file that can be read more rapidly by USU-NETWK than the format free file that it will also read, a small amount of computer time saving will also result from use of PIPEINPT. The option NETPLT=13 should be set to take advantage of this fixed format input. If a graphical display is desired then NETPLT must be changed to 14 (and also LENGON=0, unless the X- and Y- coordinates are to be added to the file that USU-NETWK will write for input to the plot program PLTNET) in the $SPECIF list when this file is to be used on an IBM compatible PC.

The basis input to USU-NETWK might be divided into the following three categories: 1. The title of the network for which the data applies. (This title is terminated by an /*.) 2. The list of options that are enclosed between $SPECIF and $SEND, and 3. The detailed data that described the layout of the network, its pipe sizes, the demands for which the solution applies, and the operation of pumps and reservoir to meet these demands. The Preprocessor program PIPEINPT utilizes these three categories of input in the order in which they are listed above.

The first question that PIPEINPT asks its user is to be supplies a file name that will contain its output file, or the input data for USU-NETWK. This name must conform to the MS-DOS file naming requirements. It can consist
of a drive designation, a path and a file name with an extension following the file name. For example C:/YY/FIL1.DAT. Generally, however, the default drive and directory will be used and, therefore, the file name with an extension will be all that is given, i.e. FIL1.DAT. Next the user is asked for a title that he wishes to give the network. The title typed in by the user will be duplicated in the file written by PIPEINPT. This title can consist of up to 10 lines, and must be terminated on a new line with a /*. Immediately upon pressing the return key after typing the /* the user is prompted with a screen full of the available options that exist in the PC version of USU-NETWK, and what their default values are set to. To change any of these options type in the number that is shown associated with an option. For those options that permit only two settings (i.e. a 0 or a 1) the screen will show this option immediately changed. For options for which 3 or more possibilities are allowed, a prompt appears at the bottom of the screen that shows the valid values, or indicates what the option means. For example if you type in 1, which is the number associated with the option NFLOW, you are shown that valid values are from 0 through 8. If you type in the number associated with VISC you are prompted by the lower line on the screen telling you this is the kinematic viscosity. For these latter type options you must next type in the appropriate value. Upon pressing the return key this value will appear after the = following the option name. You must also press the return key after entering the number associated with the option. To let program PIPEINPT know that you are through changing options you type a number such as 43, which is larger than the last number associated with the last option, and press the return key. Only those options that are changed will appear in the options list of the output file. Should you want an option to appear in this output file with the default value, you can either select it twice if it is a two valued option or type in the same value as its default is if it is a multivalued option.

Next you are prompted for the detailed data that describes the physical characteristics of the network. This data will be entered with a command name such as PIPES, NODES, etc. To remind you what these commands consist of you will see them listed on the bottom two lines, with a prompt line above this that asks you to select the command name for the type of input you wish to give next. This command name can be selected by either typing the number that is associated with the command name, or by typing the command name. If you select to type the command name make sure that the "caps lock" key is on. Upon selecting the command it will appear at the top of the screen waiting for you to enter the data called for under this command. The prompt at the bottom of the screen will list the items that must be entered under this command in the order in which they are to be entered.

In entering the data under any command it is important to note that the following three keys have special meanings: 1. the space bar (causes the value just entered to be placed into the file and the input line at the bottom of the screen now applies for the next item). 2. the return key (causes this line of input to be completed. If not all entries called for have been entered the line will be completed with values previously entered.), and 3. the divide key or / (causes the input under this command to be completed, and the bottom prompt lines to show the available commands again). Thus if you should press the return key at the beginning of the line this line will be a duplicate of the previous line, except pipe and node numbers which will be incremented by 1. Therefore, you want to be careful to not press the space bar, the return key or the / key except when appropriate. However, by pressing the space bar after a value has been completely entered, or the return key if the remaining items are the same as on the previous line you can enter the data rapidly with a minimum of key strokes. The return key should be pressed when you complete a line rather than the space bar. If you type a number in incorrectly it can be corrected using the backspace key prior to pressing the space bar or return key. Simply press the backspace key as many times as needed. Each time you press backspace the cursor will move backward over one more character. Now type in the correct value. The value that appears in the input line at the bottom of the screen will be entered in the proper column in the table shown above on the screen.

Up to twenty lines of input will appear on the screen under any given command. After this the top line will disappear from the screen as the last line is entered. A very limited editing capability of values in these 20 lines that appear on the screen can be accomplished as described below.

The last command that should be selected
is END (or its item 12), or STOP (or its item 13). Upon selecting the END command PIPEINPT assumes that you are through with the input for the network and supplies the RUN command at the end. It writes the rest of the data on its output file, i.e. the file name you supplied previously, and terminates. If the STOP is selected instead, the RUN is not added to the end of the file in anticipation that you wish to supply additional data later. If at a later time you give this same file name, you will be informed that the file already exists, and asks if you wish to add to it. Upon responding with a Y or YES, any data provided will be appended at the end of the existing file. If you terminate with an END during this session, the RUN command will be the last line written to the file.

The data file written by PIPEINPT will conform with the requirements described in the user’s manual except in the case of source pumps, or booster pumps if you select this option from PIPEINPT. The data requested from you by PIPEINPT for pumps is identical to that described in the user’s manual, namely a pipe (or node if NODESP=1) and three pair of flowrates and corresponding heads produced by the pump followed by the water surface elevation of the supply for a source pump. However, the output file will contain the coefficients of a second degree polynomial that passes through the three points that you give, the water surface elevation and the normal capacity of the pump, i.e. the middle, or second flowrate value you give. If you wish to use the option of giving the power supplied to the fluid by the pump and its normal capacity just enter a zero (0) for the next item called for. The flowrates given in connection with pumps must be in the units designated by the option NPGPM. USU-NETWK expects to read data for pumps as written by PIPEINPT when the option NETPLT=13 or 14. If NETPLT = 14, then when this data is submitted to USU-NETWK for a solution it will write a file that contains information needed by the post processor program PLTNET to graphically display the network on the screen. However, in order to use this graphical capability it is necessary that x- and y- coordinates by given in describing the physical layout of the network.

Limited Editing Capabilities

In addition to allowing you to correct a value with the backspace key before it has been entered PIPEINPT permits you to correct values in the detailed data that follows the command such as PIPES, NODES etc. as long as these values remain on the screen, i.e. are within the 20 lines under any command that are not scrolled off the top of the screen. Other changes, or editing of the file must be done with an editor or word processor.

To utilize this limited editing capability simply press one of the arrow keys on the extended pad portion of the keyboard. Pressing of these arrow key moves the cursor to an entry on the screen in the direction of that arrow key. The item on which the cursor is directed is shown blinking. You simply move to the entry that needs to be corrected by pressing the appropriate arrow keys. When you are at this value press the space bar to indicate that this is the number you wish to correct. At the bottom of the screen you will see a prompt to enter the correct value. Type in the corrected value and press the return key. If there are additional values to correct press one of the arrow keys again and repeat the above process. If you wish to delete an entire line this can be done by pressing the delete key in place of giving the corrected value. The entire line that the blinking value is on will disappear. When you are through making such corrections, press the E key to return to the position you were at in entering data prior to pressing an arrow key.

ILLUSTRATIVE EXAMPLE IN USING PIPEINPT

As a simple example to get acquainted with PIPEINPT you should execute this program to prepare the input data file to solve the network problem illustrated below. Read the following description as you execute PIPEINPT. The first prompt you receive from PIPEINPT is the name of the data file that the problem description should be written to. You might give TESTPI.DAT for this file name. You will next be ask to give a title for the network problem and to terminate giving lines of title with */. Type in the following:

Illustration to use PIPEINPT /

Next you will see the screen filled with the default options. Since all of these apply to this problem type 43. The bottom of the screen now displays the command names. Type 1 to select PIPES and enter: 1 0 1 3000 15 .005 Enter (press the space
bar between values, and the Enter key at the end of the line). For the second line of PIPES data type:

2 1 2 4000 12 Enter

Note that the coefficient for the 1st pipe has been added upon pressing the enter key. For pipe 3 just press the Enter key, and note that the data from pipe 2 is taken to fill in this line. For pipe 4 type

4 3 4 4000 8 Enter

and press only the Enter key for pipe 5.

Next press the divide (or forward slash) key / to get the command names to appear at the bottom of the screen again. Now press 2 to select the NODES command and type in the following:

1 1 500 Enter
Enter Enter Enter Enter

Note the 4 press of the Enter key for nodes 2, 3, 4 and 5 fill in the proper dat for these nodes. Now / again and type 4 to select BOOST. You will see the prompt:

Should output contain: (1) 3 pts for P. Char; or (2) Coef. for polyn.

Give 1 or 2

Type 2 and then enter the booster pumps characteristics by typing:

2 2 30 3.5 24 5 15 Enter

Press the / key again, and this time type 5 to select RESER and type

1 700 Enter

and press the / and terminate by selecting 12 for END (selection of STOP assumes you want to add to this file at a later time.) The data file that PIPEINPT has written for you should look like the following:

Illustration in using PIPEINPT
/* SPECIF NETPLT=13, SEND PIPES
1 0 1 3000.0 15.00 0.00500
2 1 2 4000.0 12.00 0.00500
3 2 3 4000.0 12.00 0.00500
4 3 4 4000.0 8.00 0.00500
5 4 5 4000.0 8.00 0.00500
6 2 4 6000.0 10.00 0.00500

NODES
1 1.000 500.00
2 1.000 500.00
3 1.000 500.00
4 1.000 500.00
5 1.000 500.00

BOOST
-2 -0.66667 -0.33333 33.3333 0.0000
3.5000 RESER 1 700.00

RUN

Now execute USU-NETWK and type in the name TESTP1DAT when ask for the input data file. You might type TTY for the output file to have the results from the solution displayed directly to the screen.
Table 1. Order and nature of input data.

<table>
<thead>
<tr>
<th>Category of Input</th>
<th>Description of Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Specifications</td>
<td>(a) Title of network up to 10 records long and terminated with a */ (or */ in columns 1 and 2 of final record. (For IBM computers the */ must be used). (b) The list of options beginning with $SPECIF (or &amp;SPECIF) in column 2 and ending with $END (or &amp;END). The list between these two consist of the parameter name followed by a comma. Only those parameters whose values are to be set different from the default need to be included.</td>
</tr>
<tr>
<td>Detailed Data of Network</td>
<td>This data consists of a command name such as PIPES, NODED, PUMPS, etc. followed on the nest record by data of the type identified by the command name. The last such command must be END or RUN. If special options are given some additional data may be required after the END or RUN in addition to the data given in categories 3, 4, and/or 5.</td>
</tr>
<tr>
<td>Data for Alternate Steady-State Analyses</td>
<td>Giving Changes to Basis Network Each group of change data begins with the word CHANGE in column 1 and terminates with the word END in column 1. Commands such as DIAMETER, LENGTH, DEMAND, etc. enter changes to the existing network. These commands are followed by appropriate change data, and occur between a change and end command. Any number of groups change of data can follow each other. Such changes are accumulative.</td>
</tr>
<tr>
<td>Data for Time Dependent Solution of Network</td>
<td>This data is read only if the option ISIM=1 is set in the $SPECIF list and includes: (a) A first record which is an options list entered with $TDATA (or $TDATA) (b) The data giving the time dependent characteristics of the network which is entered with command name suchn as DEMAND FUNCTION, PUMP RULES, etc. followed by the appropriate data. The last command is END SIML or END. (Generally categories 3 or 4 are not used for the same problem. If both are given then category #3 data follows category #4 data.)</td>
</tr>
<tr>
<td>Data for Cost Analyses</td>
<td>This data is read only if ICOST is set greater than 0 in the $SPECIF list of options. It consists of costs associated with the various components of the network. It is entered with commands such as INTEREST=i, LIFE=n, or ELECTRICITY, PIPES, etc. followed by CAPI, or UNIT and the dollar amounts, and the designation of components for which the costs apply. The cost data is terminated by the command END. These cost data precede each set of change data of category #3 and apply for the preceding analysis. Cost data follow category #4 data, if called for. If ISIML=1 and ICOST&gt;0 a cost analysis will follow each time step solution; if ICOST=2 every other time step; etc. For each such new time step cost analysis called for at least an END command must be given.</td>
</tr>
</tbody>
</table>

*This second category of input is different if the forms of input described in Appendix A or B are selected. The selection of these alternate forms of input is controlled by the option INPUTA in the $SPECIF list.
GENERAL SPECIFICATIONS
OPTIONS

INPUT LINE # 1 - TITLE

The first line, or lines, of input data consist of a title for the network. This title will be written on the output file at its beginning for identification purposes. The title can be up to 10 lines long. Each line can be up to 80 characters long. The line immediately after the title, which terminates reading of the title must contain a /* (or for larger IBM computers a */ in columns 1 and 2. In the examples in this manual the */ will be used. It is desirable practice to put in the title enough to distinguish any given "run" from other "run" that may be made so that it is not necessary to examine detail in the solution to determine what a particular output is for. For example the title may have extra lines added to the name for the water distribution system such as:

FIRE DEMANDS OF 1200 gpm AND 1000 gpm AT NODES 30 AND 54, RESPECTIVELY.

The title can contain lower and uppercase characters and numbers. Options, and command names, which are described in the following pages, must be upper case characters.

INPUT LINE #2 - PROBLEM SPECIFICATIONS OR OPTIONS

This line, or lines, starts with $SPECIF, contains any or all of the list of names, e.g. parameters, followed by an equal sign and a value that are listed in the following pages. Should The version of the program you are using utilize FORTRAN's NAMELIST capabilities, then the $SPECIF must beginning in column 2. If the version of the program you are using does its own input, then the $SPECIF may begin in column 1 or any other column. The list of parameters can be separated by either a blank or comma (unless the FORTRAN NAMELIST requires commas, as some older FORTRANS do). The terminating $END may be on the same line, or the next line, but must be given to terminate the list of options. If $END is not given the entire data file will be examined looking for it, and when not found, execution will be terminated. The $ sign can generally be replaced by & sign if desired, at least this is possible if the program does its own options input.

The length of any line listing options is limited to 80 characters. If the list is too long for one line continue onto other lines, but terminate a line with a parameter and its value. In other words the parameter, the equal sign (=) and its value should not be split between two lines. The parameters (which are listed and described in the following pages) may appear in any order. Only those parameters for which values different than those that are used as "default" values need to be included in the list, but may be included if desired even if they are given the same values that USU.NETWK uses as default if not given. At the time the program is compiled it is easy to change the default values. Generally, however, the default values will be those described in this manual.

Any of the following variations are valid specification, or option's lists:

$SPECIF NPRINT=2,NFLOW=1,NPGPM=1, VISC=.9E- 5 $END

$SPECIFNPRINT=2,NFLOW=1,VISC=.000009 NPGPM=1 $END

$SPECIF NODESP=1 $END

(Note that upper case, or capital letters, are required).
Alphabetical list of parameter names that may occur in $SPECIF$ list.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORFRO</td>
<td>The default roughness coefficient.</td>
</tr>
<tr>
<td>CORIN</td>
<td>Allows the order of data after the NODES command to be altered.</td>
</tr>
<tr>
<td>CORIP</td>
<td>Allows the order of data after the command PIPE-nodes, or the input described in Appendix A to be altered.</td>
</tr>
</tbody>
</table>

#These options are not included in the allowable $SPECIF$ list of the version of the program that only does steady-state analysis. The Calcomp graphics plotting capability may be obtained with the steady-state version of the program in which even NETPLT, HLETTE and PLOT will be included.

These options are not included in the allowable $SPECIF$ list of the PC-DOS version of USU-NETWK that does have the time-dependent solution capabilities.

CORNOD #

The list of parameters, or options, that may be included in this $SPECIF$ list are given in the following pages in alphabetical order. Their effects on the program's behavior are described in the paragraphs under these individual names. In few cases either of two names may be used with identically the same effect. These alternative parameter names are given enclosed in parenthesis. The above table lists summarized the valid parameter names.

You will find this table useful to scan for the spelling of the parameter, and then you can quickly turn to the indicated page to read about the effect of the parameter, and values it may take on.

Those names followed by # are valid only for the version of the program with time dependent solution capabilities, e.g. the simulation version of USU-NETWK. A PC version under DOS of the simulation version implements most of the options available in the full program designed for larger computers, but those that have been deleted from this PC version are denoted by a ^.

**COEFRO**

The default roughness coefficient. This roughness coefficient may be the equivalent sand roughness, e, for use in the Darcy-Weisbach equation, the Hazen-William coefficient, C, or Manning's n. It will be used as the wall roughness for pipes until a values is given under the PIPES command, or it will be used whenever a roughness values is not given on any line after the PIPE- command.

(default, $e = 0.0102$ inches for use in the Darcy-Weisbach Eq.)

CORNOD #

Allows the standard order of data after the NODES command to be altered. If CORNOD=1, then the altered order is specified. The four items whose order can be altered are: 1. the demand, 2. the node elevation, 3. the pressure or HGL if DESIGN is larger than 0, 4. the x coordinate of the node, and 5. the y coordinate of the node. Note the node number must always be the first item. Also if NETCHK=1 in the $SPECIF$ list then it is also necessary that the node oriented input that lists the pipes joining at the each node must be given last. If CORNOD=1 appears in the $SPECIF$ list, then immediately after this optional list, a list of integer numbers, giving the new order of the items, is given. For example if the elevation and demand are interchanged after the NODES command this extra line of input would consist of: 2,1,3,4 (or just 2,1/ if coordinates are not given). The 2 at the first of this list indicates that the standard second item after the node number, i.e. the elevation, is first, and the standard first item after the node number is in second position, etc.

Should CORPIN=1 also be in the $SPECIF$ list, then these integer values will be the second line after the $END$. 

CORNIP #

Allows the order of items after the command PIPE-nodes, or the input described in Appendix A to be altered. Of the four commands (CORIP, CORNOD, CORPUM & CORPIN) that allow changing the order of items after commands, this one is the most powerful. Through its proper use USU-NETWK can be made to read data that has
been entered in almost any way. Depending upon whether CORPIN = 1, 2 or 3, the following is allowed:

**IF CORPIN = 1.** The order of data after PIPE- is altered. If so, the first record after the command PIPE- must contain the new positions for the following 10 items: (1) the pipe no., (2) the diameter, (3) the length, (4) the roughness coefficient, (5) the upstream node, (6) the demand at the upstream node, (7) the elevation at the upstream node, (8) the downstream node, (9) the demand at the downstream node, and (10) the elevation at the downstream node. This data must be entered so that it can be read under the format (F5.0,3FI0.0,F5.0,FI0.0,F5.0,2FI0.0).

**IF CORPIN = 2.** The effect is the same if CORPIN=2, as that described above for CORPIN=1, except that in addition to the order of the data being changed the format of the input data must be specified using F, E or G fields. This format specification is given before the integers that define the new order. For example, if CORPIN=2, the first two lines after the PIPE- command (or immediately before the pipe data if Appendix A input is used) could be,

```
(3F5.0,3FI0.0,4F5.0)
1,5,4,6,2,7,8,3,9,10
```

These two lines will cause input to be in the order as that after the PIPE command except that it must be in the specified columns. (Formatted input under the PIPES (and other commands) also occurs if option NETPLT=13.) To understand how these last 10 values determine the new positions, list the 10 input items in their usual order as described under the command PIPE-. The 10 given values indicate the new positions for each of these 10 items, e.g. pipe no. is in position 1, diameter is in position 5, length is in position 4, etc. Items 6 (demand at the upstream node), 7 (elevation at the upstream node), 9 (demand at the downstream node) and 10 (elevation at the downstream node) are given positions 7,8,9 and 10, respectively, beyond the portion of the line that will actually contain any data. Thus these last 4 items will be read in as zeros. Since the rules under PIPE- allow these items to be omitted the zeros have no effect. The values for these items can be filled in with the NODES command.

As a simple example consider the 4 pipe network supplied by one reservoir as defined by the data below. In this data the option CORPIN=2 is used. The data after the command PIPE- is that under the PIPES command because of the new order specified by the second line of integers after the PIPE- command, except that the pipe and node numbers, as the first three items, must be right justified in fields of 5 and the remaining 3 values in fields of 10 as specified by the format. The difference is that whenever a roughness coefficient is not given by leaving columns 36-45 blank the value given by COEFRO=.005 will be used rather than that given for a previous pipe in the list.

**ILLUSTRATION OF USE OF CORPIN=2**

```
/*
$SPECIF CORPIN=2,COEFRO=.005 SEND
NODES
1 .8 100
2 1.2/
3 1/
PIPE-(3F5.0,3FI0.0,4F5.0)
1,5,4,6,2,7,8,3,9,10
1 0 1 1200. 8. .015
2 1 2 1400. 6.
3 1 2 1000. 6. .025
4 3 2 1500. 6.
RESER
1 200
RUN
```

As another example assume roughness coefficients are given after the downstream node and before the pipe length. Then if the $SPECIF list contains CORPIN=2, the first two lines after the command PIPE- would consist of:

```
(3F5.0,3FI0.0,4F5.0)
1,5,4,6,2,7,8,3,9,10
```

With CORPIN=2 the format of the input data is specified and, if desired, the usual order can be retained by giving 1,2,3,4,5,6,7,8,9,10 as the new order. Input efficiency can thus be gained by allowing fixed format input over the free format input. (See NETPLT=13, IOUTI & IOUTO, also.)

**IF CORPIN = 3.** The effect is to read the input according to the standard format (F5.0,3FI0.0,F5.0,2FI0.0) but not rearrange the order of the input. Thus if CORPIN=3 no additional lines are required after the PIPE- command. However, the data must conform to the columns specified in the above format. This restriction applies whether the data is entered via the command PIPE-, or under the input described in Appendix A resulting from including INPUTA=1 in the $SPECIF list.

Since many items of input can be omitted
using the PIPE- command and it can be used in conjunction with other commands, considerable flexibility is available. If data files exist for other network analyses programs these files may be readable by USU-NETWK almost directly. By having USU-NETWK write data file, or have the post-processor, PLTNET write such a file, this data can be converted to the standard input used by USU-NETWK for subsequent analyses after it has once been read, if desired.

CORPIP #

This option, when set to 1 (i.e. CORPIP=1), allows a new order of items after the PIPES command to be specified. This new order is specified by giving the new positions for the length, diameter and roughness coefficient immediately after the $SPECIF list. Note that the first 3 items, the pipe number, the upstream node and the downstream node cannot have their order changed. The new order is specified by giving an integer value that represents the new position of items in the list whose order can be changed. For example if data after the PIPES command consists of: 1. the pipe no., 2. the upstream node, 3. the downstream node, 4. the diameter, 5. the length, and 6. the roughness coefficient (i.e. the length and diameter are interchanged from the standard order), then this line would consist of: 2,1,3.

To understand how to designate these new position numbers, list the variables that can be changed in their standard order, i.e. 1-diameter, 2-length, and 3-coefficient. The first integer above 2. The second integer 1, since it gives position for the length, indicates the length is first, and the final integer 3 indicates that the roughness coefficient retains is 3rd position of the variables whose positions can be changed.

CORPUM #

This option, when set to 1 (i.e. CORPUM=1), allows a new order of items after the PUMPS, BPUMP and BOOST commands. The pump's designation must be first, but the order of the 3 pairs of values that define the pump curve, and the pump's sump, or wet well water surface, elevation (or inlet pressure) can be altered, or if the pump's performance is defined by its power and normal capacity, the order of these and the water surface elevation can be altered. If more than 3 points on the pump characteristic curve are provided (PCHAR3=0 in the $SPECIF list), they must occur after the reordered data. If CORPUM=1, then the first line after the $SPECIF list (or after the lines required if CORPIP=1, and CORNOD=1) must contain 7 integers that either designate (a) the new positions for (1) Q1, (2) H1, (3) Q2, (4) H2, (5) Q3, (6) H3, and (7) elevation of water surface, or (b) the new positions for (1) the power the pump supplies to the fluid, (2) its normal capacity, and (3) the elevation of the water surface. For case (b) only the first three values have any meaning but 7 integer values are required and can be given as 4,5,6 & 7. Caution must be exercised in mixing methods of specifying pump characteristics between using 3 points on the curve and power plus normal capacity, since altering the order alters the order for all pump input data regardless of the method used to designate pump performance. (See explanation under CORPIP for understanding how the new order is defined.)

For example, if the water surface elevation is to occur immediately after the pump designation (pipe no., or node no. depending upon NODESP) then the following integers would be given: 2,3,4,5,6,7,1.

DESIGN

This option is how you tell USU-NETWK that you want special design solutions rather than an analysis solution. Values that DESIGN may have are: 0 (default - analysis solution), 1 (determine diameters for NJ pipes from pressures at all nodes), 2 (special "least cost" design described in Appendix D), 3 (determine wall roughness coefficients for NJ pipes), and 4 (and 5) (specified velocities in branched pipes)

If DESIGN = 1, When DESIGN=1, then USU-NETWK is told that it is to determine the diameters of as many pipes as there are junctions (nodes), NJ, in the network, by means of a computational efficient method. (NJ = number of junctions, or nodes, of the network excluding sources if these are numbered as nodes with the option NODESP=1.) An alternative, but much poorer choice than setting DESIGN=1, is to introduce as many differential head devices as there are network nodes and specify the pressures, or HGL elevations at all nodes by using the DHEAD command. Doing the latter adds NJ additional equations to the system of corrective flowrate equations being solved. With DESIGN=1 the solution is accomplished with NJ
linear equations.

When DESIGN=1 the pressure, the pressure head, or elevation of the HGL, respectively according to whether IHGL=0, 1 or 2, must be given at all nodes of the network. When using the NODES command these pressure values are given after the nodal elevation, as a 4th item in the list of input. When using the PIPE- command these nodal pressure values are given as an additional data item containing a decimal point after either the upstream nodal elevation, or the downstream nodal elevation. A demand, and elevation must be given before the pressure when using the PIPE- command.

Since NJ pipe diameters will be determined as part of the solution, it is necessary to designate NJ pipes as having unknown diameters. The remaining, NL=No. of pipes - NJ, diameters must be known. The pipes whose diameters are unknown must be given a diameter equal to zero in the data following the PIPES command or the PIPE- command. An alternative for branched networks with as many nodes as pipes is to use the special input described under IHOL. The following additional options: IHGL, NOMSOL and NOMDIA provide control over the special design solution activated by DESIGN=1, or DESIGN=3.

In preparing data for this special design solution the following two restrictions must not be violated. 1. At least one pipe at each junction must have its diameter unknown, i.e. not all pipes at a junction can have their diameters specified, and 2. The HGL's produced from the specified nodal pressures must be consistent with the direction that the flow in the pipe must have to satisfy the demand here. A simple case of violating this second restriction would be a dead end pipe with a positive demand at its end, but an HGL elevation at its downstream end above the HGL elevation at its upstream end. A slightly more complex situation would exist at a node of degree 2 with one pipe given a diameter and the other not, and for which the flow rate in the pipe of known diameter (which is determined by its end HGL elevations, length, diameter and roughness) is less than the demand at the node, yet the HGL elevations for the other pipe requires that its flow be away from the junction. Nodes of degree 3, or higher, may violate the second restriction, also if at least one pipe does not have its flow in the direction required by continuity. It is generally difficult to not violate this second restriction, unless considerable understanding of the network's performance exists when the data is being prepared. A means of proceeding is to make adjustments after the program has indicated that your specifications do not satisfy this second restriction, and through this process acquire a better understanding of the network's performance. A better alternative is to start with a network with as many junctions as there a pipe, because you can always then determine the direction that a flow must be in and specify an appropriate slope for the HGL. Thereafter add minimum pipe diameters to those pipes that form the loops of the system. The option IHGL allow special input to make this easier for you.

**IF DESIGN = 2.** When DESIGN=2 in the $SPECIF list, then a special interactive portion of USU-NETWK (the full program, but not on PC's under DOS) is activated that assists in determining the "least cost" design. This design may suggest alternate pipe layout configurations. Its use is described in Appendix D of this manual.

**IF DESIGN = 3.** When DESIGN=3 in the $SPECIF list, then a special solution is requested to determine as many pipe wall roughness coefficients as there are junctions in the network. The same rules apply when DESIGN=3 as when DESIGN=1 with the exception that NJ coefficients must be given zero values. When DESIGN=3 all pipe diameters must be given and the roughness coefficients of the pipes are solved so that the given pressures, heads, or HGL elevations (depending on the option IHGL) are satisfied. The roughness coefficients that are solved for can be: (1) the equivalent sand roughness of the pipe for use in the Darcy-Weisbach equation, (2) the Hazen-Williams C, or (3) the Manning's n. Since coefficients are given as zero, it is necessary to provide NEQUAT or NHAZEN in the $SPECIF list with the appropriate value if either C or n are to be solved for.

**IF DESIGN = 4.** When DESIGN=4 in the $SPECIF list, then you must give the velocities that are to occur in pipes that are not included in any loops of the network rather than their diameters, and the diameters for those pipes that are in loops of the network. These velocities are given whether the diameters are called for under the PIPES or PIPE- command. DESIGN=4 is a special design in which pipes that are branches, without loop, may be sized to give desired velocities.

If the network is a branched system all pipes will have their diameters determined to give
the specified velocities. However, for looped
networks you must understand fully the layout of
the network so that as the data is prepared you
will know whether you are giving a pipe diameter,
or a velocity in a pipe. USU-NETWK will assist
you by providing you a list of pipes whose
diameters must be given and whose velocities must
be given. You request this assistance by setting
DESIGN=5, and prepare the input data without
being concerned whether a pipe is in a loop or
not. This input could give a value for the pipe
diameter in the first line after the PIPES
command, and terminate all other lines under this
command with a /. After this run with
DESIGN=5, from which you are given two lists of
pipes by USU-NETWK; (1) those pipes that are
part of loops and (2) those pipes that are not part
of any loop, you edit this data file, giving either
pipe diameters, or velocities as required. Should
you wish to specify other pipe velocities (or
diameters) than those that
USU-NETWK
defaults
to by its built-in loop finding process, you can
utilize the option LOOPRD, and provide the loop
data. However, when using LOOPRD valid and
complete loop data must always be given. Should
you wish to specify the diameters of some of the
pipes that do not have loops passing through
them, this can be done by giving a diameter
proceeded by a minus sign.

DESIGN=4 can be used in connection
with the options NOMSOL=1 (10 or 21) and
NOMDIA=1. If NOMSOL is not given a value
(i.e. the default of NOMSOL=0 is left), then the
solution will consist of diameters computed exactly
to give the specified velocities. This solution will
be given in the usual output tables, rather than
the special output tables that are given when
DESIGN=1 or =3, in other words the table will
not contain the computed diameter plus a
standard diameter. With NOMSOL=1, however,
this solution will be followed by one in which the
closest standard diameter, or nominal diameter,
will be used. If NOMSOL =10 thru 20 then the
next larger standard diameters will be used in this
follow on solution, and if NOMSOL=21 thru 30,
then the next smaller standard diameters will be
used. There is no provision that allows this pipe
to be divided into two segments one diameter
smaller than and one larger than the computed
diameter, as with a regular analysis solution. Use
of NOMDIA allows you to give a list of standard
diameters if the default values are not what you
want. In using SI units, or diameter in other than
inch sizes, a list of standard diameters must be
given because the default values are in inches.

If DESIGN=4 is given, and the network
is not given any sources of supply (i.e. no RESER
or PUMPS commands occur in the input data),
then it is necessary to give a node number, and its
HGL elevation after the RUN (or END)
command since this is then a special network with
all external flows specified.

EQUUDIA

See NEQUDI

ERROR (ER8)

The error parameter that determines the
accuracy of the solution. The iterative solution
process will terminate when the absolute sum of
changes in Q's between consecutive iterations is
less than ERROR. The error parameter can also
be set by assigning NDIGIT, an integer, equal to
n in the equation ERROR = 10^-n. The default
value is 0.01, for ES units and 0.001 for SI units.
For example the options list could contain: ERROR=.0001.

FLOWFC

See PEAKF.

GAMMA

The specific weight of the fluid in lb/ft^3 or
N/m^3 depending upon whether ES or SI units are
being used. If not given 62.4 lb/ft^3 or 9800 N/m^3
for water are assumed, respectively depending
upon whether ES or SI units are used. The
specific weight can also be entered by a command
WEIGHT. For example the $SPECIF list of
options could contain GAMMA=55 for a fluid
with a specific weight less than that of water.

HLETTE #

Determines the vertical height of the
letters that will be used on a Calcomp plot.
Default =.13 inches. A menu will be displayed if
a Calcomp plot is ask for, and one of the items
listed here allows the letter height to be changed.
Therefore HLETTE is seldom used, but can be
used so that the menu item need not be changed.
For example HLETTE=.09 could be included in
the $SPECIF list of options.
ICOST

If ICOST = 1, then a cost, or engineering economic, analysis of the network is done. If ICOST = 1, then additional cost data must be supplied after the RUN or END command, and after other data if options call for it. These additional cost data are described in a subsequent section in this manual under the title "Cost Input Data." As a minimum this additional data must consist of END that indicates that the default cost data are to be used. The only two costs that are computed if other cost data are not provided is the default costs of pipes, and the default costs of energy consumed by pumping. If ISIML = 1 also exists in the $SPECIF list, then the value given ICOST determines the time step interval for which cost analyses will be performed. If ICOST = 1, then a cost analysis will take place after each new time step solution. If ICOST = 2, then every other solution will have a cost analysis associated with it, etc. For each such requested cost analysis new cost data can be supplied. As a minimum this cost data must consist of an END. Generally cost analyses will be obtained with steady state solutions, and therefore one would only supply one set of cost data. However, if costs associated with flow from tank, or other items that are dependent upon flow rates are given then a series of costs analysis may be obtained at specified time intervals. USU-NETWK does not average, or provide a composite for all these time varying costs. The user is responsible for this composite interpretation.

IHGL

The option IHGL has meaning only if DESIGN = 0, and then it has three separate functions. The first function designates the units that are associated with the nodal pressures that must be given with DESIGN = 1 or DESIGN = 3, the second function is to inform USU-NETWK that special input follows that defines a branched network, or a branched portion of a network (for which additional loop forming pipes may be added through the regular input), and the third function is to generate the HGL elevation needed with DESIGN = 1 or 3 based upon HGL elevation values given at intersection points of a rectangular grid system superimposed over the network. In addition the command IHGL can be used to have USU-NETWK write out a data file that contains the branched portion of pipe network that can be added to include the loop forming pipes. These three functions are described separately below.

Function # 1:

If the value of IHGL is positive and less than 3, then it designates whether: (1) nodal pressures, (2) nodal heads, or (3) HGL elevation are given so that NJ pipe diameters, or roughness coefficients can be computed.

If IHGL = 0 then nodal pressures are given in psi, if NUNIT = 0, or NUNIT = 1, and if NUNIT = 2 or NUNIT = 3 then the units of pressure must be in kPa (kilo Pascals, i.e. kilo Newtons/m²). If IHGL = 1, then pressure heads, or the head above the node's elevation in feet or meters, respectively, is given depending upon whether NUNIT = 0 or 1, or NUNIT = >2 for SI units.

If IHGL = 2, then elevations of HGL, or the absolute elevation position of the piezometric surface is given, respectively, in feet or meters depending upon whether ES or SI units are used. This latter HGL elevation at any node is the sum of the elevation of the node, and the pressure head.

(default IHGL = 2 for elevations of HGL)

Function # 2:

If the value of IHGL is negative or equal to 3, then USU-NETWK is informed that special input follows. There are two different types of special input allowed, which are distinguished on the basis of whether IHGL is negative or IHGL = 3.

If IHGL = -2 or IHGL = -5, then USU-NETWK is informed that special additional data is given immediately after the $SPECIF list that defines a branched network in which all pipe diameters are sized so that a specified slope of the hydraulic grade line, HGL is satisfied. The absolute value given to IHGL determines the logical FORTRAN unit from which this additional data is read. If IHGL = -5 and the program is used from a time shared terminal, or running on a stand alone PC, you supply this special data from your keyboard in response to prompts from the computer. These prompts correspond to the input data as described below, and thus act as reminders of what you should know from reading this section of the manual. If IHGL = -2, then this special input follows immediately after the $SPECIF list of options, i.e. on the line after the $END before the first command in the same file that contains the other input data. The special data is
embedded in the regular data because when reading input from a disk file FORTRAN LOGICAL unit 2 is used. At least one regular command, and END or RUN must be given in addition to this special data which terminates with and END. Therefore, two ENDS may exist. If IHGL is another negative number, such as -7,-3 is not allowed because it is the FORTRAN output unit number, and -4 is not allowed if remarks are used at the end of lines after the PIPES or NODES commands, and -5 or -6 are not allowed because they are generally the standard terminal assignments) then this special input can be placed in a separate file that will be read as a default file for that logical unit, i.e. on a VAX operating under VMS the file name would be FOR007.DAT, or under Microsoft FORTRAN as used for the PC version of USU-NETWK the user will be prompted for the file name. Often this special data is the only data given, but additional components including, pipes and nodes can be added to define the network using the usual commands. This special input allows for large branched network to be defined with just a few lines of input. (See Example problem 18)

This special additional input data consists of the following:

First line contains the following 5 values:

(a) The HGL elevation of the first node, which must be the final trunk of the tree that forms the branched system.
(b) The discharge from (or into) this tree trunk node. If positive the flow from the other nodes move toward this node; if negative this tree trunk node supplies the network. If this given discharge does not equal the sum of the external flow from the rest of the nodes, properly accounting for sign, then a message is printed, but the discharge is corrected to equal the sum of all other external flows.
(c) The elevation of this tree trunk node.
(d) The discharge from (or into, if negative) the next group of nodes until its magnitude is changes with the command DEMAND. This value may be negative or positive, but is generally of opposite sign to that in item (a) above.
(e) The pipe roughness coefficient for pipes until its value is changed with the command COEF.

Succeeding lines consist of:

(1) The beginning node number of the tree branch, or trunk extension. 
(2) The ending node number of the tree branch, or trunk extension. These nodes must form a continuous sequence of numbers from the trunk node number 1.
(3) The slope of the HGL of this branch of the tree. If the option NPLENG=1 (length of pipes are given in 1000 ft or m), then this slope is in ft (or m) per 1000 ft (or 1000 m) of pipe length.
(4) The pipe lengths within this branch of the tree. Items (1) and (2) or items (1) through (3) can be terminated by a slash, /. If all lengths are the same, only one length needs to be given; that is this list can be terminated anywhere after item (2) with the remaining items taken from the last such value given for that item. Often the list of lengths is terminated with /. The list of lengths is limited to 99 values. These succeeding lines of input may spill over into several lines if the number of pipes between (1) and (2) is large and most of the lengths are different. A new line is given for each additional branch of the network, but any actual branch may be divided into two or more lines if one desires to specify this branch as two or more branches.

If the option LENON=0, then rather than pipe lengths this item (4) consists of a list of pairs of x & y coordinates for each succeeding node along this branch of the network. These coordinates occur in pairs, and therefore, the list can contain up to 49 pairs of values. When this list of pairs is terminated by a / (as it must be), then all remaining pipes in the branch whose coordinates are not given will be pipes of the same length and direction as those of the last pipe whose end coordinates are given. In other words the differences in the x and y coordinates of the remaining pipes will be the same as these differences for the last pipe with both end coordinates given.

For any branch of the tree in which items (c), (d) or (e) above change from the value given previously, a command ELEV, DEMAND or COEF can be given followed by the altered value. These new values for the branch's end elevation, and its nodal demands and/or pipe wall roughness coefficients must precede the data for this branch (1) through (4) above. Elevations are assumed to vary linearly from the beginning to the end of a branch.

Study the following example of a 43 node network to better understand how this special input can be used to define a branched network, or the branched portion of a network that may contain loops in other portions of it. Examples of networks in a later portion of this manual.
provide additional examples.

In addition to the above three commands (ELEV, DEMA & COEF), the command DHEAD (which stands for differential head, and can be truncated to the four characters DHEA) can be given to include a pump or any other differential head device with a constant head (i.e. a # 1 DHEAD device described under this command name). After the command DHEA two values are required: 1. the pipe number that contains the differential head, and 2. the magnitude of this differential head (positive or negative). The command DHEA must precede the branch data that contains the pipe number given in the line after DHEA.

```
TREE NETWORK
/*
$SPECIF IHGL=-2,DESIGN=1,NFLOW=1,
NPLENG=1 $END
152. 12900. 85. -300 .005
1 10 .0005 2 ./
2 22/
2 28 .0004 3. A /
COEF
.012
21 37 /
21 40 .0003 3. A /
38 42 /
38 43 .0003 1.5 /
END
RUN
```

Generally when utilizing this special input there is no reason why the node at the trunk of the tree can not be given the number 1, and the pipe connected to this node is given the number 1. However, if this is not acceptable it is possible to number the nodes and pipes either starting with a larger value, or have their values in descending order. The starting node is established by the number given the trunk, or beginning node number, or the trunk of the tree, in the first line identified as (1) above. If the ending node for the first branch is less than this value then the node numbers will be in descending order. In order to have pipe numbering start with a value different than 1 it is necessary to communicate this to USU-NETWK. This communication occurs by adding 1000 to the loss coefficient given in the first line of input, item (e) above (and possibly preceding with value with a minus sign). When 1000 is added to the loss coefficient, then USU-NETWK will read an extra line, with one integer value on it, that must contain the beginning pipe number. If the value given for the beginning pipe number is positive, then numbering of the pipes will begin with the given value, and be incremented positively by one for each succeeding pipe in the branched network. If you desire that the pipe numbers should be in descending order from the given value, then place a minus sign in front this pipe number. Pipe numbers may be in ascending order and node numbers in descending order, or vice versa. (See the example below.)

```
TREE NETWORK
/*
$SPECIF IHGL=-2,DESIGN=1,NFLOW=1,
NPLENG=1 $END
152. 12900. 85. -300 .005
1 10 .0005 2./
2 22/ 
2 21 .0004 3./
21 28 .0004 4./
COEF
.012
21 37 /
21 43 .0003 3. /A 
21 40 .0003 3. A /
38 42 /
38 43 .0003 1.5 /
END
RUN
```

The following fewer lines of input could be used to define identically the same small branched network:
roughness coefficients under the COEF command described above. The case of two different roughnesses is accommodated, however. If the roughness coefficient given in the first line of this special input is preceded by a negative sign, i.e. item (e) above is given a negative value, then an additional line immediately following this line is read that contains the following two values: (1a) the second roughness coefficient, and (1b) the diameter for which this second roughness coefficient should apply. The second roughness coefficient will apply for all larger diameter pipes as well.

If this value for the roughness coefficient is less than minus 1000, then two extra lines are required, the first consists of the beginning pipe number and the second consists of two values, a second roughness coefficient and the diameter that separates pipes with the two roughness coefficients. Pipes larger than the given diameter will have the second coefficient applied to them, and smaller pipes will have the coefficient given in the first line of special input apply to them.

Generally different roughness coefficients will not be entered for other laterals with the COEF command if this special extra input is given. However, should other roughness values be specified under the COEF command then they will apply only to pipes with smaller diameters than given by the input (1b) above, and pipes of the diameter given by (1b) and larger will be given roughness coefficients equal to the value specified under COEF plus the difference between the value given by item (1a) above and the value given as item (e) on the first line of input above.

Since this special form of input allow for large branched networks to be generated with a few lines of input, USU-NETWK allows you to write the data that defines this network in the form required under the PIPES and NODES command for future use, and modification. The usual form of data for the network can be written for a design solution with diameters equal to 0, or contain the diameters solved for from the solution produced by this special input. To have this data written to file gives values to IHGL that have -100, -200, or -300 added to its value for special input. If IHGL is between -100 and -200, then the file will be written with 0 diameters; if IHGL is between -200 and -300, then data in the file will contain diameters computed from the design solution; and if IHGL is less than -300, then both of the above types of files can be written. For any of the three above cases USU-NETWK will prompt with:

Give no. corresponding to what you want on file:
1 - Length under PIPES data but no coordinates under NODES,
2 - Length under PIPES data & coordinates under NODES after HGL,
3 - Length under PIPES data & coordinates under NODES after Elev.
4 - No lengths under PIPE & coordinates after Elev.

Example using different numbering

Example of branched network with inflow at its trunk. Elevations change.

SSPECIF NFLOW=4, NPGPM=4, NUMIT=2, NMDIA=1,
NMDST=1, ICOST=1, IHGL=2, NPLENG=1,
DESIGN=1, NPRINT=2 SEND
150 -3520 100 220 10000 0005
101
ELEV
60
117 110 .6 15. 5./
DREAD
108 20
118 108/
DREAD
112 20
118 101/
END
RUN
14 .28 .36 .565 .72 .87 1.075 1.17 1.32 1.475 1.6
1.8 2.2 2.4/
117 150
INTEREST=.08 LIFE=60
Pipes
UNIT=.12 CAPI=150000.
END

例

Example of branched network with inflow at its trunk. Elevations change.

SSPECIF NFLOW=4, NPGPM=4, NUMIT=2, NMDIA=1,
NMDST=1, ICOST=1, IHGL=2, NPLENG=1,
DESIGN=1, NPRINT=2 SEND
150 -3520 100 220 10000 0005
101
ELEV
60
117 110 .6 15. 5./
DREAD
108 20
118 108/
DREAD
112 20
118 101/
END
RUN
14 .28 .36 .565 .72 .87 1.075 1.17 1.32 1.475 1.6
1.8 2.2 2.4/
117 150
INTEREST=.08 LIFE=60
Pipes
UNIT=.12 CAPI=150000.
END
Example with ground elevations also changing

EXAMPLE OF BRANCHED NETWORK WITH INFLOW
AT ITS TRUNK. Elevations change.

/*
$SPECIF NFLOW=4, NPWM=4, NUNIT=2, NOMDIA=1,
NOMENG=1, ICOST=1, IHGL=3, NPINGLEG=1,
DESIGN=1, NPRINT=-2 $END
150 -3520 100 220 -1000.0005 .005 1.17
101
ELEV
50
117 110 .6 15. 5. /
DREAD
108 20
118 126/
118 101/
END
RUN
14 .28 .36 .585 .72 .87 1.075 1.17 1.32 1.475 1.6
1.8 2.2 2.4 /
117 150
INTEREST=0.8
LIFE=60
PIR DEN
UNII=14
.28 128 .36 162 .565 245 .72 306 .87 366 1.025 430
1.17 556 1.32 740
1.475 822 1.6 900 1.8 950 2 1000 2.2 1065 2.4 1090
PUMPS
UNIT=12
CAPI=180000.
END

Function # 3:

If IHGL=3, then special additional input data is provided that allows USU-NETWK to generate the elevations of the HGL over the network. These additional data define HGL elevations at intersection points of a rectangular grid system. When this third function is utilized the following occurs: (1) The elevation of the HGL at nodes are generated by interpolation of surfaces over the network that defined the HGL elevation. The HGL surface is defined by additional data that are described below, (2) The x- and y- coordinates must be given for each node in the regular data entered with the NODES or the PIPE- command, rather than the pressure, head or HGL elevations. The position for the nodes defined by these coordinates are used to computer the HGL elevations from the HGL surfaces of (1).

The special additional data must be given immediately after the $SPECIF list of options and consists of:
(a) The number of x = constant grid lines, NOX,
(b) The number of y = constant grid lines, NOY,
(c) The x values of each succeeding x equal constant grid lines (NOX of these values must be given),
(d) The y values of each successive y equal constant grid line followed by the HGL elevations at each of the intersections this y = constant grid line has with the NOX x = constant grid lines. While this input is free format and may be on a single line if it will fits within 80 columns, it is useful to the separate (d) above into separate lines, one for each y=constant grid line. Then

there will be NOY separate lines under (d) and each lines will contain 1+NOX values, the first of which is the y that defines this y=constant grid lines, and the remaining NOX values are the HGL elevations at the intersections with the x=constant grid lines. In illustrating this input by a small example below data items (a), (b) and (c) are on the same line, i.e. the first line after the $SPECIF list of options, and item (d) is on NOY separate lines.

This special additional input is illustrated by the small example network shown below. In this example the two x=constant grid lines are show by the two dashed vertical lines on the sketch, and the 3 y=constant grid lines are shown as dashed horizontal lines. However, the 2nd or middle y=constant lines coincides with the pipes (1),(3) & (5) and therefore is not visible.
INITIAL

In the version of USU-NETWK that also does time-dependent solutions there are two subroutines to provide initial flowrates which satisfy all junction continuity equations. The second such subroutine requires a previous valid solution. It is normally called upon to initialize flows after the first solution for time zero has been obtained, or the first of a series of alternative solutions has been obtained when using the CHANGE command to define a series of separate solutions. If INITIAL=0 is added to the $SPECIF list, then all initializations will be produced by the subroutine which does not utilize information from a previous solution. (Default, INITIAL=1).

In addition to the usage of the option INITIAL with the version of USU-NETWK that also has the capabilities to do time dependent solution in which this option determines whether the initialization of pipe flow rates is to be based on the past solution, or generated by the same subroutine that produced the first initialization, it can be used to read in the initial flow rates. This alternate use of the option INITIAL occurs if its value is negative, and is also implemented in some of the version of USU-NETWK that allow only for steady state solutions. The different allowable negative values that may be assigned to INITIAL have the following meanings:

INITIAL=-1 then the initializing flow rates are given after the RUN or END command e.g. are in the same file as the other input data. The sequential order in which these flow rates must be given is the same as the order that the pipes are entered under the PIPES or the PIPE- command. The direction of the flow is established by the first and the second nodes given after the PIPES or the PIPE- command, and a negative value indicates that the starting flow rate is in the opposite direction. The units of this flow rate are the same as that used for the demands, and is determined by the option NFLOW.

INITIAL=-2 then each initializing flow rates is preceded by its pipe number. With INITIAL=-2 the order of the initializing flow rates does not need to agree with the order of the other pipe data input, but the input consists of pairs of values; an integer that is the pipe number and a real that consists of the flow rate. The units of this flow rate are determined by the option NFLOW and this input comes after the RUN or END command.

INITIAL=-3 then the input is as with INITIAL = -1 except it must be given in another file with the name INITIAL.DAT, and is not placed after the RUN or END command. If this file does not exist USU-NETWK will prompt for the file name that contains the initializing flow rates.

INITIAL=-4 then the input is as with INITIAL = -2 except it must be given in another file with the name INITIAL.DAT, and is not placed after the RUN or the END command.

If the initializing flow rates are given in basic units, e.g. cfs when the problem is in ES units, or m$^3$/s when the problem is in SI units, regardless of the units used for the demand input as determined by the option NFLOW then 10 should be subtracted from the above values. In other words INITIAL = -11 is the same as INITIAL = -1 except that regardless of the value of the option NFLOW, the initializing flow rates will be in basic units, cfs or m$^3$/s.

The initializing flow rates must satisfy each and every junction continuity equation within the network. USU-NETWK will check that this is the case, and if the values do not satisfy the condition that the sum of all flow rates into a junction are the same as flow rates out of a junction (including the demand) then a message to this effect is written, and the solution is terminated.

INPUT #

Allows the FORTRAN logical input unit to be specified. If TTY or REMOTE is given in response to the prompt from USU-NETWK for the input file name then INPUT=5, the standard terminal or keyboard input. If a file name is given to this response, then INPUT=2, and the input is read from the given file. Should your computer installation have different logical unit number assigned, or if unit 2 cannot be used for input for some reason, then by adding INPUT to the $SPECIF list the flexibility is allowed to use these units without recompiling USU-NETWK. Similar flexibility of the output unit is allowed with the option IOUT.
INPUTA #

Allows three different forms of input data to describe a network. If 
INPUTA=0, then the standard form of input described in this main body of the manual is used. In this form a command name enters data of the specified type. (Default INPUTA=0) 
INPUTA=1, then the form of input data described in Appendix A is used. 
INPUTA=2, then the form of input data described in Appendix B is used.

IOUT #

Allows the user to specify the logical FORTRAN output unit on which the solution results will be written. If TTY or REMOTE is given in response to the prompt from USU-NETWK for a file name on which the output should be written, then IOUT=6, the usual terminal, or monitor output device. If a file name is given to this response then IOUT=3. If IOUT is included in the $SPECIF list of options then the value assigned to IOUT will be the output logical units on which the solution results are written. Thus, if needed, assigning IOUT a value allow flexibility at different computer installation in sending the output to various output devices, which may include tapes, for example, on which the output is to be stored.

IOUTI #

Reads the formatted data written by a previous execution of USU-NETWK with the option IOUTO equal to the same logical unit number or prepared according to the FORMATS indicated under the description below under IOUTO. IOUTI can only be used if this data file has been previously written. For example if during a previous execution of USU-NETWK IOUTO=20, then if IOUTI=20 during this run the input for the network will be read from this file, and this file must have the input in the FORMAT described under the option IOUTO.

IOUTO #

The option IOUTO instructs USU-NETWK to write an input data file under the built-in FORMAT. The value given to IOUTO is the FORTRAN logical unit number that will be used to write this file. For example if IOUTO=20 were included in the $SPECIF list of options than data would be written to a file using Fortran logical unit 20. Since data in this file is in specified columns, USU-NETWK can read the data in this file faster than it can a file prepared by an editor, but unformatted. However, this file can be edited to change data it contains so that the problem read with IOUTI is different than the one for which the data was written. Thus the use of IOUTO, and IOUTI above, is an alternative to using the CHANGE capability of USU-NETWK in obtaining a series of solutions for a network problem that is being studied. An important differences is that the user can examine the solution just obtained and use this as a guide in deciding what changes should be made for the next solution. An alternative means of writing an output data file is available with option NETPLT.

In making changes to this file, (1) the order of data as described below must be maintained, (2) the FORMATS for the data that are described below must be maintained, and (3) the pump characteristics must be defined by giving the coefficients for the second degree polynomial used to define the head versus flowrate curve for the pump. These coefficient are given as part of the solution output, unless suppressed with a small value of NPRINT.

The order of data in this file is as follows:

1. A line that gives the number of the following in the network using the FORMAT (1615): (a) pipes, (b) nodes, (c) source pumps, (d) booster pumps, (e) reservoirs, (f) minor losses, (g) pressure reduction valves, (h) nozzles, (i) check valves, (j) back pressure valves, (k) sum of reservoirs, nozzles, PRVs and BPVs, and if NOCHK=1 then (l) total loops, (m) pseudo loops, and (n) real loops.

2. The pipe data with the FORMAT (315,2F15.3,F15.7)

3. The node data with the FORMAT (I5,F15.5), and the following 4 items are needed on each line: (a)

4. The reservoir data. This data depends upon whether NODESP=0 or NODESP=1. If NODESP=0 the FORMAT is (15,F15.5), and the items are (a) the internal pipe number i.e. pipe sequence number under the PIPES or PIPE-command, and (b) the water surface elevation. If NODESP=1 the FORMAT is (215,2F10.5), and the following 4 items are needed on each line: (a)
the internal pipe number that connect the reservoir to the network, (b) the node number given the reservoir, (c) the water surface elevation in the reservoir, and (d) the ground elevation at the reservoir node.

(5) The pump data. This data is also different depending upon NODESP. If NODESP=0, then the FORMAT is (I5,5E15.8,E15.8) and the following items are given: (a) the internal pipe number, i.e. pipe sequence number under the PIPES or PIPE command, (b) the coefficient for the squared term for Q, (c) the coefficient for the linear term, (d) the constant coefficient all for a second degree fit of the pump characteristics modified by the water surface elevation and .5*QBA/BP, (e) the transformation term QBA, (f) the normal capacity, and (g) the water surface elevation. If NODESP=1, the FORMAT is (2I5,4E15.8,J,3EI5.8) and the following items are given: (a) the internal pipe number, i.e. pipe sequence number under the PIPES or PIPE command, (b) the node number (external value) for source pumps and 0 for booster pumps, (c) the coefficient for the squared term for Q, (d) the coefficient for the linear term, (e) the constant coefficient all for a second degree fit of the pump characteristics modified by the water surface elevation and .5*QBA/BP, (f) the transformation term QBA, (g) the normal capacity, (h) the water surface elevation, and (i) the ground elevation at the pump node.

(6) The minor loss data according to the FORMAT (15,F10.5) that consist of the pipe number, and this loss coefficient.

(7) The pressures reduction valve data according to the FORMAT (15,F15.9,F10.5) consisting of the pipe number, the distance from it upstream junction of the PRV, and the HGL elevation of the PRV setting.

(8) The nozzle data according to the FORMAT (15,F15.8) and consisting of the pipe number and the nozzle coefficient, CNOZZ.

(9) The internal pipe numbers containing check valves under the FORMAT (16I5), and

(10) The back pressure valve, BPV, data according to the FORMAT (15,2F10.5), and consisting of the pipe number, the distance from the PRV to the downstream node, and the HGL setting.

If NOCHK is greater than 0, then in addition to the above 10 types of items, the pipe numbers defining the loops of the network are also given. If PRVs or BPVs are present two sets of loops are used in the solution; the corrective flowrate and the energy loops. Only the corrective flowrate loops are written, however. When using IOUTL, this second set of loop data will be generated.

IRESID

It may be possible to save a small amount of computer time by not redefining the Jacobian Matrix and completely solving the new linear algebra problem for the final few iterations of the Newton method. This option allows for use of the last solution of the Jacobian problem to be utilized if the sum of changes is less than IRESID (a real value). Experience has indicated little if any computer time is saved and use of IRESID is discouraged. It can only be used in conjunction with the sparse matrix solution method (default IRESID=0).

ISIML#

Setting ISIML=1 indicates that the solution to the network is to be time-dependent, and therefore additional data as described under the major section "Additional Data for a Simulation Solution" is required (default, ISIML=0).

ITERA (MAX)

The maximum number of iterations that are allowed in order to obtain a solution. The default number is set to 15 (but may be different). For example if ITERA=6 were included in the $SPECIF list then the solution process would terminate after the 6th Newton iteration even if the error condition ERROR were not meet.

LENGON

This option permits lengths of pipes to be determined by providing x and y coordinates for the nodes rather than giving the pipe lengths. The default is that pipe lengths are provided, and this default corresponds with LENGON=1 (lengths of pipes on). If x and y coordinates are provided, i.e. LENGON=0, then NODESP must also be set to 1, because with NODESP=0 there is no means for providing the coordinates of sources of supply and consequently no means exist for USU-NETWK to compute the length of pipes that connect sources of supply to the network. If
LENCON=0 is included in the $SPECIF list of options, then the following changes must be made to the input data described in the section "Detailed Steady-State Data Requirements" under the PIPES, NODES and PIPE- commands. Under the PIPES command the fourth item, or the pipe length, is omitted so that only the following 5 items exist: 1. pipe no., 2. upstream node no., 3. downstream node no., 4. pipe diameter, and 5. pipe wall roughness coefficient. Under the command NODES the x and y coordinates of the node are provided after the elevation of the node so that instead of 3 items of input the input consists of the following 5 items: 1. node no., 2. demand at node, 3. elevation of node, 4. x coordinates of node, and 5. y coordinate of node. Should DESIGN=1, then the elevation of the HGL (or head or pressure) at the node is provided as the 4th item followed the x and y coordinates. Should NETPLT be given a value greater than 0 but less than 13, so that the x and y coordinates are provided as input already on the lines after the NODES command, then the only difference in input with LENDON=0 is that pipe lengths are omitted from the data after the PIPES command.

If the PIPE- command is used in place of PIPES and NODES to enter data, then the pipe lengths are omitted from their usual position after the diameter on lines after PIPE- However, the x and y coordinates are provided after the elevation given for either the upstream or downstream node. In other words there are now 4 real values that follow the node numbers on lines after the PIPE- command. Therefore, the input data under the PIPE- command is similar to that if 0 < NETPLT < 13, with the exception that the pipe length is not given. Since sources of supply must have been identified by nodes when LENDON=0, it is not permissible to have only one node per lines of data under PIPE- as is permitted with NODESP=0 if the coordinates are to be used only for plotting. Also whenever coordinates (or HGL elevations) are given it is not permitted to omit the demand at a node. Therefore, with LENDON=0 there must be 4 real values with a decimal point that follow the integer node values. If DESIGN=1 is also included, then the HGL elevation (or head or pressure) at the node must follow the elevations giving 5 such values that follow the node numbers. With LENDON left equal to the default value of 1 and DESIGN=1, then 3 real values must follow all integer node values for data entered through the PIPE- command. The use of PIPE- is not permissible with LENDON=0 with the PC steady-state version of USU-NETWK.

LENORG

In the final solution table the lengths of pipes printed will be their actual lengths plus any equivalent lengths for minor losses within them under the default options. If their actual lengths are to be printed then set LENDON=1.

LOOPOD

Is an option that allows for the loop data to be read in as input data rather than being generated internally by USU-NETWK. If LOOPOD=1 then the loop data must be supplied as input in addition to the other required data. If the standard form of input is used (i.e. that described in the body of this manual rather than in Appendixes A or B), then this loop data follows the RUN or END command immediately. The first record of additional data consists of two integers; the number of pseudo and the number of real loops (pseudo loops connect sources of supply, PRV, and/or BPV). This line is followed by loop data, each line of which lists the pipe numbers around each separate loop. One and only one line is used to define each loop. Pseudo loops must be given before real loops. The pipe number is given a positive value if the direction transversed around the loop is in the direction of assumed flow, and pipe numbers must be preceded by a minus sign if the direction traversed around the loop is opposite to the assumed direction of flow. To see how these loops are defined you should set NPRINT=1 or larger, and study the loop data that USU-NETWK places in the output file.

If the network contains pressure reduction valves, or back pressure values, then it is necessary to repeat the loop data twice (including the line of data that gives the number of pseudo and real loops). The first set of loop data is for the corrective flow rate loops and the second for the energy loops. See modification # 4 of the input for Example Problem # 1 for an example. If INPUTA=1 (that is if the form of input described in Appendix A is used) and LOOPOD=1, then the first record in front of the corrective flowrate loops are not given since this
data is given in the options under SASPECI. INPUTA=2 requires loop data. The option INPUTA=3 is used when the form of input is as described in Appendix B, with the exception that loop data are not provided but rather are generated internally. In other words INPUTA=2 or INPUTA=3 has the same effect as LOOPRD =1 or 0 if the data form described in Appendix B is used.

LOOPSE

This option allows control over the pseudo loops that will be used in setting up problems for solutions. The amount of computations required to obtain a solution depends upon the length of loops (i.e. the number of pipes in a loop). Should some loops be very long then the amount of arithmetic require for each Newton iteration in solving the linear system of equations become large especially if one of the banded methods of solution is used as described under the option NSYMMT. Pseudo loops between sources of supply (reservoir, source pumps, PRV and BPV) have the potential of being long. A long loop will occur for example if it is formed between two sources of supply that are at opposite ends of a large network. In using USU-NETWK the user has four means by which he can control loops.

1) The sources of supply can be entered in the input data in an appropriate manner, as described below, to keep pseudo loops short. (LOOPSE=0, the default).

2) The complete loops, both the corrective flowrate and energy, can be given by using the option LOOPRD.

3) The end pipes for pseudo loops between sources of supply can be specified. (LOOPSE=1)

4) USU-NETWK can be requested to optimize the set of pseudo loops. (LOOPSE=2) The last 3 of these means can be selected among through the use of the option LOOPSE. The default is LOOPSE=0 which results in (1) above. Reasons why (1) above is the default are given below.

**LOOPSE=0 (default)**

By selectively pairing reservoirs, source pumps and/or pressure control valves so that minimum sequences of pipes are needed to connect them the length of loops can be keep small allowing the banding subroutine to find a smaller band width for the Jacobian and result in reduced computer execution time. If LOOPSE=0 then USU-NETWK assume that you are giving the pairs of sources between which pseudo loops are to be formed by the order in which these sources are given in the input data. The first pseudo loop will be formed between the first and the second source pump; the next pseudo loop will connect the second to the third source pump, etc. The last source pump will have one pseudo loop between it and the first reservoir, the next loop will be between the first and the second reservoir; and finally there will be a loop from the last reservoir to the first PRV, etc. the last PRV to the first BPV, etc. If only one source pump exists then the first pseudo loop will be between it and the first reservoir etc. The above assumes that a connected path of pipes exists between the two sources, but if pressure regulating valves exist then such connected paths may not exist between all sources for the energy loops. USU-NETWK will determine this and find alternative sources to connect when it is necessary, while still maintaining the independence of the equations defined by these loops.

In setting up the system of hydraulic equations that needs to be solved two separate sets of loops are formed by the above procedure if pressure regulating valves are present or other special device exists in the network; namely the corrective flowrate loops, and the energy loops. The corrective flowrate loops do not involve PRV's or BPV's and the energy loops, about which the head losses are equated to the differences in HGL elevations at the sources, do include the PRV's and BPV's as if they were sources of supply. (See the book "Analysis of Flow in Pipe Networks" by Jeppson for an explanation of these two different loops.) Thus in entering the data one should enter the second pump after the PUMPS command as the pump closest to the first pump, and the third pump so it is close to the second, etc. The last line after the PUMPS command should represent a source that is close to the first reservoir entered under the RESER command, and the second line under RESER should be a reservoir close to the first reservoir etc. Pseudo loops for energy equations are form between the downstream side of PRV's and other sources, and between the upstream side of BPV's and other sources. Therefore, the order in which lines occur after the command VALVE should be such that the downstream side of the first PRV is close to the upstream side of the first BPV, etc.

For example if there are 5 source pumps with pipes 8, 1, 30, 200 and 121 connecting them
to the network, 4 reservoirs connected to the network by pipes 2, 10, 50 and 231, and 2 PRVs in pipes 38 and 56, then as the default USU-NETWK would attempt to form the corrective flow rate loops between the following pairs of pipes: (8,1), (1,30), (30,200), (200,121), (121,2), (2,10), (10,50), and (50,231) and the energy loops between the following pairs of pipes: (8,1), (1,30), (30,200), (200,121), (121,2), (2,10), (10,50), (50,231), (231,38), (38,56). Since the corrective flow rate loops might contain the pipes with PRVs, but the energy loops cannot, the first of these loops often are different even though they contain the same end pipes.

Should there be no path of connected pipes between the end pipes selected in this manner, then a message to this effect is printed, provided NPRINT is large enough, and an alternate pair of end pipes is chosen. Should all such pairs be exhausted and still no pipe to a particular end pipe found then additional messages are written and the solution either terminated or perhaps continued if it appears a valid single network may exist. If no path exists from a particular source of supply to any other source of supply it means that the network is actually two separate hydraulic systems. USU-NETWK assumes you have made a mistake, should this occur, and terminates the solution. You must process this dual system as two separate analyses.

For many networks the user can easily examine the map of his water system, and follow the above guideline in entering data. There are factors that do complicate the above, however. It is possible that PRVs and/or BPVs separate the network into 2 or more different pressure zones, in which event there will be no paths of connected pipes between all sources of supply and pressure control valves. In some systems it is possible to form shorter pseudo loops by connecting two or more sources to a single source. It can occur that the ordering of sources as described above to keep the corrective flow rate loops short may result in long energy loops or vice versa. (Other things being equal it is best to favor keeping the energy loops short over keeping the corrective flow rate loops short.) In some systems it is possible to form shorter pseudo loops by connecting two or more sources to a single source. Therefore, even if one gave full consideration to the above ordering it may not represent a very good solution to the problem of getting the best set of pseudo loops defined for a given network. Setting LOOPSE=1 or 2 provides other possibilities.

**LOOPSE=1**

The method referred to as (2) above, of controlling the formation of pseudo loops by providing the pipe pairs between which both the corrective flow rate loops and the energy loops are to be formed, is selected by setting LOOPSE=1 in the $SPECIF list of options. This option allows much more flexibility than ordering sources as they are entered. Since several other sources (including PRVs and BPVs) might be paired with the same source, this option does require extra thought and input data on the part of the user. These pairs of pipes are given immediately after the RUN or END command.

Each new energy pseudo loop thus specified must represent a new independent equation, e.g. that is not redundant with those already given. USU-NETWK assumes you know what you are doing when you give pairs of pipes with LOOPSE=2, and mistakes in giving these pair can cause unpredictable results. Giving pairs between which no connected series of pipes exist will result in termination of the solution, but for other mistakes on your part no checking is done.

If PRVs or BPVs are present then two such sets of integer values must be given that provide the end pipes for (1) the corrective flow rate loops, and (2) the energy loops. The number of pair of integers required with LOOPSE=1 equals the number of sources of supply (source pumps plus reservoirs minus one for the corrective flow rate pseudo loops, and the number of sources of supply including the PRVs and BPVs as artificial reservoirs (source pumps+reservoirs +PRVs+BPVs) minus one for the energy loops. When using the DHEAD command additional pseudo loops will be formed between the node whose HGL is specified and the source given, but these are not included in the above pairs. Likewise specifying unknown demands, but known HGL's with the command SETPR, etc. will add to the number of pseudo loops, but again you do not include these as pairs. Modification 5 of the input data for Example #1 illustrates the use of LOOPSE=1.

As a final means of forming pseudo loops you can request that USU-NETWK optimize the set of both corrective flow rate loops and energy loops to minimum the lengths of all of these loops. One might assume this is the best solution, and should be the default. The problem is that a considerable amount of computer effort is necessary to perform this optimization. Often it is much easier for an individual to do quite a satisfactory job by simply looking at a map of the
piping system. The computer does not view a map of the network. Rather it must form all possible loops and then make a number of decisions about which of all of these possible loops are the best sets to use. The table below shows the number of independent loops that are potentially available (without pressure zones occurring) as a function of the number of sources of supply.

With LOOPSE=3 all of these loops will be formed, or attempted to be formed, and decisions made about which of these to use so that they are all as short as possible, while still representing a set of independent equations. There are three methods that could be used to handle the information related to these loops within the program: (1) the information could all be retained in memory, (2) it would be written to a direct access file, and retrieved as necessary later, or (3) the loops could be repeatedly redetermined whenever they are needed by the optimization scheme. Increased computer times are definite disadvantages of (2) and (3). For a large number of sources, method (1) will require a large amount of memory since each loop may require a large number of word or information. The decision was to use (1) above, but discard, i.e. replace information as soon as it is apparent that a loop will not become part of the final set of pseudo loops. Furthermore, since it is not possible to determine in advance how much memory may be required, and it would be costly to check each time new loop information is obtained whether memory is still available, no checking is done within the program to terminate execution if adequate memory is not available.

If a number of solutions for a network are anticipated one might first use this option, and then later use LOOPSE=1, or better yet to extract these loops from the output obtained with NPRINT=1 or larger, and then add them to the input data with the option LOOPRD=1, or use the options IOUTO and IOUTI in obtaining these additional solutions.

<table>
<thead>
<tr>
<th>No. sources</th>
<th>2</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>30</th>
<th>50</th>
<th>75</th>
<th>100</th>
<th>160</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pseudo loops</td>
<td>1</td>
<td>10</td>
<td>45</td>
<td>105</td>
<td>190</td>
<td>135</td>
<td>1225</td>
<td>2775</td>
<td>4950</td>
<td>12720</td>
</tr>
</tbody>
</table>

**MANNI**

Manni set equal to 1, is an alternate to setting NEQUAT=2 in the $SPECIF list to specify that Manning's Equation is to be used in computing the relationships between the headlosses and flowrates in the pipes of the network. From the magnitude of the wall roughness coefficient USU-NETWK cannot distinguish whether the Darcy-Weisbach Equation or the Manning's Equation is to be used, and the former equation will be used. Therefore, it is mandatory to specify MANNI=1, or NEQUAT=2 if Manning's Equation is to be used.

**MAX**

See ITERA

**NCONTI**

NCONTI=1 causes USU-NETWK to act similar to when NOSTOP=1, however, does not terminate execution when error conditions associated with appurtenances of the network are detected.

**NELEV**

This option has a dual function; one associated with the standard input, and the second when the form of input described in Appendix B (i.e. INPUTA=2 or 3 is given in the $SPECIF list).

In connection with the standard input NELEV can be given the value -1 to specify that an extra table should be written that distinguishes pressure heads and velocity heads for the different pipes that join at each node of the network. Ordinarily NETWK does not make any distinction between the elevation of the energy line, (the sum of the elevation head, the pressure head, and the velocity head), and the elevation of the hydraulic grade line (the sum of the elevation head, and the pressure head). However, if it is important that some difference be made then this can be accomplished by setting NELEV=-1. With NELEV=-1 a special extra table will gives the pressure head and pressure at each node that joins every junction of the network by substracting the different velocity heads in the pipes from the elevation of the energy line. Since the pipes joining at a junction will generally have different velocities in them these pressure heads will be different. The reported values in the NODES DATA table will be the elevation of the energy
line, and the head reported in this extra table will be the sum of the pressure head and the velocity head. In addition if \( \text{NELEV} = -1 \) then a loss will be assumed equal to one-half the velocity head, for flow in any pipe leaving a reservoir, and a full velocity head will be assumed to be lost for any flow in a pipe entering a reservoir.

\( \text{USU-NETWK} \) is not designed to correctly solve pipe flow problems for which velocity heads dominate over pressure head, as is the case for a small diameter venturi, or aspirator. For such problems concern about what controls the elevation of the energy line in pipes upstream and downstream of nodes must exist. \( \text{USU-NETWK} \) assume the energy lines are the same for all pipes that connect at a node.

If \( \text{NELEV} \) is given a value in conjunction with \( \text{INPUTA}=2 \) or 3, then it can have the values 0 or 1. The default is \( \text{NELEV}=1 \). If \( \text{NELEV}=0 \), then the elevation of the node is read in immediately after the demand on the line of group No. 27, described in Appendix B, before the pipes which join at that node. If \( \text{NELEV}=1 \), then the elevation is read on a separate line as described in Appendix B.

**NEQUAT**

Determines what equation will be used to define the relationships between the headlosses and flowrates in pipes. If \( \text{NEQUAT}=0 \) then the Darcy-Weisbach Equation will be used; if \( \text{NEQUAT}=1 \) then the Hazen-Williams Equation will be used; and if \( \text{NEQUAT}=2 \) then the Manning's Equation will be used.

\( \text{USU-NETWK} \) will determine whether to use the Darcy-Weisbach Equation or the Hazen-Williams Equation based on the first pipe wall roughness coefficient given under the \( \text{PIPEC} \) or the \( \text{PIPE} \)- command, and therefore, \( \text{NEQUAT} \) is generally not included in the \( \text{SSPECIF} \) list. If no pipe roughness values are given so that the default value of \( \text{COEFRO} \) is used, then this value will determine the equation that will be used, and since the default value of \( \text{COEFRO} = 0.102 \), if it is not included in the \( \text{SSPECIF} \) list, the Darcy-Weisbach equation will be used. However, if the Manning's Equation is to be used either \( \text{NEQUAT} \) must be given a value of 2 or \( \text{MANNI} \) a value of 1 because the magnitude of Manning's \( n \) and the equivalent sand roughness, \( e \) cannot be distinguished. Setting \( \text{NHAZEN}=1 \) is equivalent to \( \text{NEQUAT}=1 \), as is \( \text{MANNI}=1 \) equivalent to \( \text{NEQUAT}=2 \).

**NEQUDI (EQUDIA)**

This option only has meaning if differential head devices of type \( \# \) 2 are entered with the command \( \text{DHEAD} \). If the differential head is of a magnitude so that the combined head caused by the frictional head loss plus the differential head can be achieved by another size pipe, then \( \text{USU-NETWK} \) will compute the diameter of this equivalent pipe. These equivalent diameters are the pipe sizes that the command \( \text{DHEAD} \) can be used to determine. If this special table of equivalent pipe diameters is not to be created \( \text{NEQUDI}=0 \) should be given. (Default \( \text{NEQUDI}=1 \))

**NETCHK**

This option allow the network's layout, or topology to be defined by both pipe and node oriented input data, and therefore is useful in verifying that the input data is correct that is being used by \( \text{USU-NETWK} \) in providing solutions. The default is \( \text{NETCHK} = 0 \) which indicates that the node oriented data is not given. The following four values can be given to \( \text{NETCHK} \) to allow different methods for providing this additional data that usually defines the network's layout: -1, 1, -2 or 2. If \( \text{NETCHK} \) is positive, i.e. 1 or 2 then the node oriented data is provided after the elevation of the node after the \( \text{NODES} \) command. If \( \text{NETCHK} \) is negative, i.e. -1 or -2, then this additional data follows the \( \text{RUN} \) or \( \text{END} \) command. The difference between absolute value of 1 and 2 is whether the assumed direction of flow should be checked in every pipe between this data and that given in the pipe oriented data. If \( \text{NETCHK} = [1] \) directions of flow are not checked whereas if \( \text{NETCHK} = [2] \) directions of flow must be the same in both definitions of the network.

The extra data requires if \( \text{NETCHK} \) is not equal to zero consist of the pipe number that join at each node of the network. If the value is 2 or -2, then these pipe number must be preceded by a minus sign (i.e. be the negative of the pipe number) if the assumed flow direction is into the junction.

To summarize the following is available with this option:

1. If \( \text{NETCHK} = -1 \), then after the \( \text{RUN} \) or \( \text{END} \) command extra data must be given that lists the pipes that joint at the nodes of the network. These data are on separate lines, and the data on
these lines is described as Card No. 27 using the alternate node oriented form of data described in Appendix B. Each line of this data consists of (a) the demand at this node, (b) the elevation of this node if NELEV=0, but if NELEV=1 then the elevation is not given, i.e. (b) is skipped, (c) the pipe numbers that join at this node. These lines must be in the same order as the node data given after the NODES command, and if the PIPE-command is used the order of these data must be the same as the order in which node numbers first appear.

2. If NETCHK=1, then pipe numbers that join at the node are listed after the elevation on lines after the NODES command. It is not possible to set NETCHK=1 and use the PIPE-command to enter the basis data for the network. Specified directions of flow are not checked i.e. it is not necessary to give minus values if the assumed direction of flow is into the junction.

3. If NETCHK=-2, then the input is identical to that described with NETWK=-1, except that if the assumed direction of flow for a pipe is into that node its number is preceded by a minus sign.

4. If NETCHK=2, then the input is identical to that described with NETWK=1, except that if the assumed direction of flow for a pipe is into that node its number is preceded by a minus sign.

For an examples that uses this option see Modification # 1, # 2 and # 3 for Example No. 1.

NETPLT

The option NETPLT is used to create several types of plots, or written data to files for later plotting, depending upon its value. It also is used to indicate that the input data file is in the format written by the preprocessor PIPEINPT, and therefore can be read faster as a formatted file. If NETPLT=13 or NETPLT=14 then the input data file must has the data in the columns that PIPEINPT writes in. If NETPLT equals zero, the default value, then no plot is made. Appendix C describes graphics that can be produced by USU-NETWK. This graphics is controlled by NETPLT. You should read Appendix C in connection with this description to understand how some of the graphic output can be requested, and what you must do for this to be possible. The PC version of USU-NETWK does not have some of the graphics capability described. Rather post processors may be used to do some of this same graphics, as described in Appendix C. Other versions of USU-NETWK may contain only some, or possibly none of the graphics routines, and therefore only the formatted input feature described above may be controlled by NETPLT.

If the absolute value of NETPLT=1 (i.e. 1 or -1) then a computer generated Calcomp Plot of the network will be generated. Depending upon the installation and the graphics drivers, this Calcomp Plot may be directed to other graphic devices. This plot consists of the layout of the network, with pipes and nodes labeled if desired, as well as pipe diameters and length printed. Demands at nodes may be printed. Flowrates in pipes may be printed, and pressure at nodes may be printed. A menu will appear before the plot file is written that allows selection of the items you desire to have on the plot, as well as its size, and the height of the lettering on the plot. The version of USU-NETWK designed to run on PC's under DOS will write data to a file PLTNET.DAT if NETPLT=1, that can be used by the post processor PLTNET to display the network, and selected item from its solution. See Appendix C and the explanation in using program PLTNET.

Summary of option NETPLT

This option controls:
1. Output files for graphics
2. Writing data files that can be used as input to USU-NETWK, or other graphics programs.
3. Reading formatted input files created with PIPEINPT.

THE FOLLOWING OCCURS IF NETPLT

= 0 No graphics output, or special data files written, free format input.
= 1 on a VAX: Calcomp routines are called to:
(a) Write Calcomp plot file for network & its solution. A menu will allow you to determine plot size, what to include, height of lettering, etc.
(b) Write a file that contains basis information to do graphics with another graphics software package.
(c) Both (a) and (b)
= 1 On PC: Writes a file for program PLTNET to use to display network & selected items on monitor.
= -1 on a VAX: Same as 1 except you can give message to calcomp operator(dependent upon installation) Message may be: USE INDIA INK, PEN SIZE # 2
= 2 Same as 1 except original input data does not have x & y coordinates for nodes of network. These are, therefore, provided in another file.

= - 2 Same as 2 except you can give message to operator.

= 3 More general graphics output file will be written that contains information for auxiliary programs to use such as:
(a) PLTNET3 (Requires that DISSIPLA software be present) - makes:
  (1) 3-D perspective display of network & HGL.
  (2) Contour map of pressure, etc.
  (3) 3-D perspective over contour map.
(b) PROFIILM - makes
  (1) HGL-elev. & elev. profiles thru selected sequences of pipes.
  (2) Network layout with pressure bars at nodes.
  (3) Contour map type map that shows lines of constant pressure, pressure head, or HGL-elevations.
(c) OFPLOT - draws the network and selected solution items similar to the plot routines included as part of the full simulation version of USU-NETWK. OFPLOT is a separate program for PC (or larger computer use) that consists of the same plotting subroutines that are built into the full simulation version of USU-NETWK.

= 13 & 14 Informs USU-NETWK that input data is according to format from PIPEINPT (& PLTNET). If = 14, then a data file written for PLTNET to use will be written from the solution.

NEWSTR #

When a series of solutions for a basic network are being obtained using the CHANGE command, or by running a time dependent solution, the usual procedure is to leave PRV's etc., that have shut off in that condition for the next solution or time step solution. By setting NEWSTR = 1 this procedure will be altered so that all devices will be opened, or put in the condition denoted by the original data, prior to beginning the iterative process of obtaining the next solution.

NFLOW

This option tell USU-NETWK what units are associated with the demands that are given under the NODES, or the PIPE- commands. In addition to several volumetric units for flowrate, USU-NETWK also allows weight and mass flowrates to be used.

Depending upon the value given to NFLOW the following units apply to demands:
NFLOW = 0, in cfs if ES units are used, in m³/s if SI units are used.
NFLOW = 1, in gpm (allowed only with ES units).
NFLOW = 2, in mgd (allowed only with ES units).
NFLOW = 3, in m³/s (allowed only with SI units).
NFLOW = 4, in liters/s (allowed only with SI units).
NFLOW = 5, in pounds/s (allowed only with ES units).
NFLOW = 6, in Newtons/s (allowed only with SI units).
NFLOW = 7, in slugs/s (allowed only with ES units).
NFLOW = 8, in kilograms/s (allowed only with SI units).

In solving a network problem, USU-NETWK assumes the fluid is incompressible, i.e. the internal energy or enthalphy of the fluid is not included in the energy equation. For some problems a reasonable assumption is that the internal energy does not change, but since the fluids density, or specific weight, might change significantly because of large temperature or pressure changes, an overall volumetric flow rate balance does not exist. For such problems USU-NETWK can produce a valid enough solution for many purposes by giving demands in units of weight or mass flowrates. Then continuity is satisfied, since all computations will be based on these weight or mass flowrate units. If cfs (or m³/s) are already in an input data file, weight or mass flowrates can still be specified for demands by giving PEAKF the value of 62.4 or 1.94 for ES units, or for SI units setting PEAKF=9800 or = 1000, respectively. Likewise PEAKF may modify gpm or mgd, etc. to weight or mass flowrates.

An example of a problem in which weight or mass flowrates should be used is the feed water train system of a steam turbine power plant where
the water leaving the condensor may have a temperature of 100°F or less and be at a pressure of 14.0 psia so that here the fluid's specific weight is 62.0 lbs/ft³. As the fluid passes through several stages of heaters and pumps and arrives at the boiler end, its temperature may be 400°F or more, and be at a pressure of 1000 psia or more, so that its specific weight is now 53.3 lb/ft³. The large change in the specific weight makes volumetric flowrates meaningless.

Another example is the flow of natural gas within a portion of the distribution network where internal energy is near enough constant to be assumed constant.

**NHAZEN**

If **NHAZEN**=1 USU-NETWK is told to use the Hazen-William equation for the hydraulic computations. If **NHAZEN** > 1, then Manning's equation is used. The same is achieved by setting **NEQUAT**=1.

**NOCHK #**

This option has meaning only when used in conjunction with **IOUTO** and **IOUTI**. It determines whether the input data will be checked for errors, etc., or not. If **NOCHK** is greater than 0 then the data read in from file **IOUTI** will not be checked for errors or whether a network is properly defined. Thus with **NOCHK** > 0 computer time can be saved. Not only is the data not checked for errors, but also the corrective flowrate loops are not regenerated. Thus if the data file written by setting **IOUTO** > 0 is changed so that loops must be different, then it is necessary that this loop data which is stored in the file also be changed, or **NOCHK**=0 so that these loops will be regenerated. It is not possible to set **NOCHK**=0 with **IOUTI** > 0 if **NOCHK** was set equal to 1 when the file was written with **IOUTO** > 0. The reverse is possible, however, that is **NOCHK** can be equal to 1 when using **IOUTI** > 0 even if **NOCHK**=0 when the file was created with **IOUTO**>0. (Default **NOCHK**=1)

**NODESO**

Determines whether or not the output in the **NODES** DATA table will be listed in the order of ascending nodes or in the order listed in the input data file. **NODESO**=1 list in ascending order (Default **NODESO**=1)

**NODESP**

This option must be set equal to 1 (**NODESP**=1) if source pumps and reservoirs are given node numbers. If these sources of supply are given node numbers, these node numbers must be given in the node data after the **NODES** command. If they are not numbered (**NODESP** =0, the default) then 0's are given to denote nonexistent nodes for pipes that connect a source of supply to the network when using the **PIPES** command. If nodes are numbered with **NODESP**=1, then nozzles just like reservoirs (and source **PUMPS**) are identified by their node number, and must also be numbered as nodes.

**NOMDIA**

This option has meaning only if **DESIGN** = 1, and then indicates to USU-NETWK that a list of standard pipe diameters will be given from which to select standard pipe sizes from. If **NOMDIA**=0, the default, then the default diameters given below will be used for this purpose. If **NOMDIA**=1, then immediately after the **RUN** command, this list of standard diameters must be provided. This data is provided by giving the number of new standard diameters in the list, followed by the values of these diameters all on the same line. Because of the dimensions in the program, the number in this list must be 20 or less. When using SI units this list must be given since the default standard diameters are in inches. The list of standard diameters, if the default **NOMDIA**=1 is: 0,4,6,8,10,12,15,18,20,24,30,36,42,48,54,60,72,84,96,108 inches.

**NOMSOL**

Has meaning only if **DESIGN**=1, and then determines whether USU-NETWK will stop when the special design solution is completed, or whether it will follow this design solution with an analysis solution in which the standard diameters will be used for the pipes whose diameters were not specified. If **NOMSOL**=0, the default, then, USU-NETWK stops when the solution for the unknown diameters is complete. If **NOMSOL**=1 then an analysis solution follows using the closest standard diameter to those computed.

**NOMSOL** can also be used for special computations associated with standard diameters and the design solution of pipe diameters. Generally no distinction is made between the energy line and the hydraulic grade line. If velocity heads are significant, and these head will
cause significant head and pressure differences in pipes that join at a junction, then USU-NETWK can be instructed not to ignore the velocity heads in computing nodal pressures and heads by setting NOMSOL negative. If NOMSOL = -1, then this distinction is made and an analysis based on the standard diameters does not occur. If NOMSOL = -2, then an analysis of the network using the standard diameters is performed.

The standard diameter is generally taken closest to the computed diameter. If the diameter just larger than the computed diameter is to be selected, then set NOMSOL equal to a value form 10 thru 20. If NOMSOL = 15, then the analysis based on the standard diameters will not be performed. If the diameter just smaller than the computed diameter is to be selected, then set NOMSOL = 21 through 30. If NOMSOL = 25 then the analysis solution based on the next smallest diameter will not be performed. If NOMSOL = 99, then instead of assuming a single diameter exist for the links between the nodes, the links will be composed to two pipe sizes with the standard diameters just smaller and just larger then the computed diameter. The lengths of these two pipes will be computed. No analysis will follow if NOMSOL = 99 since the specified HGL elevation are satisfied exactly.

NOPARP #

If a network contains a number of groups of parallel pipes it is possible to save computational effort in the solution by forming an equivalent pipe for these groups of parallel pipes and thus reduce the number of pipes and number of loops used in the solution. For example, if ten pipes are eliminated in this process the number of simultaneous equations that must be solved will be reduced by ten also, since each such removed parallel pipe removes a 2 pipe loop from the basic loops of the network. NOPARP allows control over whether an equivalent pipe will be created to duplicate the hydraulic characteristics of each group of parallel pipes or not. If NOPARP = 0 equivalent pipes will not be formed. If NOPARP = 1 the network will be examined for all parallel pipes and any such group of pipes with the same upstream and downstream nodes will be replaced by an "equivalent hydraulic pipe" in carrying out the solution. The removed pipes will be restored, however, at the time the solution tables are printed. All pipes in the group will be entered in the table in the position of the pipe that remains in the network as the equivalent pipe, thus parallel pipes will be easily identified in the output Table of PIPE data. The default for NOPARP = 0, since many networks do not contain parallel pipes and considerable internal checking is necessary to determine which pipes are parallel, and thus this checking is eliminated. If parallel pipes do exist in the network, especially if many parallel pipes do exist, NOPARP should be set equal to 1. It is obvious that any parallel pipes that are removed from the network cannot contain special devices such as PRVs, BPV, check valves, booster pumps, differential head devices, and minor losses. If such devices should exist in only one pipe of a group of parallel pipes the option NOPARP = 1 can be used provided this pipe occurs in the input data before the other pipes that are parallel with it, since the position of the first pipe is used for the equivalent pipe. Since the position for the other parallel pipes are removed, i.e., these pipes are not assumed to be in the network, diagnostic messages will indicate the devices have been specified in a nonexistent pipe and the solution terminated. In using either the Hazen-Williams or the Manning's equations minor losses will be handled properly with NOPARP = 1 provided only the first pipe of the group contains such a device. Since the equivalent pipe must be redetermined depending upon the total flow through a group of parallel pipes if the Darcy-Weisbach equation is used, minor losses will not be handled completely correct even if the first pipe contains this minor loss. It is also not possible to change any of the characteristics of removed pipes through the CHANGE command.

NOSTOP #

If the correct number of energy loops is not determined to equal the number of unknowns, then USU-NETWK stops execution. This condition occurs when there is an error in the specifications of the network. By setting NOSTOP = 1 this error condition will be ignored. Generally, however, execution will terminate later because of a singular matrix condition or some other condition.

NPERCT

USU-NETWK may be assisted in obtaining a better initialization by providing
estimates for the amount of inflow (or outflow as a negative inflow) from each of the sources of supply. To provide such estimates NPERC must be given a value of 1 or 5. If NPERCT=1, then estimates are in fractions of the total demand. If NPERCT=5, then these estimates are in the units used in specifying the demands. The order in which these estimates are given is the same as the order of reservoirs in the input data, followed by the order of source pumps in the input data, i.e. according to the internal reservoir and source pumps numbers. In the absence of such assistance in initializing each reservoir is assumed to supply an amount equal to their normal capacities, generally.

If NPERCT is greater than 0, the estimates of the inflow data come after the RUN or END command, i.e. are the last data line required in the input data besides the CHANGE data or SIMULATION data. This line comes after the data described if LOOPSE=1.

NPLENG

If set to a value greater than zero, i.e., NPLENG=1, then the length of the pipes are to be given in 1000 feet or 1000 meters.

NPNRES #

Applicable only if ISIML=1, and then denotes whether pump rules and changes in pump rotational speeds are to use pressure at designated nodes, or elevations of the water surfaces in the reservoirs to control the number of pumps which will be operating, i.e. whether the rule depends on the pressure at a node of a reservoir water surface elevation.

If NPNRES=0, then the pressure at the designated node control the number of pumps which will operate at a station. If NPNRES=1, then the water surface elevation in a designated reservoir controls the number of pumps which will operate at a station. This same option can be placed in the $TDATA list.

NPRINT (IOEX)

Determines the amount of extra output that will be printed. A NPRINT=-3 suppresses all such extra output and NPRINT=10 gives the maximum amount. The default value for NPRINT=0.

NPRPRV

Allows for the setting of pressure reduction valves and back pressure valves to be given in terms of the elevation of its downstream HGL elevation, or its pressure setting in psi, when using ES units, or Pascals when using SI units. If NPRPRV=0, then this setting is given in ft (or meters) of head. This HGL elevation equals $p/\gamma$ +elevation of the PRV or BPV. If NPRPRV=1 then this setting is given in psi, when using ES units, or Pascals, when using SI units. When this option is used, the elevation of the PRV or BPV is assumed at the point determined from its location in the pipe as though the pipe were a straight line between its nodes. (Default NPRPRV=0)
NPRPUM

Allows for giving pressure at the inlet of source pumps instead of the water sump elevation. If NPRPUM=1, pressure in psi is given, when using ES units, and pressure in Pascals is given when using SI units. These pressures will be the last item given after the command PUMPS. This option is useful when analyzing gas flows in networks by USU-NETWK, assuming that compressibility (i.e. thermodynamic) effects can be neglected. In such situations, the pressure of inlet sources of supply to pumps are known, and the sump elevation is an artificial quantity obtained by dividing the pressure by the specific weight of the gas plus the elevation of the pump location.

NPRRES

Allows reservoir sources of supply to have pressures specified rather than water surface elevations. If NPRRES=1, then pressure is given in psi when using ES units, and Pascals when using SI units, in place of the water surface elevation described after the command RESER. When using this option with NODESP=0, the elevation needed to convert the given pressure into an HGL elevation is taken as the elevation of the node downstream from the reservoir since no ground elevation is given at the reservoir. This option is useful for pipe networks containing gas flows (when thermodynamic effects are insignificant) and heads are artificial quantities computed by dividing pressure by the specific weight. If NPRRES=1, then pressures for all reservoirs in the network must be given as the second item on all lines after the command RESER.

NPRTRM

Allows the user control over the manner in which pressure reduction valves and back pressure values are shut-off, or removed from operation, etc. The usual mode of operation of PRV's and BPV's is to maintain the pressure, or HGL elevation setting that is given in the input data. However, should the direction of flow try to reverse, then the device will shut-off the flow when the residual becomes less than TVSUM, or if the HGL elevation supplied by the system is less than the value specified for the device, then the device is removed from operation. If this process is not finding the proper status for devices effectively, i.e. devices are repeatedly shut-off and then opened again, or removed from operation, and then placed back in operation, then NPRTRM may be given a value different from the default value of 0 to allow the user to effect some control of this process. The values of NPRTRM have the following effects:

NPRTRM=0 Do the usual thing in shutting off the flow if conditions seem to call for this, or removing PRV's and BPV's from operation.

NPRTRM=1 Print the results from the solution just obtained which indicates that the PRV should be removed from operation before redoing the solution with the PRV(s) removed. Examining this solution can give the user insight into the operation of the PRV(s), and whether its setting should be changed. (Simulation version only)

NPRTRM=2 Shut-off only one PRV at a time before performing additional iterations toward the solution. The PRV that will be shut-off, if more than 1 PRV has a negative flow rate through its pipe is the one in the pipe with the largest negative flow rate.

NPRTRM=3 Display information on the monitor showing the flow rates, etc through pipes that contain PRV's, and ask the user to response whether any PRV with a negative flow rate through its pipe should be shut-off or not. Thus the user can prevent a PRV from shutting off the flow by indicating that its flow should not be shut-off. This prompting will occur during each subsequent iteration for which the residual is less then TVSUM and a negative flow rate occurs in the pipe during this iteration. Also the user will be able to specify what PRV's should be removed from operation, or restored to operation.

NPRTRM=4 The same occurs as with NPRTRM =3 with the addition that the value of the current TVSUM (that determines when to check if PRV's should shut-off) is displayed and the user is requested to give a new value that will be used thereafter.

NPSERI

This option allows pumping stations to be references for purposes of specifying number of
pumps in series or parallel by: (a) their number (like the order of their occurrences in the input date), (b) by the pipe connecting them to the network, or (c) by their node number, if given a node number with NODESP=1. See data under commands SERIES and PARALLEL. If NPSERI=0, pumping stations are identified by the pump number, i.e. by the sequence number of the pump as determined by the order of the pump data after the PUMPS command.

NPSERI=1, pumping stations are identified by the pipe number that connects the source pump to the network or which contains the booster pump. (Default, NPSERI=1)

NPSERI=2, source pumping stations are identified by their node number. NPSERI=2 can be used only if these sources of supply are numbered as nodes by giving NODESP=1 in the $SPECIF list also. Booster pumps will still be identified by the pipe they are in.

NSORTP

This option control the order of pipes in the output table. If NSORTP=1 the pipes in the output tables will be arranged in ascending order of their numbers, i.e. the smallest pipe number will be the first entry in this table, and the largest pipe number will be the last pipe. If NSORTP=0, then the order in the PIPE DATA table will be that of the pipe data contained in the input data. (Default NSORTP=1)

NSYMMT #

This option allows selection between three separate solution methods that the full version of USU-NETWK has built into it. The PC version has but one solution method, as may be the case with other reduced version of USU-NETWK, and therefore this use of NSYMMT will have no effect. In addition NSYMMT can be used to control how USU-NETWK handles computing friction factors for use in the Darcy-Weisbach equation for turbulent flows that are close to laminar, where a discontinuity exists in the Moody diagram. See description below for this use of NSYMMT. Valid values for NSYMMT are: -1 (the default), 0,1,2,5 & 7.

One method of solution that USU-NETWK may use, which will be called method 1, is to introduce a transformation for pumps (see Jeppson's book "Analysis of Flow in Pipe Networks"). This transformation generally accelerates convergence, but destroys the symmetry of the Jacobian matrix used in the Newton Method. If NSYMMT=0, then method # 1 is used in solving the problem.

Another method, called method # 2, maintains symmetry in the Jacobian matrix, and consequently does not introduce the pump transformation. Since the Jacobian cannot be made symmetric when PRV's, BPV's or differential head devices introduced by the DHEAD command, and when pressure at nodes are specified with the SETPR command method # 2 can not be used if any such device exists in the network. Method # 2 is allowed only for problems for which a symmetric positive definite Jacobian Matrix is possible from the equations that solve for the corrective flowrates in the loops of the network. USU-NETWK will determine this if you select method # 2, and inform you. Method # 2 is selected by giving NSYMMT=1.

A third solution method, called method # 3, utilizes special sparse matrix methods in solving the linear system of equations for each Newton iteration. Method # 3 allows a symmetric, or a nonsymmetric matrix, and is the only method built into the PC version of USU-NETWK.

To summarize the use of NSYMMT in selecting the method of solution the following occurs:

If NSYMMT=-1, USU-NETWK will select whether to use method # 1, or method # 2 based on whether symmetry of the Jacobian matrix can be maintained or not.

If NSYMMT=0, method # 1 will be used.

If NSYMMT=1, method # 2 will be used if possible, otherwise method # 1 will be used if devices in the network do not allow symmetry, and a message will inform you that method # 2 cannot be utilized for this network.

If NSYMMT=5, method # 3, the sparse matrix routines will be utilized. (Default NSYMMT=5, generally, but may be NSYMMT=-1)

Note: If convergence is not achieved in using method # 2 or method # 3, and PRV do not exist, you should set NSYMMT=0 and see if convergence does occur. The use of method # 1 may require more computer time than method # 2, but generally it also is capable of converging to a solution when pump characteristics adversely influence convergence. It is possible, however, to create a network containing pumps for which no
solution exists. For example, if two pumps with flat curves supply a network in close proximity, but produce quite different heads, may result in a saddle point in the mathematics of the pump with the lower head (i.e. an imaginary solution) because it can not provide the needed head. For such situations converge to a solution can not occur or the flowrate will be negative through one of the pumps. Of course such situations should not be allowed in real piping systems.

In obtaining a time dependent solution USU-NETWK will get in trouble on an altitude valve shut off flow from (or to) a reservoir and it is attempting to obtain the solutions based on a symmetric matrix, i.e. method # 2. The program does not know in advance if an altitude valve will shut of and in the interests of efficiency selects method # 2 if NSYMMT = -1. Should this happen you will need to rerun the problem with NYSMMT = 0, or NSYMMT = 5.

In addition to the above use of NYSMMT it has the following use. By setting NSYMMT=2, or NSYMMT=7 you can request that USU-NETWK use an alternative method for evaluating the frictional head losses. This alternative method is only applicable for use with the Darcy-Weisbach equation, and will give result that differ from the regular solution only if the flow in some pipes is laminar, or very close to laminar. However, the use of this alternative method may be necessary to achieve convergence because of the discontinuity that exist in the friction factor f associated with Reynold's number around 2100. When the Darcy-Weisbach equation is used, then USU-NETWK check to determine whether the Reynold's number associated with the flow in any pipe is less than or greater than 2100 after each iteration. Should this Reynold's number be less than 2100 (i.e. the flow is laminar) then the friction factor is computed equal to 64 divided by Reynold's number, otherwise the implicit Colebrook-White equation is solved (or the wholly rough equation is solved should this be applicable). The friction factor solved from the Colebrook-White equation has quite a different magnitude than the friction factor computed under the assumption that the flow is laminar. For example at a Reynold's number of 2100 the Colebrook-White equation gives f = .050 for a 6-inch cast iron pipe (e=.0102 inches) whereas the laminar relationship f = 64/Re gives f = .030. Therefore, a discontinuity in the friction factors occurs for flows with small Reynold's numbers about 2100. In the iterative solution process the computation of the friction factor may alternate between being computed by the Colebrook-White equation and the laminar flow equation. Should this occur even for one pipe in the network, then convergence to the solution will not take place.

By specifying that NSYMMT=2, or NSYMMT=7, a continuous relationship between the friction factor and Reynold's number, i.e. the flowrate in each pipe is established. Use of this relationship should prevent the lack of convergence that would occur if the regular relationships are used. Before you can assume that the lack of convergence is due to the above switching between laminar and turbulent flow you should verify that the flowrates in some pipes may be laminar. This determination can be done by computing the Reynold's numbers associated with the flowrates given by the nonconverged solution. Another easier means for determining whether the flow is laminar is to set NPRINT larger, so that the coefficients and exponents from the exponential relationship are output, or requesting the wider single table of output by setting OUTPU1 = 0 or 2. (The PC version of USU-NETWK does not print the exponents when NPRINT is set large.) If an exponent is 2 (or one for the exponent minus 1 that is printed with NPRINT set larger) it indicates that laminar flow exist in that pipe.

To provide a continuous relationship between the friction factor and the Reynold's number (or flowrate), the laminar relationship $f=64/Re$ is modified as follows:

1. The friction factor is computed from the Colebrook-White equation for a Reynold's number of 2100 and the value for the relative roughness $e/D$ for that pipe.

2. The values of a and b in the equation $f = a/Q^{*b}$ are computed so this equation produces the $f$ computed by step 1 for $Re$ equal to 2100, and $f = .128$ for $Re = 500$ (This is the $f$ produced by $f=64/Re$), i.e. a straight line relationship is assumed on a log-log plot between the laminar relationship for a Reynold's number of 500 and that produced by the Colebrook-White equation for a Reynold's number of 2100.

3. Since the laminar flow relationship will produce an infinite value for the friction factor as the Reynold's number approach zero, a linear relationship is assumed to apply between $Re=500$ & $f = .128$ and an $f = 1.0$ for a Reynold's number of 0. The switching between the relationship from step (2) and step (3) however, may not occur until the Reynold's Number is less than 200 or possibly 150 depending upon what is set at the time the program is compiled.
In summary if the solution fails to converge, and it can be determined that the flow may be laminar in some pipes of the network, then the problem of lack of convergence may be corrected by setting NSYMMT = 2 or NSYMMT = 7. With NSYMMT=2, solution method used will be based on banding the Jacobian matrix just as if NSYMMT=0, and if NSYMMT=7, the solution will be based on the sparse matrix methods just as when NSYMMT=5. It is important to note that the above applies only when the Darcy-Weisbach equation is used. This special solution may be used for other problems for which no laminar flows occur, but with a slight increase in computation times, generally.

**NTRAND #**

This option applies only if ISIML=1, and allows control over whether the usual tables of output will be written, or a special direct access data file will be created, or both. In addition NTRAND can be used to determine the frequency, i.e. time increments at which, these files will be added to. The direct access file mentioned above can be used by the auxiliary program PRINT, described in Appendix E, to provide interactively most any desired portion of the solution, or complete tables at any time. NTRAND determines the type of output for a time-dependent solution as follows:

**NTRAND=0,** The solution tables for each time step will consist of the regular tables obtained with OUTPUT=1. A direct access file will not be created. (Default NTRAND = 0)

**NTRAND=1,** only a direct access file will be written that contains information for auxiliary program PRINT to provide the solution at any time requested.

**NTRAND=2,** both a direct access file and the creation of another file will occur which contains solution tables for each time step in the usual format of the two table form of output. Only the tables portion of the solution will be written to this extra file. The additional information usually printed to the output file besides the solution tables will still be directed to the output file given in response to the prompt requesting the name of the output file. The name of this extra file that will be created is TABLE.DAT.

In addition to the above uses NTRAND controls whether the results will be written to the files after each time step, or not. If a multiple N of 100 is added to the above values of NTRAND then USU-NETWK will complete N time step solutions between each solution whose results will be written in the usual manner of output. For example if NTRAND=200, only every second solution will have the usual table output given. If NTRAND=202 then every other time step will be added to the file TABLE.DAT, but the results for each time step will be written to the direct access file.

If NTRAND is minus 100, or minus 200 a print frequency for both the direct access file, and regular output tables can be established. Giving NTRAND values such as: -101, -102, -103, -201,-202, etc. tell USU-NETWK that the absolute multiple of 100 is the frequency that should be used to add to the direct access file, and the last digit determines the frequency of writing regular tables.

**NUMPIP**

This option must be set equal to 1 if the pipes are not given a number in the input data under the command PIPE or PIPE-. If NUMPIP=1, then the sequence order in which the pipe data occurs in the input file determines the pipe number, i.e. pipes are actually numbered sequentially starting with 1 for the first pipe. When NUMPIP=1, pipes are designated only by their two nodes in the input, but in the output are given their sequence numbers. Also for other data that requires a pipe number to be given, such as a pipe that contains a booster pump, the sequence number must be given. (Default NUMPIP=0, for giving pipe numbers in the input data.)

**NUNIT**

This parameter is used to distinguish problem using the International System of units, SI, from those based on the English units e.g. the ES system of units. It also allow for variations in units for diameters, and equivalent sand roughness coefficient with these two systems of units. The value of NUNIT has the following effect:

**NUNIT=0,** then the diameter and wall roughness, ε (for the Darcy-Weisbach Eq.) are in inches, and the length in feet (or 1000 feet if NPLENG=1). This is the default.

**NUNIT=1,** then the diameter, wall roughness, ε, and the length are in feet (or the
length in 1000 feet if NPLENG=1).

NUNIT=2, then the diameter, wall roughness, e, and the length are in meters (or the length in 1000 feet if NPLENG=1).

NUNIT=3, then the diameter and wall roughness, e, are in centimeters, and the length in meters (or 1000 meters).

NUNIT=4, then the diameter and wall roughness, e, are in millimeters, and the length in meters (or 1000 meters).

If the Hazen-Williams, or the Manning’s equation is used, then the above units designated for the wall roughness, e, are not applicable. This same control may alternatively be exercised by using the command UNITS.

OUTPU1

This option has three roles. The first is to specify the output table(s) that should be used for the solution. The second is to allow pipes and/or nodes to be identified by 8 character strings rather than numbers. The third is to control the precision, or formatting, i.e. the number of digits to the right of the decimal point of values printed in the output solution tables. The valid values for OUTPU1 are: 0, 1, 2, 11, 12, 13 (-11,-12,-13 in the case of the simulation version) and these values added to 1000, 2000, . . . 9000 to control the formatting in output tables. The first three are for the first role, the values 11, 12 & 13 denote whether "pipes & nodes", just "pipes", or just "nodes" will be identified by characters, respectively, and adding the thousands control formatting.

USE TO CONTROL OUTPUT SOLUTION TABLES

USU-NETWK can use either or both of two different forms to report the usual solution results. One form, the default forms, consists of two separate tables for the solution. The first table provides PIPE DATA, i.e. there is an entry for each pipe of the network in this table, and the second table provides NODE DATA, i.e. there is an entry for each node of the network in this table. Unless the option of using 8 characters to identify pipes, and nodes is used, these tables are restricted to 80 column in width. The second form of output consist of a single table that can used up to 132 columns across a line. This single table does, however, contain information not in the two table form of output, such as the friction factor, if the Darcy-Weisbach equation is utilized, and the coefficient and exponent in the exponential equation for defining the flowrate headloss relationship. For output control the option OUTPU1 indicates the following:

OUTPU1=0, then the single wider table is used for the output of the solution.

OUTPU1=1, (the default) then the two separate tables, one contain PIPE DATA and one containing NODE DATA are used for the output of the solution.

OUTPU1=2, then the solution is duplicated by writing the solution to both the single, as well as the two tables. For time-dependent solutions the two table form is required if special output or the direct access file is to be written. Also selection of string for pipe and node identification automatically selects the two table form of output as described below.

USE TO IDENTIFY PIPES AND NODES BY ALPHA-NUMERIC STRINGS

PIPEC and NODES may be identified by alpha-numeric strings of up to 8 characters in length, if desired, rather than using numbers (integers). To communicate this usage to USU-NETWK the option OUTPU1 in the $SPECIF list should be given a value of 11, 12 or 13. The following applies:

1. If OUTPU1=11 then both PIPES and NODES are identified by 8 character strings.
2. If OUTPU1=12 then only PIPES are identified by 8 character strings.
3. If OUTPU1=13 then only NODES are identified by 8 character strings.

When using the version of USU-NETWK that also allows time dependent solutions, then the above values can be preceded by a minus sign with the effect that in the solution tables the internal pipe and node numbers will be given in addition to the alpha-numeric identifiers. Thus, if OUTPU1=-11 then in the PIPE DATA table (given as part of the solution) the first three columns will contain: 1. the internal pipe number, 2. the upstream internal node number, 3. the downstream internal node number, 4. the pipe string identifier, 5. the upstream node string identifier, 6. the downstream node string identifier, and the other usual items of data. Each line in the NODE DATA table will contain the following columns: 1. the internal node number, 2. the node identifiers, and the other usual items of data. If OUTPU1=-12, then the internal pipe numbers will be given in the first column before the pipe identifiers. If OUTPU1=-13, then after the pipe
numbers in the PIPE DATA table two extra columns will be included for the internal upstream and downstream node numbers, and the NODE DATA table will contain the internal node numbers in addition to the node string identifiers. Presently the PC version of USU-NETWK accepts only positive values for the option OUTPU1.

Furthermore, if PIPES and/or NODES are identified by alpha-numeric strings then the input data file to USU-NETWK must adhere to the following special requirements:

1. NODES must be the first command that is given after the $SPECIF list of options. The exception is if only PIPES are given as character strings (OUTPU1=12); then the PIPES command may be first.

2. Since up to 8 characters, including blanks, can be used for the character identifiers it is necessary that this data be given in 8-character field widths. This requirement means that after the PIPES command that the first 8 positions of each input line are for the pipe identifier, and if NODES are also identified as character strings then positions 9 through 16 and 17 through 24, respectively, are reserved for the two nodes at the ends of this pipe. In identifying devices such as source PUMPS, RESERvoirs, and NOZZLes the character string identifier for the pipe or for the node must be in the first 8 columns of these lines of input data depending upon whether the option NODESP=0, or NODESP=1, respectively. If NODESP=0, and pipes are numbered (not identified by a character string), then the usual free format applies to the later integer data, and if NODESP=1, and nodes are numbered, then the usual free format applies to these lines, and the node number (integer) is the first value on the line. For devices such as VALVES (PRV's), and VALVC (check valves), that are always in pipes, the 8 character fields must be used for the pipe identifies for these input lines if OUTPU1=11 or =12, but these lines of input retain their usual free format input if pipes are numbered, i.e. OUTPU1=13. A / at the beginning of the next field terminates the list.

3. The PIPE-nodes command cannot be used to enter any data if either PIPES or NODES are identified by character strings. Rather the commands PIPES and NODES must be used.

4. Since any character string is accepted in the first column for the identifier, except a % which denotes that this line does not constitute a line of input, it is necessary to terminate data of a given type. This termination is done by the word STOP in columns 1 through 4. Thus each new command, including the RUN and END must be preceded by the word STOP. The exceptions are the first command, NODES, and if only pipes are identified by character strings, then the first command PIPES is not preceded by STOP.

5. It is not possible to use the "CHANGE" command to define alternate analyses by using PIPES or NODE character identifiers. If CHANGE data does change pipe diameters, lengths, or demands at nodes, then the internal number for the pipe or node must be given. Internal numbers are described later.

6. When giving lists of pipe identifiers under such commands as VALVC (check valves), or MINOR (minor losses), it is necessary that each succeeding identifier be within the next field of 8 character width. This list of pipe identifier must be terminated by a / in the first character position of the next 8 character field. If pipes are numbered in the usual manner (OUTPU1=13), then this list may be the usual free format for the integers it contains.

7. The options NSORTP and NODESO for having entries in the PIPE DATA and NODE DATA output tables listed in ascending order, now have no meaning. Both of these tables will retain the order of the input data under the PIPES and NODES commands, respectively. Furthermore, it is not possible to use the single table output (e.g. that given if OUTPU1=0, or =2).

8. If the option NODESP=0 (the default unless set to 1 in the $SPECIF list), then upstream nodes for pipes that connect sources of supply to the network do not have an upstream node. In using numbers for nodes these non-existent nodes are identified by a zero in the data after the pipes command. When using alpha numeric identifier for nodes ($OUTPU1=11 or 13) the same zero can be used provided it occurs in the first column of the 8 width field for this node identifier. The other alternative is to leave the full 8 character field for this non-existent node identifier blank. If NODESP=0, and pipes are numbered with a alpha numeric identifier ($OUTPU1=11 or 13), then the identifier for the pipe that connects a reservoir or a source pump to the network must be used in identifying pipes under the RESER and PUMPS commands.

9. The PIPE DATA output table will exceed 80 columns and, therefore, if the output is sent to a monitor with only 80 columns across a line directly by giving TTY for the output file name then each line will be broken with the last portion of the line displayed below the first.
portion of the line. Thus the table will be difficult to read. The same is true if output files are displayed with a TYPE, or COPY DOS command. (A terminal or PC with the capability of displaying 132 characters across a line can be set to this mode and show the tables properly.) When printing output files written by USUNETWK with abs(OUTPUT1)> 10 (but without thousands added) either wider paper than 8 1/2 inch paper must be used, or a font should be used that prints 16.5 characters per inch. This applies only if PIPES are designated by character strings. See the description of the auxiliary program PRNT to accomplish this.

10. Much of the extra output that can be requested by setting the option NPRINT larger than 0 will show pipes and nodes as internal integer numbers rather than their character strings. These internal numbers are established by the order in which the input data occurs after the PIPES and the NODES commands. These internal numbers can be displayed associated with the alpha-numeric identifiers by setting the option NPRINT=5 or larger, and as noted above can be shown in the solution tables by placing a minus in front of 11, 12 or 13 when using USUNETWK on larger computers.

11. When specifying number of pumps in series or parallel at stations with the SERIES or PARAL commands, the option NPSER1 will automatically be set to 0, and it is necessary that the data after these commands use the pump station numbers, e.g. the order of the data after the PUMPS and BOOSTer commands. Also it is necessary that the command BOOSTer be used rather than BPUMP for booster pumps.

Use of character strings does not change the manner in which USUNETWK internally keeps track of pipes, nodes, loops, etc. for the network. This internal book keeping is done with the internal numbers. These internal numbers are established by the order in which PIPES and NODES are first entered in the input data. The use of character strings is a convenience to the user who wishes to have a more easily understood mean for separating pipes and nodes in different parts of the distribution network. An alternative to the use of character strings is to use numbers starting with different hundreds, or possibly thousands, but this limits the number of zones, or pipes within a designated zone.

Example of input data files for a small network that identify: (1) both PIPES and NODES, (2) only PIPES, (3) only NODES with character strings and (4) the same as (1) except the option NODESP=0 of not having sources of supply numbered as nodes are given below.

Both PIPES and NODES are identified by character strings.

```bash
/* SPECIFIC NODESP=1, NFLQW=1, NGPUM=1,
 OUTPUT1=11, PEAKF=2.5 SEND
 NODES
 NODELN1 0 30/
 NODELN2 200/
 NODELN3 200/
 NODELN4 150/
 NODELN5 200/
 NODELN6 150/
 NODELN7 0/
 NODELN8 0/
 STOP
 PIPES
 PIPELN1 NODENL8 MODEL1 500 8 130
 PIPELN2 NODENL1 NODENL2 500/
 PIPELN3 NODENL2 NODENL3 500/
 PIPELN4 NODENL1 NODENL4 1200 6/
 PIPELN5 NODENL3 NODENL4 2000 6/
 PIPELN6 NODENL4 NODENL5 2500 6/
 PIPELN7 NODENL6 NODENL5 2500 8/
 PIPELN8 NODENL6 NODENL3 1500/
 PIPELN9 NODENL7 NODENL6 500/
 STOP
 RESER
 NODENL8 100
 NODENL7 120
 STOP
 VALVC
 PIPELN3 PIPELN7 PIPELN6 /
 STOP
 RUN
*/

Only PIPES are identified by character strings.

```bash
/* SPECIFIC NODESP=1, NFLQW=1, NGPUM=1,
 OUTPUT1=12, PEAKF=2.5 SEND
 NODES
 1 0 30/
 2 200/
 3 200/
 4 150/
 5 200/
 6 150/
 7 0/
 8 0/
 STOP
 PIPES
 PIPELN1 8 1 500 8 130
 PIPELN2 1 2 /
 PIPELN3 3 2 /
 PIPELN4 4 2 500 6/
 PIPELN5 3 4 2000 6/
 PIPELN6 4 5 2500 /
 PIPELN7 5 6 2500 8/
 PIPELN8 6 3 1500/
 PIPELN9 7 6 500/
 STOP
 RESER
 8 100
 7 120
 STOP
 VALVC
 PIPELN3 PIPELN7 PIPELN6 /
 STOP
 RUN
*/

Only NODES are identified by character strings.

```bash
/* SPECIFIC NODESP=1, NFLQW=1, NGPUM=1,
 ICOST=1, OUTPUT1=13, PEAKF=2.5 SEND
 NODES
 MODENL1 0 30/
 MODENL2 200/
 MODENL3 200/
 MODENL4 150/
 MODENL5 200/
 MODENL6 150/
 MODENL7 0/
 MODENL8 0/
 STOP
 PIPES
 1 MODENL8 MODENL1 500 8 130
 2 MODENL1 MODENL2 500/
 3 MODENL2 MODENL3 500/
 4 MODENL1 MODENL4 1200 6/
 5 MODENL3 MODENL4 2000 6/
 6 MODENL4 MODENL5 1500/
 7 MODENL6 MODENL5 2500 8/
 8 MODENL6 MODENL3 1500/
 9 MODENL7 MODENL6 500/
 STOP
 RESER
 MODENL8 100
 MODENL7 120
*/
```
STOP VALVC
3 7 6 /
STOP
INTEREST=.12
LIFE=80
RESERVOIRS
CAPI=3000
WATER
UNIT=3000, RESER, 1
UNIT=3000, RESER, 2
UNIT=12000, NODES, 1-5
OPERATING
CAPI=2000
UNIT=2000, PIPES, 1-4
ADDITIONAL
MAINT.
UNIT=4000, RESER
UNIT=2500, FLOWS
UNIT=2500, PIPES, 1-4
END

Pipes and Nodes identified by character strings
/* $SPECIFI
NFLOW=1, NPGPM=1, ICOST=1
OUTPU1=12, PEAKF=2.5 SEND

NODES
NODEL1 0 30/
NODEL2 200/
NODEL3 200 /
NODEL4 150 /
NODEL5 200 /
NODEL6 150 /
END

PIPES
PIPEL1 0
PIPEL2 NODEN2 NODEN3 NODEN4 NODEN5 NODEN6 NODEN7
PIPEL3 NODEN2 NODEN3 NODEN4 NODEN5 NODEN6 NODEN7
PIPEL4 NODEN2 NODEN3 NODEN4 NODEN5 NODEN6 NODEN7
PIPEL5 NODEN2 NODEN3 NODEN4 NODEN5 NODEN6 NODEN7
PIPEL6 NODEN2 NODEN3 NODEN4 NODEN5 NODEN6 NODEN7
PIPEL7 NODEN2 NODEN3 NODEN4 NODEN5 NODEN6 NODEN7
PIPEL8 NODEN2 NODEN3 NODEN4 NODEN5 NODEN6 NODEN7
PIPEL9 0

RESER
PIPELM1 100
PIPELM2 120
STOP
VALVC
PIPELM3 PIPELM7 PIPELM6 /
STOP
RUN
INTEREST=.12
LIFE=80
RESERVOIRS
CAPI=3000
WATER
UNIT=30000, RESER, 1
UNIT=45000, RESER, 2
UNIT=12000, NODES, 1-5
OPERATING
CAPI=20000
UNIT=20000, PIPES, 1-4
ADDITIONAL
MAINT.
UNIT=4000, RESER
UNIT=2500, FLOWS
UNIT=2500, PIPES, 1-4
END

USE TO CONTROL FORMATTING IN TWO TABLES FORM OR OUTPUT

It is possible to change the number of digits printed beyond the decimal for the following quantities when using the two table form of

data:
units digit (3) controls length format
10th digit (2) controls diameter format
100th digit (1) controls flowrate format

Table showing which parameters can have their printed formats altered using values of the option OUTPU1 equal to 1000 or greater.

<table>
<thead>
<tr>
<th>Thousands</th>
<th>Controls FORMAT (digits beyond decimal point) in position*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digit</td>
<td>PIPE DATA</td>
</tr>
<tr>
<td>1000+</td>
<td>length-3,diameter-2,flowrate-1</td>
</tr>
<tr>
<td>2000+</td>
<td>flowrate-1</td>
</tr>
<tr>
<td>3000+</td>
<td>diameter-2,flowrate-1</td>
</tr>
<tr>
<td>4000+</td>
<td>diameter-2</td>
</tr>
<tr>
<td>5000+</td>
<td>length-3</td>
</tr>
<tr>
<td>6000+</td>
<td>length-3,diameter-2</td>
</tr>
<tr>
<td>7000+</td>
<td>length-3,diameter-2</td>
</tr>
<tr>
<td>8000+</td>
<td>length-3,diameter-2</td>
</tr>
<tr>
<td>9000+</td>
<td>diameter-3</td>
</tr>
</tbody>
</table>

* OUTPU1 = x,xxx (Value from 1000 to less than 9999)
output in (a) the PIPE DATA: length, diameter, and pipe flowrate; and (b) the NODE DATA: first column of demand, elevation, and pressure. The control over these formats is by means of the option OUTPU1 as given in the tables below. In control over these formats is by means of the option OUTPU1 as given in the tables above. In understanding the information in these tables the option OUTPU1 is given a value between 1000 and less than 9999. Starting from the left side of this value the digits after the thousands are numbered from 1 to 3. The value give to these 3 individual digits specifies the number of digits that will be printed after the decimal point. The thousands denotes what will be controlled, i.e. 1000 indicates only PIPE DATA formats will be controlled; 2000 indicates only NODE DATA formats will be controlled, etc. as shown in the second table below. The values given to the digits numbered 1 to 3 control different parameters. The first (or No. 1) digit, or the hundreds digit, controls either the format of the flowrate in the PIPE DATA, or the format of the pressure in the NODE DATA. The second (or No. 2) digit, or the tens digit, controls either the format of the diameter in the PIPE DATA, or the format of the elevation in the NODE DATA output solution table. Finally the No. 3, or units digit, controls either the format of the length in the PIPE DATA, or the demand in the first column of the NODE DATA.

Anytime the value given to OUTPU1 is 1000 or greater, two things are communicated to USU-NETWK. First it should use the default two table form or output, and second that the number of digits printed after the decimal point of three parameters is specified by the values in the three numbered digits. Thus if it is desired that the length have two digits printed beyond the decimal point, and the diameter have zero decimal points, then OUTPU1=1202 should be included in the $SPECIF list of options. The first 2 after the 1 for the thousand simply keeps the standard 2 digits of precision for the pipe flowrate. The 1000 indicates the format of the parameters in the first column of the table below are being specified, i.e. the parameters in the PIPE DATA.

**PCHAR3**

This option is used to communicate if three pairs of points (flowrate, Q, and a corresponding head, H, as a pair), or more then three points are to be used to define pump characteristics. The operation of pumps is defined by passing a second degree polynomial through 3 pairs of values of Q and H. This polynomial understanding the information in these tables the option OUTPU1 is given a value between 100 and less than 9999. Starting from the left side of this value the digits after the thousands are numbered from 1 to 3. The value give to these 3 individual digits specifies the number of digits that will be printed after the decimal point. This becomes the mathematical description that replaces the "pump curve" and defines its operating characteristics. Three points are generally adequate to properly define the operation of a pump if for the solution requested it is operating near its normal capacity (where the pump achieves its maximum efficiency). However, if a pump is poorly suited to do the job needed in a network, it may be forced to operate with a flowrate, and head quite different from its maximum efficiency point. Under such conditions, or under vastly varying conditions at different times during a time-dependen solution, 3 pairs of points on a pump curve may not be adequate. By adding the option PCHAR3 = 0 to the $SPECIF list of options up to 10 pairs of values of Q versus H can be given to define pump characteristics. When more than 3 pairs of values are provided, USU-NETWK still fits a 2nd degree polynomial between 3 pairs of consecutive points, but the 3 points will be altered as the solution progresses so that the flow in the pipe containing the pump during the iterative solution processes will be bracketed by the 1st and last of these three points, unless the flowrate is less than or larger then the smallest Q and largest Q in the list. When giving more than three points they must be arranged in ascending order of magnitude of Q.

If PCHAR3 = 1 (the default), only three pairs of values for flowrate Q and pump head H can be given. The order of these points if then unimportant.

If a solution shows negative heads produced by some pumps, or fails to converge properly the problem might be eliminated by defining the pump characteristics with more than three points. When doing this it is even permissible to provide some heads that are associated with negative flowrates, as an extension of the pump characteristic curve to the left of the origin. This extension may be made artificially steep, but should be smooth enough to ensure that a 2nd degree polynomial fits the data without
wild behavior between the points. An alternative to using more than three points is to spread the points out over a wider range of flowrates. USU-NETWK assumes that the middle of the 3 points represents the normal operating conditions for the pump, and if the solution shows considerable deviation from this operation a message is written informing the user of this.

A utility program PUMCUM, which is described in Appendix F is designed to help you easily visualize how the pump data you give is interpreted by USU-NETWK in defining how much head a pump produces as a function of the flowrate passing through the pump. Should convergence to a solution not occur, or negative flows occur in pipes containing pumps, then you might well be specifying pumps that are incompabible in their combined operation. Such situations should be avoided in practice since a pump under these conditions will only consume energy but not contribute to the flow through the network.

PEAKF (FLOWFC)

All external demands are multiplied by this value. Giving this option different values for consecutive solutions allows the total demand level to be changed without preparing new input data. Only one value needs to be changed. This option might also be used to convert from any units which the demand may be given in, into cfs, etc. For example, if demands are in ac-ft per day PEAKF should equal 1.9835 to convert to cfs with NFLOW=-0. Likewise giving PEAKF=1.55 is equivalent to setting NFLOW=2. Also if the convention of the data is minus for demands out from the network (opposite to the convention of USU-NETWK) this can be allowed by setting PEAKF=-1. (See the commands PEAKF, andDFRAC as well as command under CHANGES for other possible method for changing demands from those in the input data file on a more selective basis.)

PLOTH #

Determines the height (in the y direction) of the plot on the Calcomp Plotter in inches. The maximum and minimum y-coordinates will be found, and the x and y coordinates scaled so that this specified height is achieved. A menu will appear just before the plot is made that allows you to change this value. Default is PLOTH=20 inches.) See Appendix C for more information.

PRESMA

If a value is given to PRESMA other than zero, then the pressure at each node will be examined, and any pressure greater than this value will be flagged. All such flagged values will be written to the output file before the summary table(s) which give the solution results. The pressure given by PRESMA is in psi, when using ES units, and Pascals when using SI units.

PRESMI

If a value is given to PRESMI other than zero, then the pressure at each node will be examined, and any pressure less that this value will be flagged. All such flagged values will be written to the output file before the summary table(s) which give the solution results. The pressure given by PRESMI is in psi, when using ES units, and Pascals when using SI units.

PRVLOS

Under special conditions the upstream pressure may drop below the pressure setting of PRV's. Should this condition occur the PRV becomes inoperative, but acts like a minor loss device. PRVLOS is the minor loss coefficient, K, in the equation \( H_L = K(V^2/2g) \) for such minor losses. Default is PRVLOS=10.

REYNUM

If the Darcy-Weisbach equation is used to determine head losses as a function of flowrate and if minor losses are present, the equivalent length of pipe to duplicate the minor loss device is computed on the basis of a specified Reynolds Number. This equivalent length is not adjusted during the solution process if LENORG=1. REYNUM is the Reynolds for which the equivalent length is first computed. If LENORG=0, the value given to REYNUM will have no effect on the final solution results. (Default REYNUM=500,000.)
TVSUM

This option contains the value of the absolute sum of changes in flowrates between consecutive iterations at which the flow in pipes containing PRV's is checked to see if the flow direction is reserved. If a negative flow occurs in any such pipe the PRV, etc. is shut-off. No check is made to determine if a PRV, etc. should shut-off if this sum of changes is greater than TVSUM. The default value is $2 + \frac{\text{No. pipes}}{10}$, but not more than 40.

TVSUM1 #

In order to prevent an occurrence in which a PRV, or check valve, continues to cycle open and then close between consecutive iterations, the value of TVSUM is reduced each time a device changes its condition of operation. TVSUM1 is the factor by which TVSUM is multiplied if a device shuts-off. (Default $\text{TVSUM1}=0.85$)

TVSUM2 #

This option serves the same purpose as TVSUM1, except it is the multiplying factor when the device opens again after being closed. (Default $\text{TVSUM2}=0.85$)

VELMAX

If a value other than zero is given to VELMAX, then the velocity in each pipe will be examined. Any velocity whose magnitude exceeds the value of VELMAX in feet per second, when using ES units, or in m/s when using SI units, will be flagged. All such flagged values will be written, and identified in the output solution file between the summary tables which give the PIPE DATA results and the NODE DATA results. These velocities will be examined only if the two table form of output (OUTPU1=1 or 2) is selected.

VELMIN

If a value other than zero is given to VELMIN, then the velocity in each pipe will be examined. Any velocity whose magnitude is less than the value of VELMIN in feet per second, when using ES units, or in m/s when using SI units, will be flagged. All such flagged values will be written, and identified in the output solution file between the summary tables which give the PIPE DATA results and the NODE DATA results. These velocities will be examined only if the two table form of output (OUTPU1=1 or 2) is selected.

VISC

The kinematic viscosity of the fluid in ft$^2$/sec or m$^2$/s depending respectively whether ES or SI units are used. The default value is 0.00001217 ft$^2$/sec (water at 60°F), if ES units are used or 0.000001131 m$^2$/sec (water at 15.6°C) in SI units are used.

NOTES RELATED TO OPTIONS
DETAILED STEADY-STATE DATA REQUIREMENTS

Commands that enter data

Numeric data which describe the network in detail are introduced by a command, or descriptive name, that precedes the numeric data of a specific type. The first 5 characters of these commands must be identical to those described in this portion of the manual. Furthermore, these commands must be given in upper case letters, so when making up the input data files, etc. make sure that the "caps Lock" is on. The individual commands are (listed in alphabetical order): BOOSTer, BPVAIne, CHECK input, DESIGN, DFRAC, DHEAD, END, ERROR, ITERAtion, LIPE, NLPE, LNODE, NLNODE, MINOR, NODES, NOZZLes, PARALlel, PEAKF, PIPES, PIPE-nodes, PRINT, PUMPs, QGIVE, RESERvoir, ROTATional, RUN, SERIES, SETPressure, UNITS, VALVcheck, VALVE (prv), VISCOsity, WEIGHT. Characters beyond the fifth character are not examined by the computer and, therefore, command names may be added to if additional characters help clarify the name. There must be at least one line containing data after the command if this name implies data follows. The three commands that do not call for data are: CHECK, RUN and END. Generally all the data of this type will follow a single command. However, as many of the same commands as desired may be included in the input data file. The exceptions to this rule are that the data given on the last of the following lines will be used: ITERA, UNITS, ERROR, VISCO and WEIGHT, because these commands introduce a single value. These same values can be introduced by options in the $SPECIF list. RUN and END are exceptions also since they indicate that the data for describing the network has all been given, and no more commands of those listed above are read after an END or RUN.

Use of % in column 1

In doing several different analyses of a given network in which some components of its description are changed it may be desirable to keep the original data available. Two methods of keeping data from previous analyses in the file are: (1) Use an editor to move these lines of the file to after the last END of the data file, and (2) placing a % in the first column in front of the lines.

The latter method is most convenient, especially if subsequent analyses may wish to utilize this data again. NETWK will ignore any line in the input file that contains a % in column #1. USU-NETWK also will ignore any information that occurs after a / anywhere in a line unless this / is followed immediately by an R in which case the text thereafter will be taken as a remark as explained in the next paragraph.

Use of "remarks" at the end of input lines
(Not implemented in Steady-State PC program)

Remarks can occur on any data line after a /. If these remarks occur on lines after either the PIPES, NODES, or PIPE- commands they will be printed in the solution tables at the end of these same pipe or node lines, provided that an R follows the /. In other words to tell USU-NETWK that you wish to have these remarks retain and printed in the solution tables an R must follow the / without any blanks between. Thus if a line after the PIPES command is: 12 4 7 2400. 20./R THIS PIPE NEEDS TO BE REPLACED the remark "THIS PIPE NEEDS TO BE REPLACED" will appear at the end of the line for pipe 12 in the solution table for PIPE DATA. Likewise, any remark on a NODES line preceded by an /R after the required data will be written in the output file after the line for this node in the NODE DATA solution table. Similarly remarks after a /R on a line after the PIPE- command will appear on the appropriate line in the table for PIPE DATA. Remarks are restricted to 32 (or 36 depending on the computer used) characters after the /R, and if they occur on lines that are not truncated by a /, then the /R must follow the last number without any blanks, or a comma, between. These remarks can be used only in connection with the two table form of output, that is the option OUTPUT=1, or OUTPUT=2 must be in effect. The ability to retain, and write remarks in the solution tables does not exist in the PC steady-state version of USU-NETWK.

General requirements, free format, and truncation of lines

Every data file must contain a PIPES and NODES command, or the command PIPE- with appropriate data following them, since every
network has pipes and nodes. The other commands are optional, depending upon the configuration of the network, and the type of analysis being requested. The data after all commands is free of any format requirements, unless special options are included in the $SPECIF which alter this. If OUTPU1>11 an exception occurs so that data fields 8 characters in width are reserved for the pipe and/or nodes that are identified by a string of characters rather than by integers. Another common exception is if NETPLT=13 or =14, indicating that the data is formatted according to that produced by the preprocessor PIPEINPT. The free format data can separate individual items in the list with a blank or more between them, or a comma, or a comma followed by one or more blanks.

It is not necessary to provide all of the items call for under the commands. The rule that USU-NETWK follows is that if a line of input is terminated with a / before all of the items called for under the command are given, then the missing items should equal the value last given to that particular item. In other words a / in a line of input causes two things to occur: 1. the reading of this line is stopped, and 2. items called for are given values equal to those last given in the input file. Generally it is not necessary to terminate a shorten list with a /, since for most of the input USU-NETWK expect all called for items to be on a single line of input. After reading 20 consecutive blanks termination in reading a line occurs as if a / existed. However, for some lists, such as the list of pipes containing check valves, several lines may be read, and these lines must be terminated with a /.

The exception to the rule that data at the end of a list can be omitted by giving a /, is the data after the command PIPE-. Under this command decimal are required to be given with all real values, and decimal points are not given with integer values such as pipe and node numbers. Here data in the list can be left out from within the line. With the other data after a command, the default order of item under the command are arrange so that generally the most frequently repeated items occur at the end. For example after PIPES the last item in the list is the roughness coefficient. If, therefore, all pipe in the network have the same roughness coefficient, it need only be given on the first line after PIPES. Then if a subsequent analysis is to use a different roughness coefficient, only one value need be changed in the input data file to define this new problem. Diameter is the second from the last item. Many pipe will have the same diameter, and the above rule indicates that they need not be given provided the diameter last given is still correct. The first item in the list, the pipe number, is different for each pipe, and therefore it can never be truncated.

List of commands and what they enter

The type, and order of data, which follow each of the commands are described in the following paragraphs. The items described by the numbers 1, 2 ... under each of the heading, which are the commands, established the default order of data under this command. In the case of the simulation version of USU-NETWK this default order can be changed by the options whose names begin with COR. The descriptions of these commands are given below in alphabetical order to make it easier to find a given command, rather than according to their frequency of usage. See the first paragraph in this major section for a list of commands that are available.

BOOSTer pump or BPUMP

The input under the command BOOST or BPUMP is identical to that described under PUMPS with the exception that nodes cannot be used to identify booster pumps even if NODESP=1, and no elevation of the source of supply exists. The direction of flow through the pump must be specified by the "upstream" and "downstream" nodes in the pipe data for this pipe as entered under the command PIPES or PIPE-. The input consists of use (a), (b), or (c) below.

(a) using three (or more points, see PCHAR3) points to define the pump's characteristics,
1. The pipe number (which is the pipe identifier if OUTPU1=11, or OUTPU1=12) that contains the booster pump,
2. The flowrate $Q_1$,
3. The head, $H_1$, produced by the pump corresponding to $Q_1$,
4. The flowrate, $Q_2$, corresponding to most efficient operation of pump
5. The head, $H_2$, produced by the pump corresponding to $Q_2$,
6. The flowrate $Q_3$,
7. The head, $H_3$, produced by the pump corresponding to $Q_3$.
This 7th item is terminated with a / or a O as an extra value for the water surface elevation.
(b) giving power the pump supplies to the fluid and its normal capacity,
   1. The pipe number (or identifier) that contains the booster pump,
   2. The power of the pump times its efficiency in units of horsepower when using ES units, and in units of KiloWatts when using SI units,
   3. The normal capacity of the pump, i.e. its flowrate at peak efficiency.
This third item should be terminated with a /
For additional explanation about the input data after the BOOST or BPUMP command see the description under PUMPS.

(c) giving coefficients to 2nd degree polynomial. (See explanation under command PUMPS for this input.)

Example of booster pump data
BOOST
20 1.2 400. 1.0 420. .8 430.0

BPVAL (back pressure valve)

A back pressure valve, BPV, is used to maintain a constant pressure at the upstream side of it, thus maintaining a pressure in the upstream portion of the network that would become too small if the BPV were not installed. A BPV accomplishes this by discharging only sufficient flowrate through the pipe in which it is installed to maintain the desired upstream pressure.

The maintenance of a constant upstream pressure may be considered the normal mode of operation of a back pressure valve. Obviously, should the pressure on the upstream side of the BPV drop below its pressure setting, the value can only shut off the flow, but cannot maintain the pressure setting. A shut-off BPV is a secondary mode of operation. Finally, as a third mode of operation, some back pressure valves permit reversed flow from the downstream to the upstream sides of the valve. In this mode the BPV becomes effectively inoperative, but since it is an obstruction in the line it becomes a minor loss. USU-NETWK determines the mode of operation and does what is necessary to accommodate how a BPV actually works in practice.

Data for a back pressure valve consists of:
1. The pipe number (integer) that contains the back pressure valve.
2. The distance between the back pressure valve and its pipe's downstream node in feet or meters depending upon whether ES or SI units are used, respectively.
3. The pressure setting of the BPV, expressed in elevation of the hydraulic grade line, HGL, immediately upstream from the valve. The HGL elevation is in feet or meters depending upon whether ES or SI units are used respectively. If the option NPRPRV=1 then pressure is given instead of the elevation of the HGL, but this HGL is computed on the basis of the pipe being a straight line between its end nodes.

Example of back pressure valve data
BPVAL
5 500 340
15 300 360

CHECK

This command does not have any lines of data following it. If the command CHECK is in the input data file, it informs USU-NETWK that it should not attempt to obtain a solution, but rather check the data to find any error. This command may be used when you first try "running" a new network problem, and you know from your past experience that you will undoubtedly have made some mistakes. The command END must be the last of the detailed data requirements when a CHECK is given. RUN and CHECK imply opposites.

DESIGN

The command DESIGN is an alternative to setting DESIGN=1 in the $SPECIF list of options. The command DESIGN must appear before the NODES, or PIPES command or the PIPE-nodes command since USU-NETWK must be informed that pressure are provided following the nodal elevations and it is not to terminate when a zero is given for a pipe diameter. DESIGN indicates to USU-NETWK that a special design solution is requested in which it is to solve for NJ pipe diameters or roughness coefficients, etc. as defined under the option DESIGN. (NJ=the number of junctions in the network). Values after the command DESIGN can be: 0, 1, 2, 3, 4 or 5. The input data accompanying this special design solution requested with DESIGN equal to 1 or 3 must (a) give NJ pipe diameters equal to zero, and (b) provide the pressures (or HGL elevations), as well as, the demands at all nodes of the network.
DFRAC

The command DFRAC has one value that follows it. This value is a new peaking factor, or multiplier of demands. Any nodes entered in the input list after DFRAC will have their demands multiplied by the given value. Thus the command DFRAC allows the value given to PEAKF (or FLOWFC) in the $SPECIF list to be altered repeatedly through the input data. An additional NODES command (or the PIPE-nodes command) would follow each occurrence of the command DFRAC, unless DFRAC is used as a replacement of PEAKF, and occurs before the NODES or PIPE-command.

DHEAD

This command is used to introduce a differential head device which allows a solution to situations in which both demands and pressures are specified at some nodes in the network. This command is useful to size network components, such as heads and capacities of pumps, setting for pressure reduction valve, or determine pipe diameters, i.e. design network components.

A differential head device may create a positive head in a pipe line such as a booster pump, or a negative head, such as a pressure reduction valve, in addition to the fluid frictional head loss. There are two types of differential head devices, as far as USU-NETWK is concerned. The first type, or type # 1, is one for which the amount of the differential head is specified. The second, or type # 2 is referred to as a design differential head device for which the amount of the differential head is unknown. The magnitude of this differential head is to be determined such that the pressure, or HGL elevation, at a specified node equals some given amount. The demand is also specified at this node. The command DHEAD enters both types # 1 and type #2 differential head devices. The device is type # 1 if input items 3 and 4 below are given zero values. For a type # 2 device an additional energy equation will be written between the node of the specified HGL elevation (or pressure) and one of the sources of supply. You must give both the source of supply as well as the HGL (or pressure) at the node where this is to be specified. That is, a differential head device of type # 2 allows both the head and demand at a designated node to be specified. In order to satisfy this dual condition, an additional unknown is introduced in a pipe somewhere in the network, which is the differential head device.

The type # 2 differential head device is very useful in designing a network in that it allows both the pressure (or HGL elevation), as well as the demand, to be specified at a given node. In order to specify both of these, something in the network must increase or decrease the head, such as a pump, or a pressure reduction valve. However, the amount of this increase or decrease in head is unknown. The magnitude of the differential head will be provided as part of the solution. From this magnitude an appropriate device can be selected to provide the incremental head indicated. If the magnitude is positive, a booster pump is called for, or if the frictional loss is greater in magnitude than the positive differential head, a larger size pipe can be used. If the magnitude is negative a pressure reduction valve can be inserted in the pipe or a smaller size pipe used to give the same frictional head loss as the given size pipe, plus the differential head loss. If the option NEQUDI=1, then USU-NETWK provides a table of such equivalent pipe sizes, whenever an equivalent pipe can be used. Thus, in addition to providing design information for pumps, or head reduction devices, the type # 2 differential head device can be used to size pipes.

The user must use sound judgement in specifying differential head devices of type # 2, because a problem can easily be specified for which no solution exists. For example, it is obvious that an HGL elevation, or pressure, at a node may be specified too small for even an infinite resistance to flow in a pipe, to meet, especially if this pipe is far removed from the node where the pressure is specified. The flow simply comes to the node through other paths, and arrives with too large a pressure.

There are many other possible situations for which solutions are not possible. For instance, the pressure at the node downstream from a single supply source for a network cannot be specified unless the differential head is inserted in that pipe, since this is already uniquely determined by the reservoirs water surface elevation, the total demand which must come through this pipe, and the pipe headloss properties. Likewise, it is not possible to specify pressures at two nodes from this node, if no other pipes join at this node, without these pipes containing a differential head device. USU-NETWK does not attempt to determine all the situations that you might specify for which solutions are not possible. This burden is left to you. However, if convergence to a
solution does not occur it is likely that you have an impossible situation specified. The user is, therefore, wise to limit the number of iterations allowed by using ITER or MAX as options. Before using a large number of differential head devices in a single solution the user should have some knowledge about the network's performance.

Input data consist of:
1. The pipe number containing the differential head device,
2. The amount of this differential head. In the case of type # 1 devices, this head (in feet or meters depending upon whether ES or SI units are used, respectively) will be added to the frictional head loss in the pipe. If type # 2 devices are being introduced this is only an estimate of the differential head loss. A head loss is negative and an increase in head, or a positive differential head, increases the HGL elevation from the upstream node of the pipe toward the downstream node of the pipe. Obviously, upstream and downstream nodes for pipes with DHEAD devices in them are important, as are specified flow directions for pipes containing booster pumps, PRV's, check valves, and back pressure valves.
3. The node number where the HGL elevation (or pressure) is specified for a type # 2 device. If this value is preceded by a minus sign (i.e. is negative), then USU-NETWK assumes that pressure is being given in psi for ES units, and in Pascals (N/m²) for SI units, rather than as a HGL elevation. A zero for this third item indicates to USU-NETWK that a type # 1 device is intended and, therefore, it does not set up the additional equation needed to solve for the differential head for a type # 2 device.
4. The designation of a source of supply for type # 2 devices (i.e. a source pump, or reservoir) that USU-NETWK should use to form an energy loop to, from the node given in item # 3. This designation is a node number assigned to the source. Otherwise it is the pipe number that connects the source to the network. In order to save computations this source should be in the proximity of the node of item # 3, since it is used to obtain an additional energy equation to allow for a differential head in the pipe of item # 1 to be solved for. It should also not be a source that has little influence on the flow in the pipe of item # 1 or the pressure at the node of item # 3. If this is to be a type # 1 differential head device, the value should be zero.
5. The elevation of the HGL in feet, or meters, depending upon whether ES or SI units are used, respectively, at the node in item # 3. If item # 3 is negative this value will be taken as the pressure (in psi or Pascals) and USU-NETWK converts the given value to an HGL elevation by dividing the pressure (appropriately converted to psf) by the specific weight of the fluid to get pressure head and adds this amount to the elevation of the node. If type # 1 device is intended, this item is given a zero, or truncated by a /.

Example of input under the DHEAD command
DHEAD
5 -10.5 0/10 5. 4 11 165.

END

The command END denotes the end of the detailed data, and upon encountering END, USU-NETWK stops looking for additional commands. Depending upon options that may be set there can be data after the END.

If NPERCT=1 in the $SPECIF list, then the fraction of the total demand coming from each source of supply is required as further data after END.

Also if end pipes for pseudo loops are specified as denoted by setting LOOPSE=1 in the $SPECIF list, then this data also follows the END. If all external flow are specified to a network, i.e. no reservoirs, or source pumps exit then it is also necessary to supply a node number and an HGL elevation as a starting value for the HGL elevations and pressures of nodes to be computed. These two values also follow the end for such special networks.

ERROR

This command introduces a single value after it, which is the error parameter that determines when the Newton iterative solution process will be terminated. When the absolute sum of the changes in flowrates is less than this amount, the iteration stops. Use of the command ERROR is an alternative method to setting ERROR= to this value in the $SPECIF list of options.

Example of the use of ERROR
ERROR .0005
FORMAt

If the option NETPLT=13, or = 14 then USU-NETWK is told that the input data is formatted. The default formatting for this input data, that is expected ordinarily, is that written by the preprocessor PIPEINPT. For input under the command PIPES this expected format is as follows:

(3I4,F8.2,F10.2) if LENGON=0
(3I4,F8.1,F8.2,F10.2) if LENGON=1

and if NUMPIP=1 then the PIPES format is:
(2I4,F8.2,F10.2) if LENGON=0
(2I4,F8.1,F8.2,F10.2) if LENGON=1

The default formatting for input under other commands such as NODES, RESER, PUMPS, etc. that have one integer followed by real variables is:

(I5,F10.2)

The command FORMAT allows any other format to be given prior to reading data that is in columns specified by this given format. The given format information must begin with a left parenthesis, (, and end with a right parenthesis, ), and contain the information as it would in a FORTRAN FORMAT statement. The length of this format information is restricted to 48 characters or less. For example if the data is formatted that contains the items expected under the PIPES command and is in the following columns: (1) diameter 1-10, (2) length 11-20, (3) roughness coefficient 21-30, (4) pipe number 31-35, (5) upstream node number 36-40, and (6) downstream node number 41-45, the following two lines of input must precede the PIPES command for USU-NETWK to properly read this input:

FORMA
(T31,3I5,T1,3F10.2)

The FORMAT command followed by the proper format information can precede any other command, and the given format information will apply until it is changed by another FORMAT command. For the given format to be utilized it is necessary that the option NETPLT also be set to 13 or 14, informing USU-NETWK that the input is in given columns, and FORTRAN FORMATTED input is to be utilized rather than free format information. Otherwise the format information will be ignored and the input will be read as if it is free format. When using formatted input it is not possible to truncate items from the ends of input lines, as can be done when using free format input, however.

GAMMA

Has same action as WEIGT. (see WEIGT)

ITERAtion

This command is an alternative to using ITERA or MAX in the $SPECIF list of options, and as such it sets the maximum number of Newton iterations that will be allowed in obtaining the solution. If not given in either place, the default value of 8 or 15 will be assumed.

Example of the use of ITERA
ITERA
10

LPIPE,NLPIPE,LNODE,NLNODE

These four commands are listed together because they are all designed to permit the user to not have all pipes or nodes of the network appear in the two tables form of output. Generally it will be desired to have all pipes and nodes listed in the final solution tables. However, for large networks especially if the analyses are later ones of a series, or steps in time dependent solutions it may be desirable to not have all pipes, or nodes appear in the solution tables. The commands LPIPE (standing for list pipe), and NLPIPE (standing for not list pipe) allow the user to either give a list of pipe numbers that should be listed in the final pipe data table from the two table form of output, or give a list of pipe numbers that should not be contained in this final pipe data table, respectively. The pipe numbers can be listed as individual values separated by a blank or comma, or a range of pipes can be given by including a - between the beginning and ending pipe numbers of the consecutive series, or any combination of these. Pipe numbers given in this list that do not actually exist in the network will be ignored. Likewise the commands LNODE (standing for list node) and NLNODE (standing for not list node) allow for specifying node numbers to include, or exclude from the node data table from the two table form of output, respectively. If NODESP=1, such that the sources of supply are numbered as nodes, then
these extra nodes will always appear in the node data table and cannot be deleted with the command NLNODE. Furthermore, when using the NODESP=1 option it is necessary that these nodes occur in the input list of nodes after all real nodes. If it is desired that sources of supply not be listed in the output table, then the option NODESP=0 should be used. The program anticipates that you will either use LPIPE or NLPIPE, but not both. Attempts to use both commands will result in unpredictable results. Likewise either LNODE or NLNODE should be used but not both.

Example of using LPIPE & LNODE
LPIPE
1 3-7 30-50 140/
LNODE
4-7 24 40 50-70/

MINOR

The command MINOR provides the loss coefficient for devices, such as possibly globe valves, that produce extra losses in a pipe in addition to fluid frictional losses. Devices that cause small losses such as large radius bends, gate valves, etc. are handled as part of the frictional loss in a pipe, generally. In fact, the common practice is to use "extra" losses such as those introduced by MINOR sparingly in defining a network. Often the magnitude of loss coefficients are "guessed at", and consequently an equally satisfactory procedure is to accommodate extra "minor losses" as part of the pipe wall roughness coefficient. The following two items occur on each line after the command MINOR:

1. The pipe number which contains the minor loss device, and
2. The coefficient K in the minor loss equation,
   \[ h_L = K \left( \frac{V^2}{2g} \right). \]

Example of the command MINOR
MINOR
2 2.5
18 .5

NODES

The command NODES supplies information about the demands (consumptions or if negative inflows) that are assumed to take place at the junctions of the network, and the elevations of the pipes here so that pressure and pressure heads, etc. can be computed. These two items of information are not contained in the items after the PIPES command, and therefore constitute additional information not logically entered under the pipe oriented form of input. In addition if x and y coordinates are to be given for plotting of the network, or used to define the layout of the network from which pipe lengths are computed they are also items that are supplied to USUNETWK with the NODES command. Furthermore, with the option DESIGN=1 or DESIGN=3 HGL elevations are given, and these become an item entered with the NODES command. Also "data to check the network" might be given with the option NETCHK=2, and these come after the NODES command. Therefore, the list of items under the NODES command will be expanded upon depending upon the options in the $SPECIF list. The first three items listed below are always entered with the NODES command, and the others dependent upon whether an option call of it (them) or not.

1. The node number (an integer unless OUTPUT=11 or 13, in which case an 8 character identifier replaces the number) for which this data applies. The node numbering does not need to contain all numbers from 1 to the total number of nodes in the network, or be in consecutive order. Depending upon the dimensions of the program these integer node identifiers must be less than some limit such as 5000, 10000 or 15000.

2. The external flow, or demand, at this node. If inflow occurs at this node, the demand is a negative value, and outflows are positive. The units of this demand must be those designated with the option NFLOW. If the option NODESP=1 in the $SPECIF list, such that source pumps and reservoirs are identified as nodes in the network the demands given for these "artificial" nodes must be zero. In other words you do not provide an estimate of the flowrate from supply pumps, or reservoir, or into reservoirs as a demand. Flowrates from sources of supply are part of the solution. Therefore, a demand is considered an external flow from (to) the network. (If you want to supply estimates of flowrates from sources see the option LOOPSE.)

3. The elevation of the node with units of feet or meters, depending upon whether ES or SI units are used, respectively. If NODESP=1, then the elevation given for sources should be the ground elevation. The water surface elevations for reservoirs, and the sump elevations for source
pumps are given under the commands RESER and PUMPS. The default of NODESP=0 does not provide an entry in the NODES DATA table when giving the solution for sources of supply since they are not numbered as nodes, and no ground elevation is needed then.

Example of node data without options set that require more data after the NODES command

NODES
1 1.2 500.
2 .8/
5 1.1 520.
8 0./

4. The next possible item after the NODES command is the elevation of the HGL elevation (or pressure head, or pressure depending upon IHGL), if and only if the option DESIGN=1 or DESIGN=3 in the $SPECIF list. If DESIGN=0, but other options call for more information under the NODES command, the items listed below move into this fourth position.

5. If either the option LENGON=0, or 0 < NETPLT < 13, which indicate that pairs of x and y coordinates are going to be given, then the x coordinates in feet, or meter, respectively depending upon ES or SI units are used, is given as extra data after the NODES command.

6. The y coordinate of the node corresponding with the x coordinate given as item # 5.

7. If the option NETCHK is given in the $SPECIF list then the pipe numbers that join at this node are includes as this item of data.

8. Remarks may occur at the end of any line of input if /R precedes this remark. The R after the / tells USU-NETWK that you want the remark recorded. Otherwise any comments you place on lines of the input file after a / will be ignored. Remarks are restricted to 32 (or possibly 36) characters. Similar remarks are permitted after the PIPES and the PIPE- commands. This input is listed as item 8. However, regardless of the options it is always the last item if given. For example if a line under NODES were as follows: 8 240. 220./R NODE AT INTERSECTION #10 then in the NODE DATA table node 8 will have NODE AT INTERSECTION #10 at the end of its line.

Example of data after NODES with the option NETCHK=2, but DESIGN=0

NODES
1 1.2 500. -6 2 4 7/
2 .8 500 -2 8 10/
5 1.1 520 3 6/
8 0. 520 7 6 -3/

Example of data after NODES with the option LENGON=0 (The last two items are the x & y coordinates)

NODES
1 1.2 500. 0. 1000.
2 .8 500 1200. 500.
5 1.1 520 1000. -400.
8 0. 520 1800. -300.

Examples of data after NODES with the option LENGON=0 & DESIGN=1

NODES
1 1.2 500 620 0 1000
2 8 500 618 0 1200 500

NOZZL es

The command NOZZL introduces data describing a sprinkler nozzle, or orifice at the end of a pipe. Besides the more obvious uses of this command, it might be used to determine how much flow the network might be able to provide at a fire hydrant if it were fully opened. One would need to know the hydrants opening size, and its discharge coefficient. Data after the NOZZL command consists of:

1. The pipe number (integer) at whose end the nozzle exits. Nozzles exit at the ends of pipes just like sources of supply do. One might view a nozzle as a negative source, but with different characteristics. Therefore, if NODESP=0 the ends of pipes where nozzles exist are not given a node number and after the PIPES command this end would be denoted by a 0. If NODESP=1, then the nozzle ends of pipes must be given a node just as the sources of supply are given a node. Under this option this item is the node number given to the nozzle end of the pipe.

2. The diameter of the nozzle, or orifice in the same units that are used for pipe diameters. This diameter is assumed to be short in length, i.e. no frictional loss is computed for a pipe of some length with this diameter.

3. The discharge coefficient of the nozzle or the orifice. For an orifice this will be
approximately 0.6.

4. The elevation of the nozzle or orifice.
   (Given only if NODESP=0)

5. If desired, an estimate of the HGL upstream from the nozzle may be given. If not given the starting HGL elevation is generated internally in USU-NETWK. If this item is not given, then # 4 is terminated by a /. Likewise, if other entries are as previously given they may be omitted by ending with a /.

Example of nozzle data with NODESP=0
NOZZLES
8 2 .58 100./
12 1.5 .62 80. 120.

If NODESP=1 (i.e. Nozzles numbered as nodes)
If node numbers are used to identify nozzles, then the input item no. 4 above will be given on the NODES line for this node, and consequently item # 5 above replaces item #4, but might be deleted by ending item #3 with a /.

Example of nozzle data with NODESP=1
NOZZLES
15 2./
18 2.3 .56/

PARALLEL

It is possible to specify that any number of pumps of the same curve characteristics as given by the PUMPS of BOOST command are operating in parallel (or in series) at a station. Two items of data are needed to specify pumps in parallel, and are:

1. The designation of the pumping station according to whether NPSERI=0, 1 or 2 by pump number, pipe number, or node number (if a source pump station and NODESP=1).

2. The number of pumps operating in parallel at this station.

Example of the PARALLEL command
PARAL
3 2
4 1

PEAKF

This command does not replace the option PEAKF, but could be used to do so. After the command PEAKF there are two lines of data that constitute the needed information. The first line is the peaking factor and the second is the list of node numbers for which this peaking factor applies. The command PEAKF thus accomplishes the same as the option PEAKF (FLOWFC) only if all nodes are listed such as 1-320, but allows greater flexibility in that different groups of nodes can have the given demands multiplied by individual factors. Any number of groups of two lines can occur after the command. The first of each pair consists of a single real value, the multiplier or peaking factor. The second of the pair, containing the node numbers, may consist of single integer values separated by blank(s) or a comma, or by a - (minor sign) between two integers to denote the range from the first through the last node. If the - is used no blank spaces can occur between the two integers. The second line must be terminated with a /

The command PEAKF must be entered after the NODES command (or PIPE-command) used to enter the nodes for which it gives peaking factors. The command PEAKF can be used in connection with PEAKF in the specification list $SPECIF if desired. When PEAKF appears in both places all given demands are multiplied by the value given to PEAKF in the $SPECIF list first. Thereafter, the new resulting demand is again multiplied by the value entered for the designated nodes through the command PEAKF. The same node number may appear under more than 1 group of two lines under the PEAKF with the effect of having the multiplication accumulative.

Example of using the command PEAKF
PEAKF
1.5
1 3 10-16 20 31-45.
.8
2 22/

PIPES

The command PIPES enters data for each pipe in the network as well as its nodes, length, diameter and roughness coefficient and this data describes the systems connectivity and physical characteristics. The items under the pipes command consist of:

1. The number (integer) associated with this pipe. If the option NUMPIP=1 is included in the $SPECIF list, then this first item is omitted, and the pipe numbers associated with pipes in the output data will be according to the sequence number of the pipe in the input data. Pipes do
not need to be numbered sequentially, nor do all numbers from 1 to the total number of pipes in the network need to exist in the input data file. Duplicate pipe numbers will be identified by a message to this effect, but processing of the solution will continue. The maximum number that can be assigned to a pipe will vary depending upon the dimensions allowed by the version of USU-NETWK that you are using, but will be 5000, or larger.

If the option OUTPU1 > 10, then this first item under the PIPES command must be an 8 character string (which may include blanks). See the description under OUTPU1 in the section of this manual that describes options for more detail about this string of 8 characters as a pipe identifier.

2. The number (integer) of the node at the upstream end of this pipe. If OUTPU1=11 or 13, then this will be an 8 character string. For most pipes the user may not know the direction of flow, and for these pipes upstream and downstream have no significance. USU-NETWK will determine which is the correct upstream and downstream nodes as the solution is obtained. However, for pipes that contain booster pumps, valves, etc. upstream and downstream nodes must be given correctly.

In the two table form of output an asterisk (*) precedes the numeric data for the pipe if you select the incorrect direction for flow, as a help should you want to know if you have the direction of flow reversed in some pipes.

3. The number (integer) of the node at the downstream end of this pipe. The same applies as stated above for item # 2.

4. The length of this pipe in feet, or meters depending upon whether ES or SI units are used, respectively. Two options have an effect on this length. If NPLENG=1, then these lengths must be given in 1,000 feet or 1,000 meters. If the option LENGON=0, then this item is omitted from this list, and the pipe length will be computed from the x and y coordinates of the nodes which are required as other input under the NODES command.

5. The diameter of this pipe in the units designated by the value given to the option NUNIT. (If the option DESIGN=4 (or =5) is used then for any pipe that is not part of a loop this item will be the velocity specified for this pipe. The pipe diameter for this non-looped pipe will be solved for to provide teh specified velocity. See the option DESIGN=4 for further explanation.)

6. The wall roughness coefficient for this pipe. Depending upon whether you wish the flowrate, headloss relationship to be bases on (a) the Darcy-Weisbach, (b) the Hazen-William, or (c) the Manning's equation, this coefficient will be: (a) the equivalent sand roughness in the units designated by NUNIT, (b) the Hazen-William's C, or (c) Manning's n for the given pipe. If you don't use NEQUAT in the $SPECIF list, then USU-NETWK will decide whether the coefficient is for use in the Darcy-Weisbach equation of the Hazen-Williams equation according to the magnitude of the first coefficient encountered in the input.

7. Remarks. Generally remarks are not given associated with a pipe, but if you wish to have a remark in the solution table for the PIPE DATA you must follow the last given item with a /R. The R after the / tells USU-NETWK that you want the remark recorded. Otherwise any comments you place on lines of the input file after a / will be ignored. Remarks are restricted to 32 (or possibly 36) characters, and similar remarks are permitted after the NODES and the PIPE commands. For example if a line under pipes were as follows:

```
12 4 7 2400. 20./R THIS PIPE NEEDS TO BE REPLACED
```

then in the PIPE DATA table pipe 12 will have THIS PIPE NEEDS TO BE REPLACED at the end of its line. (Only the simulation version of USU-NETWK allows for remarks.)

Example of pipe data with LENGON=1 (the default)

```
PIPED
3 8 2 1500. 6 .008
7 1 4 2000. 8/
9 2 4 1000/
10 3 8/
12 4 7 2500 10/
```

Example of pipe data with LENGON=0, and NUMPIP=1

```
PIPED
8 2 6 .008
1 4 8/
2 4/
3 8/
4 7 10/
```

```
PIPE-nodes
```

The command PIPE- introduces both the pipe and node data on the same line, one line for
each pipe. Therefore, it can replace both the PIPES and the NODES command in defining the 
physical characteristics of a network. The 
command PIPE- may be used exclusively for 
supplying all the information for pipes and nodes 
or it may be used to provide only a portion of this 
data, in which case the 
commands PIPES and NODES would be needed 
to supply the remaining data.

The general form of data entered by the 
PIPE- command is given by the numbered items 1 
through 10 below. However, depending upon 
options in the $SPECIF list, this list of items is 
added to or deleted from. Many of the items 
listed may be omitted as described later. With all 
default options the list of item entered by the 
PIPE- command are (FORTRAN variable names 
given for each item that are referred to in a 
table that follows):

1. The pipe number, NOP(I)
2. The pipe diameter, D(I) (If the option 
   DESIGN =4, then instead of pipe 
diameter this will be the specified 
   velocity for the non-looped pipe as 
described under the option DESIGN.)
3. The pipe length, L(I)
4. The wall roughness coefficient, E(I).
   This value will be the equivalent sand roughness, 
   c, for the Darcy-Weisbach equation, C for the 
   Hazen-Williams equation, and n for Manning's 
   equation.
5. The upstream node number of the 
   pipe, L11
6. The demand, or external flow, at the 
   upstream node, QJ(L11)
7. The elevation of the upstream node, 
   ELEV(L11)
8. The downstream node number of the 
   pipe, L22
9. The demand, or external flow, at the 
   downstream node, QJ(L22)
10. The elevation of the downstream 
    node, ELEV(L22).

In providing these data decimal points 
must be included with real variables, and cannot 
be included with integers. Items 1, 5 and 8 are 
integers. The rest are reals. Therefore, decimal 
points must be given with items 2, 3, 4, 6, 7, 9 and 
10.

As is the case with other data, this data is 
free format; that is, the data items are separated 
by one or more blanks, or a comma which may be 
followed by blanks. If a line is followed by a /R 
any remarks following this up to 32 characters will 
be kept track of, and written in the solution table 
for PIPE DATA at the end of the line for the 
given pipe. A line cannot exceed 80 characters in 
length. By examining items with decimal points 
and those without decimal points, USU-NETWK 
permits many of the items listed above to be 
omitted for any line. The items which may be 
omitted, or must be omitted are as follows:

Item #1, the pipe number, NOP(I), must 
be omitted if and only if NUMPIP =1 in the 
$SPECIF list of options.

Item #4, the roughness coefficient, E(I), 
may be omitted on any line. Whenever E(I) is 
not given on any line the standard value given by 
the option COEFR0 will be assigned as the 
roughness coefficient for this pipe. The default 
value for COEFR0 is 0.0102 inches for a cast 
iron pipe in ES units.

Item #5, the upstream node number, 
L11, if a source of supply (or sink) such as a 
reservoir, source pump, or nozzle exists at an end 
of this pipe, and NODESP=0 (the default) in the 
$SPECIF list. If desired a 0 (zero) may be 
entered for this new existent node.

Item #6, the demand, QJ(L11) may be 
 omitted on any line. If QJ(L11) is zero this 
 demand does not need to be given on any line. If 
QJ(L11) is not zero, it must be given at least 
one, but may be duplicated on other lines if 
desired. If different values of QJ(L11) are given 
on different lines, the last value given will be 
taken as this demand.

Item #7, the nodal elevation, 
ELEV(L11), may be omitted on any line if its 
value is given on another line. IF NODESP=0 
ELEV(L11) must be omitted as must QJ(L11) if 
this upstream node number L11 is omitted.

Item #9, the demand at the downstream 
node, QJ(L22), may be omitted on any line. If it 
is zero this demand does not need to be given on 
any line. If QJ(L22) is not zero, it must be given 
on at least one line, but may be duplicated on 
other lines if desired. If different values are given 
for QJ(L22), the last value will be taken.

Item #10, the elevation at the 
downstream node, ELEV(L22), may be omitted on 
any line. The same rules apply as for ELEV(L11).

One important rule that must be kept in mind for 
QJ(L11), ELEV(L11), QJ(L22), and ELEV(L22) 
is that if only one decimal point number follows 
a node number (L11 or L22), then this will be 
assumed to be the elevation and not the demand. 
The reason for this rule is that many demands 
may be zero, and therefore need not be included 
as zeros, but may be included as a zero if desired.

A summary of the items which may occur
Table of possible entries and the order of these items (with CORPIN=0) after the PIPE- command. (Note that options LENGON=0 dictates that pipe lengths L(I) be omitted; NUMPIP=1 dictates that NOP(I) be omitted; DESIGN=1 or =3 dictates that an HGL elevation, pressure head, or pressure be given, and this will value will follow the elevation ELEV(L11) or ELEV(L22); and if 0<NETPLT<9 or LENGON=1 then x and y coordinates are needed, and these follow the HGL elevation of elevation of the node. Furthermore, the roughness coefficient E(I) may be omitted from any line.

<table>
<thead>
<tr>
<th>Item of input*</th>
<th>Case no.</th>
<th>#2</th>
<th>#3</th>
<th>#4</th>
<th>#5</th>
<th>#6</th>
<th>#7</th>
<th>#8</th>
<th>#9</th>
<th>#10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NOP(I)</td>
<td>D(I)</td>
<td>L(I)</td>
<td>E(I)</td>
<td>L11</td>
<td>QJ(L11)</td>
<td>ELEV(L11)</td>
<td>L22</td>
<td>QJ(L22)</td>
<td>ELEV(L22)</td>
</tr>
<tr>
<td>2</td>
<td>NOP(I)</td>
<td>D(I)</td>
<td>L(I)</td>
<td>E(I)</td>
<td>L11</td>
<td>ELEV(L11)</td>
<td>ELEV(L11)</td>
<td>L22</td>
<td>QJ(L22)</td>
<td>ELEV(L22)</td>
</tr>
<tr>
<td>3</td>
<td>NOP(I)</td>
<td>D(I)</td>
<td>L(I)</td>
<td>E(I)</td>
<td>L11</td>
<td>L22</td>
<td>QJ(L22)</td>
<td>ELEV(L11)</td>
<td>L22</td>
<td>ELEV(L22)</td>
</tr>
<tr>
<td>4</td>
<td>NOP(I)</td>
<td>D(I)</td>
<td>L(I)</td>
<td>E(I)</td>
<td>L11</td>
<td>QJ(L11)</td>
<td>ELEV(L11)</td>
<td>L22</td>
<td>QJ(L22)</td>
<td>ELEV(L22)</td>
</tr>
<tr>
<td>5</td>
<td>NOP(I)</td>
<td>D(I)</td>
<td>L(I)</td>
<td>E(I)</td>
<td>L11</td>
<td>ELEV(L11)</td>
<td>ELEV(L11)</td>
<td>L22</td>
<td>QJ(L22)</td>
<td>ELEV(L22)</td>
</tr>
<tr>
<td>6</td>
<td>NOP(I)</td>
<td>D(I)</td>
<td>L(I)</td>
<td>E(I)</td>
<td>L11</td>
<td>L22</td>
<td>QJ(L22)</td>
<td>ELEV(L11)</td>
<td>L22</td>
<td>ELEV(L22)</td>
</tr>
<tr>
<td>7</td>
<td>NOP(I)</td>
<td>D(I)</td>
<td>L(I)</td>
<td>E(I)</td>
<td>L11</td>
<td>L22</td>
<td>QJ(L22)</td>
<td>ELEV(L11)</td>
<td>L22</td>
<td>ELEV(L22)</td>
</tr>
<tr>
<td>8</td>
<td>NOP(I)</td>
<td>D(I)</td>
<td>L(I)</td>
<td>E(I)</td>
<td>L11</td>
<td>ELEV(L11)</td>
<td>ELEV(L11)</td>
<td>L22</td>
<td>ELEV(L22)</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>NOP(I)</td>
<td>D(I)</td>
<td>L(I)</td>
<td>E(I)</td>
<td>L11</td>
<td>L22</td>
<td>QJ(L22)</td>
<td>ELEV(L11)</td>
<td>L22</td>
<td>ELEV(L22)</td>
</tr>
<tr>
<td>10</td>
<td>NOP(I)</td>
<td>D(I)</td>
<td>L(I)</td>
<td>E(I)</td>
<td>L11</td>
<td>L22</td>
<td>QJ(L22)</td>
<td>ELEV(L11)</td>
<td>L22</td>
<td>ELEV(L22)</td>
</tr>
<tr>
<td>11</td>
<td>NOP(I)</td>
<td>D(I)</td>
<td>L(I)</td>
<td>E(I)</td>
<td>L11</td>
<td>L22</td>
<td>QJ(L22)</td>
<td>ELEV(L11)</td>
<td>L22</td>
<td>ELEV(L22)</td>
</tr>
<tr>
<td>12</td>
<td>NOP(I)</td>
<td>D(I)</td>
<td>L(I)</td>
<td>E(I)</td>
<td>L11</td>
<td>L22</td>
<td>QJ(L22)</td>
<td>ELEV(L11)</td>
<td>L22</td>
<td>ELEV(L22)</td>
</tr>
</tbody>
</table>

\* NOP(I) - pipe number; D(I) - pipe diameter; L(I) - pipe length; E(I) - roughness coefficient; L11 - upstream node no.; L22 - downstream node no; QJ(L11) - demand at upstream node; QJ(L22) - demand at downstream node; ELEV(L11) - elevation at upstream node; ELEV(L22) - elevation at downstream node.

On any line after the PIPE- command is given in the table above. In this table items are lined up in columns, but in the input data need be separated only by a single blank, or comma.

Since E(I) may be left out of any line this table actually represents 24 different cases that are allowed under the PIPE- command. Furthermore, with the default of NODESP=0 (source of supply not numbered as nodes) it is possible to enter a 0 for L11 to denote this non existent node if this is preferable or leaving it out. It is important to note that the 12 cases given in this table apply only under the default options, and that several options do change items in the list. With LENGON=0 item # 3, L(I), must be omitted, and x and y coordinates may be given after the elevation for the upstream, or for the downstream, or both nodes (not permitted with the PC steady-state version of USU-NETWK). These elevations must contain a decimal point. If NETPLT is given a value greater than zero, but LENGON=1, then L(I) remains but the x and y coordinates follow either ELEV(L11),or ELEV(L22), or both. When giving x and y coordinates the restriction is that both the demand and elevation must be given. If DESIGN=1 or DESIGN=3, then the elevation of the HGL, the pressure head, of the pressure (depending upon the option I HGL) must follow the nodal elevation ELEV(L11), ELEV(L22) or both. If NETPLT > 0 or LENGON =0 along with DESIGN > 0, then after the HGL elevation the x and y coordinates are given, so that 5 values containing decimal points may follow either the upstream node, L11 or the downstream node, L22. The restriction for this latter situation is that x and y coordinates cannot be given unless the nodal pressure is given first, and both the nodal demand and elevation must also be given.
Example of use of the PIPE- command under default options
1 12. 5000. .01 3 1.2 148. 4 3.0 150.
2 8. 1000. 1 5 1.5 165.
5 14. 6000. 7 1.6 180.

Example of use of the PIPE- command with NUMPIP=1
12. 5000. .01 3 1.2 148. 4 3.0 150.
8. 1000. 1 5 1.5 165.
14. 6000. 7 1.6 180.

Example of use of the PIPE- command with DESIGN=1
1 12. 5000. .01 3 1.2 148. 220. 4 3.0 150. 248.
2 8. 1000. 1 5 1.5 165. 255.
5 14. 6000. 7 1.6 180. 275.

Example of use of the PIPE- command with LENGON=0
8 24. 140. 1 2. 1000. 0. 0. 2 1.5 1040. 1500. 0.
9 18. 135. 2 3 1.3 1010. 1500. 2000.
10 12. 130. 4 1.2 1040. 0. 2000. 3

PRINT

The command PRINT allows an alternative means for controlling the amount of output just as the option NPRINT (or IOEX) in the options list does. The command PRINT is followed by a single integer value from -3 to 10. The command PRINT changes the print option after it is entered, and may appear as many times as wanted. Thus if one desires that only the NODES information that is given in the input data file be written to the output file PRINT followed by 4 may precede the NODES data, and PRINT followed by 0 may follow the NODES data. If one desired all of the input data file to be written to the output file the option NPRINT=10 could be in the $SPECIF list, and then just before the RUN command the command PRINT could be followed by -3.

Example of use of the command PRINT
PRINT
5

PUMPS

The command PUMPS enters data for source pumps. An alternative to the use of the command PUMPS would be to replace each line under it by lines under the RESER and BOOST commands, since a source pump consists of a pump in a pipe line, like a booster pump, plus a reservoir with a known water surface elevation. The data after the PUMPS command mathematically describes the operation of a pump, i.e. provides information so that the head produced by the pump can be defined as a function of the flowrate in the pipe that contains the pump. There are two means for doing this. The best is to provide pairs of flowrate Q and corresponding head H produced by the pump to define this relationship. The second method is to provide the power the pump supplies the fluid (its power rating multiplied by its efficiency) and its normal capacity. These are described below starting with item 2.

Data after the PUMPS command consists of:

1. The pump station identifier. Under the default options this identifier is the pipe number that connect the source pump to the network, i.e. the pipe that contains the pump. If NODESP=1 then this identifier is the node number assigned to the source pump, instead of the pipe number. If NUMPIP=0 and NODESP=0 then the pipe number given to identify this pumping station is the sequencial number of this pipe after the PIPES or PIPE-command.

If the option OUTPU1=11 or 12 is in the $SPECIF list, such that pipes (and/or nodes) are identified by 8 character strings, then the identifier given for this item is that identifier i.e., the 8 character string for this pipe (or node). Not only are the characters used important but also their position in the field of 8 character width, i.e. a blank in the pipe identifier before PIPENO2 is not the same as PIPENO2 with a blank after it. See the description under the option OUTPU1 for more explanation about use of character strings for identifiers.
First method (pairs of $Q$ versus $H$)

2. The flowrate, $Q_1$, at the first point on the pump's characteristic curve in the units designated by the option NPGPM.
3. The head, $H_1$, produced by the pump corresponding to $Q_1$.
4. The flowrate, $Q_2$, at the second point of the pump's characteristic curve. This flowrate is assumed to be the normal capacity (i.e. point of greatest efficiency) for the pump, provided that only 3 points are given to define the pump's operating curve.
5. The head, $H_2$, produced by the pump corresponding to $Q_2$.
6. The flowrate, $Q_3$, at the third point of the pump's characteristic curve.
7. The head, $H_3$, produced by the pump corresponding to $Q_3$.

(All heads are in feet when using ES units and in meters when using SI units.)

This list of pairs of $Q$ versus $H$ can be expanded to 10 pairs if the option PCHAR3=0 in the $SPECIF$ list of options. Under this option the next item, i.e. the water surface elevation, must be terminated with a $\backslash$, and the normal capacity is taken at the flowrate nearest the middle of the list of pairs. Furthermore under the option PCHAR3 it is necessary that the flowrates be listed in ascending (increasing) order of magnitude.

8. The elevation of the well, or reservoir water surface from which the pump receives its supply in feet or meter, depending upon whether ES or SI units are used, respectively. This supply surface is often called the sump water surface elevation. If the option NPRPUM=1, this is the pressure at the pump's inlet instead of the elevation of the water surface.

See Appendix F, that describes a utility program PMPCUR, to help you understand and visualize how a given set of input data actually defines a pump's operating characteristics, and how you might provide data to more nearly duplicate the operation of a pump.

Example of data after PUMPS (first method)
PUMPS
5 1 300 1.2 290 1.4 270 3000
9 2 350 1.8 355 1.6 357 2800

Second method (power & normal capacity)

This alternate is used if a pump curve is not available but the pump's power rating is known, and its normal capacity is known (or assumed). When using this alternative USU-NETWK generates a 2nd degree polynomial equation that defines the head produced by the pump as a function of its flowrate typical of most common mixed flow pumps. However, each pump has its unique characteristics of operation, and consequently the results may differ from your pump when you use this option, and the solution shows a flowrate through this line quite different from the normal capacity of the pump.

2. The power supplied the fluid by the pump in horsepower, or kilowatts, depending upon whether ES or SI units are used, respectively.
3. The normal capacity of the pump in the units designated by the option NPGPM.
4. The elevation of the source of supply as described under 8 above. This fourth item must be terminated with a $\backslash$ so that USU-NETWK can determine that you are using this alternative method of input.

Example of data after PUMPS (using the power and normal capacity)
PUMPS
1 12. 1.5 3200/
15 15. 1. 3100./

Third method (giving coefficient of 2nd degree polynomial)

If the pump station identifier is negative, then USU-NETWK will read in the constants a, b and c in the second degree polynomial equation given below to define the pump curve.

\[ H_p = aQ^2 + bQ + c \]

in which $H_p$ is the head produced by the pump and $Q$ is the flowrate. Both head and flowrate are in basis units. The preprocessor PIPEINPT will obtain, and write the pump curve giving these coefficients rather than 3 points. When this is done then the preprocessor does the work of fitting the second degree polynomial to the 3 pairs of data rather than having USU-NETWK do these computations. Upon using the utility program...
PMPCUR described in Appendix F of this manual you may wish to also provide these values. If so the input data consists of:

1. The pump identifier (pipe no. if NODESP=0, or node no. if NODESP=1 preceded by a negative sign.
2. The coefficient a in the above equation.
3. The coefficient b in the above equation.
4. The coefficient c in the above equation.
5. The elevation of the fluid source for the pump.

USU-NETWK will print the coefficients a, b and c under the default level of NPRINT for each pump.

QGIVE

This command permits flowrates in selected pipes to be specified. It is sometime desirable to be able to specify the flowrate that must exist in a pipe. While QGIVE permits such flowrates to be specified, there are situations for which QGIVE is not applicable, and therefore the following explanation if provided. QGIVE cannot be used to specify the flow in a dead end pipe, because the demand at its end, that is not connected to another pipe, already specify what the flowrate in it must be. If a specified flowrate is to come from a source of supply it should be handled as a dead end pipe with a negative demand. If the pipe is "interior" so that loops of the network pass through it, then its flowrate can be specified through the command QGIVE. A special application for which QGIVE might be used is in connection with a valve in a pipe that controls the flowrate by being partly closed. For such applications there is a maximum flowrate that is possible when the valve is fully opened. In other words no positive head (or head increase in the direction of flow) is permitted to achieved the specified flowrate; rather only negative heads, or headlosses in addition to the fluid frictional losses are allowed. This application can be communicated to USU-NETWK by placing a minus sign immediately before the pipe number described below. If a negative pipe number is given and the solution for this pipe indicates a positive head is required, then USU-NETWK: (1) will reject this as the desired solution to the network, (2) remove this pipe as a pipe whose flowrate is specified, and (3) recompute the solution to this new network without the flow in this pipe specified. Therefore, by providing negative pipe numbers as input under QGIVE the user communicates to USU-NETWK that he is not certain whether the flowrate in a pipe is actually specified. It will only be specified if it is possible to control the flowrate by a valve. Should the flowrate in this pipe be less than that desired, then the actual achievable flowrate is accepted.

The solution for problems that use the QGIVE command to specify flowrates in interior pipes will report, as extra data, the heads (positive or negative) required in these pipes for the selected flowrates to be possible. Data after the command QGIVE include:

1. The pipe number that contains the specified flowrate.
2. The specified flowrate. This flowrate will be assumed from the upstream node to the downstream node. If a reversed direction is desired this flowrate can be negative.

Important note: If remarks are provided after data following the PIPES or PIPE commands, then it is necessary that the pipes with specified flowrates must occur in the input data file after the pipe with the last remark associated with it. The reason for this restriction is that USU-NETWK does not consider pipes with specified flowrates the same as regular pipes and they are removed in its counting of pipes.

Example of the QGIVE command
QGIVE
1 1.1
100 1.2

RESERvoirs

The command RESER is used to enter data that define the water surface elevations in reservoirs or tanks. As such a reservoir serves two functions from the viewpoint of hydraulics of the network. First, it provides a source of supply (which may be out from the network, as well, but whose magnitude is unknown), and second, it fixes the elevation of the HGL at this point. For steady-state analyses, this elevation is constant. For time-dependent solutions the initial reservoir water surface elevation is specified in the definition of the network, and a stage-volume or storage relationship (given under the time-dependent data) controls how this elevation may change with time. Only one pipe may connect a
reservoir to a network, and this pipe cannot contain devices such as a pressure reduction valve, for example. If it did then the PRV would effectively specify the head at this position. Data after the RESER command consist of the following two items:

1. The identifier for the reservoir (an integer unless pipes or nodes are given as 8 character strings). If the option NODESP=0 (the default), then this identifier is the pipe number that connects this reservoir to the network. If NODESP=1 in the $SPECIF list, then this identifier is the node number assigned to this source of supply.

2. The elevation of the water surface in the reservoir, or tank in feet or meters, depending upon whether ES or SI units are used, respectively. If the option NPRRES = 1 in the $SPECIF list, then the pressure is given instead of the water surface elevation.

The assumption USU-NETWK makes in initializing the problem for the Newton iterative solution is that the flow is from reservoirs to the network unless you indicate otherwise. Should you assume the flow is from the network into the reservoir, i.e. it is filling, there are two method you can use to tell USU-NETWK this. The first is to give a negative pipe number for # 1 above (negative node numbers are not permitted). A second better method is to give this as the downstream node in the data entered with the PIPES or PIPE- command. Denoting flows into reservoirs, where this is most likely the case, will result in USU-NETWK providing a better flow initialization, and generally result in fewer iterations for a solution, therefore.

**ROTAT**

The command ROTAT stands for rotational speed of pumps. This command can be used in entering original data for a network. The same command, under CHANGE data, and similar commands associated with time dependent data allow for different rotational speeds for pumps to be specified. Use of this command with the original input data allows the information associated with pumps under the PUMPS or the BOOST command to be for a different rotational speed than the pumps are operating under for the requested solution. For example the data you have from the pump curve may be for 1200 rpm, but the pump is actually operating at 1500 rpm. One method would be to scale up the flow rates, and corresponding heads that you read from the pump curve before entering the data for USU-NETWK to use under the PUMPS or the BOOST command. Alternatively you can give the data as read from the pump curve, and then use the ROTAT command and have USU-NETWK scale up the flow rates and heads.

Dimensional analysis can be used to develop the following two similarity relationships for the operation of homogenous pumps:

Flow rate - \( \frac{Q}{ND^3} \) = \( \frac{Q}{ND^3} \)

Pump head - \( \frac{H}{N^2D^2} \) = \( \frac{H}{N^2D^2} \)

in which \( Q \) is the flow rate through the pump, \( N \) is its rotational speed, \( D \) is its impeller diameter, and \( H \) is the head produced by the pump. The subscripts 1 and 2 represent two different operating conditions, or two homogenous pumps. The applications of the above two similarity laws in USU-NETWK is limited to changing the rotational speed \( N \). The first similarity law above indicates that the flow rate \( Q \) varies proportional to the rotational speed \( N \), and the head produced by the pump \( H \) varies as the square of the rotational speed.

The input under the command ROTAT consists of pairs of numbers. The first of each pair consists of the pump designation, and the second the ratio of the desired rotational speed of the pump to the rotational speed for which the input data under the PUMPS or BOOST command apply. The designation of the pump is:

(a) according to pump number, i.e. its order in the input data, (b) the pipe number that connect the source pump to the network, or that contains the booster pump, or (c) the node number for source pumps if NODESP=1 in the $SPECIF list according to whether NPSERI=0, =1, or =2, respectively. Thus pumps under the ROTAT command are identified the same as they are under the SERIES or the PARALLel commands. This list of pairs of values must be terminated with a / and generally will only be one line long, but may extend unto other lines if desired. If you prefer each pair of values may be on a separate line. For the example above for which pump data is given for 1200 rpm, but the operation of the pump is at 1500 rpm, and if the pump designator is 1, the input line after the ROTAT command would be 1 1.25/

Example

ROTAT
The command RUN tell USU-NETWK to: 1. terminate reading data that defines the network, and 2. to obtain a solution. The END command caused the same. For any problem it is possible to have the following data appear after the RUN or END commands, depending upon options set, and the nature of the network:

1. Node oriented input used to duplication the definition of the network's layout to assure correctness of the input data (see the option NETCHK=-1 or =-2).

2. Data needed for time-dependent solutions. This data is expected only by the simulation version of the program, and then only if the option ISIML=1 in the $SPECIF$ list. This input data is described in the subsequently section entitled "Detailed Dynamic Simulation Data Requirements."

3. The node number and the HGL elevation at this node that gives the starting value for head and pressure computations. This additional line of data is required if, and only if, the network is the special simple network without any sources of supply such as a source pump or reservoir. For such networks all exterior flows are given and the sum of positive demands equals the sum of negative demands. Fluid mechanics text books often give these simple network as illustrative problems. This node and HGL elevation one needed if IHGL=-l, for example if no PUMPS or RESER are given and NOMSOL=1 is included in the $SPECIF$ list of options.

4. Pairs of end pipes to be used to define the pseudo loops. These values are given if, and only if, LOOPSE =1 in the $SPECIF$ list. Should PRV's exist in the network, it is necessary to give both pairs of end pipes for the corrective flowrate loops, and the energy loops.

5. Estimates of the fraction of total demand, or the flowrates coming from each reservoir and source pump of the network. These values are given if, and only if, NPERCT =1 in the $SPECIF$ list.

6. The cost data called for if ICOST=1 or greater in the $SPECIF$ list.

7. CHANGE data for an alternative analysis to define a network with some elements altered but which is still the same basic network.

These data are described in the next section of this manual under the heading "Change Data."

**SERIES**

The command SERIES is used to specify that any number of pumps are in series at a given pumping station. Two items of data are need in each line after this command. They are:

1. The designation of the pumping station according to whether NPSERI=0,1, or 2 by pump number, pipe number or node number (only if NODESP=1, also). If a booster pumping station has more than one pump in serier (or parallel), it cannot be designated by node number. If NPSERI=2, booster pumps are still designated by pipes.

2. The number of pumps in series.

Example of the command SERIES

2 3/
4 2/

SETHGL and SETPRe

In the analysis of a pipe networks pressures may be known rather than demands. The command SETPR and SETHG allow for a nodal pressure, or HGL elevation to be given and have USU-NETWK compute the demand (which is assumed unknown) at some other node, or the same node, for each entry. The commands SETPR (standing for setting pressure) and SETHGI (standing for setting HGL elevation) can be used to either specify the pressure or the elevation of the hydraulic grade line at nodes, respectively. If the command SETPR is used the pressure will be given in psi when using ES units for pipe diameters and lengths (e.g. NUNIT less than 2) and in kilo pascals, kPa, when using SI units for diameters and lengths (e.g. NUNIT= 2 or =3). When pressures are given the pressure head is added to the elevation of the node to establish the hydraulic grade line elevation. If the data is entered with the command SETHG, then the elevation of the hydraulic grade line is given in feet or meters dependent, respectively whether ES or SI units are used for the problem. In order to satisfy the specified pressure at a node it is necessary that a demand is adjusted somewhere in the network. The input data under these commands consist of:

1. The node number where the pressure (or elevation of the HGL) is specified.
2. The node whose demand is to be adjusted, e.g. solved for as part of the solution. This node can be the same node as the node in #1, but may be any other real node (e.g. not a source node if these are numbered with the use of the NODESP=1 option).

3. The pipe number that connect a source of supply, e.g. a reservoir or source pump, to the network. This item is required to save effort in the program to find the source with the smallest paths from the nodes given in #1 and #2. This source of supply should be as close as possible to the nodes given in #1 and #2 to minimize computations involved in the solution process. The pipe number is required and no option allow for the source number, or its node number if NODESP=1.

4. The pressure, or elevation of the HGL, that is being specified at the node given by #1, respectively depending whether SETPR or SETHG is used.

The use of the commands SETPR and SETHG is different from the use of the DHEAD command. With the latter command the solution determines a differential head needed in a pipe, such as a pump if positive, or a loss device if negative, or a different pipe size that will satisfy the specified pressure. With the commands SETPR and SETHG and external flow to or from the network will be solved for in order to satisfy the given pressure. Generally the demand at the node of #3 above will be given a value of zero under the NODES or PIPE- command, since this demand is designated as unknown. However, a demand may be given if desired. If a demand is given the value will be utilized in the initialization of the flowrates needed for the iterative solution process, and a good guess of the amount may reduce the number of iterations needed for a solution.

Example of using SETPR and SETHG

SETPRE SETHGL
2 5 1 40
7 7 3 45

Example of using SETPR and SETHG

SETPRE SETHGL
2 5 1 40
7 7 3 45

UNITS

The command UNITS enters a single integer value. It is an alternative to the use of the option NUNIT in the $SPECIF list. If the value following the command UNITS equals:

= 1, then the diameter, wall roughness, and length are in feet.
= 2, then the diameter, wall roughness and length are in meters,
= 3, then the diameter, wall roughness are in cm, and the length is in meters.

**VALVCheck**

The command VALVC enters data for a check valve, which prevents flow in a direction opposite to that specified from the upstream to the downstream node of the pipe. The data are:

1. The list of pipe numbers (integers) that contain the check valves, terminated with a /.

Since only one item of information, the pipe number, is need for each check valve, this data is different in that it is not necessary to have a different line for each additional check valve, but they can, and generally are all on a single line. A check valve is different than a pressure reducing valve (PRV). The PRV holds the pressure constant downstream from it regardless of the upstream pressure, whereas a check valve only prevents reverse flow. A PRV acts as a check valve should the flow in its pipe reverse its flow direction.

Example of data after the VALVC command:

4 6 8 2/

**VALVE** (pressure reduction valve, PRV)

Pressure reducing valves (PRVs) or pressure regulators are used to reduce pressure to a specified amount regardless of the flowrate through the pipe. If the flow reverses in a pipe line that contains a PRV, then it shut off the flow, and acts like a check valve. Should the pressure upstream from the PRV be less than the pressure setting, the valve becomes inoperative, i.e. opens as wide as it can, but it can not maintain the pressure setting. Under wide open operation a PRV has the effect of a minor loss, very similar to a globe valve. USU-NETWK will determine which of the above 3 conditions should exist at a PRV, and obtain the correct solution accordingly. PRV's cannot exist in consecutive (connected) pipes, or pipes from reservoirs or source pumps for obvious reasons. Data entered after the command VALVE consists of:

1. The pipe number (integer) which contains the pressure reduction valve.
2. The distance between the upstream
node of this pipe and the valve in feet or meters, depending upon whether ES or SI units are used.

3. The pressure setting of the PRV expressed in elevation of the hydraulic grade line (HGL-elevation), immediately downstream from the valve. This HGL-elevation is in feet or meters, depending upon whether ES or SI units are used. If the option NPRPRV=1, the pressure is given instead of the HGL elevation. To compute the HGL-elevation from this pressure USU-NETWK assumes that the pipe is a straight line between its end nodes to find its elevation at the location of the PRV.

Example of PRV data after the command VALVE
VALVE
2 100 340
3 160 370

VISCOSity

The command VISCO enters the kinematic viscosity of the fluid in units of ft²/sec or m²/sec, depending upon whether ES or SI units are used. The kinematic viscosity can alternatively be specified different from the default with the option VISC in the $SPECIF list. The viscosity will be the same for all pipes, i.e. it is not possible to enter VISCO intermittently between pipes and have its value changes.

Example of the VISCO command
VISCO
1.275E-6

WEIGHT

The command WEIGHT allows the specific weight of a fluid to be given as an alternative to using the option GAMMA in the $SPECIF list. One value follows the command WEIGHT, e.g. the specific weight of the fluid in lb/ft³ or N/m³, depending upon whether ES or SI units are used, respectively. If not given, 62.4 lb/ft³ is assumed if ES units used, and 9800 N/m³ is assumed if SI units are used by setting NUNIT=2 or greater.

Example
WEIGHT
47.5

NOTES RELATED TO COMMANDS
CHANGE DATA

USU-NETWK allows changes to be made to the basic network, and obtains a solution for the new network thus defined without having to make those changes in the original input data file. Solutions obtained in this manner are referred to as alternate solutions. Generally less computations is required to obtain an alternate solution than the original solution. The computer effort in reading the input data file is always saved. Depending upon the changes specified, the alternate solution may be obtained with only a fraction of the amount of computations needed for the original solution. This saving is accomplished by branching back to an appropriate point in the program. The point of this branching depends upon the nature of the changes. For example, if the change consists of multiplying all demands by a constant, then initialization is obtained by multiplying all previous flowrates by this constant, and then branching to the equation solving part of USU-NETWK. If only the elevation of the junctions are changed, then the previous solution is used to get the new heads and pressures, etc. If individual demands, and/or pipe lengths or diameters, are changed, then a new initialization and solution are obtained. The version of USU-NETWK that can also do time-dependent analyses contains two subroutines to provide initial flowrates in all pipes that satisfy the junction continuity equations. The second such subroutine is designed to use the previous solution for the initialization process, and it will be call on to generate the initializing flowrates for the alternate solutions, as well as for subsequent time step solutions. None of the changes will result in the basic loops being determined again, or a new banding of the Jacobian matrix, unless a PRV, etc. shuts-off or opens again in the alternate solution.

Any number of alternate analyses can be obtained of the basis network. For each such new alternate analysis the word CHANGE introduces the changes, and the word END terminates the changes. In other words each group of changes are included between the commands CHANGE and END and these must start in column 1 of separate lines. The changes are accumulative; that is whatever changes were made to obtain the previous alternative solution will be carried over into the next alternate analysis. If you want to go back to the original problem, it is necessary to change back as part of the new change instructions. The change option is exercised the same regardless of whether the form for the

detailed input data is that in the body of this manual, or according to the rules described in Appendix A or Appendix B.

Changes to the network are specified by providing pairs of lines for each type of change. The first line of each pair consists of a command, or descriptive name, which describes the change, and the second provides the "change" data that are introduced with this command. The commands which are allowed, and a brief explanation of what they change are given in Table 1 below. These names must be spelled correctly to 5 characters, and if desired more characters can be added. These first 5 characters must be upper case, and are shown in upper case in Table 1 with added characters in lower case to identify what the command does better. A command can appear between the CHANGE and END any number of time for a given alternate analysis. The units of the changes are the same as those used in the original detailed input data for the network, e.g. those indicated in the $SPECIF list of options.

Table 1. Commands for making changes to the basic network.

<table>
<thead>
<tr>
<th>Command</th>
<th>What it Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALLCOEeffic.</td>
<td>All roughness coeff.</td>
</tr>
<tr>
<td>COEFFicient.</td>
<td>Individual roughness coeff.</td>
</tr>
<tr>
<td>DEMANdS</td>
<td>Individual demands</td>
</tr>
<tr>
<td>DFRAcUion</td>
<td>Multiplier of all demands</td>
</tr>
<tr>
<td>DIAMEters</td>
<td>Individual diameters</td>
</tr>
<tr>
<td>DHEAD</td>
<td>Differential heads; both type # 1 &amp; type # 2 handled</td>
</tr>
<tr>
<td>NLNODeS</td>
<td>Elevation of nodes</td>
</tr>
<tr>
<td>ELPUMPs</td>
<td>w.s. elev. of pump sump</td>
</tr>
<tr>
<td>ELREServoirs</td>
<td>w.s. elev. of reservoirs</td>
</tr>
<tr>
<td>LENGTH</td>
<td>Individual pipe lengths</td>
</tr>
<tr>
<td>MINOR</td>
<td>Coef. for minor losses,(not PC)</td>
</tr>
<tr>
<td>PARALtec</td>
<td>Number of pumps in parallel</td>
</tr>
<tr>
<td>PEAKF</td>
<td>Multiplier of demands for ranges of nodes</td>
</tr>
<tr>
<td>PRVHG</td>
<td>HGL setting of PRV's</td>
</tr>
<tr>
<td>QGIVE</td>
<td>Specified flowrates in pipes</td>
</tr>
<tr>
<td>ROTATional</td>
<td>Ratio of rotational speed of pumps to that of pump curve</td>
</tr>
<tr>
<td>SERIEs</td>
<td>Number of pumps in series</td>
</tr>
<tr>
<td>SETPR</td>
<td>Specified pressure at nodes</td>
</tr>
<tr>
<td>SETHGI</td>
<td>Specified HGL-elev. at nodes</td>
</tr>
</tbody>
</table>
A single value follows the two commands ALLCO and DFRAC, and these values are the new coefficient, and the new peaking factor, respectively. All roughness coefficients are changed to the value given after ALLCO, regardless of whether all pipe had the same coefficient in the previous solution or not. The value after DFRAC is a multiplier of all demands from the previous solution. Thus if a peaking factor of 1.5 were used in the original input data using the option PEAKF=1.5 in the $SPECIF$ list, or the command PEAKF, then a value of 0.5 after DFRAC will result in the peaking factor of 0.75 for the alternative analysis. It is not possible to change computations from being carried out from the Darcy-Weisbach equation to the Hazen Williams equation or vice versa.

Examples using ALLCO and DFRAC

<table>
<thead>
<tr>
<th>Command</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALLCO</td>
<td>.005</td>
</tr>
<tr>
<td>DFRAC</td>
<td>.5</td>
</tr>
</tbody>
</table>

The commands DHEAD, DIAME, LENG, COEFF, DEMAN, ELNOD, ELRES, ELPR, PVRHG, MINOR, QGIVE, SETPR and SETHG are followed by pairs of values. The first value of each pair is the pipe number, node number or device number, and the second value of each pair is the new value. The command names denote what the new value is. The first of each pair must be an integer without a decimal point, and the second may contain a decimal point. These pairs of values can extend across a line up to and including column 80. The list of these pairs of values must be terminated with a /. In using DHEAD the pipe designated on the first of the pair of values must contain a differential head in the original data. If this is a type # 1 device, then the second of the pairs of values is the new specified differential head. If this is a type # 2 differential head device, then the second of the pair of values is the new specified HGL-elevation (e.g. the changed value for the last, or 5th, item in the original input data) at the node given as the 3rd item in the original input data. If this second item of the pair is negative, then it will be taken as a pressure (psi for ES units, and Pascals for SI units) rather than an HGL-elevation. See alternate to Example Problem # 2 for an example.

Examples using the above commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIAME</td>
<td>2 8. 6 12. 30 10/</td>
</tr>
<tr>
<td>DEMAN</td>
<td></td>
</tr>
</tbody>
</table>

The commands PARALLEL and SERIES change the number of pumps operating in parallel or series at a pumping stations. Three items are needed (grouped together) instead of two as with the commands above. These three values consist of: (1) the number of the pumping station, (2) the old (or previous) number of pumps in series or parallel, and (3) the new number of pumps to be placed in series or parallel operation. Note if simultaneous changes in the water surface from which a source pump obtains the fluid and the number of pumps which operate in parallel or series at this same station are to be made, then it is necessary that the change in the number of pumps in parallel or series by given before the change in water surface elevation.

Example of the SERIES command

SERIES 1 1 2 3 3 2 2 1 0/

Two lines of data are required after the command PEAKF, instead of the single line like the other commands. The first line consists of the peaking factor, i.e. the multiplier of the demands at the nodes given in the next line, and the second the list of nodes. This list of nodes may consist of groups of nodes identified by a - between the beginning and last value of the group, or individual node numbers. This command is not implemented in the PC version of USU-NETWK.

Example of use of the PEAKF command

PEAKF 1.5
20 28 50-70 10 80-85/

The command TITLE is used to introduce a new title for the alternate analysis. The exact manner in which this command has been implemented is different in USU-NETWK currently than was done previously. Earlier versions of USU-NETWK had the new title follow the command TITLE on the same line. If more than a single line of new title was desired, then the command TITLE was repeated on several lines. However, to make this title change consistent with the title at the beginning of the input data file USU-NETWK now has the lines of new title follow the command TITLE. The new title is terminated by a */ or */ (the latter a must for large IBM computers) on the next line.
as is the original title terminated.

Example of new title for alternate analysis

```
TITLE
FIRST ALTERNATE ANALYSIS RELATED TO THE
STUDY OF THE KERNS SYSTEM
Fire demands at nodes 2, 7 and 8
```

The following gives an example of change data that will cause two alternate analyses to follow the original solution to a network problem. These lines of input occur after the RUN command, or the last line of input required for the original problem.

```
CHANGE
TITLE
First alternate analysis, all previous demands multiplied by 2.5
DFRAC
2.5
END
CHANGE
TITLE
Second alternate analysis, diameters of pipes 1, 8 and 50 changed
and the demand at node 10 changed to 1.5 cfs
```

NOTES RELATED TO CHANGES
Introduction

In addition to providing solutions to hydraulic problems giving flowrates and pressures throughout pipe networks, USU-NETWK will also provide an engineering economic analysis of the system for which the hydraulic solution has just been obtained. The costs given as part of this analysis can be as applicable to any given system as are the data provided as "cost input data." Or only default cost data built into USU-NETWK may be utilized, without providing any data, in which case the results from the cost analysis would provide only a rough comparison of relative costs associated with various alternatives. Stated another way, the cost analysis can be as complete and realistic as you, as a user, are willing to provide appropriate input data, or if no data are provided, beyond the request to perform an economic analysis, then only two items are included: the costs of pipes and the costs of energy, if pumps exist, and these are based on default values for (a) cost per unit length of different size pipes, (b) energy costs, (c) life expectancy, and (d) interest rate.

The capability of USU-NETWK to provide an engineering economics analysis makes it easy to determine the cost effectiveness of various alternatives. Therefore, as a user of USU-NETWK, at least as you approach the final design of a system you will probably find it very useful to request that a cost analysis also be done. Cost data should be collected and prepared so that USU-NETWK's cost analysis is meaningful.

This section of the manual describes the input data that are needed for such an engineering economic analysis. This input data has been designed to be as simple as possible, while permitting a maximum amount of flexibility in assigning both reoccurring costs, such as maintenance, energy and costs for water and/or one time capital investment costs to entire groups of network components, or provide separate values to each individual component. Basic default values for capital costs for pipe and unit costs of electricity are built into USU-NETWK, allowing a minimum cost analysis to be performed by the single command END, which denotes the end of the cost data. Other costs can be added to these by providing the appropriate data, and these default values can be overridden by providing values to replace them. Thus, as the user, you have control over how detailed the cost analysis is to be. At the simplest level, the default values will be used for capital costs of pipe and unit costs of electricity at the default interest rate and life will be used. At a much greater level of complexity, you can supply different interest rates, life expectancies, and/or different capital and unit costs to various individual components or groups of components of the network.

In writing this section of the manual the assumption has been made that you have completed a course in Engineering Economics, or at least have had experience in evaluating alternatives by making comparisons of costs on either the basis of the "present worth" or on the basis of "reoccurring series costs." However, to refresh you memory about basics such as capital recover and present worth factors, the following subsection is included in this manual.

Synopsis of Engineering Economic Analysis

The costs associated with a water distribution system roughly fall within the category of a capital investment, or a reoccurring cost. The acquisition of the right of way, installation of pipe, purchase of pumps and their installation, and construction of storage tanks or reservoirs are examples of capital investments. With these there is a present cost to acquire and/or get the facility on line. Often the facility will be paid for over many years. If so the total amount paid will include the initial cost plus interest on the money borrowed. Paying for energy consumed by pumps, buying water, maintaining the system, and paying salaries of personnel associated with the system's operation are examples of reoccurring costs.

In order to have a convenient basis of comparison of the costs of alternatives in the design of a water system, either the capital investment costs need to be converted to reoccurring costs and then added to these costs, that do actually reoccur; or the reoccurring costs need to be converted to present worth values, and added to the capital investment costs. Using the former as the basis of comparison, the alternative with the smallest reoccurring sum of costs would
be selected if all other consideration are equal, and based on the latter, the most attractive alternative would have the smallest composite present worth. The choice of the reoccurring composite cost, or the composite present worth comparison is a matter of personal choice.

A capital investment cost is converted to a reoccurring cost by multiplying its amount by the capital recovery factor, crf. The formula that gives the crf is,

\[
\text{crf} = \frac{1}{(1+i)^n - 1}
\]  

in which \(i\) is the interest rate as a decimal, and \(n\) equals the number of reoccurrences of the payment. A reoccurring cost (that is assumed to be constant in amount) is converted to a present worth by multiplying it by the present worth factor, pwf, which is the reciprocal of crf, and is given by the formula,

\[
\text{pwf} = \frac{1}{(1+i)^n}
\]

For example assume it costs $100,000 to get a new pump on line. However, since $100,000 is not available the money must be borrowed at an interest rate of 10%, with an agreement to pay the loan back over a 10 year period with equal payment at the end of each year. The table below shows the cash flow associated with this capital investment.

<table>
<thead>
<tr>
<th>Yr.</th>
<th>Amount owed at start of year</th>
<th>Interest Payment</th>
<th>Reduction in amount owed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100000.00</td>
<td>10000.00</td>
<td>16274.54</td>
</tr>
<tr>
<td>2</td>
<td>93725.46</td>
<td>9372.55</td>
<td>16274.54</td>
</tr>
<tr>
<td>3</td>
<td>86823.47</td>
<td>8682.35</td>
<td>16274.54</td>
</tr>
<tr>
<td>4</td>
<td>79231.27</td>
<td>7923.13</td>
<td>16274.54</td>
</tr>
<tr>
<td>5</td>
<td>70879.86</td>
<td>7087.99</td>
<td>16274.54</td>
</tr>
<tr>
<td>6</td>
<td>61693.31</td>
<td>6169.33</td>
<td>16274.54</td>
</tr>
<tr>
<td>7</td>
<td>51588.10</td>
<td>5158.81</td>
<td>16274.54</td>
</tr>
<tr>
<td>8</td>
<td>40472.37</td>
<td>4047.24</td>
<td>16274.54</td>
</tr>
<tr>
<td>9</td>
<td>28245.07</td>
<td>2824.51</td>
<td>16274.54</td>
</tr>
<tr>
<td>10</td>
<td>14795.04</td>
<td>1479.50</td>
<td>16274.54</td>
</tr>
</tbody>
</table>

The amount of the payment, $16,274.54 shown in column 4 of this table is obtained by multiplying the crf, which from Eq. 1 equals 0.0162754 by $100,000. Interest shown in column 3 has accumulated over the year from the balance due after the last payment. The charges for the use of the money are listed in column 3, and are obtained by multiplying column 2 by 0.1. The last column, which is the difference between the payment, column 4, and the interest, column 3, is subtracted from the first column to obtain the next value in the first column for the next year. The 10 payments exactly equals the amount of interest plus the initial $100,000. Therefore, the reoccurring amount of $16,274.54 is equivalent to the $100,000 capital costs of the pump. If the pump needs to be replaced after these 10 years of operation, then the $100,000 present worth of the pump is identical to a $16,274.54 annually reoccurring cost.

The annual payment amount of $16,274.54, however, is not the total cost for having the new pump on line. It uses energy and requires maintenance that, let's assume equals $20,000 per year. One method for evaluating the yearly costs for the pump is to add these two reoccurring amounts to give $31,228.80/yr. An alternative is to multiply the $20,000 by the pwf and add this amount to the $100,000 capital cost of the pump. The pwf based on a 10 percent interest rate and number of payments equals 9.090909 so that the total composite present worth of the pump equals $281,818.18.

Any number of other capital investment, or reoccurring costs can be added to get the total cost. Before they can be added it is necessary to convert them to the same base, e.g. a reoccurring cost or a present worth amount. Since the values of crf and pwf depend upon the interest rate, and the number of payments \(i\) and \(n\) must be known. If the amounts are to be interpreted as those necessary to keep the water system in continual service, then the number of payments can be interpreted as the life expectancy of that component of the system. If payments are made twice a year, the crf can be computed using \(n = 2 \times (\text{life expectancy})\) and \(i = 1/2 \times (\text{annual interest rate})\). Life expectancies of different components of the system need not be the same, nor do interest rates associated with them need to be the same. The crf and/or pwf must however be computed based on the proper \(n\) and \(i\) for that network component.

There are numerous variations that might be used to the above. Salvage values may be
included, but if the salvage value is subtracted from the cost of the replacement the same final result is obtained. Inflation may be factored into the analysis by subtracting its rate from that of actual interest rate. For practically all water distribution system analysis constant reoccurring series costs is realistic. USU-NETWK assumes that all reoccurring costs are constant for all payment periods.

**Nature of Cost Data USU-NETWK Understands**

You communicate to USU-NETWK that an engineering economic analysis is to be done by including ICOST=1 in the $SPECIF list of options. With ICOST set greater than zero, USU-NETWK expects to read cost data after the RUN command.

The input data for an engineering economic analysis of a network utilizes the same general format as other input data. A name (command) enters data of a given type. The cost data are entered with any, or all of 18 following available commands. The names are words you can relate the type data they enter to. These commands consist of: INTEREST, LIFE, PAYMENT, ELECTRICITY, PIPES, RESERVOIRS, APPURTENANCES, WATER, OPERATING, POWER, PUMPS, ADDITIONAL, NPRINT, EFFICIENCY, DAYS, RESPOWER, DEMCOSTS, and END. The first 4 characters in each command are significant, and each may be truncated to only 4 characters. For example ADDI, EFFI, etc are sufficient. In addition the words CAPI= and UNIT= preceded values being assigned to capital and unit costs, respectively. Furthermore, the names EQ-C and EQ-A are used to introduce coefficients for equations that will be used to computer capital, and reoccurring costs, respectively. The uses of these 18 commands and the additional 4 names are summarized in Table 1. All commands and names must begin in column 1.

The first three commands listed above, namely INTEREST, LIFE, and LIFE, are in a different category than the rest in that they introduce parameters that are used to compute a new crf and pwf, and they are followed immediately with an equal sign, which in turn is followed immediately with a value. These parameters can occur individually, or together, and may be repeated any number of times in the form shown below.

- INTEREST = i simple annual interest rate (without compounding)
- LIFE = life expectancy in years
- PAYMENT = time between payments in years (or fractions thereof)

To get the i used in equations 1 and 2 above INTEREST is multiplied by PAYMENT, or i = INTEREST $\times$ PAYMENT, and the value of n is computed by dividing LIFE by PAYMENT. In other words if PAYMENT is 1 the values entered by the commands INTEREST and LIFE are equal to i and n in Equations 1 and 2; otherwise they are not. For example, if the desire is to compute costs based on making monthly payments and an annual interest rate of 10 percent exits, then $i = \frac{.10/12}{12} = 0.00833333$, and payment $= \frac{1}{12} = 0.0833333$. Every new entry of any of these three commands will result in new values for crf and pwf being computed, and these new values will be used in computing costs and summing each new computed cost into the accumulated "present worth" and accumulated "annual cost" until any one of them is re-entered, at which time a new crf and pwf will be computed. The default values will be used in any such computations until one of these commands is first entered.

The commands EFFI, DAYS, RESP and NPRINT are followed by a single value that have the following meanings:

- **EFFI** - This command enters a combined motor pump efficiency that will be used to compute the energy used by the pumps. In other words, the energy that a pump adds to the fluid will be divided by this value in computing the kilowatt-hours of energy that will by multiplied by the unit cost values entered by UNIT= after the ELEC or PUMP commands. The default for EFFI is 1. Therefore, if EFFI is not given as part of the cost data the values entered under pumps by UNIT= should equal the electrical rate paid per kilowatt-hour divided by the combined efficiency of the pump and its motor. The unit entered by EFFI will be used in all energy cost computations until it is given a new value.

- **DAYS** - This command enters the number of days of operation for pumps throughout a year. The default is 365. Therefore, if a pump operates 50% of the time the command DAYS followed on the next line by 182.5 should be given before the ELEC or PUMP command for this pump. An alternative would be to reduce the unit energy costs to one-half the actual amount.

- **RESP** - The default meaning of UNIT=amount after the RESER command is the
A demand charge amount that will occur charges, it is possible to get the costs printed command, and again a other components of the network. An alternative meaning for UNIT=amount is achieved by giving the name RESP, that denotes the flow from reservoirs is to be considered water that has been pumped into them and, therefore, the outflow is to cost the same as pump power or energy, a value different from zero. This alternative meaning is accomplished by including the following in the cost data:

RESP
1

To change back to the default meaning, a 0 can follow RESP subsequently in the list of cost data. After the command RESP followed by a 1 the amount after UNIT is the cost of the energy associated with this water flowing out of reservoirs. Should the flow be into the reservoir, no cost is associated with it, since this cost will be taken into account by the pumps that supply the flow to the network. The energy is computed as for pumps, e.g. the flow rate is multiplied by the head (the difference between the water surface elevation and the ground elevation) and an appropriate conversion factor to give kilowatt-hours. The number of hours per year can be reduced from 365 by providing an amount after the command DAYS. Also if an amount is given after the command EFFI then this given efficiency is divided into the above computed energy before multiplying by the unit costs for energy.

**NPRI** - This command changes the amount of extra information that will be written to the output file. In other words this command, as part of the cost data, overrides the value set by the option NPRINT in the $SPECIF list, or the last value entered by the command PRINT. Its use as part of the cost data allows more, or less, extra output be obtained. For example, if NPRINT is used to enter a 6 just before the PIPE command, and again a 0 before the next command, it is possible to get the costs printed out for every pipe in the network on a separate line, but not this amount of detailed cost data for other components of the network.

**DEMC** - This command allow for demand charges associated with pumps to be included as part of the engineering economic analysis. Either pumps that are included in the input data under the commands PUMPS and BOOSTER can be included, or negative demands at nodes might be considered to be pumps. There is only one line (which may extended into several lines) that follows the command DEMC. The items on this line consist of:

1. \( KW_{lim} \) - The limit of kilowatts that can be used by the pump without a demand charge. The amount of this demand charge is computed by the equation below. (An exception to having zero demand charges occurs if the kilowatts, \( KW \), used by the pump are less than \( KW_{lim} \), but the next parameter \( T_{dc} \) is not zero).
2. \( T_{dc} \) - A demand charge amount that will occur if the quantity \( (KW-KW_{lim}) \), as defined below, is negative. The amount \( T_{dc} \) is multiplied by the days of operation to obtain the demand charge.
3. \( C_{dc} \) - The constant in the quadratic equation below that computes the demand charge. The linear coefficient in the quadratic equation below that computes the demand charge.
4. \( B_{dc} \) - The coefficient that multiplies the squared term in the equation below that computes the demand charge.
5. \( A_{dc} \) - The coefficient that multiplies the squared term in the equation below that computes the demand charge.
6. \( S_{dc} \) - A demand charge that is a dollar amount for each pump included in the list, but does not depend upon the kilowatts of power consumed by the pump, and is not multiplied by the number of days of operation.
7. A list of pump numbers (i.e. the order of the pumps in the input data with booster pumps following source pumps), or the list of node numbers preceded by a minus sign if a negative demand represents pumped water. This list of pump numbers, or node numbers) must be terminated by a / The / allows this list to be extended unto several lines if needed.

The equation that computes the demand charge rate is as follows:

\[
\text{$/\text{day}} = C_{dc} + B_{dc}(KW-KW_{lim}) + A_{dc}(KW-KW_{lim})^2
\]

in which \( KW \) is the kilowatts of power required by the pump.

Should the quantity \( (KW-KW_{lim}) \), which is the kilowatts of power required by the pump minus the limited value given or parameter \#1, be negative, then the amount \( T_{dc} \) given as item \#2 above is taken instead of the amount \$/day computed by the above equation. To get the reoccurring costs the amount computed by the above equation is multiplied by the days of pump operation per year, or 365 unless the command DAYS appears before the command DEMC.
In addition to the cost computed by the last multiplication the value of $S_{dc}$ or item # 6 above (without multiplying it by the number of days of operation) is added to the above amount. This latter value $S_{dc}$ is added whether the kilowatts of power required by the pump exceed the limit $KW_{lim}$ or not. Any of the coefficients in the above equation may be assigned zero values should this be appropriate in defining the methods used to obtain the demand charges. If all negative demands represent pumped water, and all of these are to have the demand charges associated with them then rather than listing all of these node numbers, as item 7 above, this item can be given as -.9999 following by /.

END - The command END tells USUNETWK that no more cost data are being provided, and therefore the cost analysis should be terminated. In obtaining time-dependent solutions an additional END command is needed for each succeeding time step solution for which a cost analysis is requested.

The remaining seven commands must be followed on the next line or lines with the name UNIT=value (unit cost), or CAPITAL=amount (or CAPI=amount). The values following the equal signs are the amounts to be used as the unit cost or amount of the capital investment. An exception is the line after the command PIPE if it is UNIT=N. In this exception N indicates the number of pairs that follow giving the cost of pipe per foot of length (or meter of length if SI units are used) and the diameter in inches (or meters) for which that cost applies. The other exception is the command ADDI. After ADDI the next line provides a name for that additional cost. This name is followed by UNIT=unit cost or CAPI=amount in the usual manner. Commands WATER, OPERATING and ADDITIONAL, when followed by UNIT=unit, have a comma followed immediately by one of the words: FLOW, PIPES, NODES, RESER, PUMPS, or VALVE (all 5 characters long). All of these are unit costs that multiple flowrates in basic units (cfs, m$^3$/s, lb/sec, slug/sec, N/s or Kg/s). The word FLOW indicates the total demand on the network. The word PIPES indicates the flowrate in pipes. The word NODES indicates the positive demands from the nodes (flowrates into the network, i.e. negative demands, are ignored in computing costs under these commands). The word RESER indicates flowrates from/to reservoirs. This cost for reservoirs flowrate is accumulated regardless of whether the flow is from or into the reservoir. The word PUMPS indicates flowrates through pumps. The word VALVE indicates flowrates through those pipes containing pressure reduction valves.

After the commands UNIT=value or CAPI=amount a comma may be given followed by a range of values. In Table 1 this is shown as , <range of ... > in which the ... are two integers separated by a - that identify the range of pipes, reservoir, etc. The range is given by an integer value, a dash followed by a second integer without any intermediate spaces. This range is optional, but if given then only those pipes, etc. within the range will have the given costs attached to them. Should the range include pipes, etc. that do not exist, the non exist numbers will be ignored. If the dash and second integer of the range are omitted then only the one given value constitute the range. For example UNIT=.05,PUMPS,1 indicates the unit cost of .05 applies only for the flowrate through the line containing pump 1. A 1-1 in place of 1 would have exact the same effect. Giving the following after the command WATE: UNIT=.04,NODES would indicate that a unit cost of $.04 per cfs of flowrate should be charged for all demands throughout the network, whereas UNIT=.04,NODES,5-25 would indicates that a unit cost of $.04 per cfs of flowrate should be charged for the demands at nodes 5 through 25 for those node numbers that exist within this range. After the command ELEC a line UNIT=.09 would indicate that the energy in kilowatt-hours used by all pumps in the network should be multiplied by .09 to compute these costs, but a line UNIT=.09, 2-4 would indicate that this unit cost applies only for pumps 2 through 4.

It is important to understand that in doing an engineering cost analysis, USUNETWK carries out the computations in the same sequence as the data are given, and that the costs are accumulated as the computations are completed. If no PIPES or ELEC commands are included, then USUNETWK will carry out these two costs analyses based on the default costs for these items. Since commands may be repeated as many times as desired, with the cost for that item accumulated, you have considerable flexibility in defining the costs and what should be included. Both CAPI= and UNIT= may follow a command, and either or both of these may be repeated any number of times after a given command. Furthermore, ranges may be given. If ranges overlap between consecutive UNIT’s the effect is to add the two separately given unit costs to the overlapping numbers. Thus for example if you desire you
could use the PIPES command to separate the installation of pipes into: (1) the actual cost of the pipe, (2) the installation of the pipe in the ground, and (3) the cost of acquiring the right-of-way. For item (1) there would be considerable different unit costs for different pipe sizes and the given data would reflect this. For item (2) there would likely be smaller differences in unit costs for the small versus large diameter pipes, and for item (3) all unit costs would be the same regardless of the pipe diameter.

You should carefully study Table 1. After you have become familiar with cost input data to USU-NETWK you will find that a glance at this table is about all you need to decide on what you want to include in the cost data portion of the input. The following explanation of the meaning of terms will help you in understanding Table 1 better.

Example of cost data

PIPES  CAPl=25000  UNIT=8,1-40
4 2 6 4 8 6 10 8 12 10 15 12 18 14 20 20
UNIT=8,41-60
4 4 6 6 8 10 12 12 12 20 15 25 18 45 20 60
RESER  CAPl=12000,1
UNIT=200
ELECTRICITY  CAPl=5000,2
UNIT=.08
WATER  UNIT=1, FLOWS
UNIT=.5, PUMPS, 4-5
CAPl=10000
OPERATING  UNIT=.05, PIPES
ADDITIONAL
REPLACEMENT  UNIT=.02, FLOWS
END

1. ELECTRICITY represents electrical energy consumed by pumps, and must be used for costs associated with any pump in the original data. If pumps exist but they have not been entered in the original data with the command PUMPS or BOOSTER but rather have been given as a negative demands, then the command PUMPS must be used rather than ELEC.

2. UNIT= enters a unit cost, or a multiplier of the item indicates by the command to get the reoccurring costs. When given after the command RESER this multiplication will produce a one time cost that is added to the initial capital investment cost. When used after any other commands (except pipes) the multiplication of the unit cost by the item indicated will be considered a reoccurring cost, or constant series cost that occurs each payment period in years. If PAYMENT=1 (the default), then this will be a cost each year.

3. CAPl= enters a single capital investment cost. When CAPl=value is given after the commands ELEC, PUMP and RESER without giving a range, then this amount will be applied to all such components in the network. For example if there are 5 pumps in the network, and "value" is 15000, then the total capital cost will be 5 x 15000 = $75,000 for the 5 pumps. CAPl=value after other commands such as WATER, etc., will cause the amount of only one "value" to be added to the costs. If you wish to have the amount of "value" included only once after a command such as ELEC and several pumps exist, then simply add a comma followed by 1 for the range.

4. In the case of the energy costs associated with pumping no efficiency of motor and pump is used unless you have previously given the command EFFI and an efficiency (fraction) following it. Therefore, if you don't use the EFFI command the actual costs for electricity should be divided by the combined efficiencies of the motors and the pumps.

5. By changing the interest rate, and life repeatedly with the commands INTEREST=rate and LIFE=life before giving other cost data it is possible to use different interest rates, etc. as the basis for computing reoccurring costs from given capital costs, and present worth, from reoccurring cost so that the interest rates, and lives can duplicate what the actual situation is. The default for LIFE is 50 years, and the default interest rate is 0.10.

6. The value given after the equal sign following the command PAYMENT is the fraction, or multiple of years when the reoccurring uniform payments occur. The default is 1, meaning that uniform series costs occur once each year. If you want to have the computations based on monthly compounding of interest, the command PAYMENT=.08333 could be given for example.

7. The default capital costs for pipes consist of the following:
<table>
<thead>
<tr>
<th>Pipe dia.</th>
<th>Cost/unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>4&quot;</td>
<td>$3.67</td>
</tr>
<tr>
<td>6</td>
<td>$5.33</td>
</tr>
<tr>
<td>8</td>
<td>$7.67</td>
</tr>
<tr>
<td>10</td>
<td>$10.67</td>
</tr>
<tr>
<td>12</td>
<td>$16.67</td>
</tr>
<tr>
<td>15</td>
<td>$24.00</td>
</tr>
<tr>
<td>18</td>
<td>$43.33</td>
</tr>
<tr>
<td>20</td>
<td>$56.67</td>
</tr>
<tr>
<td>24</td>
<td>$80.00</td>
</tr>
<tr>
<td>30</td>
<td>$100.00</td>
</tr>
<tr>
<td>36</td>
<td>$120.00</td>
</tr>
<tr>
<td>42</td>
<td>$145.00</td>
</tr>
<tr>
<td>48</td>
<td>$170.00</td>
</tr>
<tr>
<td>54</td>
<td>$200.00</td>
</tr>
<tr>
<td>60</td>
<td>$235.00</td>
</tr>
<tr>
<td>72</td>
<td>$270.00</td>
</tr>
<tr>
<td>84</td>
<td>$300.00</td>
</tr>
<tr>
<td>96</td>
<td>$350.00</td>
</tr>
<tr>
<td>108</td>
<td>$390.00</td>
</tr>
</tbody>
</table>

The command PIPE is used to supply a different list of pipe diameters, and associated unit costs. This is done by following the command PIPES on the next line with UNIT=N, in which N is the number of different pipe diameters for which data will be supplied. N cannot exceed 20 with current program dimensions, and then on the next line (or lines) provide the list of N pairs of values with the diameters followed by the costs. This list can be spread across a line up to and including column 80, or can be listed with only two values per line. If a pipe diameter in the network is not equal to one of the given pipe diameters, i.e. between two of the entries of the given pipe data, then the costs associated with that pipe will be interpolated linearly between the costs for the two pipe diameters that bracket its size. Costs associated with pipes with either smaller diameters, or larger diameters than in the list of diameters will have their unit costs equal to the end values of the cost data, i.e. the pipe costs data are not extrapolated. The diameters must be entered in ascending order of magnitude, i.e., the smallest diameter first and the largest diameter last.

8. The POWER command is the only one that produces a revenue, or negative cost, associated with the network unless negative values are entered following CAPI= or UNIT=.

POWER provides the capability to determine revenue that can be obtained through generation of power. The amount given after CAPI= following the POWER command represents the capital costs associated with the installation, etc. of the power generating facilities, and is a costs. The amount provided after the UNIT= is the amount that each kilowatt hour of power can be sold for. This energy is computed based on the flowrates, i.e. demands at the nodes and heads available at the nodes as determined by the solution. The range of nodes (which will generally be a single value) must be given. No efficiency of the hydraulic turbine-generator is used in the program. Therefore, the unit cost data supplied must reflect this combined efficiency plus any transmission losses.

9. It is possible to provide costs associated with ELECTRICITY, PIPES, RESERVOIRS, POWER, and PUMPS by providing parameters in equations that are commonly used for this purpose. Table 2 below gives these equations. The name EQ-C (which can replace CAPI) following the above commands enters the parameters K, a and b shown in the equations, and the given equation is used to compute a capital cost. The name EQ-A (which can replace UNIT) following the above commands enters the parameters K, a and b shown in the equations, and the given equation is used to compute a reoccurring cost. The parameters are entered by leaving a space after the name EQ-C or EQ-A and then giving: A=value,B=value,K=value; <range>. Note that capital A & B in the actual input correspond to a & b in the equations in the table below. If any of the parameters (letters) followed by an = is not given, then the current default value for that parameter will be used, and if the range is not given all devices of that type are assumed. The starting default values are shown in the table below, and any value given for a device becomes the default thereafter.

The following is an example of using equations to compute costs.

<table>
<thead>
<tr>
<th>PIPES</th>
</tr>
</thead>
<tbody>
<tr>
<td>EQ-C K=25,A=1.5</td>
</tr>
<tr>
<td>ELEC</td>
</tr>
<tr>
<td>UNIT=.09</td>
</tr>
<tr>
<td>EQ-C A=.4,K=580,B=-.35;1-2</td>
</tr>
<tr>
<td>CAPI=32000,1-4</td>
</tr>
<tr>
<td>RESER</td>
</tr>
<tr>
<td>EQ-C K=15000,A=500,B=.5;2</td>
</tr>
<tr>
<td>EQ-A K=750,A=500,B=.5;2</td>
</tr>
<tr>
<td>END</td>
</tr>
</tbody>
</table>
Table 1. Summary of input to provide data for an engineering economic cost analysis of the network. (Commands and names that enter data are in upper case letters.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>I. INTEREST=rate</td>
<td>Enters the interest rate, default=0.1</td>
</tr>
<tr>
<td>II. LIFE=life</td>
<td>Enters the life of the network components, default=50 years</td>
</tr>
<tr>
<td>III. PAYMENT=x</td>
<td>Enters the time in years, or fractions thereof between payments of the uniform reoccurring series amount, default=1.0</td>
</tr>
<tr>
<td>IV. ELECTRICITY</td>
<td>For pumps in detailed data of network</td>
</tr>
<tr>
<td>UNIT=cost/kwhr, &lt;range of pump numbers&gt;</td>
<td>(default =0.10)</td>
</tr>
<tr>
<td>CAPI=amount, &lt;range of pump numbers&gt;</td>
<td></td>
</tr>
<tr>
<td>EQ-A eq. parameters; &lt;range of pump numbers&gt;</td>
<td>(see Table 2)</td>
</tr>
<tr>
<td>EQ-C eq. parameters; &lt;range of pump numbers&gt;</td>
<td></td>
</tr>
<tr>
<td>V. PUMPS</td>
<td>For negative demands at nodes, assumed pumped</td>
</tr>
<tr>
<td>UNIT=cost/kwhr, &lt;range of node numbers&gt;</td>
<td></td>
</tr>
<tr>
<td>CAPI=amount, &lt;range of node numbers&gt;</td>
<td></td>
</tr>
<tr>
<td>EQ-A eq. parameters; &lt;range of node numbers&gt;</td>
<td></td>
</tr>
<tr>
<td>EQ-C eq. parameters; &lt;range of node numbers&gt;</td>
<td></td>
</tr>
<tr>
<td>VI. POWER</td>
<td>For income received from power generated</td>
</tr>
<tr>
<td>UNIT=revenue/kwhr, &lt;range of node numbers&gt;</td>
<td>(is a cost)</td>
</tr>
<tr>
<td>CAPI=amount, &lt;range of node numbers&gt;</td>
<td></td>
</tr>
<tr>
<td>EQ-A eq. parameters; &lt;range of node numbers&gt;</td>
<td></td>
</tr>
<tr>
<td>EQ-C eq. parameters; &lt;range of node numbers&gt;</td>
<td></td>
</tr>
<tr>
<td>VII. PIPES</td>
<td>Enters unit costs of different pipe diameters</td>
</tr>
<tr>
<td>UNIT=N, &lt;range of pipes&gt;</td>
<td>(N=number of different pairs of diameters and unit costs in list that follows)</td>
</tr>
<tr>
<td>dia_1 cost/ft (or m)</td>
<td></td>
</tr>
<tr>
<td>dia_2 cost/ft (or m)</td>
<td></td>
</tr>
<tr>
<td>. . . (see table in text for default unit costs for pipe)</td>
<td></td>
</tr>
<tr>
<td>dia_N cost/ft (or m)</td>
<td></td>
</tr>
<tr>
<td>CAPI=amount, &lt;range of pipes&gt;</td>
<td></td>
</tr>
<tr>
<td>EQ-A eq. parameters; &lt;range of pipe numbers&gt;</td>
<td></td>
</tr>
<tr>
<td>EQ-C eq. parameters; &lt;range of pipe numbers&gt;</td>
<td></td>
</tr>
<tr>
<td>VIII. RESERVOIRS</td>
<td></td>
</tr>
<tr>
<td>UNIT=cost/ft of elev. above ground, &lt;range of reservoirs&gt;</td>
<td></td>
</tr>
<tr>
<td>CAPI=amount, &lt;range of reservoirs&gt;</td>
<td></td>
</tr>
<tr>
<td>EQ-A eq. parameters; &lt;range of reservoir numbers&gt;</td>
<td></td>
</tr>
<tr>
<td>EQ-C eq. parameters; &lt;range of reservoir numbers&gt;</td>
<td></td>
</tr>
<tr>
<td>IX. APPURTEYNANCES (PRV,BPV,check valves &amp; other controls)</td>
<td></td>
</tr>
<tr>
<td>CAPI=amount</td>
<td></td>
</tr>
<tr>
<td>X. WATER</td>
<td>Costs of water</td>
</tr>
<tr>
<td>UNIT=cost/unit of water, FLOWS</td>
<td></td>
</tr>
<tr>
<td>UNIT=cost/unit of water, PIPES, &lt;range of pipes&gt;</td>
<td></td>
</tr>
<tr>
<td>UNIT=cost/unit of water, NODES, &lt;range of nodes&gt;</td>
<td></td>
</tr>
<tr>
<td>UNIT=cost/unit of water, RESER, &lt;range of reservoirs&gt;</td>
<td></td>
</tr>
<tr>
<td>UNIT=cost/unit of water, PUMPS, &lt;range of pumps&gt;</td>
<td></td>
</tr>
<tr>
<td>XI. OPERATING</td>
<td></td>
</tr>
<tr>
<td>CAPI=amount</td>
<td></td>
</tr>
<tr>
<td>UNIT=cost/unit of water, FLOWS</td>
<td></td>
</tr>
<tr>
<td>UNIT=cost/unit of water, PIPES, &lt;range of pipes&gt;</td>
<td></td>
</tr>
<tr>
<td>UNIT=cost/unit of water, NODES, &lt;range of nodes&gt;</td>
<td></td>
</tr>
<tr>
<td>UNIT=cost/unit of water, RESER, &lt;range of reservoirs&gt;</td>
<td></td>
</tr>
<tr>
<td>UNIT=cost/unit of water, PUMPS, &lt;range of pumps&gt;</td>
<td></td>
</tr>
<tr>
<td>XII. ADDITIONAL</td>
<td>Up to 5 additional costs may be defined</td>
</tr>
<tr>
<td>NAME OF ADDITIONAL COST</td>
<td>(up to 12 characters long)</td>
</tr>
<tr>
<td>UNIT=cost/unit of water, FLOWS</td>
<td></td>
</tr>
<tr>
<td>UNIT=cost/unit of water, PIPES, &lt;range of pipes&gt;</td>
<td></td>
</tr>
<tr>
<td>UNIT=cost/unit of water, NODES, &lt;range of nodes&gt;</td>
<td></td>
</tr>
<tr>
<td>UNIT=cost/unit of water, RESER, &lt;range of reservoirs&gt;</td>
<td></td>
</tr>
<tr>
<td>UNIT=cost/unit of water, PUMPS, &lt;range of pumps&gt;</td>
<td></td>
</tr>
<tr>
<td>XIII. EFFICIENCY</td>
<td>Enters combined efficiency of pumps &amp; motors decimal value (default=1.0)</td>
</tr>
<tr>
<td>XIV. DAY</td>
<td>Enters days of pump operation through year value (default=365)</td>
</tr>
<tr>
<td>XV. RESPOWER</td>
<td>Water coming from reser. is assumed pumped</td>
</tr>
<tr>
<td>1 or 0 (1 denotes power, 0 costs based on flowrate)</td>
<td></td>
</tr>
<tr>
<td>XVI. NPRINT</td>
<td>Changes amount of output written integer -3 thru 10 allowed, with -3 least</td>
</tr>
<tr>
<td>XVII. END</td>
<td>Terminates reading of cost data and writes summary table giving present worth and annual costs</td>
</tr>
</tbody>
</table>
Table 2. Allowable Equations that can be used to define costs

<table>
<thead>
<tr>
<th>Item</th>
<th>Equation</th>
<th>Initial Default Values</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>ELEC</td>
<td>$=KQ^aH^b$ or $=KE^a$ if $b=0$ or $=Kd^aQ</td>
<td>b</td>
<td>$ if $b&lt;0$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>E is kilowatt of power.</td>
</tr>
<tr>
<td>POWER</td>
<td>(same as ELEC)</td>
<td>K=635,a=.642,b=-.453</td>
<td></td>
</tr>
<tr>
<td>PUMP</td>
<td>(same as ELEC)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PIPE</td>
<td>$=Kd^a\exp(bD)$</td>
<td>K=23.18,a=1.6875,</td>
<td>D is dia. in basic units</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>i.e., feet or meters</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>a = vol. of tank in 1000 gal for example.</td>
</tr>
<tr>
<td>RESE</td>
<td>$=K_d^a$</td>
<td>K=12912,b=.556,a=500</td>
<td></td>
</tr>
</tbody>
</table>

If the command is EQ-C then the $'$s are the present worth values; if the command is EQ-A then $ stands for cost of reoccurring payment.

NOTES RELATED TO COSTS DATA
DETAILED TIME DEPENDENT DATA REQUIREMENTS

(Appplies only to the simulation version of USU-NETWORK)

Introduction

From its first implementation in 1976, USU-NETWK has allowed the performance of a piping system to be simulated over any designated period of time. The literature dealing with the subject of modelling water distribution systems now often refers to such time-dependent solutions as "extended time simulations." Such time dependent solutions perform a series of steady-state analyses at specified time increments until the total time desired in covered. Reservoir elevations are adjusted for each succeeding solution based on the elevation-storage characteristics of the reservoirs; demands are adjusted to agree with time varying magnitudes as determined by dimensionless time dependent demand functions; or rules based on nodal pressures or tank water surface elevations; the number of pumps that operate at pumping stations are adjusted according to time schedules; and/or rules based on pressures at nodes or water surface elevations in reservoirs; etc. so that the actual performance of the system over the designated period of time is simulated. The accuracy of the simulation is dependent upon how well such "time dependent" input data reflect the actual conditions under which the network is performing.

USU-NETWK does not do a transient analysis of pressure surges that result from very rapid closure of valves, or on/off operation of pumps. Such transient analyses must take into account the inertia of rapidly decelerating-accelerating the fluid, as well as the elastic properties of the fluid, and the pipes that contain this fluid, and accommodate surge tanks, air chambers, etc. that are used to control the large "water hammer" pressures resulting therefrom. In fact USU-NETWK does not account for the pressure differences needed over the length of a pipe to accelerate, or decelerate, a "solid column" of water that does not compress. Such analyses based on solid water column theory require the solution of a system of first order ordinary differential equations. USU-NETWK solves the same system of nonlinear algebraic equations that define a steady-state problem. The difference is that this system of equations is modified for each new time step to reflect a new steady-state condition at this time. The determination of the conditions at each new time are based on projecting conditions, as defined by the current solution, ahead one increment in time. This technique is analogous to a "predictor" in solving ordinary differential equations, numerically. In solving differential equations a "corrector" is generally subsequently applied. The corrector technique is not used by USU-NETWK to adjust conditions at the new time step since this would result in an iteration of iterative solutions by the Newton Method, and greatly increase computer times required. Furthermore, the needed logic would be considerably more complex and if not extensive enough could result in a flip/flop situation of repeatedly turning a pump on and then off again, for example. Therefore, should conditions vary very rapidly in time, the user must specify an appropriately small time step for the results obtained from USU-NETWK to agree well with what actually occurs.

When should time-dependent solutions be obtained?

The answer to the question "when should I obtained a time-dependent solution to my network problem" is very subjective. Considerably more data is required for such simulations than a steady-state analysis, and the implication is that you know what the time varying demands, etc. actually are. Physical information about a network such as pipe diameters and lengths, locations of pumps, and tanks, etc. are relatively easy to obtain versus information related to time-dependent demands. When a steady-state solution is obtained it represents a hypothetical condition such as an average daily demand condition, which likely seldom if ever occurs since the time when the average demand occurs at one location is likely different than when it occurs elsewhere. When fire demands are extracted at selected nodes, the solution is used to answer a "what if" question, that helps evaluate the adequacy of the network to deliver under unique circumstances, that again likely never will occur. However, a simulation over a day's time period implies that you have information that not only describes how demands change over a typical day, but that you have information related to tank water surface levels, pumps operating, etc. at the beginning of that day. True the day may be hypothetical, but the time varying conditions through the simulation period should represent a typical real condition, or else the results...
from the computer solution have little resemblance to what happens in the actual network.

Therefore, generally time-dependent solutions are not obtained first in studying the performance of a piping system. Rather steady-state solutions are obtained first, and they are used to verify that the model thus created by the computer does describe the actual system reasonably well. The knowledge acquired from real data as well as studying individual steady-state solutions, might then be used to define what the time varying problem most like should be.

While considerably more information is required for a time-dependent solution than individual steady-state solutions; the amount of information obtained about the networks performance is also much greater. In fact the information included in a single time-dependent solution supersedes many individual steady-state solutions, because conditions, such as water surface elevations in tanks, and pumps on or off are automatically adjusted to reflect what should actually occur. Thus some of the hypotheticalness of the individual steady-state solutions is replaced by what might actually occur. Perhaps the best answer to the question of "when should time-dependent solutions be obtained?" is only if you have considerable knowledge and insight into actual operations and conditions imposed on the system. However, once this knowledge exists time dependent solutions should be used rather than individual steady-state analyses because the results therefrom will provide more information about the network's performance.

The time-dependent input data to USU-NETWK has been made as flexible as possible, while still making it as simple as possible for you. The best way for you to decide what information, should be assembled for a time-dependent analysis is to study the input allowed, and gain some experience in obtaining time-dependent solutions for simple example problems that you make up. You will also need to put more effort in understanding what the time dependent solutions tell you about the performance of the network. USU-NETWK provides you tools that assist in this regard as well. You can have special extra "time-dependent" tables created to examine pressures at selected nodes that are most important, flow rates in selected pipes, and/or water surface elevations in tanks, etc. Also the auxiliary program PRINT (described in Appendix E) allows you to interactively ask questions about selected pressures, flow rates, head losses, etc. at any time. The auxiliary graphics program PLTTIM (see Appendix C) permits you to display the variation of pressure, flowrates, water surface elevation in tanks, etc. as functions of time. Generally most information related to time variations are prepared before hand, and placed in the input data file that USU-NETWK reads in defining the problem. However, you can provide some of this information interactively as you examine the solutions from previous time steps, i.e. you can turn pumps on or off depending upon what you see is occurring during the simulation, for example.

The time-dependent input data have two major categories similar to that of the initial input data; namely a list of options followed by the detailed data that describes the time variations. The data in each of these major categories are described in separate sections below. The time dependent data follows the RUN or END command from the detail data, provided options have not dictated that other data follow this command.

**Time Dependent Options**

Options, or specifications, that provide general information related to the time dependent-solution are included between $TDATA and $END, and this list must preceded the other detailed data for the time-dependent solution. This $TDATA list of options is similar to the $SPECIF list of options. Valid parameter names followed by an equal sign and the value given to the parameter are given for those options for which the default values are not appropriate for the problem being solved. The difference is that the list under $TDATA apply only for the time-dependent solution. The parameter entered in the $SPECIF list still apply. For example an abbreviated list of options may be, $TDATA DTIME=1,INCHR=1,ALTV=1 $END

The names that can be included in this options list are given in alphabetical order below.

**Parameters that may be included in $TDATA list of options:**

**ALTV:** This option permits water surface elevations in reservoirs to be limited by altitude valve settings at the top of the tank, and the bottom of the tank or not be limited, i.e. assume that the tank has no top and bottom. If ALTV=0, then the storage elevation data given for reservoirs will be extrapolated as necessary under the assumption that the reservoir cannot overtop or become empty. A time-dependent analysis with ALTV=0, for example, could indicate what elevations the top and bottom of the storage tank should have to accommodate conditions described in this simulation so that it will not empty or overfill. If ALTV=1, then the largest elevation given for a given reservoir's storage elevation curve (entered with the command STORAGE) will be assumed to be the highest water surface elevation allowed. Should the water surface elevation
attempt to rise above this value, then the altitude valve will shut off the flow into the reservoir. Likewise, the smallest elevation given in the storage elevation data will limit the lowest water surface elevation that is allowed. Flow out from this reservoir after the smallest elevation exists will result in the flow from the tank being shut-off. The value given to ALTV applies to all reservoirs in the network, e.g. it is not possible to have some tank "topless" and "bottomless" and flows from other tanks controlled by altitude valves. (Default ALTV=1)

DTIME: The total time of analysis in days that the time-dependent solution should cover. A value of DTIME=1 for example is equivalent to HTIME=24. (Default DTIME=1)

HTIME: The total time of the analysis in hours that the time-dependent solution should cover. (Default HTIME=24)

INCHR: The time increment, in hours, between consecutive time-dependent solutions. (Default INCHR=1)

It is possible to change the time increment that will be used during different portions of the time-dependent solution by giving either this option INCHR or the option INCMIN a negative value. The effect is as follows:
(A) If INCHR & INCMIN are positive then the simulation will use a constant time increment for the entire time-dependent solution.
(B) If either INCHR<0 or INCMIN<0, then the first time increment will be the absolute value of the negative value(s) given these options. In addition at least one line of additional input is needed that defines when the new time increment(s) should be changed, and what the new time increment(s) should be. This added line must come immediately after the $TDATA list on the next line and this added input consists of:
1. The number of new time increments that will be supplied in the list that follows. This list is limited to 8 or less pairs. If this number is zero, then the list below is not given but rather after each new time step in the solution the user will be prompted for a new time increment in seconds. This invokes option (C) below.
2. The time when the new time increment should start. The units of time used for this time are those designated by the option NTIMED.
3. The new time increment in the units denoted by the option NTIMED.
4. The next time for a changed increment.
5. The corresponding new time increment.
6. etc. with the number of pairs given equal to the integer given in # 1 above.
(C) If either INCHR or INCMIN is negative, and input 1. from (B) above is given a zero value, i.e. no pair of time and new time increment is given, then after each time-step solution the user will be prompted for the new time increment that should be used for the next time step. These time increments must be given in seconds.

INCHP: The time increment, in hours, between which the results will be written to a file (printed). This increment cannot be smaller than INCHR, but could be twice as large as INCHP if the results from only every other solution are to be written to the output file. See the option NPRTAB for a description of what will, or will not, be written for every other such time increment.

INCMIN: The time increment, in minutes, between consecutive time-dependent solutions. (Default INCMIN=60)

See the description under INCHR for how variable time increments can be used.

INCMIP: The time increment, in minutes, between which the results will be written to a file (printed). This increment cannot be smaller than INCMIN, but could be twice, or three times as large as INCMIN if the result from every second, or third solution are to be printed. See option NPRTAB for a description of what will, or will not, be written on this increment.

ISUNIT: This options indicates what units are being used to give reservoir storage volumes under command STORAGE.
If ISUNIT
=0, storage volumes are in cubic feet.
=1, storage volumes are in acre-feet.
=2, storage volumes are in million gallons.
=3, storage volumes are in billion gallons.

LINEAR: Permits interpolation of demand function curves, storage function curves, and flow rule curves to be linear between consecutive pairs of values, or based on a second degree polynomial passing through 3 consecutive values. If LINEAR=1 interpolation is linear, if = 0 the interpolation is quadratic. (default=0)

MTIME: The total time of the analysis in minutes that the time-dependent solution should cover. For example if one wished to cover a time period of 12.5 hours, this
could be done by setting HTIME=12.5, or by setting MTIME=750, or by setting HTIME=12 and MTIME=30.

NOPUMP: This option allows you to control whether the flow rate in a pipe that contains a pump should shut-off, e.g. be exactly zero, if all of the pumps at the station in this pipe shut-off. If NOPUMP=1, then the flow rate is not set to zero when the number of pumps in series or parallel that are in this pipe becomes zero. In other words even though the pumps are all shut off in a given pipe there can be flow in this pipe. A by pass line as often exits at a pumping station would be modelled by setting NOPUMP=1.

NPUNOD: Allows pumps and reservoirs to be referenced by three options when giving time-dependent data: If NPUNOD =0, pumps (including booster pumps) are referenced by number, e.g. the order in which they occur in the input data. The numbering of booster pumps occurs after the last source pump. Thus if there are 3 source pumps, the first booster pump after the command BOOST will be number 4.

=1, source pumps and reservoirs are referenced by node number (allowed only if NODESP=1 in the $SPECIF list of options). Booster pumps are referenced by their pipe number. (Default = 1 if NODESP=1)

=2, pumps and reservoirs are referenced by the pipe number that connects the source to the network. Booster pumps are in pipes and therefore are referenced by pipe number whether NPUNOD=1 or =2.

NPRTAB: This option has meaning in conjunction with options INCHRP and INCMIP by indicating what tables these latter options have control over writing at the indicated frequencies. If NPRTAB =0, then the specified frequencies apply for both the special tables requested by PRINTT>0 and the regular tables of output. If NPRTAB=1, then the specified frequencies apply only to the regular tables of output. For example if NPRTAB=1, INCMIN=15 and INCMIP=30, then any requested data from PRINTT>0 would be written to the special tables every time step of 15 minutes, but only every other time step, or every 30 minutes would the solution be written to the regular tables of output. If NPRTAB=2, then INCMIP has no control on the regular tables, but controls the frequency of writing data to the special tables requested by PRINTT>0. Note control of printing intervals can also be had with the option NTRAND in the $SPECIF list of options. Conflicting specifications between these two controls can result in unpredictable printing of solutions or no solution results printed at all. Therefore, if NTRAND is used to control printing frequencies, then don't use INCMIP, INCHRP and NPRTAB.

NPNRES: This same option is permitted in the $SPECIF list with the same effect as when it is included in the STDATA list of options. The option determines whether a nodal pressure, or reservoir water surface elevation controls pump rules. If NPNRES=0, then nodal pressures at selected nodes control the number of pumps on and off in PUMP rules and in FLOW rules. If NPNRES=1, then the water surface elevation in selected reservoirs control these given rules.

NTIMED: This option allows different units of time to be used in the demand functions and pump schedules given under the commands DEMAND FUNCTION and PUMP SCHEDULE. If NTIMED =0, time is in minutes.

=1, time is in hours (default)

=2, time is in days.

PRINTT: This option allows for special (extra) tables to be written to files under the filename SPECIAL*.DAT (where * = 10, 11, 12 .. ). (When running USU-NETWK under DOS on a PC this name is shortened by leaving out the L to stay within the 12 characters limit allowed.) These special tables contain time as the first column, and flow rates or head losses in selected pipes, or pressures at selected nodes as described below. These tables allow easy examination of what happens over time at selected positions within the network. Graphics described in the graphics Appendix C of this manual allows data from these special tables to be displayed using the utility program PLTITM. If PRINTT is different from 0, then special tables are requested as described below.

If PRINTT > 0, then it is necessary that the appropriate extra commands, PIPE TABLE and/or NODE TABLE follow the STDATA list immediately as dictated by the value given PRINTT. If PRINTT =1, tables of pressure at designated nodes are generated. In this (ese) table(s), and all other special tables, the first column contains time. The remaining columns are pressures at the nodes given under the NODE TABLE special command.

=2, tables of flow rates in designated pipes are generated.

=3, tables of both pressure at designated nodes and flow rates in designated pipes are generated.

=4, tables of head losses in designated pipes are generated.

=5, tables of pressure at designated nodes and head losses in designated pipes are generated.

If PRINTT>0 then according to the tables
requested the following special commands must occur:

**PIPE TABLE**

The command PIPE TABLE is used to introduce pipe numbers that are to be included in a special table of flow rates, or head losses, versus time. This command or one of the following special commands must occur immediately after the STDATA options if PRINTT>0. A list of pipe numbers, or the word ALL, follows PIPE TABLE. ALL indicates that all pipes in the network should be included in the special tables. In order to give PIPE TABLE, the option PRINTT must = 2, 3, 4 or 5.

Example
PIPE TABLE
1 10-15 21/

**NODE TABLE**

The command NODE TABLE is used to introduce nodes to include in the special table called for with PRINTT=1, 3 or 5. A list of node number must follow the command or the word ALL.

Example
NODE TABLE
4 7 15-20/

**RESER. TABLE**

The special command RESER. Table allows the user to designate which reservoirs this (ese) special table(s) is (are) to be created for in the same manner as PIPE TABLE or NODE TABLE cause special tables of data. Depending upon the value of NPUNOD the designation of the reservoirs will be by: (0) number, (1) node provided NODESP=1, or (2) by pipe number. In the special output table(s) the heading for water surface elevation or water depths in the reservoir will be this reservoir designation. The list designating the reservoir, which is the second record after the RESER. TABLE command can be the word ALL, if all reservoirs are to be included. If the reported values in this special table for reservoirs is to be the depth of water rather than the water surface elevation, then the command BOTTOM (with no data following it) should be one of the commands used for special tables. If BOTTOM is given, then the bottom of the reservoir will be taken as the smallest value provided in the data for this reservoir's STORAGE FUNCTION. Should no STORAGE FUNCTION be provided for this reservoir, then the water surface elevation will be reported even if the command BOTTOM is given. The command RESER. TABLE can be used with any combination of special tables, i.e. PRINTT can be given any value from 1 through 5. If a special reservoir table(s) is to be created without any other special tables, then PRINTT should be given the value of 10.

Example (PRINTT=3)
PIPE TABLE
1 5 10 12-16/
NODE TABLE
5-15/
RESER. TABLE
ALL
END TABLES

Example (PRINT=10)
RESER. TABLE
1-10/
BOTTOM
END TABLES

(see examples 10, 11, 12, 20 and 21)

**END TABLES**

This special command indicates the end of list of numbers for special tables. It must be given if and only if PRINTT>0.

**VCOEMA:** This options introduces the maximum minor loss coefficient that can exist for a valve or minor loss devices introduced by commands such as SMINOR, SVALVE, etc. before the flow in the pipe is shut-off completely. The default value is 1000. This loss coefficient, is the multiplier of the velocity head that produces the head loss due to the minor loss. The reason for this maximum loss coefficient is to prevent numerical problems should a resistance for a pipe becomes far out of line with those of other pipes in the network. If a few resistance coefficient do become extremely large, then the solution will be subject to truncation errors associated with the Jacobian matrix used in the Newton iterative method of solution. If the loss coefficient exceeds the value of VCOEMA, then the same routines are called on to shut-off the flow in a pipe if it try to reverse in a pipe containing a PRV or check valve. During the simulation if the loss coefficient becomes less than VCOEMA, and the flow in a pipe was previously shut-off, then the flow will be restored.

**Commands that Enter Time Dependent Data**

As with the original data that defines a network, commands enter data that describe what is to
happen with time for a time-dependent solution by USU-NETWK. In the pages that follow there is a complete explanation of the nature of the data that each command enters under the command as a heading. These headings are listed in alphabetical order to make it as easy as possible for you to turn to a given command. However, the alphabetical listing has the disadvantage that commands that accomplish similar functions are not together except for a few cases where the commands are very similar. Therefore, to give you an overview of what the commands do a numbered list of information that can be provided to USU-NETWK to control a time dependent solution is given immediately below. The commands are capitalized.

1. Demands changing,
   (a) according to time: DEMAND FUNCTIONS
   (b) according to pressures at nodes or water surface elevations in reservoirs: FLOW RULES

2. Volumes stored in reservoirs, or tanks, as functions of water surface elevations: STORAGE FUNCTIONS

3. Changing the number of pumps that are operating in series or in parallel at pumping stations,
   (a) as a function of time: PUMP SCHEDULES
   (b) as a function of the pressure at a node, or water surface elevation in a reservoir: PUMP RULES
   (c) by changing the rotational speed of a pump as a function of time: ROTSCH
   (d) by changing the rotational speed of a pump as a rule: ROTRUL

4. Changing differential head devices: DHEAD and HGLSET

5. Changing the flow rate specified in interior pipes,
   (a) as a function of time: SQGIVE
   (b) according to a 2nd degree equation whose coefficients are given based on pressures at nodes or water surface levels in tanks: RQGIVE

6. Change valve setting and/or minor loss coefficients,
   (a) as functions of time: SVALVE, SVALVC, and SMINOR
   (b) as functions of pressure or water surface elevations in tanks: RVALVE, RVALVC, RVAL+P and RMINOR.

The command DATVAL enters data that provides the loss coefficient for a valve as a function of its opening.

7. Special level and pressure control algorithms: LCALGO and PCALGO

8. Specifying lists of pipe and/or nodes that should, or should not, be included in the solution tables: LPIPE, LNODE, NLPipe and NLNODE.

The description of the commands understood by USU-NETWK for entering time dependent data, and what this data consists of are described in the following pages. Each command must be spelled correctly to 6 characters.

Before describing individual commands let's examine a possible small time-dependent simulation. Before describing individual commands let us examine a possible small time dependent simulation problem that involves the more commonly encountered commands. As you read this you should look ahead and read the descriptions under the commands that are used in providing time dependent data to USU-NETWK. The network problem that will be dealt with is Example 1 given in the subsequent section under ILLUSTRATIVE EXAMPLES, and the original input that describes the network will be as given in this latter section using the PIPE- command in describing the basic network. This network consists of 11 pipes, 6 nodes, two reservoirs, and a source pump. Assume that the demand at node 1 in this network remains constant at 1 cfs, and that the demands at nodes 2, 5 and 6 vary in time such that the given values need to be multiplied by the factors given in the first table below to define these time variations, and that the demands at nodes 3 and 4 vary as given in the other table below. The variation in demand between table entries is linear.

<table>
<thead>
<tr>
<th>Time Peaking (hrs)</th>
<th>Multiplier of Demands at nodes 2,5 &amp; 6</th>
<th>Time Peaking (hrs)</th>
<th>Multiplier of Demands at nodes 3 &amp; 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>1.2</td>
<td>3</td>
<td>1.3</td>
</tr>
<tr>
<td>8</td>
<td>1.0</td>
<td>6</td>
<td>1.5</td>
</tr>
<tr>
<td>12</td>
<td>0.8</td>
<td>10</td>
<td>1.0</td>
</tr>
<tr>
<td>16</td>
<td>0.5</td>
<td>14</td>
<td>0.6</td>
</tr>
<tr>
<td>20</td>
<td>0.6</td>
<td>18</td>
<td>0.5</td>
</tr>
<tr>
<td>24</td>
<td>1.0</td>
<td>24</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Assume that four pumps exist in series at the pumping station, and that for the steady state analysis given in the example problem # 1 that the given pump curve applies for three of these pumps operating. The
operation of the pumps is controlled by the water surface elevation in tank (reservoir) attached to the network through pipe 11, such that when its water surface elevation gets down to 348 ft the fourth pump is put into operation. The complete operating rule for this pump station is:

Water Surface Elevation, Tank 11 348 352 355 358
No. of pumps operating in series 4 3 2 1

The tanks are circular with the one attached through pipe 11 with a diameter $D = 70.7$ ft (a cross-sectional area of 3927 ft$^2$) and the other tank has a diameter $D = 282.4$ ft (a cross-sectional area of 62832 ft$^2$). The smaller tank (off pipe 11) has its bottom at elevation at 345 ft, and it is 15 feet high. The other larger tank has its bottom at elevation 392 ft, and it is 10 feet high. A constant inflow of 8 cfs occurs to this larger tank. The input for this problem can consist of:

**EXAMPLE NUMBER 1 OF SIMPLE NETWORK**

```
$SPECIF NPRINT=-1,NODESP=0,ISIML=1 SEND
PIPE:
1 12. 500. 7 -8. 300. 2 2. 300.
5 10. 1500. 2 1. 280.
3 8. 1600. 3 4. 6. 200.
4 10. 2000. 1 4
2 8. 2200. 2 3 1.5 280.
6 8. 2200. 5 3
8 8. 3000. 5 4 6. 200.
9 6. 1400. 5 6
7 12. 1000. 5 1. 270.
11 12. 500. 1
12 12. 1. 7
PUMPS
7 3 80 4 77 5 72 350
RESER
12 400
11 350
MINOR
4 10
VALVE
8 1500 320
RUN
$TDATA NPNOD=2,NPNRES=1,INCHR=2,ISUNIT=0,LINEAR=1,PRINTT=3 $END
PIPE TABLE
ALL NODE TABLE
ALL RESER. TABLE
1 11/ END TABLES
DEMAND FUNCTION
1 4 1.2 8 1 12.8 16 .5 .2 20 .6 24 1/
2 5 6/
2 3 1.3 6 1.5 10 1 14 .6 18 .5 24 1/
3 4/
PUMP RULES
7 2 11 3 348 4 352 3 355 2 358 1/
STORAGE FUNCTION
1 345 0 350 19635 355 39270 360 58905/
```

You should study each line of this input data in connection with the descriptions under the individual commands so you understand what each value communicates to USU-NETWK. Note the following:

The constant inflow to the larger tank (reservoir) is handled by adding an extra node 7 (with a negative demand of 8 cfs), and pipe 12 with a 1 ft length to the input data description in Example Problem # 1. Special Tables are requested with the option PRINTT=3 in the $TDATA list. Since the PUMP RULE is tied to the water surface elevation in tank 11, the option NPNRES=1 is included in the $TDATA list of options. The time dependent solution will be completed over a 24 hour period (the default) in 2 hour increments.

**DATVAL**

This command is used to enter valve characteristics for valves whose opening will be controlled through input under the commands LCALGO and PCALGO that provide input data for special level and pressure control, respectively. This command is not implemented in the PC version of USU-NETWK. DATVAL should not be used unless LCALGO or PCALGO are also used. If all valves are identical then only one entry under DATVAL is used by USU-NETWK to determine the head loss caused by the valves as a function of their openings, even though there may be several entries in subsequent input lines that control valves. The methods used for determining the head losses caused by partly closed valves are applicable to most butterfly valves whose "head loss coefficient", $K$, or "flow coefficient", $c_v$, plot as a straight line against the valve's opening on semilog graph paper. Figure 1 shows these relationships for a butterfly valve that was calibrated at the Utah Water Research Laboratory to illustrate typical calibration data. Both the "flow coefficient" and the "loss coefficient" are used in practice. Furthermore, sometimes the valve opening is expressed as a fraction from 0 (closed) to 1 for fully open, or as percent with fully open equal to 100%, as well as from 0 to 90 degrees as shown in Figure 1. USU-NETWK will
accommodate any of these methods for defining a valve's characteristics, according to the following 4 cases:

**Case # 1.** The input data describes the valve's "flow coefficient" $c_v$ and the valve's opening is from 0 to 1. For this case the following equations apply.

\[
\begin{align*}
  c_v &= c_o e^{c_1} \quad \cdots \quad (1) \\
  Q &= c_v \sqrt{\Delta p} \quad \cdots \quad (2)
\end{align*}
\]

in which $x$ is the valve opening with $0 \leq x \leq 1$ and $c_o$ and $c_1$ are input values as described below. These values are used to determine the head loss from the equations,

\[
\begin{align*}
  h_L &= K \left( \frac{V^2}{2g} \right) \quad \cdots \quad (3) \\
  K &= \frac{C(2gA^2)}{(\gamma c_v^2)} \quad \cdots \quad (4)
\end{align*}
\]

in which $\gamma$ is the specific weight of the fluid, $g$ is the acceleration of gravity, and $C$ is a coefficient to accommodate different units for the flow rate $Q$, and pressure change $\Delta p$. USU-NETWK evaluates $C$ based on the flow rate used for the demands as designated by the option NFLOW, and the pressure as described below.

**Case # 2.** This second case is identical to Case # 1 with the exception that the opening of the valve is defined between 0 and a given amount $D_{open}$ and is denoted by $D$ in the equations below. In this case $0 \leq D \leq D_{open}$. $D_{open}$ is typically 90 degrees for a butterfly valve, but may be given a different value by the user.
Case # 3. The head loss coefficient is defined by input data directly by providing the coefficient \( c_o \) and \( c \) (positive) in Equations 5 below and 3 above.

\[
K = c_o e^{-c x} \quad \text{in which} \quad 0 \leq x \leq 1 . \tag{5}
\]

Case # 4. The head loss coefficient, \( K \), is defined by input data directly as with Case # 3, however, the valve opening is between 0 and \( D_{\text{open}} \) as in Case # 2, as shown by the equation below.

\[
K = c_o e^{-c D} \quad \text{in which} \quad 0 \leq D \leq D_{\text{open}} . \tag{6}
\]

Case # 5. The flow coefficient, \( c_v \), varies as a straight line on a semi-logarithmic plot over the majority of the valve's closure positions, but varies linearly over the nearly closed positions. For this case as in Case # 1 and # 3, the valve opening is normalized and varies from 0 to 1. For this case the flow coefficient, \( c_v \) in Eq. 2 is given by,

\[
c_v = c_{\text{max}} e^{-c x / a} \quad \text{for} \quad a \leq x \leq 1 \quad \text{and} \quad \tag{7}
\]

\[
c_v = c_{\text{min}} + d x c_{\text{max}} \quad \text{for} \quad 0 \leq x \leq a . \tag{8}
\]

in which \( c_{\text{max}} \) equals the flow coefficient, \( c_v \) for 100 percent open, \( c_{\text{min}} \) equals the flow coefficient, \( c_v \) for the fully closed, and is therefore the closed valve's leakage \( c_v \), and \( c \) is related to \( a \) by,

\[
c = \log_a (1 / a) \quad \text{for} \quad 0 \leq x \leq a . \tag{9}
\]

and \( d \) is computed from

\[
d = e^{a c} - c_{\text{min}} / (a c_{\text{max}}) . \tag{10}
\]

For this Case # 5, the two input items 3 and 4 given below is the usual input under DATVAL are replaced by the 3 items \( c_{\text{max}}, c_{\text{min}} \) and \( a \), and item 6 does not exist, i.e. the input consists of:

1. A valve identification,
2. The number 5 for Case # 5,
3. coefficient, \( c_{\text{min}} \)
4. coefficient, \( c_{\text{max}} \)
5. value for parameter \( a \), and
6. the diameter of the valve.

The input data that is entered under the command DATVAL consists of:

1. A valve identification number that will be used with the commands LCALGO and PCALGO. This number is a positive integer.
2. The Case # 1 through # 5 as described above.
3. The coefficient \( c_o \) in the above equations.
4. The exponent \( c \) in the above equations.
5. The diameter of the valve needed to compute the area in Eq. 4 above. This diameter is provided only if Case # 1 or # 2 is given by item 2 above, and must be in basic units, i.e. feet or meters.
6. The fully open valve position, \( D_{\text{open}} \) for Case # 2 and Case # 4 above. If Case # 4 is given by item 2 then this becomes the 5th item of input. If \( D_{\text{open}} \) is not given with Case # 2 or Case # 4, then 90 degrees will be assumed. 0-degrees is closed and \( D_{\text{open}} \) is fully open.

The list of input items above should be terminated with a slash /. Note: \( C \) in Eq. 4, will be determined internally in USU-NETWK and is a constant used to convert units. If \( \Delta p \) in Eq. 2 is in basic units (psf for ES units or \( N/m^2 \) for SI units) and \( Q \) is in basic units (cfs for ES units or \( m^3/s \) for SI units), then \( C \) is unity. If for example, however, \( Q \) is in gpm and \( \Delta p \) is in psi, then \( C = 448.83^2 (144) = 2.901 \times 10^9 \). The units that will be used to determine \( C \) are taken identical to those in the SSPECIF list as follows: NUNIT=0, then \( \Delta p \) in psi; NUNIT=1, then \( \Delta p \) in psf; NUNIT=2, then \( \Delta p \) in kPa; NUNIT=3, then \( \Delta p \) in kPa. The units of \( Q \) in Eq. 2 are determined by NFLOW to agree with units used for demands.

(see example network 19)

DEMAND FUNCTION

This command enters data that describe dimensionless demand functions, e.g. how the demands at nodes vary with time. A dimensionless demand function always begins with unity at time zero. In other words, the demands given in the original solution, including the multiplier, PEAKF determine the demands for the initial \( T=0 \) solution. The term DEMAND FUNCTION could be thought of as a DEMAND SCHEDULE in that the demands are specified to vary with time. If rules change demands, then the command FLOW RULES must be used. There are pairs of two records or lines of data for each dimensionless demand
function. The items in these records consist of:

Record # 1:
1. The number of the demand function, or integer identifier. This number is not used by USU-NETWK, and could be eliminated from the input requirements, but is there for the user's convenience in identifying different dimensionless demand functions.
2. Time in units specified by NTIMED.
3. The peaking factor, which multiplies the demands given in the original input data. This peaking factor will apply to the nodes whose numbers are given in the second record. This peaking factor applies for the time given by 2 above, and will be interpolated linearly or quadratically depending upon LINEAR for other times.
4. Time
5. Peaking factor at the time given by 4.
6. Etc. Pairs of time, peaking factors and terminate the list with /.

Record # 2:
A list of node numbers. This list of nodes, as well as other lists under other commands may consist of individual values, or a range of values separated by a - (minus sign). Entries in the list may be separated by a comma, a blank(s) or both. No space, or blanks, are permitted between the two integers separated by -. Any number of individual entries or ranges may be specified provided the information does not exceed 80 columns on a single line. Example of such lists are:
1 10-15 21 40-50/
4,7,20-40,80/

Example of data after DEMAND
DEMAND FUNCTION
1 1 .6 6 .1 12 1.4 18 2.1 24 .8/
1 2 13-24 32 41-52/
2 1 .6 6 1.2 12 1.8 18 1.9 24 .6/
3-12 25-31 33-40 53/
(see example networks 10, 11, 12,13, 14, 15, 20, 21, and 22)

DHEAD

This command and the command HGLSET permit changes for type # 1 and # 2 differential head devices, respectively, that were introduced into the original data by means of DHEAD. DHEAD used as a command to enter simulation data allows for type # 1 DHEAD devices to have their differential head amounts changed with time. Type # 1 DHEAD devices introduce a specified differential head (positive or negative) in a pipe. Data after the simulation command DHEAD specifies how this specified differential head is to change with time. Data include:
1. The pipe number that contains the differential head device.
2. Time
3. The differential head associated with the above time.
4. Time
5. The differential head associated with the above time.
6. Etc. terminating the list with a slash /.

Times do not need to correspond to the time steps of the solution. Linear interpolation is used in obtaining differential heads at the time steps of the solution.

Example of DHEAD
DHEAD
5 1 40 6 50 8 45 24 40/
10 2 30 5 40 24 38/
(see example network 22)

FLOW RULES

A FLOW RULE allows the demand at a designated node to be given as a function of either (1) the pressure at a selected node, or (2) the water surface elevation in a selected reservoir. A FLOW RULE differs from a demand function in much the same manner that a pump rule differs from a pump schedule. One defines the change with respect to time and the other defines changes as a function of the pressure somewhere in the network, or as a function of water surface elevations in a tank. Thus flow rules permit demands, which in practice will generally be inflows, or negative demands, to be controlled according to conditions that may exist in the network as indicated from the solution that has just been obtained for the current time step. In specifying a DEMAND FUNCTION, the given values are multiplying factors, i.e. are dimensionless demand functions. However, the given values for a FLOW RULE are the actual demands (positive or negative), in the units used in giving the original demands, since this is useful for
inflow controlled by rules. Also since the common application for a flow rate is an individual node, no provision allows for a range of nodes to be given to which a single rule applies. The items after the command FLOW RULES are included on a single input record, that may extended into a second line if necessary and are:

1. The node number where the FLOW RULE applies. The FLOW RULE will be interpolated using quadratic, or linear, interpolation depending upon whether LINEAR = 0 or = 1, respectively. If the intent of the FLOW RULE is to have it a step function whose flow rate remains constant until the given pressure (or water surface elevation) cross the given value, then a minus must precede this node number.
2. The node number whose pressure controls the rule, or the designation of the reservoir, whose water surface elevation control the rule. If this integer is negative, the later is assumed, i.e., that a reservoir's water surface controls the rule. For this latter rule a reservoir may be designated by: (a) its number (preceded by a minus), (b) by its node number preceded by a minus, provided NODESP=1, or (c) the pipe number preceded by a minus, depending upon whether NPUNOD=0, 1 or 2, respectively.
3. The demand flow in cfs, gpm, mgd, m$^3$/s, etc. according to NFLOW. This value is negative if the flow is into the network.
4. The pressure, or water surface elevation, that controls the flow rate.
5. The demand flow
6. The pressure, or w.s. elevation corresponding to the flow in # 5, etc. and terminating the list with a /.

If 10000 is added to the node number, then rather than computing what the demand is at this node by the rule that follows, the user will be requested to supply the demand in basic units for each new time increment. A prompt from USU-NETWK will inform him that a new demand should be entered from the keyboard by giving the flow rule number, the time, the node where this demand is requested for, and the current demand value. The rest of the data items listed above that are required under each FLOW RULE must be given, but the values are ignored, and therefore, can be any numbers other than zeros, with the exception that the second value must be a valid node or reservoir.

Example of FLOW RULE input data:
FLOW RULE
25 -30 -500. 1580 -1000 1570 -1500 1500/
(see example network 15A)

HGLSET

This command allows for the HGL elevation (or pressure) that was specified at a node by introducing a type # 2 DHEAD device in the original data to be changed with time. A type # 2 DHEAD device determines what magnitude of differential head must exit in a specified pipe in order to maintain the elevation of the HGL (head, or pressure) at the specified magnitude at a designated node. HGLSET allows for changing the elevation of the HGL at this designated node with time. When using HGLSET it is necessary that the original input data contain a DHEAD command that introduces the type # 2 device into the network. The node given by HGLSET must agree with the node number given by the DHEAD command in the original data, i.e. the third item of the original input data must be the first item given under HGLSET. If the node number is not referred to by the HGLSET command, then the HGL elevation at this node does not change with time. Data after the HGLSET command include:

1. The node number where the HGL elevation (or pressure) is specified.
2. Time
3. The HGL-elevation (or pressure) associated with the above time. If the value given is negative, then it will be interpreted as a pressure in psi for ES units, or Pascals for SI units.
4. Time
5. The HGL elevation (or pressure) associated with the above time, etc, terminate with a slash, /.

Times do not need to correspond to the time steps of the solution. Linear interpolation is used in obtaining the HGL-elevations at the time steps of the solution.
Example of the HGLSET command
HGLSET
4 500 4 550 12 490 24 480/
14 3 480 6 500 15 510 24 500/
(see example network 22)

LCALGO

(This command and the command PCALGO are
handled by a subroutine in USU-NETWK and this
subroutine is not included in the reduced simulation
version of USU-NETWK designed for PC's such as the
IBM PC-RT.)

The command name LCALGO stands for
"Level Control Algorithm" and it enters data that
defines the operation of rules that can be adapted to
situations where a control attempts to keep the water
surface elevation in a reservoir at some desired level.
The piping configuration at the reservoir whose water
level the algorithm is attempting to control is as shown
in the sketch below.

```
   valve (2) (3)
   X (4)
   [1]
   (1)
```

This configuration consists of a pipe with the
capability of provide a reasonable large flow rate and
contains a valve that will open or close in response to
the water level in the reservoir. This pipe is # 2 in the
above sketch. At least 4 pipes join at the node [1].
Pipe #4 connects the reservoir to the network through
this node [1] and one additional pipe conveys water to
this node, e.g. pipe # 1, and pipe # 3 conveys water
from node [1]. Thus a typical situation exist where a
valve controls the flow into the reservoir, but the
reservoir delivers water to the system, and may receive
water from other sources. If in a real situation all
these pipes don't actually exit, they can be make of
small diameter, so that they have little effect on the
overall solution. However, it is possible to have a
situation for which the algorithm cannot accomplish its
intended purpose. If it is assumed that the flow rates
in Pipes 1, 2 and 3 are monitored, and these flow rates
are used to determine the position of the valve opening
in pipe #2, as dictated by equations that are given
below. The rate of valve movement can be limited by
the speed at which it can be closed or opened. Also
the control of the valve's opening can be transferred
from the level control algorithm to manual control
during any designated periods of time. When such
manual override occurs then the valve's opening does
not change regardless of what the equations might
indicate as changes that should occur to the valve's
opening.

The input after LCALGO consist of two lines, or
records, for each level control algorithm; the first of
which provides the coefficients, etc. that are used in the
equations, and the second determines if and during
which time period, or periods manual override of the
valves movement is to occur. The order of input data
are given immediately below. Thereafter, the equations
are given that use these input values in
controlling the water surface elevation in the reservoir.

Record # 1

1. The number for pipe # 4 in the above
sketch that connects the reservoir to the
network. It is the reservoir at the end of pipe
# 4 whose water level is being controlled by
the algorithm.
2. The number for pipe # 2 in the above
sketch that contains the valve that opens or
closes in attempting to control the level of the
reservoir.
3. The number of pipe # 1 in the above
sketch that also brings flow into the reservoir
in addition to pipe # 2.
4. The number of pipe # 3 in the above
sketch that takes water from the reservoir.
5. The valve identification (ID) that is given
as the first item under the command DATVAL
to provide the characteristics of the valve in pipe # 2.
6. \(L_{span}\) in the equations below.
7. \(K_{lc}\) in the equations below.
8. \(L_{set\cdot point}\) from the base elevation item 16 below.
9. \(LC_{bias1}\) in the equations below.
10. \(F_{i\cdot span}\) in the equations below.
11. \(LC_{bias2}\) in the equations below.
12. \(K_{fc}\) in the equations below.
13. \(FC_{bias}\) in the equations below.
14. \(TVOPT\), the total opening time for the valve. These times are in seconds, minutes, or hours depending upon NTIMED.
15. \(TVCLT\), the total closing time for the valve (same units as above)
16. Base elevation of the reservoir. The head which is computed in USU-NETWK will be referenced from this elevation when comparing with the head in the equations below that will determine how much the valve should be opened or closed.
17. The valve opening at time zero. This value should agree closely with the opening of the valve that would give the minor loss coefficient entered in the network data under the command MINOR. If a minor loss is not given in the network data, and this opening is not provided, then a diagnostic message will occur and the valve will be fully opened at time zero. If this entry is not given by ending item 16 with a slash, then USU-NETWK will compute the initial opening based on the minor loss coefficient given. If a value is given that is not within reasonable agreement with the opening that agrees with the \(K\) given under the command MINOR, then a message will inform the user of the discrepancy.
18. The RESET RATE to provide integral control action. The RESET RATE is limited to a maximum of 0.5. If this # 18 item is given and the initial valve opening is to be computed based on the minor loss coefficient given in the original data, then # 17 must be -1.0.

Record # 2
This second record consists of groups of 3 values that define if and when manual control is to take over the operation of the valve controlled by the level control algorithm. The 3 values are: (a) the beginning time for manual control (in seconds, minutes or hours depending on NTIMED), (b) the ending time for the manual control, and (c) the minor loss coefficient that is to apply over this time interval. Any number of groups of these three values can be given with the list terminated with a slash /. If the ending time of the first group of three entries is zero, then no manual override of the level controller will occur. If a negative value is given for the minor loss coefficient it tells USU-NETWK that the same minor loss coefficient should be used that is now in existence as determined by the level control algorithm up to the given beginning time given as item (a) above.

Level Control Algorithm (e.g. equations that control)

The level control algorithm determines the opening of the valve that is designated by input item 5 above. The equations that are given below determine what this control consists of. The same symbols used in these equations are the same as given for the input data items above.

The level controller provides the following output based on the input data and the water level in the reservoir above the base elevation:

\[ LC_{output} = K_{lc} \left( L_{set\cdot point} - L_{measured} \right) / L_{span} + LC_{bias1} \]

This output should be between 0.0 and 1.0.

The output from the level controller along with monitored flow rates goes into a summer that provides the following output:

\[ S_{output} = Q_3 - Q_1 + 2F_{i\cdot span} \left( LC_{output} - LC_{bias2} \right) \]

The output from the summer, input data and the flow rate in pipe # 2 that contains the valve determines the desired change in opening of the valve as follows:

\[ FC_{output} = K_{fc} \left( S_{output} - Q_2 \right) / F_{i\cdot span} + FC_{bias} \]

The actual change in the opening of the valve will reflect the speed at which it can be opened or closed. If the desired change in opening \( FC_{output} \) is less than the maximum change that is permitted during the simulation time increment, then \( FC_{output} \) is the amount of valve opening that will occur. If the desired change is larger, then the maximum allowed change controls.
The maximum allowed changes may be different for a valve that is opening versus a valve that is closing according to:

\[ x_{\text{max}} = \text{Simulation time increment } X \]
\[ \times \frac{100\% \text{valve movement/TVOPT}}{100\% \text{valve movement/TVCLT}} \]

Example: See example problem 19 in a later section of this manual for data after the LCALGO and PCALGO commands.

**LNODE, LPIPE, NLNODE, & NLPIPE**

These four commands are listed together because their roles are very similar. They can be used as part of the input for a time dependent solution, and exercise essentially the same control over the output as the command with these same names do when entering the basis data; that is they allow for only a portion of the pipes, or nodes to be listed in the final output tables. In using them as part of the time-dependent input data the first value given is the time when the new listing should begin to apply. The remaining values given denote pipes (or nodes) that will (or will not be) included in the output tables. The commands beginning with N (NLNODE AND NLPIPE) indicate do not list these pipes in output tables. LNODE and LPIPE indicate "list these nodes and pipes, respectively. The output tables will contain the pipes and nodes that are designated (which will be all if no LPIPE, etc. commands are used in the original input data) until these time given by the first input value is equalled or exceeded in the simulation solution. Thereafter, the new list will apply until the time given on the next list occurs. To separate one list from another list the command must be given again. Otherwise more than one line of input will be interpreted as continuation of the list of pipes or nodes.

For example if it is desirable to have only pipes 30 through 50 appear in the pipes data output table starting with time 10 hours through 14 hours, and starting with time 15 hours through the remainder of the simulation have pipes 40 through 60 listed the input would consist of:

```
LPIPES 10 30-50/
LPIPES 15 40-60/
```

If most of the pipes or nodes are to be included, but a few excluded, then it will be more convenient to use the commands NLPIPE and NLNODE. For example, assume that starting at time eight hours, nodes 12 through 120 are to be excluded from the node data output tables to the end of the simulation, then the following command could be used:

```
NLNODE 8 12-120/
```

**PCALGO**

(See note in parenthesis after LCALGO)

The name of this command stands for "pressure control algorithm", and it is designed to control the pressure at a node between specified limits by opening or closing a valve. This command and the command LCALGO are two commands with similar functions; LCALGO activates a valve to maintain a constant water surface level, whereas PCALGO activates a valve to control pressure at a node. You should read the description under LCALGO along with this description in deciding how to describe a controller that may exist in your piping system. Generally the "Pressure Controller" will activate a different valve in a different pipe than the valve controlled by a "Level Controller." However, they may both control the same valve in the same pipe, and it is possible to have the "Pressure Controller" and the "Level Controller" pass control from one to the other depending upon conditions that may exist in the pipe network. As is the case with a level controller, it is possible to have manual control override the change indicated by the pressure control algorithm. The input for each "Pressure Controller" will consist of either two or three records. It will consist of two records unless this "Pressure Controller" can alternate with a "Level Controller" in determining the opening of the valve. In the latter situation three records are required. The means of communicating to USU-NETWK that this "Pressure Controller" may alternate in being active with a "Level Controller" is by placing a minus in front of the pipe number (the first item of input described below).

The input data after the command PCALGO consists of the following: (see description under LCALGO for
meaning of symbols.)

Record # 1
1. The pipe number that contains the valve controlled by this algorithm (This is pipe #2 in the sketch under LCALGO). If this number has a minus immediately in front of it, then this pressure controller may alternate in being active with a level controller designated in the third record of input described below.
2. The node number that controls the algorithm.
3. The valve identification given as the first input under the command DATVAL for the valve that this pressure controller can change the opening of.
4. The low pressure limit.
5. The dead band of pressures.
6. TVOPT, the full stroke opening time for this valve. This time is in seconds, minutes, or hours, depending upon NTIMED.
7. TVCLT, the full stroke closing time for this valve (same units as above).
8. The controller gain.
9. The maximum change in the valve opening position.
10. The maximum value of $c_v'$
11. The initial valve opening $x_o'$.

Record # 2
The second record consists of groups of 3 values as described as the second record under LCALGO.

Record # 3
This record exists only if a negative pipe number is given as the first item under record # 1. This third record consists of two items for each pressure controller. These are:
1. The low reservoir level that determines whether control is turned over to "Level Control." If the reservoir water surface drops below this value, then level control takes over.
2. The high reservoir level that determines whether control which was previously given over to level control will be passed back to pressure control. Whenever the reservoir water surface exceeds this value and level control is active, then control will be passed back to pressure control.

Note: If this third record is given, then it is necessary that data for a level controller has already been entered using the command LCALGO, that designates that its pipe # 2, that contains the valve have the same number as given in item # 1 under record # 1 under the command PCALGO. It is by finding this pipe that USU-NETWK decides which level controller may take over control.

(see example network 19)

PUMP RULES

Pump operating rules turn pumps in series or parallel on or off according to the pressure at a selected node, or according to the elevation of a water surface in a selected reservoir, respectively depending upon whether NPNRES = 0 or = 1. A pump rule is contrasted from a pump schedule in that the latter determines the number of pumps operating as a function of time. The items of data included after the command PUMP RULES consists of:
1. The designation of the pumping station for which the rule applies. This designation is (a) the pump number, (b) the node number, or (c) the pipe containing the pumps depending respectively whether NPUNOD = 0, 1 or 2.
2. An indicator (1 or 2) of series or parallel operation: 1 indicates series operation, and 2 indicates parallel operation.
3. The node number whose pressure controls pump operation. If NPNRES=1, then this will be the reservoir designation whose water surface elevation determines the number of pumps to be placed in operation.
4. The number of pumps on initially.
5. Pressure setting or water surface elevation (according to NPNRES) to operate the number of pumps specified in 6 below.
6. Number of pumps on.
7. Pressure setting (psi or Pascals) or water surface elevation (feet or meters) to operate the number of pumps specified in 8.
8. Number of pumps on.
9. Etc. The pressure or water surface elevation and corresponding number of pumps are entered in ascending order of magnitude. Terminate the record with a slash /.

Up to 20 sets of pressure-number of pumps data may be
Example of the PUMP RULE command

PUMP RULES
10 2  50 1 10, 20, 40, 1/

The interpretation of a pump rule, i.e. the process of turning pumps on or off differs, depending upon whether one considers turning more pumps on from the initial number on or turning pumps off from those initially operating. These two different viewpoints are illustrated below under viewpoints # 1 and # 2. USU-NETWK makes the decision that the user is basing the data on viewpoint # 1 if the number of pumps on initially (4th value in the input list above) is closest to the least number of pumps on in the list than the largest number of pumps on in the list. On the other hand, USU-NETWK assumes viewpoint # 2 if the initial number on is closest to the largest number of pumps as given in the list. This default way of determining whether viewpoint # 1 or # 2 are being used can be overridden. Viewpoint # 2 will govern the operation of the pumps regardless of the above, if the station designation (i.e. the 1st item in the input record) is preceded by a minus sign. Viewpoint # 1 will govern if 10000 is added to the station designation. USU-NETWK does not switch from viewpoint # 1 to # 2 if more pumps are turned on or if pumps are being turned off, respectively.

Example of PUMP RULES, Viewpoint # 1 for two pumps but Viewpoint #2 for the third pump.

PUMP RULES
10 2  5 0 40, 4 50, 3 70, 2 90, 1 100, 0/
10 2  5 0 100, 0 90, 1 70, 2 50, 2 40, 4/
10010 2 5 2 100, 0 90 1 70 2 50 3 40 4/

Example of PUMP RULES, Viewpoint # 2 for two pumps but Viewpoint #1 for the third pump

PUMP RULES
10 2  5 4 40, 0 50, 4 50, 3 70, 2 90, 1 100, 0/
10 2  5 4 100, 0 90, 1 70, 2 50, 3 40, 4/
-10 2  5 0 40, 4 50, 3 70, 2 90, 1 100 0/
(see example networks 10 and 15)

PUMP SCHEDULES

Pump operating schedules are used to specify on-off operations of pumps as a function of time. Thus a pump schedule is analogous to a demand function and a pump rule is analogous to a flow rule.

The data item after the command PUMP S are:

1. Pump designation for which the schedule applies. This designation will be according to the option NPSERI. If NPSERI=0 then pump number; if NPSERI=1 node number; and if NPSERI=2 then one pipe number.
2. An indicator of series or parallel operation by giving a 1 or 2. 1 indicates that the pumps operate in series, and a 2 indicates that they are in parallel.
3. Time in units specified by NTIMED.
4. Number of pumps operating at the time specified in 3 above.
5. Time
6. Number of pumps operating at the time
7. Etc. Up to 50 sets of time-number of pumps data are permitted. Terminate the record with a slash, /.

If 10000 is added to the pump designation, then the user will be ask, at each new time increment, to supply information from the keyboard (terminal) to define the operation of this pump. When provided at least through item # 4 and that the line be terminated with a slash /, but his data will not have any effect on the pump characteristics.

The user will be allowed to select from the following menu items in defining the pump's operation for each succeeding time step: (1) Define the new pump curve by providing three pairs of flowrate and head values plus the water surface elevation of the water supply (which is zero for booster pumps), (2) Give the power, normal capacity, and water surface elevation, (3) Give the three coefficients that define the parabola that defines the pump characteristic curve, (4) Shut the pump off (This shut-off causes a zero flowrate in the pipe that contains the pump), or (5) Make no changes to the pump's operation for the next time step. If the pump has been previously shut-off an item (6) can be selected to turn the pump on again by using any of the items (1) through (4) above. The user will be prompted each time increment to select one of the above items for each such specially designated pump, and then to supply the required data according to his selection.

Example of PUMP SCHEDULE
PUMP S
5 1 2 2 4 3 6 2/
6 2 1 2 3 3 10 2/
(see example networks 11, 13, and 20)

**RMINOR, SMINOR, RVALVE & SVALVE**

These four commands are for entering data associated with time-dependent solution that change the additional head loss, beyond the pipe’s frictional loss, due to a device in a pipe. They are described in general in this paragraph, and then listed separately below. All four commands allow control of the network’s performance by altering the flow resistance in pipes. The commands with an S as the first letter, SMINOR and SVALVE enter data giving this additional resistance as a function of time, i.e. a schedule. The commands with R as the first letter enter data that define rules that control the resistance to flow in the pipe. The commands with MINOR as the remaining 5 characters, SMINOR and RMINOR, allow changes of the minor loss coefficients K to occur. After SMINOR these values of K are given as a function of time, and after RMINOR values for K are determined from the data as a function of the head at a selected node, or from the water surface elevation in a reservoir. The commands with VALVE as the remaining 5 characters, SVALVE and RVALVE are used to control the opening of a valve. The valve opening in turn establishes the minor loss coefficient by means of additional data. The input entered by these command are as follows in the order in which they are numbered. Values are linearly interpolated or extrapolated as needed regardless of the value given to LINEAR.

(see example networks 10, 20 and 21)

**RMINOR**

1. The pipe number that contains the minor loss device, with the same other conditions as described below under SMINOR.
2. The reservoir number, or the node number if preceded by a minus sign, whose water surface elevation, or whose head, controls the rule. Placing a negative immediately before this integer communicates that this nodal head (HGL-elevation of node) governs rather than a reservoir water surface elevation.
3. A water surface elevation of a reservoir, or the head at a node, depending upon whether # 2 is positive or negative, respectively.
4. The minor loss coefficient corresponding to the value given in # 3.
5. A water surface elevation, or head.
6. A corresponding minor loss coefficient, etc. terminating with a slash, /.

Note: In # 2 above that reservoirs must be identified by number, i.e. their order in the original input data, and setting NPSERI has no effect on this command. Allowing them to be identified by pipe or node as with the command STORAGE FUNCTION has not been implemented. It is necessary that the given heads are given in ascending order. Values are linearly interpolated if necessary, and when using nodal heads,
the values must be heads and not pressures as is allowed with other commands.

**SMINOR**

1. The pipe number that contains this minor loss device. It is not necessary to enter a loss coefficient in the original input data through the command MINOR, etc. The first, or time zero solution will not contain a minor loss device if it is not entered with the MINOR command, even if a K is given associated with time zero in this time dependent input data.
2. Time in the units specified by NTIMED in the $TDATA list.
3. The minor loss coefficient K associated with the previous time.
4. Time
5. This minor loss coefficient, K, etc. (The time must be in ascending order. Terminate this list with a slash, /.

If the desire is to specify the values of the minor loss coefficients interactively, then instead of supplying the list of times and minor loss coefficients, give negative values to the times. With a negative value for time USU-NETWK will prompt the user at the absolute value of this time for the minor loss coefficient to be supplied from the keyboard (terminal from which the program is being executed). The prompt for this requested input is the pipe number that contains the minor loss, and the starting time in hours for the new loss coefficient. If the value given to this new coefficient is larger then VEOMA, then the flow in this pipe will be completely shut-off. Therefore, the use of this option allows for flows to be shut-off and started again in pipes of the network during time-dependent solutions.

**RVALVE**

(See description and equations below under SVALVE for a definition of symbols)

1. The pipe number that contains the valve.
2. The reservoir number whose water surface controls the rule given by items 8, 9, etc. below, or if a negative value, then the node number whose HGL-elevation controls the rule.
3. \( c_0 \)
4. c
5. \( K_o \)
6. \( K_I \)
7. b
8. \( H_j \), A value of water surface elevation, or head for the rule.
9. \( x_j \), the valve opening corresponding to \( H_j \).
10. \( H_2 \)
11. \( x_2 \), etc. terminating with a slash, /.

The heads \( H_j, H_2 \) etc. must be listed in ascending order of magnitude. The coefficients in items 3 through 7 above have the same meaning as these coefficients in items 2 through 6 under the command SVALVE.

**SVALVE**

(See explanation and equations below for a definition of symbols)

1. The pipe number that contains the valve.
2. \( c_0 \)
3. c
4. \( K_0 \)
5. \( K_I \)
6. b
7. \( T_j \), time in the units specified by NTIMED.
8. \( x_j \), valve opening corresponding to \( T_j \).
9. \( T_2 \)
10. \( x_2 \), valve opening corresponding to \( T_2 \), etc. terminating with a slash /.

The times, \( T_j \) must be given so that \( T_j \) is less than \( T_2 \) is less than \( T_3 \), etc. The valve openings are linearly interpolated, or extrapolated if needed. The items 2 through 6 under SVALVE and item 3 through 7 under RVALVE are defined by the equations below. The valve loss coefficient \( K \) is related to the valve opening by the equations:

\[
K = c_0 e^{-cx} + K_0 + K_I x^b .
\]  
(1)
in which e is the base of natural logarithms, and K gives the head loss by means of,

\[ h_L = K \frac{V^2}{2g} = KQ^2/(2gA^2) \]  

(2)

in which \( c_0 \), c, \( K_o \), \( K_l \) and b are given coefficients, and x is the valve opening provided by the rule, e.g. the list of numbers given by items 7, 8, 10 etc. above. x will be linearly interpolated between the values listed above, and the times must be given in ascending order. If \( K_o \), \( K_l \) and b are given as zeros, then Eq. 1 describes the loss coefficient for a butterfly valve (see Figures 1 and 2 under DATVAL). For describing the opening of a butterfly valve x is in degrees and 0 degrees is closed and 90 degrees is opened. For describing other types of valves \( c_0 \) can be given a zero valve and c = 1, and the other coefficients used to describe the valves loss coefficient, or \( c_0 \) and c may also be used. Thus for a gate valve, for example, x may be the stem movement, or the fraction of the allowable stem movement, as the user desires.

A possible alternative to using any of these commands is to use the special rule that can be invoke through the command RQGIVE. In using any of valve or minor loss commands numerical problems can occur as the valve approaches complete closure or the specified loss coefficients become too large. USU-NETWK solves a large matrix problem to obtain the solution. If the resistance to flow is large in a pipe, the elements of the matrix associated with this pipe will become very large also. Truncation errors in the matrix algebra can invalidate the results when some elements of the matrix are very large in comparison with the other elements, despite the fact that methods are used that minimize truncation errors. Also convergence to the solution is slower if pipes of vastly different resistance to flow exist in a network. Consequently, in preparing data for any of these commands care should be given so that K's, either specified directly, or through valve opening, do not become very large, unless you wish the flow to be shut off completely. If the loss coefficient exceeds VCOEMA then USU-NETWK shut the flow in the off.

Examples using these commands

SMINOR
14 1. 100. 2. 10. 3. 0. 24. 0/
20 1.5 10. 3. 5. 10. 1. 24. 1/
RMINOR
16 -9 60. 200. 70. 100. 80. 25/

ROTRUL

The command ROTRUL under time dependent data is similar to the command ROTAT in the original input data, in that it allows that the rotational speed of pumps can be changed according to a rule that is dependent upon the pressure at a node, or the water surface elevation in a reservoir. The command ROTSCH, described below allows this rotational speed to be varied according to a time schedule. In much the same manner as PUMP RULE, or FLOW RULE provides controlling rules, the rule given under the command ROTRUL determines how the rotational speeds change according to how the pressure may vary at a node, or according to how the water surface elevation in a reservoir may vary as determined by the time dependent solution. The rotational speed of the pump is given as the ratio of its rotation to the rotational speed for which the pump characteristics were given by the data under the PUMPS or BOOSTer command. The input data under ROTRUL consist of:

1. Pump designation for which the rule applies. If NPUNOD=0 then this designation is the pump number (i.e. its order in the original input data); if NPUNOD=1, then this designation for source pumps is their node number (NODESP=1 must be set in the original input, and booster pumps are designated by the pipe containing them, and if NPUNOD=2, then all pumps are designated by the pipe containing them.

2. An indicator of: 1 - discrete, or 2 - continuous operation. Discrete operation means that the rotational speed changes to the new given ratio of rotation as a step function when the given time is equalled or exceeded. This new speed remains constant until the next given time, etc. Continuous operation means that the rotational speed will be determined by linear interpolation according to the time of the solution and the times given to define the schedule. This second item of input must be either 1 or 2; 1 for discrete operation, and 2 for continuous operation.

3. The node number whose pressure controls the pump's rotational speed (if NPNRES=0) or the
reservoir designation whose water surface controls the pump's rotational speed (if \(NPNRES=1\)) Reservoir designations are determined according to the option \(NPNRES\) also. If for a given pump it is desired to reverse what controls the rule, for this given pump, from that indicated by \(NPNRES\), then precede this value by a minus value. For example if \(NPNRES=0\), and a given pump is to be controlled by the water surface in a reservoir, rather than by a nodal pressures as indicated by \(NPNRES=0\), then give minus this reservoir designation. If \(NPNRES=1\) and a pump is controlled by a nodal pressure, then the controlling node number would be given as minus.

4. The pressure setting, or water surface elevation for the rule,

5. The ratio of pump rotational speed to that given by the original input data corresponding to the value in item \(#4\).

6. The pressure setting, or water surface elevation,

7. The ratio of pump's rotational speed

8. Etc. Terminate the record with a /.

**Example**

```
ROTRUL
  5 1 100 1.3 90 1.2 80 1.0 70 .9 60 .8 50 .75 /
  8 2 120 1.5 100 1.3 70 .8 /

ROTSCH
```

The command ROTSCH under time dependent data is similar to the command ROTAT in the original input data, in that it allows that the rotational speed of pumps be changed as a function of time, i.e. according to a time schedule. The command ROTSCH does allow the rotational speed to be changed according to a time schedule, and the command ROTRUL (discussed previously) allows for this change to be governed by a rule. The input data after this command are similar to those after the command PUMP SCHEDULE, and consist of the following:

1. Pump designation for which the schedule applies. If \(NPUNOD=0\) then this designation is the pump number (i.e. its order in the original input data); if \(NPUNOD=1\), then this designation for source pumps is their node number (NODESP=1 must be set for the original input), and booster pumps are designated by the pipe containing them, and if \(NPUNOD=2\), then all pumps are designated by the pipe containing them.

2. An indicator of: 1 - discrete, or 2 - continuous operation. Discrete operation means that the rotational speed changes to the new given ratio of rotation as a step function when the given time is equalled or exceeded, and this new speed remains constant until the next given time. Continuous operation means that the rotational speed will be determined by linear interpolation according to the time of the solution and the times given to define the schedule. This second item of input must be either 1 or 2; 1 for discrete operation, and 2 for continuous operation.

3. Time in units specified by NTIMED.

4. Ratio of pump rotational speed to that given by the original input data after the PUMPS, or the BOOSTer commands that define the pump characteristic curve, corresponding to the time in item \(#3\).

5. Time

6. Corresponding ratio of pump rotation

7. Etc. Terminate the record with a slash /.

As with the command ROTAT, that can be used in the original data, only the rotational speed, and not the diameter that occurs in the two similarity laws, that are described in that previous section, can be changed. The ratios of rotational speeds given under the commands ROTSCH and ROTRUL are the specified rotational speeds divide by the rotational speed defined by the pump characteristic data of pairs of \(Q & h's\), or power and normal capacity given under the PUMPS or the BOOSTer commands. By using the command ROTAT is in the original data, pump operation for the initial solution, i.e. time equal zero step, can be different from that defined by the pump
characteristics. However, if the command ROTAT is used, then the ratio of rotational speed given under the command ROTAT is based on the data under the PUMPS (or BOOST) command and not the altered speed given by the command ROTAT. Giving a time = 0 under ROTAT will be ignored, i.e. cannot be a substitute for using the command ROTAT.

Example

ROTSCH
1 1 2.93 .8 6 1. 10 1.4 15 1.3 18 1/
2 1 2.83 .7 6 .9 10 1.5 15 1.4 18 1/

RQGIVE and SQGIVE

These are commands that allow the flow rate which has been specified in selected interior pipes through the command QGIVE to be altered during a time dependent solution. The S at the beginning of QGIVE stands for schedule. SQGIVE specifies flow rates vary in the designated pipes as a function of time. The R at the beginning of QGIVE stands for rule. A rule controls the flow rate by an equation whose solution depends upon the conditions in the network. The rule may be associated with a reservoir, and is thus governed by the water surface elevation in the reservoir, or it may be associated with a node and then the head at this node governs. The rule is created by providing the coefficients CO, C1, C2 and C3 in the equation described below under the heading of RQGIVE.

RQGIVE

There are three quite different means available under the command RQGIVE for providing rules that determine flow rates in interior pipes. The first method requires that coefficients for a quadratic equation be provided that gives the flow rate as a function of the pressure head above a specified base head. The second has more specific use for butterfly valves. With its use you adjust the position of the value by giving values to constants in equations. The third possibility is to specify the flowrate by giving data during the simulation that determines the flowrate as a function of the water surface elevations in selected reservoirs. You communicate to USU-NETWK which of these three rules to apply by giving a positive, or negative value for the pipe number, or adding 10000 to this number whose flow rate is specified. These three rules are discussed under the separate subheadings RQGIVE # 1, RQGIVE # 2 and RQGIVE # 3 below, but the # is not part of the command.

RQGIVE # 1: The first rule for specifying flow rates consists of the equation:

\[ Q = CO + C1(H-C3) + C2(H-C3)^2 \quad (1) \]

in which CO, C1, C2 and C3 are supplied as input, and H is either the water surface elevation of the reservoir, or the pressure head (units of length) at the node, depending upon which is the control of the rule.

The data entered by the RQGIVE command consists of:

1. The pipe number (positive) whose flow rate is specified.
2. The designation of the control for this rule. This designation consists of either the reservoir number or the node number preceded by a negative sign. The negative value tells USU-NETWK that the head at the node controls rather than a water surface elevation in a reservoir.
3, 4, 5, and 6. The four coefficients, CO, C1, C2 and C3.

In using RQGIVE, it is not possible to designate reservoirs by node number or pipe number through the option NPUNOD. Reservoirs must be identified by their number in using RQGIVE, e.g. their order in the original data. The units of Q (flow rate) in the above equation are those designated by NFLOW. Thus if NFLOW=1, which indicates gpm, then the given coefficient must produce the desired flow rate in gpm. It should be noted that Eq. 1 is a quadratic equation in terms of the difference between the water level, or the head H and a base level or head given by C3. If this relationship is to be linear, then C2 is given a value of zero. CO represents a base flow. Giving CO = 0 and C2 = 0 and C1 a positive value causes the flow rate in the designated pipe to be proportional to the difference H-C3; with Q negative if H becomes less the C3. By giving C1 a negative value Q will be positive if H is less than C3. While it is not a necessary requirement of USU-NETWK, the flow rate specified in the original data under the command QGIVE should satisfy Eq. 1.
Example using command RQGIVE
RQGIVE
3 1 .03 .001 .0 500
3 1 .035 .001 .0 500.

RQGIVE # 2: The second type rule that can be applied for controlling the flow rate in a pipe is designed more specifically for a butterfly valve. Instead of specifying the flow rate directly as a function of a head difference this rule specifies the position of the valve opening. The rule works in the following manner.

A. The opening of the valve is adjusted by Equations 3, 4 or 5 below.

B. Based on this valve opening a new flow rate will be computed for the next time increment such that the minor loss created by the valve plus the frictional loss in the pipe are equal to the current difference in the HGL elevation at the upstream and downstream nodes of this pipe.

C. The new flow rate will apply for the following time increment. The procedure in these three steps is repeated for each new time increment.

To understand how to apply this rule a little background in valve head loss relationships is given. The head loss across a valve is commonly given by,

$$H_v = KV^2/(2g) = KQ^2/(2gA^2) \quad (1)$$

For a butterfly valve the loss coefficient $K$ is described well by,

$$K = c_o e^{-c x} \quad (2)$$

(See figure 1 under DATVAL)

in which $c$ is the base of natural logarithms, $c_o$ and $c$ are constants for a given valve, and $x$ is the degrees of opening of the butterfly valve with 0 degrees closed and 90 degrees opened. (The rule will actually completely open the valve if $x$ is 80 degrees or larger.)

The degree of valve opening $x$ in this rule is controlled by any of the following equations:

(a) For $H$ greater than $H_2$

$$x = x_o - (1-C_2 \Delta t)(C_o + C_2(H-H_2)) \quad (3)$$

(b) For $H$ less then $H_1$

$$x = x_o + (1-C_1 \Delta t)(C_o + C_2(H_1-H)) \quad (4)$$

(c) For $H_1 < H < H_2$ \quad (5)

$$x = x_o$$

in which $H_1$, $H_2$, $C_o$, $C_1$, and $C_2$ are coefficients (constant for any valve), that are provided as input as described below. $H$ is the head at the node specified in the rule whose head controls, and $x_o$ is the current opening of the valve in degrees, and for the first time step is provided as input, and $\Delta t$ is the time increment in seconds between individual solutions and is specified by input through INCHR or INCMIN. After the new valve opening $x$ is computed from one of the above equations, then the new flow rate, $Q$, which is specified the next time increment, is computed to satisfy the implicit equation,

$$K_pQ^n + KQ^2/(2gA^2) - H_t = 0 \quad (6)$$

in which $K_p$ and $n$ are the current coefficient and exponent, respectively that describe the frictional head loss in the pipe, and $H_t$ is the current total head loss (the difference between the upstream and downstream HGL elevations) in the pipe with the specified flow rate, which is in the pipe containing the butterfly valve.

The input data for this second rule for specifying flow rates indirectly through a rule which controls the valves opening consists of:

1. Negative of the pipe number whose flow rate is controlled by the rule.
2. Negative of the node number where the head $H$ is the control of the incremental valve setting.
3. The lower head, $H_j$.
4. The upper head, $H_2$.
5. The constant $c_o$ in Eq. 2, that determines the valve head loss.
6. The constant $c$ in Eq. 2.
7. An upstream HGL-elevation, $H_u$, that if given a value other than zero, will be used as the HGL-elevation at the upstream end of this pipe in place of the actual current HGL-elevation.
8. A downstream HGL-elevation, $H_d$, that if given a value other than zero, will be used as the HGL-elevation at the downstream end of this pipe in place of the actual current HGL-
Generally $H_u$ and $H_d$ will be given as zeros. However, if one wishes to have the specified flow rate computed based upon either of these variables, then both $H_u$ and $H_d$ would not be given for the same pipe (even though this could be done) because this would have the same effect as keeping the flow rate in the pipe constant.

9. The constant $C_0$ that controls the valve opening in Eq. 3 or 4.
10. The constant $C_1$ that controls the valve opening in Eq. 3 or 4.
11. The constant $C_2$ that controls the valve opening in Eq. 3 or 4.
12. The valve position at time zero, $x_o$ (0 degrees closed, 90 degrees opened)

If the initial valve opening $x_o$ is not known, then any negative value should be given for this input item. A negative value indicates to USU-NETWK that $x_o$ for use in Eqs. 3, 4 and 5 should be computed for the first time increment. Position $x_o$ must be between 0 and 90 degrees. Often the user will have a number of steady-state solutions available for the system at the time he prepares the time dependent data. It is from this steady-state information of the network's performance that he/she specifies the flow rate in the pipe, and then he/she can compute the valve opening by using the printed value from this solution for the head across the valve and the flow rate. First the loss coefficient $K$ is computed, and then $x$ from Eq. 2 is computed, and this becomes this 12th item of input.

**RQGIVE # 3:**

The third possibility of specifying flowrates is to specify what this flowrate should be based on input data. To select this method for specifying the flowrate in a designated pipe 10000 is added to the pipe number. As is the case with all input under SQGIVE and RQGIVE it is necessary that the flowrate in this (these) pipe(s) be specified in the original data by the command QGIVE. The data entered by the RQGIVE command consists of the following if the flowrate is to be given as a list of values related to reservoir water surface elevations.

1. The pipe number plus 10000
2. The reservoir number whose water surface controls the flowrate in the pipe identified by item # 1. This must be the reservoir number, and not the pipe or node that might identify the reservoir, i.e. NPUNOD is not implemented with this command. (Reservoir numbers are established by the order in which they appear in the original input data.)
3. The water surface elevation for the reservoir designated by item # 2 for which the following given flowrate in item # 4 applies.
4. The flowrate corresponding to the water surface elevation given in item # 3.
5. A higher or lower water surface elevation than given by item # 3 in the designated reservoir of item # 2.
6. The flowrate corresponding to the water surface elevation given by item # 5.
7. etc., a water surface elevation and a corresponding flowrate as pairs. The list must be terminated with a /. The water surface elevations in the reservoirs must be either ascending or descending in magnitude, i.e. they cannot be in random order. The flowrates will be linearly interpolated between the values, and if the water surface elevation drops below the smallest value given, or rises above the largest value given a linear extrapolation will occur in determining the value of the specified flowrate.

Example of RQGIVE command

```
RQGIVE
8 4 2000 200 0 450/ 
-10 -6 80 85 244 .05675 0. 0. 0. 10. 40.3/ 
-14 -4 85 88 244 .057 0. 0. .2 12. 60./
(see example network 20)
```

**SQGIVE**

The input data after the command SQGIVE consists of:

1. The pipe number whose flowrate is specified.
2. The time.
3. The new flowrate.
4. The new time.
5. The new flowrate, etc. with the list terminated with a ./

The times are in units as specified by NTIMED, and the flowrates are in the units denoted by NFLOW. Whenever the time at the next simulation solution equals or exceeds the time in the above list, then the flowrate in this pipe is changed to the new value. The flowrate may be negative, as well as positive.
with the meaning that the flow is reversed from the
direction given by the original upstream and
downstream nodes.

Example:

SQGIVE
5 8 1.5 12.0 15 0.5/
9 4 20 10.3 12 2.5/

SETPRS, SETHGS, SETPRR & SETHGR

These commands allow changes in specified pressures (or HGL-elevations) for simulation solutions. The two commands with PR as the 4th and 5th characters, respectively (e.g. SETPRS & SETPRR) are to be used if the pressure at a node is to change from time step to time step, and the two commands with HG as the 4th and 5th characters, respectively (e.g. SETHGS & SETHGR) are to be used if the elevation of the hydraulic grade line at a node is given. The last letter consisting of and "S" or and "R", respectively in these commands denotes "schedule" and "rule" respectively. That is the commands SETPRS & SETHGS introduce simulation data that provides the time schedule of changing pressures at a node, and the commands SETPRR & SETHGR introduce simulation data that provides the rules governing how pressures change at a node based on the hydraulic grade line elevation at some other node, or based on the water surface elevation in a reservoir.

In order to utilize these commands it is necessary that the pressure (or HGL elevation) be specified at the same nodes in the original input. Only nodes whose pressure (or HGL-elevation) were fixed in the original input can have their pressures change during a time-dependent solution. The demand at the node given in the original input is computed to satisfy the specified pressure. It is not possible to have the node whose demand is computed change to another node during a time-dependent solution.

SETPRS and SETHGS have the following input data after them:

(1) The node number whose pressure (or HGL-elevation) is to change as a function of time. This node number must be under the SETPR or SETHG command in the data providing the definition of the network.
(2) Time
(3) The pressure (or HGL-elevation) that is to apply at this time. If pressure is given then it must be in psi when using ES units and kiloPascal when using SI units.
(4) Time
(5) The pressure (or HGL-elevation) that is to apply at this time, etc. terminated with a 

The times must be given in increasing order, and the pressure (or HGL-elevation) given with the first time will apply until the simulation reaches this time. The last value given will apply for simulation times beyond the time associated with this value. The pressure or elevation of the HGL will not be changed until the simulation time exceeds the values given in the above list. In other words no interpolation, or extrapolation of the time schedule occurs.

SETPRR and SETHGR have the following input data after them:

(1) The node number whose pressure (or HGL-elevation) is to change as determined by this rule. This node number must be under the SETPR or SETHG command in the data providing the definition of the network.
(2) A node number (or a reservoir number) whose hydraulic grade line elevation controls the rule. A minus sign precedes this value if it is a reservoir number rather than a node number. In the later case the value must be the reservoir number, i.e. the order of its entry the original data, and not the pipe number that connects it to the network, or its node number if sources are numbered using the option NODESP=l.
(3) The elevation of the hydraulic grade line in feet when using ES units or meters when using SI units at the node (or reservoir number) given by item # 2 above.
(4) The corresponding pressure (or HGL-elevation) that is specified at the node of item # 1 above. When using ES units giving pressure it must be in psi, and when using SI units the pressure must be in kiloPascals.
(5) & (6) etc. terminated with a 

The rule will be linearly interpolated between
the values given in the above list regardless of the setting of LINEAR in the $TDATA list. The rule must be given such that the elevations of the hydraulic grade line at node of item # 2 above are in ascending order of magnitude. The value associated with the smallest elevation will apply until the elevation at node # 2 (or reservoir water surface elevation denote by #2) is exceeded. Likewise the last pressure (or HGL-elevation) in the list will apply whenever the first value in the last pair of values for the rule is exceeded.

Note: Use of a rule based on the HGL elevation at a node in the close proximity of the node whose pressure is being specified can result in a problem in which the rule forces pressure downward or upward rapidly during succeeding time steps. For example a rule that specifies a decreasing pressure as the pressure decreases at the node of item # 2 above will result in an increased demand to satisfy the decrease in pressure but will also reduce the pressure at node # 2, and the cycle will continue. The use of rules makes more sense generally if the pressure at node # 2 is not related directly to the pressure at node # 1.

Example

SETPRS
 2 1 78 2 75 3 70/
 4 28 80 4 78/
SEITRR
 2 91 4364 76 4366 78 4368.5 80 4371 82/
 4 55 4294 81 4296.5 83 4298.9 83 4301 84/
(see example network 23)

STORAGE FUNCTION

This command defines the volume of water in storage in reservoirs, or tanks as a function of the water surface elevation. If this relationship is not given for any tank, then its water surface will be held constant during the entire period of the time-dependent solution, as if the tank had an infinite capacity. The information that defines this volume-w.s. elevation relationship consists of two records, or lines, as described below.

Record # 1

1. The number that identifies the storage function.
2. A water surface elevation in the reservoir.
3. The volume of reservoir storage corresponding to the elevation in (2). The units of this storage volume are as specified by the value of ISUNIT in the $TDATA list of options.
4. A water surface elevation in the reservoir.
5. The volume of reservoir storage corresponding to the elevation in (4).
6. Etc. At least two sets of elevation-storage data are required. Terminate the input record with a slash, /.

Record # 2

A list of reservoir numbers, node numbers, or pipe numbers according to that indicated by the option NPUNOD that specify which reservoirs the above relationship apply to. This list will be a single reservoir designator unless more than one reservoir or tank is of the same size, and set at the same elevation.

Records (1) and (2) may be repeated as pairs for each different storage function.

Example of data after STORAGE FUNCTION
STORAG
1 284. 10. 288. 14. 292. 18./
1-3 5 7/
(see example networks 10, 11, 12, 13, 14, 15 and 20)

SVALVC, RVALVC & RVAL+P

Are commands similar to commands SVALVE and RVALVE in that the head loss caused by a valve is controlled. In SVALC, RVALVC and RVAL+P the data that are provided control the change in the opening of the valve from its current opening rather than the opening itself. Often in practice sensors, that activate a motor driven valve, determine whether the valve should be opened or closed further from the present opening. To keep the magnitude of transient pressures small these motor driven valves open and close slowly, especially when closing the valve. The commands described in this section permit more accurate simulation of the performance of such systems, especially if a small time increment is used in obtaining the time-dependent solution. The command SVALVC permits you to alter the equations that determine valve movement as a function of time. The command
RVALVC permits you to change these equations dependent upon reservoir water surface elevations, or nodal heads that exist at the current time step. If the equations that govern the change in valve opening do not change with either time or a condition elsewhere in the network, then either use of the command SVALVC or RVALVC will give identical results.

The command RVAL+P permits the control of the valve to change from a reservoir to the head at a node or vice versa. In using this command different coefficients can be given to the equations that control the valve's movement from a reservoir than those that govern the valve's movement from a node. Furthermore, the water level in the reservoir may always determine if and when control of the valve is switched from a reservoir to a node and back again depending upon conditions. The manner in which the opening or closing of the valve occurs is described by input data. This input data also dictates whether the reservoir or nodal head conditions control the switching and is described below under the heading RVAL+P.

SVALVC

The input data under the command SVALVC consists of the following items in the order given by the numbers with the list terminated by a slash, /.

1. The pipe number that contains the valve whose opening is being controlled.
2. The reservoir number whose water surface elevation controls the change in opening of the valve, or if a negative value is given, it is the node number whose head controls the change in opening of the valve. (The pipe that connects the reservoir to the network, or the node number of the reservoir (if NODESP=1) cannot be used for this value; it must be the reservoir number, e.g. the sequence number of the reservoir in the original input data.)
3. \( c_p \), the coefficient in Equation 1 below that determines the head loss due to the valve.
4. \( c \), the exponent in Equation 1 below that determines the head loss due to the valve.
5. \( K_p \), the coefficient in Equation 1 below that determines the head loss due to the valve.
6. \( K_f \), the coefficient in Equation 1 below that determines the head loss due to the valve.
7. \( b \), the exponent in Equation 1 below that determines the head loss due to the valve.
8. \( x_0 \), the initial opening of the valve. The range of this opening must be 0 for completely closed to 90 for fully open, e.g. the opening applies strictly to a butterfly valve with degrees of opening, but may be used for other valves as long as 90 represents fully open.
9. \( C_f \), a closing time coefficient as described by Equation 2 below that modifies the desired amount of closure dependent upon the time increment used in the time-dependent solution. \( \Delta t \) in Equation 2 is always in seconds. A value of \( C_f = 0 \) causes the full desired change in opening to take place. If \( C_f \) is given a value of one-half the reciprocal of the simulation time increment (i.e. for a 1-hour increment \( C_f = 0.5/3600=0.00013889 \), then one-half the desired opening will take place. \( C_f \) should be given a value so that the quantity \( (1 - C_f \Delta t) \) is within the limits of 1 to 0.
10. \( C_{pb} \), the constant in Equation 2 and 3 below.
11. \( C_2 \), the multiplier of the head difference in Equation 2 and 3 below.
12. \( C_3 \), an opening time coefficient as described by Equation 3 below. The same restrictions apply to \( C_2 \) as described under 9 above for \( C_3 \).
13. \( H_L \), the lower limit of head as described by the equations below.
14. \( H_2 \), the upper limit of head as described by the equations below.
15. The number of the parameter above (2 through 14) whose value is to change as a function of time as described by the data from item 16 and beyond.
16. Time 1 (the time when a parameter value is to change).
17. The new value of this parameter to be changed at time 1.
18. Time 2
19. The new value of this parameter to be changed at time 2, etc. terminating the list with a slash, /.

If item 14 is terminated with a slash, /, then no changes in any of the parameters will occur with time.

The head loss coefficient for the valve is dependent upon its opening as given by the following...
equation:

\[ K = c_0 e^{-cx} + K_0 + K_1 x^b \]  \hspace{1cm} (1)

in which e is the base of natural logarithms, and K is the loss coefficient in the equation \( h_L = KV^2/(2g) \), \( x \) is the opening of the valve with 0 closed and 90 fully open, and the other variables are as described above.

The equation that determines the new valve opening are:

(a) for \( H \) greater then \( H_2 \) (i.e. the valve is being closed)

\[ x = x_o - (1-C_1 \Delta t)(C_0 + C_2(H - H_2)) \]  \hspace{1cm} (2)

(b) for \( H \) less than \( H_1 \) (i.e. the valve is being opened)

\[ x = x_o + (1-C_2 \Delta t)(C_0 + C_2(H_1 - H)) \]  \hspace{1cm} (3)

(c) for \( H \) between the lower limit \( H_1 \) and upper limit \( H_2 \), i.e. \( H_1 \leq H \leq 6,22 \)

\[ x = x_o \]  \hspace{1cm} (no change in valve opening)  \hspace{1cm} (4)

Note from the nature of the above equations that if \( C_0 \) and \( C_2 \) are positive that \( x \) decreases, i.e. the valve closes when \( H \) is larger then \( H_2 \), and that \( x \) increases, i.e. the valve opens, when \( H \) is smaller than \( H_1 \). This operation applies if the pipe containing the valve supplies flow to the reservoir, and its closure reduces the flow into the reservoir. If the opposite control is desired in which an opening of a valve is to take place when \( H \) is larger than \( H_2 \), then this can be achieved by giving negative values to \( C_0 \) and \( C_2 \), etc.

Figure 1 under DATVAL shows the calibration of a butterfly valve, that was done at the Utah Water Research Laboratory at Utah State University. This figure shows the two common methods for describing the loss that a partly closed valve creates in a pipe line. One method gives the "flow coefficient" \( C_v \) and the other gives the "loss coefficient" \( K \). The above input requires that \( K \) be defined by the data, and not \( C_v \). Sometimes the valve opening is given as a fraction from 0 to 1 or from 0% to 100%. The input values above must be given so that 90 degrees is fully open.

RVALVC

The input data after the command RVALVC are identical to those described above for the command SVALVC up through and including item 15. Starting with item 16 the following apply:

16. The Head 1 in the reservoir, that if exceeded will cause the value of one of the item 2 through 14 in the input to be altered.
17. The new value of this item.
18. The head 2 in the reservoir, that if exceeded, will cause the value of one of the items 3 through 14 in the input to be altered.
19. The new value of this item, etc. terminating the list with a slash, /.

The list of heads and parameter values in item 16 through the slash / constitute a rule for the selected parameter, item 15. Linear interpolation will be used to determine the actual value of the parameter used in the above equations. The list of heads must be given in ascending order of magnitude.

RVAL+P

The command RVAL+P permits introduction of a valve that may be open or closed to control the flow in a pipe in response to either the water surface elevation in a reservoir or the head at a node depending upon conditions in the network. Control can switch between the two. For example, the control of the valve's movement may start with the water surface elevation in a reservoir. Later as this level drops below ( or rises above) a designated amount the control can be turned over to the equations based on the head at a node or vice versa. The accomplishment of the transfer of control is by means of what will be referred to as a "switch" in the description below. Input data controls the operation of the switch as well as the valve's movement. In understanding the operation of the switch you must separate its operation from the control valve. The switch can be pointing to the reservoir for example, but because of the level of the water in the reservoir, the control of the valve's movement may depend upon the head at a node. Considerable flexibility is permitted in determining under what conditions the switch changes from pointing to the reservoir to pointing to the node, as well as how the valve will be opened or closed. Table 1 summarizes the 24 different cases that are available. The selection of these cases is through the values provided to the input parameters \( H \) (the reservoir switch control head).
and $H_p$ (the nodal switch control head), through negative values for either the reservoir or node number, and through adding 2000 to these number designations. In preparing data to enter under the RVAL+P command you should find the case in Table 1 that agrees with the manner in which you want the valve’s movement to be alternately controlled by the reservoir and node and then make sure the correct values are provided as input.

If any negative value is given to $H_p$, then the switch will always point to the reservoir (Case 1 through 4 in Table 1). If the reservoir number is positive, with $H_p$ negative, then whenever the reservoir water surface elevation is above the given head, $H_r$, the control of the valve is by the reservoir, but whenever the reservoir water surface is below $H_r$ then the valve’s movement is controlled by the nodal head. (Case 1 and 2 in Table 1.) If this is to be reversed, i.e. the nodal head controls the valve whenever the water surface elevation in the reservoir is above $H_r$, and the reservoir controls whenever its water surface drops below $H_r$ then a minus should precede the reservoir number $N_r$. The latter possibilities are Case 3 and 4 in Table 1.

On the other hand if $H_r$ is negative, then the switch always points to the node, as given by Cases 5 through 8 in Table 1. If both the reservoir number, $N_r$, and the nodal number, $N_p$, are positive, then the node controls the valve whenever the head at the node is less than $H_p$, but the reservoir controls the valve whenever this head exceeds $H_p$. The reverse of this valve control is accomplished by placing a minus immediately in front of the node number, $N_p$ (Case 7 and 8).

In Cases 9 through 16 the switch changes from pointing to the reservoir or node to coincide with where the control of the valve resides. If the pointing of the switch is to change only if both the reservoir level and nodal pressure are both met, then 2000 is added to the node number, $N_p$, or the reservoir number, $N_r$, as shown by Cases 7 and 8.

The input data under the command RVAL+P consists of the following:

1. The pipe number that contains the valve whose opening is to be controlled.

2. The reservoir number whose water surface acts as a controller of the movement of the valve’s opening.

3. $c_o$ - the coefficient in Equation 1 above (under SVALVC) that determines the head loss due to the valve.

4. $c$ - the exponent in Equation 1 above that determines the head loss due to the valve.

5. $K_o$ - the coefficient in Equation 1 above that determines the head loss due to the valve.

6. $K_f$ - the coefficient in Equation 1 above that determines the head loss due to the valve.

7. $b$ - the exponent in Equation 1 above that determines the head loss due to the valve.

8. $x_0$ - the initial opening of the valve. The range of this opening must be 0 for entirely closed to 90 for fully open, i.e., the opening applies strictly to a butterfly valve with degrees of opening but may be used for other valves as long as 90 represents fully opened.

9. $C_I$ - A closing item coefficient as described by Equation 2 above, that modifies the desired amount of closure dependent upon the time increment used in the simulation solution. $\Delta t$ in Equation 2 is always in seconds. A value of $C_I = 0$ causes the full desired change in opening to take place. If $C_I$ is given a value of one-half the reciprocal of the simulation time increment (i.e. for a 1-hour increment $C_I = .5/3600 = 1.38889 \times 10^{-4}$), then one-half the desired opening will take place. $C_I$ should be given a value so that the quantity $(1 - C_I \Delta t)$ is within the limits of 1 to 0.

10. $C_0$ - the constant in Equation 2 and 3 above.

11. $C_2$ - the multiplier of the head difference in Equation 2 and 3 above.

12. $C_3$ - an opening time coefficient as described by Equation 3 above. See the
explanation of $C_7$ above to assign appropriate values to $C_3$.

13. $H_1$ - the lower limit of head as described by the equations above.

14. $H_2$ - the upper limit of head as described by the equations above.

15. The node number that is to be used to control the movement of the valve's opening. The parameters given as 18 through 22 below apply for this modal control and can have different values than those above that apply for reservoir control.

16. $H_r$ - the reservoir head that is used to control the pointing of the switch to either the reservoir or the node.

17. $H_p$ - the nodal head that is used to control the pointing of the switch to either the reservoir or the node.

18. $C_7$ - the closing time coefficient.

19. $C_9$ - the constant in Equations 2 and 3 above.

20. $C_2$ - the multiplier of the head difference in Equations 2 and 3 above.


22. $H_l$ - the lower limit of head as described by the equations above.

23. $H_2$ - the upper limit of head as described by the equations above.

The coefficients, etc. given as items 9 through 14 above are used to determine the incremental opening or closing of the valve whenever it is controlled by the reservoir water surface elevation, and therefore items 13 and 14 are reservoir water heads. Items 18 through 23 determine the incremental opening or closing of the valve whenever it is controlled by the head at the node given as item 15. Don't get confused by the duplication of symbols $C_9$, $C_2$ and $C_3$ in items 10 through 12 and items 18 through 21, or $H_l$ and $H_2$ or items 13 and 14, and $H_r$ and $H_2$ or item 22 and 23. The first apply in the equations for reservoir control and the latter for same equations by nodal head control. The values of the C's items 9 through 12 and items 18 through 21, may be the same but generally will be different. Likewise the same is true for the H's.

Table 1. Different cases that can be selected from in controlling the opening of a valve by either the water surface elevation in a reservoir, or the head at a node, and how the switch which points to one of these controllers is directed through input data.

<table>
<thead>
<tr>
<th>Case</th>
<th>Res.</th>
<th>Node</th>
<th>$H_r$</th>
<th>$H_p$</th>
<th>Current Control Pointing at switch</th>
<th>Sketch showing the condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$-H_r$</td>
<td>+</td>
<td>res.</td>
<td>res.</td>
<td>$H_r$</td>
<td>Sketch showing the condition</td>
</tr>
<tr>
<td>2</td>
<td>$-H_r$</td>
<td>+</td>
<td>node</td>
<td>res.</td>
<td>$H_r$</td>
<td>Sketch showing the condition</td>
</tr>
<tr>
<td>3</td>
<td>$-H_r$</td>
<td>+</td>
<td>node</td>
<td>node</td>
<td>$H_r$</td>
<td>Sketch showing the condition</td>
</tr>
<tr>
<td>4</td>
<td>$-H_r$</td>
<td>+</td>
<td>node</td>
<td>node</td>
<td>$H_r$</td>
<td>Sketch showing the condition</td>
</tr>
<tr>
<td>5</td>
<td>$-H_r$</td>
<td>+</td>
<td>node</td>
<td>node</td>
<td>$H_r$</td>
<td>Sketch showing the condition</td>
</tr>
<tr>
<td>6</td>
<td>$-H_r$</td>
<td>+</td>
<td>node</td>
<td>node</td>
<td>$H_r$</td>
<td>Sketch showing the condition</td>
</tr>
<tr>
<td>7</td>
<td>$-H_r$</td>
<td>+</td>
<td>node</td>
<td>node</td>
<td>$H_r$</td>
<td>Sketch showing the condition</td>
</tr>
<tr>
<td>8</td>
<td>$-H_r$</td>
<td>+</td>
<td>node</td>
<td>node</td>
<td>$H_r$</td>
<td>Sketch showing the condition</td>
</tr>
<tr>
<td>9</td>
<td>$+H_r$</td>
<td>+</td>
<td>res.</td>
<td>res.</td>
<td>$H_r$</td>
<td>Sketch showing the condition</td>
</tr>
<tr>
<td>10</td>
<td>$+H_r$</td>
<td>+</td>
<td>node</td>
<td>res.</td>
<td>$H_r$</td>
<td>Sketch showing the condition</td>
</tr>
<tr>
<td>11</td>
<td>$+H_r$</td>
<td>+</td>
<td>node</td>
<td>node</td>
<td>$H_r$</td>
<td>Sketch showing the condition</td>
</tr>
<tr>
<td>12</td>
<td>$+H_r$</td>
<td>+</td>
<td>node</td>
<td>node</td>
<td>$H_r$</td>
<td>Sketch showing the condition</td>
</tr>
<tr>
<td>13</td>
<td>$+H_r$</td>
<td>+</td>
<td>node</td>
<td>node</td>
<td>$H_r$</td>
<td>Sketch showing the condition</td>
</tr>
<tr>
<td>14</td>
<td>$+H_r$</td>
<td>+</td>
<td>node</td>
<td>node</td>
<td>$H_r$</td>
<td>Sketch showing the condition</td>
</tr>
<tr>
<td>15</td>
<td>$+H_r$</td>
<td>+</td>
<td>node</td>
<td>node</td>
<td>$H_r$</td>
<td>Sketch showing the condition</td>
</tr>
<tr>
<td>16</td>
<td>$+H_r$</td>
<td>+</td>
<td>node</td>
<td>node</td>
<td>$H_r$</td>
<td>Sketch showing the condition</td>
</tr>
<tr>
<td>17</td>
<td>$+H_r$</td>
<td>+</td>
<td>node</td>
<td>node</td>
<td>$H_r$</td>
<td>Sketch showing the condition</td>
</tr>
<tr>
<td>18</td>
<td>$+H_r$</td>
<td>+</td>
<td>node</td>
<td>node</td>
<td>$H_r$</td>
<td>Sketch showing the condition</td>
</tr>
<tr>
<td>19</td>
<td>$+H_r$</td>
<td>+</td>
<td>node</td>
<td>node</td>
<td>$H_r$</td>
<td>Sketch showing the condition</td>
</tr>
<tr>
<td>20</td>
<td>$+H_r$</td>
<td>+</td>
<td>node</td>
<td>node</td>
<td>$H_r$</td>
<td>Sketch showing the condition</td>
</tr>
<tr>
<td>21</td>
<td>$+H_r$</td>
<td>+</td>
<td>node</td>
<td>node</td>
<td>$H_r$</td>
<td>Sketch showing the condition</td>
</tr>
<tr>
<td>22</td>
<td>$+H_r$</td>
<td>+</td>
<td>node</td>
<td>node</td>
<td>$H_r$</td>
<td>Sketch showing the condition</td>
</tr>
<tr>
<td>23</td>
<td>$+H_r$</td>
<td>+</td>
<td>node</td>
<td>node</td>
<td>$H_r$</td>
<td>Sketch showing the condition</td>
</tr>
<tr>
<td>24</td>
<td>$+H_r$</td>
<td>+</td>
<td>node</td>
<td>node</td>
<td>$H_r$</td>
<td>Sketch showing the condition</td>
</tr>
</tbody>
</table>

-109-
ILLUSTRATIVE EXAMPLES OF NETWORKS AND INPUT DATA USING THE STANDARD FORM OF INPUT

The following pages contain small networks to illustrate data requirements. The input is given on the same page or next page to the network and generally the data is illustrated twice. Once using the PIPES and NODES command and once the PIPE-command to replace these two commands, or other possible variations of input. These are very small networks.

Examples of data using the other forms of input are given in Appendix A and B.

EXAMPLE PROBLEM 1

The input for the network below is given in two different forms below the sketch. In the first column the commands PIPES and NODES are used to define the networks layout. In the second column the command PIPE- is used to enter this same data. The solution output using either of these forms of data is shown on the next page. After this solution the data for this same network has been prepared numbering the source of supply as Nodes 10, 11, and 12 and indicating this with NODESP = 1. The solution from this data is shown on the page thereafter.

<table>
<thead>
<tr>
<th>PUMP CHARACTERISTICS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q₁ (cfs)</td>
</tr>
<tr>
<td>----------</td>
</tr>
<tr>
<td>3.0</td>
</tr>
</tbody>
</table>

TWO ALTERNATIVE DATA DECK FOR THIS NETWORK

**EXAMPLE NUMBER 1 OF SIMPLE NETWORK**

`$/ SPECIFIC NFLOW=0,NPGPM=0,NPRINT=-1, OUTPUT=0 $END`  

**PIPES**

1 0 2 500 12 .0102  
5 2 1 1300 10/  
3 3 4 1600 8/  
4 1 4 2000 10/  
2 2 3 2200 8/  
6 3 5/  
8 5 4 3000/  
10 4 6 2000/  
9 5 6 1400/  
7 0 5 1000 12/  
11 0 1 500/  

**NODES**

2 2 300  
1 1 230  
3 1.5 280  
4 6 200  
5 1 270  
6 3 200  

**PUMPS**

7 3 80 4 77 5 72 350  

**RESER**

1 400  
11 350  

**MINOR**

4 10  

**VALVE**

8 1500 320  

**RUN**

---

EXAMPLE NUMBER 1 OF SIMPLE NETWORK

`$/ SPECIFIC NFLOW=0,NPGPM=0,NPRINT=-1 $END`  

**PIPES**

1 12. 500. 2 2. 300.  
5 10. 1300. 2 1 1. 280.  
3 8. 1600. 3 4 6. 200.  
4 10. 2000. 1 4  
2 8. 2200. 2 3 1.5 280.  
6 8. 2200. 5 3  
8 8. 3000. 5 4 6. 200.  
9 6. 1400. 5 6  
7 12. 1000. 5 1. 270.  
11 12. 500. 1  

**PUMPS**

7 3 80 4 77 5 72 350  

**RESER**

1 400  
11 350  

**MINOR**

4 10  

**VALVE**

8 1500 320  

**RUN**

---
Pipe Network Analysis Based on the Corrective Flowrate Equations

**Title Given to Network**

---

**Example No. 1 of Simple Network**

*All demand flows are multiplied by 1.0000*

<table>
<thead>
<tr>
<th>PIPES</th>
<th>NODES</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>6</td>
</tr>
</tbody>
</table>

- **Source Pumps:** 1
- **Booster Pumps:** 0
- **Reservoirs:** 2
- **Minor Loss:** 1
- **Pumps:** 0

**Pressure Receiving Valves & Control**: 1

**Pipe Network Head Elevation**

1. 400.00
2. 350.00

**Model Type**: 6

**Minor Loss Coefficients**

**Flow from Pumps and Reservoirs**

14,500

**No. of Real Loops**: 2
**No. of Pseudo Loops**: 3

---

**Results of Solution**

<table>
<thead>
<tr>
<th>Pipe No. (IDPA)</th>
<th>0.03525</th>
<th>0.2500</th>
<th>8.358</th>
<th>1.0000</th>
</tr>
</thead>
<tbody>
<tr>
<td>517</td>
<td>100.00</td>
<td>0.2500</td>
<td>8.358</td>
<td>1.0000</td>
</tr>
<tr>
<td>518</td>
<td>100.00</td>
<td>0.2500</td>
<td>8.358</td>
<td>1.0000</td>
</tr>
<tr>
<td>519</td>
<td>100.00</td>
<td>0.2500</td>
<td>8.358</td>
<td>1.0000</td>
</tr>
<tr>
<td>520</td>
<td>100.00</td>
<td>0.2500</td>
<td>8.358</td>
<td>1.0000</td>
</tr>
<tr>
<td>521</td>
<td>100.00</td>
<td>0.2500</td>
<td>8.358</td>
<td>1.0000</td>
</tr>
<tr>
<td>522</td>
<td>100.00</td>
<td>0.2500</td>
<td>8.358</td>
<td>1.0000</td>
</tr>
<tr>
<td>523</td>
<td>100.00</td>
<td>0.2500</td>
<td>8.358</td>
<td>1.0000</td>
</tr>
<tr>
<td>524</td>
<td>100.00</td>
<td>0.2500</td>
<td>8.358</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

---

**Example No. 1 of Simple Network**

- **Numbering Sources As Nodes**
- **Specific Nplegn=1, Nodesp=1**

---

**END**

---

**Table:**

<table>
<thead>
<tr>
<th>Pipe</th>
<th>Head</th>
<th>Flow</th>
<th>Pressure</th>
<th>Head</th>
<th>Pressure</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>517</td>
<td>100.00</td>
<td>8.358</td>
<td>1.0000</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>518</td>
<td>100.00</td>
<td>8.358</td>
<td>1.0000</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>519</td>
<td>100.00</td>
<td>8.358</td>
<td>1.0000</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>520</td>
<td>100.00</td>
<td>8.358</td>
<td>1.0000</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>521</td>
<td>100.00</td>
<td>8.358</td>
<td>1.0000</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>522</td>
<td>100.00</td>
<td>8.358</td>
<td>1.0000</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>523</td>
<td>100.00</td>
<td>8.358</td>
<td>1.0000</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>524</td>
<td>100.00</td>
<td>8.358</td>
<td>1.0000</td>
<td></td>
</tr>
</tbody>
</table>
PIPE NETWORK ANALYSIS BASED ON THE CORRECTIVE FLOWRATE EQUATIONS

TITLE GIVEN TO NETWORK

EXAMPLE NO. 1 OF SIMPLE NETWORK
ALL DEMAND FLOWS ARE MULTIPLIED BY 1.0000
PIVES 11
NODES 9
SOURCE PUMPS 1
BOOSTER PUMPS 0
RESERVOIRS 2
MINOR LOSS 1
PRVS 1
NOZZLES 0
CHECK VALUES 0
BACK PRESS V 0

PUMPS:
PIPE 2ND ORDER COEF LINEAR COEF SHUT-OFF HEAD SUMP ELEV
7 -1.000 4.000 77.00 350.00

RESERVOIRS PLUS NOZZLES:
NO. PIPE NODE ELEVATION
1 1 10 400.00
2 11 12 350.00

MINOR LOSS PIPES AND THEIR COEFFICIENTS:
PIPE K
4 10.00

PRESSURE REDUCING VALUES (1):
PIPE DOWNSTREAM DOWNSTREAM DISTANCE HGL
2 400.00 320.00

MINOR LOSSES:
PIPE MINOR LENGTH + EQUIVALENT NO. LOSS LENGTH
4 10.00 2410.40

FLOW FROM PUMPS AND RESERVOIRS EQUALS 14.500

NO. OF REAL LOOPS 2 NO. OF PSEUDO LOOPS 3 NL= 3

ITERATION= 1 SUM OF DIFFERENCES= .1214002
ITERATION= 2 SUM OF DIFFERENCES= .1444001
ITERATION= 3 SUM OF DIFFERENCES= .3744001
ITERATION= 4 SUM OF DIFFERENCES= .1534001
ITERATION= 5 SUM OF DIFFERENCES= .1184000
ITERATION= 6 SUM OF DIFFERENCES= .863-003

PUMPS:
NODE PIPE HEAD FLOW
11 7 63.81 6.15

ELEVATION OF HGL UPSTREAM AND DOWNSTREAM OF PRVS:
PIPE UPSTREAM DOWNSTREAM HGL HGL
8 366.85 320.00

UNITS OF SOLUTION ARE:
DIAMETERS - INCH
LENGTH - FEET
HEADS - FEET
ELEVATIONS - FEET
PRESSURES - PSI
FLOWRATES- CFs
DARCY-WEISBACH FORMULA USED FOR COMPUTING HEAD LOSS

PIPE DATA:

<table>
<thead>
<tr>
<th>PIPE</th>
<th>NODES</th>
<th>LENGTH</th>
<th>DIAM</th>
<th>COEF</th>
<th>FLOW RATE</th>
<th>VELOCITY</th>
<th>HEAD LOSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>500.</td>
<td>12.0</td>
<td>.010200</td>
<td>8.31</td>
<td>10.58</td>
<td>14.79</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>2200.</td>
<td>8.0</td>
<td>.010200</td>
<td>1.83</td>
<td>5.24</td>
<td>30.63</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>1600.</td>
<td>8.0</td>
<td>.010200</td>
<td>2.49</td>
<td>7.14</td>
<td>40.99</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>2410.</td>
<td>10.0</td>
<td>.010200</td>
<td>3.53</td>
<td>6.46</td>
<td>38.43</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1300.</td>
<td>10.0</td>
<td>.010200</td>
<td>4.48</td>
<td>8.21</td>
<td>33.22</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>2200.</td>
<td>8.0</td>
<td>.010200</td>
<td>2.14</td>
<td>6.20</td>
<td>42.72</td>
</tr>
<tr>
<td>7</td>
<td>11</td>
<td>5000.</td>
<td>12.0</td>
<td>.010200</td>
<td>6.13</td>
<td>7.83</td>
<td>18.52</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>3000.</td>
<td>8.0</td>
<td>.010200</td>
<td>1.15</td>
<td>3.29</td>
<td>16.86</td>
</tr>
<tr>
<td>9</td>
<td>6</td>
<td>1400.</td>
<td>8.0</td>
<td>.010200</td>
<td>1.83</td>
<td>9.34</td>
<td>87.42</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>2000.</td>
<td>10.0</td>
<td>.010200</td>
<td>1.17</td>
<td>2.14</td>
<td>3.71</td>
</tr>
<tr>
<td>11</td>
<td>12</td>
<td>500.</td>
<td>12.0</td>
<td>.010200</td>
<td>.05</td>
<td>.06</td>
<td>0.00</td>
</tr>
</tbody>
</table>

NODE DATA:

<table>
<thead>
<tr>
<th>NODE</th>
<th>DEMAND (CFs)</th>
<th>ELEV</th>
<th>HEAD</th>
<th>PRESSURE</th>
<th>HGL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.000</td>
<td>280.</td>
<td>70.0</td>
<td>30.33</td>
<td>350.00</td>
</tr>
<tr>
<td>2</td>
<td>2.000</td>
<td>300.</td>
<td>83.21</td>
<td>36.06</td>
<td>381.21</td>
</tr>
<tr>
<td>3</td>
<td>1.500</td>
<td>280.</td>
<td>72.56</td>
<td>31.44</td>
<td>352.56</td>
</tr>
<tr>
<td>4</td>
<td>6.000</td>
<td>200.</td>
<td>111.57</td>
<td>48.35</td>
<td>311.57</td>
</tr>
<tr>
<td>5</td>
<td>1.000</td>
<td>270.</td>
<td>125.28</td>
<td>54.29</td>
<td>395.28</td>
</tr>
<tr>
<td>6</td>
<td>3.000</td>
<td>200.</td>
<td>107.86</td>
<td>46.74</td>
<td>307.86</td>
</tr>
<tr>
<td>11</td>
<td>-6.147</td>
<td>290.</td>
<td>123.81</td>
<td>53.65</td>
<td>413.81</td>
</tr>
<tr>
<td>10</td>
<td>-3.000</td>
<td>300.</td>
<td>100.00</td>
<td>43.33</td>
<td>400.00</td>
</tr>
<tr>
<td>12</td>
<td>-3.064</td>
<td>280.</td>
<td>70.00</td>
<td>30.33</td>
<td>350.00</td>
</tr>
</tbody>
</table>

-113-
This page illustrates four modifications that might be used in preparing the data for the network of example number 1. In the first two modifications the network's topology is being defined twice to help assure that no errors were made in preparing the input data. In the first modification this additional data is provided on the records after the NODES command. In the second modification this additional data is provided on separate records after the RUN command, allowing for their easy removal after the initial analysis has verified that the input data does properly define the network. If NETCHK had been set = -1 (or -1) instead of 2 (or -2) it would not have been necessary to distinguish whether the flow is assumed into the junction by preceding the pipe number with a minus sign.

In the third modification the command PIPE- replaces the commands PIPES and NODES and finally in the fourth modification loops are specified by external data rather than internally determined by the program.

**EXAMPLE PROBLEM 1A**

The input data listed below shows what is needed to obtain a special design solution for the network in Example Number 1, in which the pipe 8 containing the PRV has been removed. This special design solution gives the diameter of as many pipes as there are nodes in the network in order to satisfy the pressures that are specified at all nodes of the network. The following changes between the data that follows and the input data for Example Number 1 should be noted: 1. The inclusion of DESIGN = 1 (denoting that a special design solution is to be obtained), HGL ELEV (denoting that HGL elevations will define nodal pressures), and NOMSOL = 1 (denoting that an analysis using the closest nominal diameter is to follow the design solution). 2. Diameters for 6 pipes, 1, 5, 4, 6, 10 and 9 are given diameters equal to zero (the number of such zeros must equal the number of nodes in the network). 3. The nodal pressures (given as HGL elevations) follow the elevation on the cards entered through the NODES command. Note that special devices such as PRV's, check valves, BPV, and minor loss devices cannot be included as part of the network when requesting a special design solution. The use of the Hazen-Williams equation instead of the Darcy-Weisbach equation, as was used in Example 1, is only to illustrate that different friction loss equations can be used.
EXAMPLE NO 1 USING DESIGN CAPABILITY
/
$SPECIF NFLOW=0,NPGPM=0,IHGL=2,
DESIGN=1,NOMSOL=1$END
PIPEC
1 0 2 500 0. 130
5 2 1 1300 /
3 3 4 1600 8 /
4 1 4 2000 0 /
2 2 3 2200 8 /
6 3 5 2200 0 /
10 4 6 2000 0 /
9 5 6 1400 /
7 0 5 1000 10 /
11 0 1 500 12 /
NODES
2 2 300 383
1 1 230 348
3 1 5 280 353
4 6 200 312
5 1 270 395
6 3 200 308
PUMPS
7 3 80 4 77 5 72 350
RESER
1 400
11 350
RUN
SOLUTION TO DESIGN #1A
TITLE GIVEN TO NETWORK
-----------------------------
EXAMPLE NO 1 USING DESIGN CAPABILITY

ALL DEMAND FLOWS ARE MULTIPLIED BY 1.0000

PIPEC
10
NODES
6
SOURCE PUMPS
1
BOOSTER PUMPS
0
RESERVOIRS
2
MINOR LO OSSE
0
PRVS
0
NOZZLES
0
CHECK VALVES
0
BACK PRES V.
0
DIF. HEAD DEV
0

PUMPS:
PIPE 2ND ORDER COEF LINEAR COEF SHUT-OFF HEAD SUMP ELEV
7 -1.000 4.000 77.00 350.00

THE FOLLOWING PIPE NUMBERS WERE NOT USED IN YOUR NETWORK:

RESERVOIRS PLUS NOZZLES:
NO. PIPE NODE ELEVATION
1 1 0 400.00
2 11 0 350.00

NODE NO. DEMAND PIPES AT NODE
-----------------------------
1 2 2.000 -1 5 2
2 1 1.000 -5 4 -11
3 3 1.500 -3 -2 6
4 4 6.000 -3 -4 10
5 5 1.000 -6 9 -7
6 6 3.000 -10 -9

-115-
## DESIGN PIPE DIAMETERS

<table>
<thead>
<tr>
<th>PIPE DIAMETER</th>
<th>DIAM. NOM. AREA</th>
<th>LENGTH</th>
<th>COEF.</th>
<th>FLOW RATE</th>
<th>HEAD LOSS</th>
<th>VEL.</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO. FROM TO</td>
<td>DIA.</td>
<td>FLOW</td>
<td>VEL.</td>
<td>LOSS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 0 2</td>
<td>10.00 0.624</td>
<td>10.0</td>
<td>500.</td>
<td>130.</td>
<td>6.684</td>
<td>10.72</td>
</tr>
<tr>
<td>5 2 1</td>
<td>8.04 0.352</td>
<td>8.0</td>
<td>130.</td>
<td>130.</td>
<td>2.782</td>
<td>7.89</td>
</tr>
<tr>
<td>3 3 4</td>
<td>8.00 0.349</td>
<td>1600</td>
<td>130.</td>
<td>130.</td>
<td>2.674</td>
<td>7.66</td>
</tr>
<tr>
<td>4 1 4</td>
<td>10.60 0.613</td>
<td>2000</td>
<td>130.</td>
<td>130.</td>
<td>4.631</td>
<td>7.56</td>
</tr>
<tr>
<td>2 2 3</td>
<td>8.00 0.349</td>
<td>2200</td>
<td>130.</td>
<td>130.</td>
<td>1.902</td>
<td>5.45</td>
</tr>
<tr>
<td>6 5 3</td>
<td>7.99 0.348</td>
<td>2200</td>
<td>130.</td>
<td>130.</td>
<td>2.272</td>
<td>6.53</td>
</tr>
<tr>
<td>10 4 6</td>
<td>10.28 0.577</td>
<td>2000</td>
<td>130.</td>
<td>130.</td>
<td>1.902</td>
<td>5.45</td>
</tr>
<tr>
<td>9 5 6</td>
<td>7.99 0.348</td>
<td>2200</td>
<td>130.</td>
<td>130.</td>
<td>2.272</td>
<td>6.53</td>
</tr>
<tr>
<td>7 0 5</td>
<td>10.00 0.545</td>
<td>1000</td>
<td>130.</td>
<td>130.</td>
<td>4.967</td>
<td>9.11</td>
</tr>
<tr>
<td>11 0 1</td>
<td>12.00 0.785</td>
<td>500.</td>
<td>130.</td>
<td>130.</td>
<td>2.849</td>
<td>3.63</td>
</tr>
</tbody>
</table>

### NODE DATA

<table>
<thead>
<tr>
<th>NODE</th>
<th>DEMAND</th>
<th>ELEV.</th>
<th>HEAD</th>
<th>PRESSURE</th>
<th>HGL</th>
<th>ELEV.</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2.000</td>
<td>300.</td>
<td>83.00</td>
<td>35.97</td>
<td>383.00</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1.000</td>
<td>230.</td>
<td>118.00</td>
<td>51.13</td>
<td>348.00</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1.500</td>
<td>280.</td>
<td>73.00</td>
<td>31.63</td>
<td>353.00</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>6.000</td>
<td>200.</td>
<td>112.00</td>
<td>48.53</td>
<td>312.00</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1.000</td>
<td>270.</td>
<td>125.00</td>
<td>54.17</td>
<td>395.00</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>3.000</td>
<td>200.</td>
<td>108.00</td>
<td>46.80</td>
<td>308.00</td>
<td></td>
</tr>
</tbody>
</table>

Flow from pumps and reservoirs equals 14.500

Iteration 1: Sum of differences = 0.109E+01

Iteration 2: Sum of differences = 0.513E-01

Iteration 3: Sum of differences = 0.109E-03

Losses due to fluid friction in all pipes:

Power loss = 113.800 H.P. = 84.895 K.WATTS.

Energy loss = 2037.479 KWHRS / DAY

Pumps:

<table>
<thead>
<tr>
<th>PIPE</th>
<th>HEAD</th>
<th>FLOW</th>
<th>HORSEPOWER</th>
<th>KILOWATTS</th>
<th>KWATTS-HRS/DAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>70.19</td>
<td>5.29</td>
<td>42.11</td>
<td>31.41</td>
<td>753.94</td>
</tr>
</tbody>
</table>

Units of solution are:

- Diameters: Inch
- Length: Feet
- Heads: Feet
- Elevations: Feet
- Pressures: PSI
- Flowrates: CFS

Hazen-Williams formula used for computing head loss.

Pipe data:

<table>
<thead>
<tr>
<th>PIPE NO.</th>
<th>NODES</th>
<th>FROM</th>
<th>TO</th>
<th>LENGTH</th>
<th>DIAM</th>
<th>COEF</th>
<th>FLOW</th>
<th>RATE</th>
<th>VELOCITY</th>
<th>HEAD</th>
<th>LOSS /1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0 2</td>
<td>500.</td>
<td>10.0</td>
<td>130.0</td>
<td>6.43</td>
<td>1.91</td>
<td>5.47</td>
<td>30.25</td>
<td>13.75</td>
<td>21.97</td>
<td>43.95</td>
</tr>
<tr>
<td>2</td>
<td>2 3</td>
<td>2200.0</td>
<td>8.0</td>
<td>130.0</td>
<td>2.69</td>
<td>7.70</td>
<td>41.38</td>
<td>25.86</td>
<td>20.85</td>
<td>29.93</td>
<td>23.03</td>
</tr>
<tr>
<td>3</td>
<td>4 1</td>
<td>2000.0</td>
<td>10.0</td>
<td>130.0</td>
<td>4.30</td>
<td>7.89</td>
<td>41.69</td>
<td>20.85</td>
<td>20.85</td>
<td>29.93</td>
<td>21.02</td>
</tr>
<tr>
<td>4</td>
<td>1 4</td>
<td>1300.0</td>
<td>8.0</td>
<td>130.0</td>
<td>2.52</td>
<td>7.23</td>
<td>29.93</td>
<td>20.85</td>
<td>20.85</td>
<td>29.93</td>
<td>23.03</td>
</tr>
<tr>
<td>5</td>
<td>2 1</td>
<td>2200.0</td>
<td>8.0</td>
<td>130.0</td>
<td>2.28</td>
<td>6.52</td>
<td>41.85</td>
<td>19.02</td>
<td>19.02</td>
<td>29.93</td>
<td>23.03</td>
</tr>
<tr>
<td>6</td>
<td>3 5</td>
<td>1000.0</td>
<td>10.0</td>
<td>130.0</td>
<td>5.29</td>
<td>9.70</td>
<td>40.56</td>
<td>30.56</td>
<td>20.85</td>
<td>29.93</td>
<td>23.03</td>
</tr>
<tr>
<td>7</td>
<td>0 5</td>
<td>1300.0</td>
<td>6.0</td>
<td>130.0</td>
<td>2.01</td>
<td>10.24</td>
<td>30.56</td>
<td>30.56</td>
<td>30.56</td>
<td>29.93</td>
<td>23.03</td>
</tr>
<tr>
<td>9</td>
<td>6 4</td>
<td>1400.0</td>
<td>6.0</td>
<td>130.0</td>
<td>1.99</td>
<td>1.81</td>
<td>1.37</td>
<td>1.37</td>
<td>1.37</td>
<td>30.56</td>
<td>30.56</td>
</tr>
<tr>
<td>10</td>
<td>4 6</td>
<td>2000.0</td>
<td>10.0</td>
<td>130.0</td>
<td>0.99</td>
<td>1.81</td>
<td>1.37</td>
<td>1.37</td>
<td>1.37</td>
<td>30.56</td>
<td>30.56</td>
</tr>
</tbody>
</table>

Node data:

<table>
<thead>
<tr>
<th>NODE</th>
<th>DEMAND</th>
<th>(CFS)</th>
<th>ELEV.</th>
<th>HEAD</th>
<th>PRESSURE</th>
<th>HGL</th>
<th>ELEV.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.000</td>
<td>448.83</td>
<td>230.</td>
<td>118.09</td>
<td>51.17</td>
<td>348.09</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2.000</td>
<td>897.66</td>
<td>300.</td>
<td>78.03</td>
<td>33.81</td>
<td>376.03</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1.500</td>
<td>673.24</td>
<td>280.</td>
<td>67.78</td>
<td>29.37</td>
<td>347.78</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>6.000</td>
<td>2692.98</td>
<td>200.</td>
<td>106.40</td>
<td>46.11</td>
<td>306.40</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1.000</td>
<td>448.83</td>
<td>270.</td>
<td>119.63</td>
<td>51.84</td>
<td>399.63</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>3.000</td>
<td>1346.49</td>
<td>200.</td>
<td>103.66</td>
<td>44.92</td>
<td>303.66</td>
<td></td>
</tr>
</tbody>
</table>

---

-116-
EXAMPLE PROBLEM 1B

This is a another variation of problem # 1 in which the option LENGON=0 has been turned off by giving this parameter a value of zero. By so doing the network program is told that: 1. the lengths of pipes will not be given in the data that follows the PIPES or the PIPE- command, and 2. the x and y coordinates of the nodes will be given following the elevation after the NODES command, or if the PIPE- command is used then the x and y coordinates will be provided after the elevation wherever this elevation is given. The lengths of the pipes are computed from the x and y coordinates.

The use of this option is especially attractive if you plan to have the computer make a plot of the network and its solution, because in making such a plot the x and y coordinates are needed anyway.
In this variation to Example Problem 1 we wish to specify the magnitude of the flowrates in pipes 2 and 10 to be 2.5 cfs and 0.5 cfs respectively. In order for these flowrates to occur it will be necessary that either a booster pump or valve (or other energy dissipating device) exist in these pipes, depending upon whether the flowrate would otherwise be less than or greater than these amounts, respectively. In addition to giving the solution of the network with these specified flowrates, the computer program will give the required amounts of head in those pipes whose flowrates have been specified through the command QGIVE. Such given positive heads will require a booster pump that will produce the given head with the specified flowrate through it. Such given negative heads will require a valve or other loss device to dissipate this amount when the flowrate equals the amount specified. In the solution to the input data below, the amount of head needed in pipes 2 and 10 are 45.68 feet and -78.80 feet respectively. (Solution in this manual.)

EXAMPLE NO. 1 OF SIMPLE NETWORK NUMBERING SOURCES AS NODES

```plaintext
/*
$SPECIF NPLENG=1,NODESP=1 $END
PIPES
1 10 2  .5 12 .0102
5 2 1 1.3 10/
3 3 4 1.6 8/
4 1 4 2 10/
2 2 3 2.2 8/
6 3 5/
8 5 4 3/
10 4 6 2 10/
9 5 6 1.4 6/
7 11 5 1 12/
11 12 1 .5/
NODES
2 2 300
1 1 280
3 1.5 280
4 6 200
5 1 270
6 3 200
10 0 300
11 0 290
12 0 280
QGIVE
2 2.5
10 .5
PUMPS
11 3 80 4 77 5 72 350
RESER
10 400
12 350
MINOR
4 10
VALVE
8 1500 320
RUN
```

EXAMPLE NO. 1 OF SIMPLE NETWORK NUMBERING SOURCES AS NODES

```plaintext
/*
$SPECIF NPLENG=1,NODESP=1 $END
PIPE-
1 12 .5 10 300. 2 2. 300.
5 10 1.3 2 1 1. 280.
3 8. 1.6 3 1.5 280. 4 6. 200.
4 10. 2. 1 4
2 8. 2.2 2 3
6 8. 2.2 5 1. 270. 3
8 8. 3. 5 4
10 10. 2. 4 6 3. 200.
9 6. 1.4 5 6
7 12. 1. 11 290. 5 1. 270.
11 12. .5 12 280. 1
QGIVE
2 2.5
10 .5
PUMPS
11 3 80 4 77 5 72 350
RESER
10 400
12 350
MINOR
4 10
VALVE
8 1500 320
RUN
```
EXAMPLE NO. 1 OF SIMPLE NETWORK
NUMBERING SOURCES AS NODES

ALL DEMAND FLOWS ARE MULTIPLIED BY 1,000

PIPES 11
NODES 9
SOURCE PUMPS 1
BOOSTER PUMPS 0
RESERVOIRS 2
MINOR LOSSES 1
PRVS 1
NOZZLES 0
CHECK VALVES 0
BACK PRES V. 0
DIFF. HEAD DEV 0
SPECIFIED Q 2

PUMPS:
<table>
<thead>
<tr>
<th>REASON COEF</th>
<th>LINEAR COEF</th>
<th>SHUT-OFF HEAD</th>
<th>SUMP ELEV</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.000</td>
<td>-1.000</td>
<td>4,000</td>
<td>77.00</td>
</tr>
</tbody>
</table>

RESERVOIRS PLUS NOZZLES:
<table>
<thead>
<tr>
<th>NO. PIPE NODE ELEVATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 1</td>
</tr>
<tr>
<td>2 11</td>
</tr>
</tbody>
</table>

MINOR LOSS PIPES AND THEIR COEFFICIENTS:
<table>
<thead>
<tr>
<th>PIPE K</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 10.00</td>
</tr>
</tbody>
</table>

PRESSURE REDUCING VALVES (1): PIPE DOWNSTREAM DOWNSTREAM DISTANCE HGt ELEV
| 8 | 1500.00 | 320.00 |

MINOR LOSSES:
<table>
<thead>
<tr>
<th>PIPE LENGTH + EQUIVALENT NO. LOSS K LENGTH</th>
</tr>
</thead>
<tbody>
<tr>
<td>14.50</td>
</tr>
</tbody>
</table>

FLOW FROM PUMPS AND RESERVOIRS EQUALS 14.50

ITERATION-
1 SUM OF DIFFERENCES = 0.356E+02

ITERATION-
2 SUM OF DIFFERENCES = 0.956E+01

ITERATION-
3 SUM OF DIFFERENCES = 0.298E+01

ITERATION-
4 SUM OF DIFFERENCES = 0.191E+00

ITERATION-
5 SUM OF DIFFERENCES = 0.270E+00

LOSES DUE TO FLUID FRICTION IN ALL PIPES

POWER LOSS = 141,174 H.P. = 105,316 KWATTS, ENERGY LOSS = 2527,583 KWHRS /DAY

PUMPS:
<table>
<thead>
<tr>
<th>NODE PIPE</th>
<th>HEAD FLOW HORSEPOWER KILOWATTS KWATT-HRS/ DAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>11 7 66.53 5.80 43.81 32.68 784.34</td>
<td></td>
</tr>
</tbody>
</table>

HEADS NEEDED TO PRODUCE SPECIFIED FLOWRATES IN PIPES
<table>
<thead>
<tr>
<th>NO. PIPE FLOW RATE</th>
<th>HEAD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2 2.50 45.68</td>
<td></td>
</tr>
<tr>
<td>2 10 0.50 -78.00</td>
<td></td>
</tr>
</tbody>
</table>

ELEVATION OF HGt UPSTREAM AND DOWNSTREAM OF PRVS:
<table>
<thead>
<tr>
<th>PIPE UPSTREAM DOWNSTREAM HGt HGt</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 390.29 320.00</td>
</tr>
</tbody>
</table>

UNITS OF SOLUTION ARE:
Diameters - Inch
Length - Feet
Heads - Feet
Elevations - Feet
Pressures - PSI
Flowrates - CFS

DARCY-WEISBACH FORMULA USED FOR COMPUTING HEAD LOSS

| PIPE NO. | FROM TO LENGTH DIAM COEF FLOW RATE VELOCITY HEAD HLOSS |
|----------|-------------|---------|-----------|----------------|--------|
| 1        | 10 2 500.0 12.0 0.010200 8.83 11.24 10.94 37.87 |
| 2        | 2 3 2200.0 8.0 0.010200 2.50 7.16 56.69 25.77 |
| 3        | 3 4 1600.0 8.0 0.010200 2.81 8.04 51.75 32.34 |
| 4        | 1 4 2410.0 10.0 0.010200 3.20 5.86 31.70 13.15 |
| 5        | 2 1 1300.0 10.0 0.010200 4.33 7.94 31.06 23.89 |
| 6        | 5 3 2200.0 6.0 0.010200 1.81 5.17 29.94 13.61 |
| 7        | 11 5 1000.0 12.0 0.010200 5.00 7.39 16.54 16.54 |
| 8        | 5 4 3000.0 8.0 0.010200 0.50 1.43 1.34 1.13 |
| 9        | 5 6 1400.0 6.0 0.010200 2.50 12.73 161.24 115.17 |
| 10       | 4 6 2000.0 10.0 0.010200 0.50 0.92 0.75 0.37 |

* 11 1 12 500.0 12.0 0.010200 0.13 0.17 0.01 0.01 |

AV. VEL = 7.56 AV. HGt/1000 = 31.096 MAX VEL = 12.73 MIN VEL = 0.17

AV. HGt = 7.56 AV. HGt/1000 = 31.096 MAX HGt = 12.73 MIN HGt = 0.17

NODE DATA:

<table>
<thead>
<tr>
<th>NO.</th>
<th>DEMAND (CFS)</th>
<th>DEMAND (GPM)</th>
<th>ELEV</th>
<th>HEAD</th>
<th>PRESSURE</th>
<th>HGt ELEV</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.000</td>
<td>448.83</td>
<td>280.0</td>
<td>70.01</td>
<td>30.34</td>
<td>350.01</td>
</tr>
<tr>
<td>2</td>
<td>2.000</td>
<td>897.66</td>
<td>300.0</td>
<td>81.06</td>
<td>35.13</td>
<td>381.06</td>
</tr>
<tr>
<td>3</td>
<td>1.500</td>
<td>673.24</td>
<td>280.0</td>
<td>90.05</td>
<td>39.02</td>
<td>370.05</td>
</tr>
<tr>
<td>4</td>
<td>6.000</td>
<td>2692.98</td>
<td>200.0</td>
<td>118.30</td>
<td>51.26</td>
<td>320.0</td>
</tr>
<tr>
<td>5</td>
<td>1.000</td>
<td>448.83</td>
<td>270.0</td>
<td>129.99</td>
<td>56.33</td>
<td>399.99</td>
</tr>
<tr>
<td>6</td>
<td>3.000</td>
<td>1346.49</td>
<td>200.0</td>
<td>38.76</td>
<td>16.79</td>
<td>238.76</td>
</tr>
<tr>
<td>11</td>
<td>-5.804</td>
<td>-2604.79</td>
<td>290.0</td>
<td>126.53</td>
<td>54.83</td>
<td>416.53</td>
</tr>
<tr>
<td>10</td>
<td>-8.930</td>
<td>-3963.11</td>
<td>300.0</td>
<td>100.00</td>
<td>43.33</td>
<td>400.0</td>
</tr>
<tr>
<td>12</td>
<td>0.133</td>
<td>59.87</td>
<td>280.0</td>
<td>70.00</td>
<td>30.33</td>
<td>350.00</td>
</tr>
</tbody>
</table>

AV. HEAD = 88.03 AV. HGt = 343.03 MAX HEAD = 129.99 MIN HEAD = 38.76
EXAMPLE PROBLEM 1D

In this variation of Problem # 1 a couple of the options that allow the order of input to be different than that described under the PIPES and NODES commands is illustrated. Assume the pipe data is in the following order: (1) the pipe no., (2) the upstream node no., (3) the downstream node no., (4) the pipe diameter, (5) the roughness coeff., & (6) the pipe length.

The node data has the following order: (1) the node no., (2) the elevation, & (3) the demand. Proper input under these circumstances is as shown below.

EXAMPLE NO. 1 IN WHICH THE ORDER OF ITEMS HAVE THE DIAMETER, ROUGHNESS, & LENGTH AFTER NODES NUMBERS, AND ELEVATION AND DEMAND ARE INTERCHANGED AFTER NODE NUMBERS.

/*
$SPECIF NODESP=1, CORPIN=2 $END

PIPPES
1 10 2 12 .0102 500
5 2 1 10 .0102 1300
3 3 4 8 .0102 1600
4 1 4 10 .0102 2000
2 2 3 8 .0102 2200
6 3 5 8 .0102 2200
8 5 4 8 .0102 3000
10 4 6 10 .0102 2000
9 5 6 6 .0102 1400
7 11 5 12 .0102 1000
11 12 1 12 .0102 500

NODES
2 300 2
1 280 1
3 280 1.5
4 200 6
5 270 1
6 200 3
10 300 0
11 290/
12 280/

PUMPS
11 3 80 4 77 5 72 350

RESER
10 400
12 350

MINOR
4 10

VALVE
8 1500 320

EXAMPLE PROBLEM 1E

This variation of problem 1 illustrates how the use of the options that allow the order of items under the PIPE- command might be used to read formatted input that was prepared for another program. Assume that the data exists as follows: (a) pipe diameters in F5.0, (b) length in F10.0, (c) roughness coefficient in F10.4, (d) the next field of 10 data that is not needed by this program, (e) pipe number in IS, (f) upstream node in IS, and (g) downstream node in IS. Since the option CORPIP does not allow for the order of the pipe number to be other than the first item it is necessary to utilize CORPIN and the PIPE- command. The additional data consist of: (a) the node number in IS, (b) the demand in F5.1, and (c) the elevation in F10.5. Since this formatted data can be read directly with the NODES command it needs no special attention. Thus by adding a few additional statements to an existing data file, and also deleting a few statements it is not necessary to reenter the data.

EXAMPLE 1 WITH DIFFERENT ORDER OF ITEMS ON PIPES CARD.

/*
$SPECIF NPRINT=10, NODESP=1, CORPIN=2 $END

PIPE-(F5.0,2F10.0,10X,7F5.0)
4 1 2 3 5 7 8 6 9 10
12. 1000. .01020
10. 1300. .01020
8. 1600. .01020
10. 2000. .01020
8. 2200. .01020
8. 2200. .01020
8. 3000. .01020
10. 2000. .01020
6. 1400. .01020
12. 1000. .01020
12. 5000. .0102

NODES
2 300 2
1 280 1
3 280 1.5
4 200 6
5 270 1
6 200 3
10 300 0
11 290/
12 280/

PUMPS
11 3 80 4 77 5 72 350.

RESER
10 400
12 350

MINOR
4 10/

VALVE
8 1500 320

RUN
EXAMPLE PROBLEM 2

EXAMPLE NO. 2

```plaintext
/* $SPECIF NODESP=1,NFLOW=1,NPGPM=1,ICOST=1 $END
PIPES
1 8 1 500 8 130
2 1 2/
3 2 3/
4 1 4 1500 6/
5 3 4 2000 6/
6 4 5 2500/
7 6 5 2500 8/
8 6 3 1500/
9 7 6 500/
NODES
1 0 30/
2 200/
3 200/
4 150/
5 200/
6 150/
7 0/
8 0/
RESER
8 100
7 120
VALVC
3 7 6/
RUN
INTEREST=.12
LIFE=80
RESERVOIRS
CAPI=30000
WATER
UNIT=30000,RESER,1
UNIT=45000,RESER,9
END
*/
```

EXAMPLE NO. 2

```plaintext
/* $SPECIF NODESP=1,NFLOW=1,NPGPM=1 $END
NUMPIP=1,COEFRO=130 $END
PIPE-
8. 500. 8 1 30.
8. 500. 1 2 200. 30.
8. 500. 2 3 200. 30.
6. 1500. 1 4 150. 30.
6. 2000. 3 4
6. 2500. 4 5 200. 30.
8. 2500. 6 5
8. 1500. 6 3
8. 500. 7 6 150. 30.
RESER
8 100
7 120
VALVC
3 7 6/
RUN
CHANGE
DFRAC
1.5
END
CHANG
ALLCO
150.
END
*/
```

1. In this alternate input two alternate analyses are included. The first of which multiplies all previous demands by 1.5. The second leaves the demands at this increased amount and increases all Hazen-Williams roughness coefficients to 150.
RESERVOIRS PLUS NOZZLES:

<table>
<thead>
<tr>
<th>NODE</th>
<th>NO.</th>
<th>ELEVATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>6</td>
<td>100.00</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>100.00</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>120.00</td>
</tr>
</tbody>
</table>

CHECK VALVES ARE IN FOLLOWING PIPES

<table>
<thead>
<tr>
<th>NODE</th>
<th>NO.</th>
<th>DEMAND</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>0</td>
<td>7</td>
<td>7</td>
</tr>
</tbody>
</table>

FLOW FROM PUMPS AND RESERVOIRS EQUALS 900,000

ITERATION I = 1 SUM OF DIFFERENCES = 0.281E+01
DURING ITERATION NUMBER 2 CHECK VALVE IN PIPE 3 HAS SHUT OFF FLOW.

ITERATION I = 2 SUM OF DIFFERENCES = 0.821E+01
ITERATION I = 3 SUM OF DIFFERENCES = 0.367E+00
DURING ITERATION NUMBER 4 CHECK VALVE IN PIPE 3 HAS SHUT OFF FLOW.

ITERATION I = 4 SUM OF DIFFERENCES = 0.926E+00
ITERATION I = 5 SUM OF DIFFERENCES = 0.324E+00
ITERATION I = 6 SUM OF DIFFERENCES = 0.180E+00
ITERATION I = 7 SUM OF DIFFERENCES = 0.100E-02
LOstS DUE TO FRICTORion IN ALL PIPES

POWER LOSS = 2.264 H.P. = 1.689 KWATTS. ENERGY LOSS = 40.541 KWHRS/DAY

UNITS OF SOLUTION ARE:

<table>
<thead>
<tr>
<th>DIAMETERS</th>
<th>LENGTH</th>
<th>HEADS</th>
<th>ELEVATIONS</th>
<th>PRESSURES</th>
</tr>
</thead>
<tbody>
<tr>
<td>INCH</td>
<td>FEET</td>
<td>FEET</td>
<td>FEET</td>
<td>PSI</td>
</tr>
</tbody>
</table>

PIECE DATA:

| PIPE | NODS | FROM TO LENGTH OAM | COEF FLOW RATE VELOCITY HEAD LOSS |
|------|------|--------------------|-------------------------------|----------------------------------|
| 0    | 0    | 600.00 0.00        | 66.97 10.00                  | 65.97 20.00                     |
| 1    | 0    | 600.00 0.00        | 66.97 10.00                  | 65.97 20.00                     |
| 2    | 0    | 600.00 0.00        | 66.97 10.00                  | 65.97 20.00                     |
| 3    | 0    | 600.00 0.00        | 66.97 10.00                  | 65.97 20.00                     |
| 4    | 0    | 600.00 0.00        | 66.97 10.00                  | 65.97 20.00                     |
| 5    | 0    | 600.00 0.00        | 66.97 10.00                  | 65.97 20.00                     |
| 6    | 0    | 600.00 0.00        | 66.97 10.00                  | 65.97 20.00                     |
| 7    | 0    | 600.00 0.00        | 66.97 10.00                  | 65.97 20.00                     |
| 8    | 0    | 600.00 0.00        | 66.97 10.00                  | 65.97 20.00                     |
| 9    | 0    | 600.00 0.00        | 66.97 10.00                  | 65.97 20.00                     |

COSTS ASSOCIATED WITH THIS NETWORK:

<table>
<thead>
<tr>
<th>ITEM</th>
<th>TYPE</th>
<th>PRESENT WORTH SERIES AMOUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ELEC</td>
<td>0.00</td>
</tr>
<tr>
<td>2</td>
<td>PIPE</td>
<td>78000.00</td>
</tr>
<tr>
<td>3</td>
<td>RESE</td>
<td>80000.00</td>
</tr>
<tr>
<td>4</td>
<td>WATE</td>
<td>9041.94</td>
</tr>
</tbody>
</table>

TOTAL 147041.94 17647.07
EXAMPLE PROBLEM 2A

Suppose in the network in Example Problem 2 that the reservoir that is connected to the piping system by pipe no. 1 represents an actual reservoir of almost infinite supply, but the reservoir at the end of pipe no. 3 is a relatively small storage tank, and would not be able to supply the flowrate given in the previous analysis for very long. Therefore, a pump is needed to cause more flowrate to come from the former reservoir. This pump will be placed in pipe no. 1 and its head is to be determined so that the pressure at node no. 2 is 40 psi. The use of the command DHEAD is well suited to size this pump. The data below illustrates how information related to the required pump head and capacity can be determined for both the original demands as well as for larger demands equal to 1.5 times those given in the original data. The second analysis is accomplished by added CHANGE data for an alternative analysis in which: (1) all previous demands are multiplied by 1.5, and (2) the pumps should produce the head necessary for the pressure at node 2 to equal 45 psi.

Input data for Example 2A.

EXAMPLE NO. 2A
/*
$SPECIF NODESP=1,NFLOW=1,NPGPM=1 NUMPIP=1,COEFRO=130 $END
PIPE.
8. 500. 8 1 30.
8. 500. 1 2 200. 30.
8. 500. 2 3 200. 30.
6. 1500. 1 4 150. 30.
6. 2000. 3 4
6. 2500. 4 5 200. 30.
8. 2500. 6 5
8. 1500. 6 3
8. 500. 7 6 150. 30.
RESER
8 100
7 120
VALVC
3 7 6/
DHEAD
1 40 -2 7 40
RUN
CHANGE
DFRAC
1.5
DHEAD
1 -45/
END
END

EXAMPLE NO. 2A
/*
$SPECIF NODESP=1,NFLOW=1,
NPRINT=-3,NPGPM=1 $END
PIPER
1 8 1 500 8 130
2 1 2/
3 2 3/
4 1 4 150 6/
3 3 4 200/
6 4 5 2500/
7 6 5 2500 8/
8 6 3 1500/
9 7 6 500/
NODES
1 0 30/
2 200/
3/
4 150/
5 200/
6 150/
7 0/
8/
RESER
8 100
7 120
VALVC
3 7 6/
DHEAD
1 40 -2 7 40
RUN
CHANGE
DFRAC
1.5
DHEAD
1 -45/
END
END
EXAMPLE PROBLEM 3

PUMP CHARACTERISTICS

<table>
<thead>
<tr>
<th>No.</th>
<th>Q₁ (gpm)</th>
<th>h₁₁ (ft)</th>
<th>Q₂</th>
<th>h₂₂</th>
<th>Q₃</th>
<th>h₃₃</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>900</td>
<td>230</td>
<td>1800</td>
<td>220</td>
<td>2700</td>
<td>200</td>
</tr>
<tr>
<td>2</td>
<td>900</td>
<td>50</td>
<td>1800</td>
<td>45</td>
<td>2700</td>
<td>38</td>
</tr>
</tbody>
</table>

INPUT DATA

EXAMPLE PROBLEM WITH INPUT GIVEN WITHOUT NUMBERING PIPES

* $SPECIF NPGPM=1,NUMPIP=1
* IOEX=-3 $END

PIPES
0 1 500 12 .0102
1 2 500 6/
2 3 600/
1 4 500 12/
1 5 800 12/
5 6 600 8/
6 7 600/
7 8 600/
10 7 800/
10 10 500 14/
10 11 2000 12/
10 9 2000 10/
7 9 2000 6/
9 11 1200 6/
9 12 1200 8/
8 12 1200 8/
5 8 1000 10/
11 21 200 12/
21 22 1000 10/
10 22 1000 6/
22 23 1000 6/
22 24 500 6/
22 26 1500 8/
26 27 500 6/
26 27 600 6/
27 28 600 6/
27 28 500 6/

RESER
1 400

PUMPS
9 900 230 1800 220 2700 200 300

BPUMP
15 900 50 1800 45 2700 38/

VALVE
10 1200 250

END

CHANGE
DFRAC
1.3

END

CHANGE
ELRES
1 420/
PARALLEL
1 1 2/
END

NODES
1 1 250
2 5/
3 1/
4 5/
6 2/
5 5.5/
6 1.5/
7 1/
10 0/
11 0/
9 1/
21 0 100/
27 5/
28/
22 1.5/
25 6/
25 8/
24 6/

EXAMPLE PROBLEM WITH INPUT GIVEN WITHOUT NUMBERING PIPES

* $SPECIF NPGPM=1,NUMPIP=1
* IOEX=-3,NPLENG=1 $END

PIPE
12 .5 1 1.250.
6 .5 1 1.2 .5 250.
6 .6 2 3 1.250.
12 .5 1 4.5 250.
12 .8 4 5 .5 250.
8 .8 6 6 1.5 250.
8 .8 6 7 1.250 6
8 .6 10 250 .7
14 .5 10
12 2.1 10 11 250.
10 2.2 10 9 1.250.

6 2 7 .9
6 1 2 9 11
8 1.2 8 2.250 .9
10 1.5 8
12 .2 11 21 100.
10 .1 21 22 1.5 100.
6 .1 22 25 .5 100.
6 .5 23 24 .6 100.
8 .1 5 22 25 .6 100.
6 .5 27 .5 100.25
6 .6 21 27
6 .5 27 26 .5 100.

RESER
1 400

PUMPS
9 900 230 1800 220 2700 200 300
BOOSTER
15 900 50 1800 45 2700 38 0/

VALVE
10 1200 250

END

CHANGE
DFRAC
1.3

END

CHANGE
ELRES
1 420/
PARALLEL
1 1 2/
END

-122-
EXAMPLE PROBLEM 4

This example contains six source pumps that are filling the reservoir under the no demand conditions shown. Other analyses of this network would place demands as the nodes to determine the systems performance when the reservoir assists the pumps in supplying the demand. The data shown for this network has been prepared using the option of not numbering the pipes. The data has been entered in the same sequential order as the pipes are number on the sketch, and therefore even though the pipes are given no numbers in the input, they will be numbered in the output as they are on the sketch. This occurs because the program will number pipes in the order in which the data is entered if the option NUMPIP=1 in the $SPECIF list.

All pumps identical with the following characteristics

<table>
<thead>
<tr>
<th>No.</th>
<th>Q (gpm)</th>
<th>Head (feet)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>400</td>
<td>351</td>
</tr>
<tr>
<td>2</td>
<td>600</td>
<td>285</td>
</tr>
<tr>
<td>3</td>
<td>700</td>
<td>234</td>
</tr>
</tbody>
</table>
EXAMPLE PROBLEM 4A

This Problem 4A is the same problem as Problem 4. The input however is prepared using the option of numbering sources of supply as nodes, i.e. the parameter NODESP=1 is placed in the $SPECIF list. This option requires the following changes in the input data: 
1. Under the command PIPES the source nodes are now numbers other than 0 (for non-existent node), 
2. Under the command PIPE every line now contains two node numbers, 
3. Under the command NODES the new source nodes are added. These must all have a zero demand, and 
4. Under the RESER and PUMPS commands the reservoirs and source pumps are identified by there node number and not the pipe that connects them to the network.

All pumps identical with the following characteristics

<table>
<thead>
<tr>
<th>Pt. No.</th>
<th>Q (gpm)</th>
<th>Head (feet)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>400</td>
<td>351</td>
</tr>
<tr>
<td>2</td>
<td>600</td>
<td>285</td>
</tr>
<tr>
<td>3</td>
<td>700</td>
<td>234</td>
</tr>
</tbody>
</table>
EXAMPLE PROBLEM 4B

This example illustrates the special design applied to the main lines of Example 4. The 5 nodes of this water transmission system allow the 5 pipes 3, 4, 7, 9, and 11 to be sized. In determining what pressures to specify at the nodes the assumption was made that the slope of the hydraulic grade line should be constant from node 3 to the reservoir, i.e. losses in pipes 3, 5, 7, 9, and 11 should be proportional to their lengths, and that the elevation of the HGL at node 3 should be as given by the previous solution or 5620 ft.

COLORADO SPRINGS-DESIGN
/*
$SPECIF NOMSOL=1,NFLOW=1,
NPGPM=1,NUMPIP=1,NODESP=1,
IDX=2,DESIGN=1,INGL=0 $END
PIPECOLORADO SPRINGS-DESIGN
/*
$SPECIF NOMSOL=1,NFLOW=1,
NPGPM=1,NUMPIP=1,NODESP=1,
IDX=2,DESIGN=1,INGL=0 $END
PIPETABLE 8.
1 1670. 1
2 1200
3 2600
4 1200
5 1860
6 1200
7 2340
8 1200
9 1840
10 2500
11 4500
NODESTABLE 3
3 0 5380
5 0 5392
7 0 5410
9 0 5412
11 0 5420
0 0 5350
2 0 5340
4 0 5365
8 0 5370
10 0 5414
12 0 5545
PUMPSTABLE 1
1 400 351 600 285 700 234 5354
2 400 351 600 285 700 234 5345
4 400 351 600 285 700 234 5370
6 400 351 600 285 700 234 5372
8 400 351 600 285 700 234 5374
10 400 351 600 285 700 234 5420
RESERVETABLE 12
12 5560
RUN

PIPE NETWORK ANALYSIS BASED ON THE CORRECTIVE FLOWRATE EQUATIONS

BY ROLAND W. JEPSSON
UTAH STATE UNIV.
LOGAN, UTAH 84322

TITLE GIVEN TO NETWORK
---------------------------
COLORADO SPRINGS-DESIGN

ALL DEMAND FLOWS ARE MULTIPLIED BY 1.0000
PIPECOLORADO SPRINGS-DESIGN
PIPES 11
NODES 12
SOURCE PUMPS 6
BOOSTER PUMPS 0
RESERVOIRS 1
MINOR LOSSES 0
PRVS 0
NOZZLES 0
CHECK VALVE 0
BACK PRES. V. 0
DIF. HEAD DEV 0

-125-
### Pipe Data

#### Pipe Table

<table>
<thead>
<tr>
<th>No.</th>
<th>From</th>
<th>To</th>
<th>Length</th>
<th>Dia.</th>
<th>Coef.</th>
<th>Flow Rate</th>
<th>Velocity</th>
<th>Head</th>
<th>Loss</th>
<th>1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.00</td>
</tr>
</tbody>
</table>

#### Unit of Solution

- **Diameters**: Inch
- **Length**: Feet
- **Heads**: Feet
- **Elevations**: Feet
- **Pressures**: PSI

### Flow Rates (GPM)

<table>
<thead>
<tr>
<th>Node</th>
<th>Demand</th>
<th>Elev</th>
<th>Head Pressure</th>
<th>MGL ELEV</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0.00</td>
<td>5380.</td>
<td>240.00</td>
<td>104.04</td>
</tr>
<tr>
<td>5</td>
<td>0.00</td>
<td>5392.</td>
<td>218.03</td>
<td>94.48</td>
</tr>
<tr>
<td>7</td>
<td>0.00</td>
<td>5410.</td>
<td>194.12</td>
<td>84.12</td>
</tr>
<tr>
<td>9</td>
<td>0.00</td>
<td>5412.</td>
<td>178.92</td>
<td>84.12</td>
</tr>
<tr>
<td>11</td>
<td>0.00</td>
<td>5420.</td>
<td>160.62</td>
<td>69.60</td>
</tr>
</tbody>
</table>

**Flow from Pumps and Reservoirs:**

<table>
<thead>
<tr>
<th>Flow</th>
<th>ELEV</th>
<th>MGL ELEV</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>12</td>
<td>4500.00</td>
</tr>
</tbody>
</table>

#### Node Data

<table>
<thead>
<tr>
<th>Node</th>
<th>Demand</th>
<th>Elev</th>
<th>Head Pressure</th>
<th>MGL ELEV</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0.00</td>
<td>5380.</td>
<td>240.00</td>
<td>104.04</td>
</tr>
<tr>
<td>5</td>
<td>0.00</td>
<td>5392.</td>
<td>218.03</td>
<td>94.48</td>
</tr>
<tr>
<td>7</td>
<td>0.00</td>
<td>5410.</td>
<td>194.12</td>
<td>84.12</td>
</tr>
<tr>
<td>9</td>
<td>0.00</td>
<td>5412.</td>
<td>178.92</td>
<td>84.12</td>
</tr>
<tr>
<td>11</td>
<td>0.00</td>
<td>5420.</td>
<td>160.62</td>
<td>69.60</td>
</tr>
</tbody>
</table>

#### Pumps

<table>
<thead>
<tr>
<th>Node</th>
<th>Pipe</th>
<th>Flow</th>
<th>Horsepower</th>
<th>Kilowatts</th>
<th>Kwatt-Hrs/Day</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>200</td>
<td>608.81</td>
<td>43.24</td>
<td>32.26</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>350</td>
<td>980.45</td>
<td>73.77</td>
<td>57.31</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>130</td>
<td>475.05</td>
<td>43.22</td>
<td>32.24</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>254</td>
<td>681.01</td>
<td>42.11</td>
<td>31.41</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>300</td>
<td>706.47</td>
<td>41.12</td>
<td>30.68</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>196.26</td>
<td>762.15</td>
<td>37.81</td>
<td>28.21</td>
</tr>
</tbody>
</table>

### Reservoirs

- **Head**: 5560.0
- **Flow from Pumps and Reservoirs**: 0.00
EXAMPLE NUMBER 4C

This example examines the time dependent operation of the system in problem # 4 that has 6 pumps filling the reservoir. The operation of the pumps are controlled by the water surface in the reservoir. Three of these pumps, that are in pipes 1, 6 and 10, have their operation controlled by a rule that indicates how many pumps are operating in parallel as a function of the water surface elevation in the reservoir. At these stations there are actually 4 parallel pumps all of are operating initially when the reservoir is empty. Assume the other three pumps, that are in pipes 2, 4 and 8 are variable speed pumps that have their rotational speeds controlled. Pumps in pipes 2 and 4 have this speed controlled by a rule related to the water surface elevation in the reservoir, and the pump in pipe 8 has its speed controlled as a function of time. The pump in pipe 2 has a speed control that various it speed continuous as the depth changes. Its operation is denoted as continuous by giving a 2 as the second input value. The pumps in pipes 4 and 8 have these operation discretely controlled, e.g. their speed are changed only when a given water surface elevation is exceeded; thus a 1 is used as the second input value for the rotational rule for the pump in line 4 and for the rotational schedule for the pump in line 8. The bottom of the reservoir is at elevation 5500 ft, and it contains 8 million gallons when its water surface elevation is at 5560 ft.

Input data for solving this problem is given below. Note the following: (a) Since the option NPSERI=0 is given in the $SPECIF list that the pump number identifies the number of parallel pumps in pipes 1, 6 and 10. These identifiers are 1, 4 and 6. (b) The pump characteristics for pumps 1, 4 and 6 are for individual pumps since the PARAllel command indicating that 4 pumps are operating at each of these stations is (An alternative would be to give 4 times as large a flow rate for the 3 points, as in the previous input data, and leave out the PARAllel command.) (c) The option NPUNOD in the $TDATA list indicates that when referring to pumps and reservoirs their numbers will be used. Thus the reservoir at the end of pipe 11 at node 12 is identified by a 1 under the STORAGE FUNCTION command, and under the command PUMP RULES, RORTRUL and ROTSCH that pump and reservoir numbers are used. Setting the option NPNRES=1 tells USU-NETWK that rules based on water surface elevations in reservoirs will be used rather than HGL elevations at nodes.

Only the special time dependent tables requested by PRINTT=3, and the commands PIPE TABLE, NODE TABLE and RESER. TABLE are given below the input data.

Input data to above time-dependent problem
Simulation of Colorado Spring pumps
/* $SPECIF NPRINT=-2,NFLOW=1,NPGPM=1, NODESP=1,COEFR0=120,NPSERI=0,ISIML=1 $END PIPE*
  1 8. 1670. 1 5350. 3 5380.
  2 8. 1200. 2 5540. 3
  3 12. 2600. 3 5 5392.
Controlling the operation of some pumps by a PUMP RULE that determines the number of pumps operating in parallel, and other pumps by changing their rotational speed in this example intend to illustrate some of the capabilities of USU-NETWK, and does not represent a desirable mode of operation necessarily, and can easily result in a situation in which some pumps would not be pumping water into the reservoir, or might be operating at very low efficiencies. The efficiency of parallel pumps at a station will remain near the highest efficiency if the head remains quite constant but the flow rate varies. For example if 3 parallel pumps are operating at maximum efficiency with a flow rate of 0, then if the head remains constant and the flow rate is to reduce to 0/3 then only one pump should operate. If the pump curves are very flat keeping all three pumps in operation for 0/3 would increase the head slightly, if at all depending upon how flat these pump curves are. This would result in 3 times the energy consumption for pumping the water over having only one pump operating.

On the other hand changing the rotational speed of a pump changes the head produced by the pump proportional to the square of this change and the flow rate only in proportion to this change. Thus reducing the rotational speed by 1/2 results in the pump wanting to only produce 1/4 as much head, and 1/2 as much flow rate. The effect of changing rotational speeds of pumps is midway between changing the number of pumps in parallel and changing the number of pumps in series at a station.

In this illustrative example it is possible to reducing the rotational speed too much so that a pump controlled by ROTRUL or ROTSCH would not be capable of pumping against the heads produced by other pumps controlled PUMP RULES, resulting in either back-flow through this pump, or no solution existing for the system of simultaneous equations that USU-NETWK solves. Should the latter be the case then the Newton Method used by USU-NEWTON would fail to converge regardless of how many iterations may be allowed by changing the option ITERA (or MAX) in the $SPECIF list.

The utility computer program PMPCUR described in Appendix C is designed to assist you in understanding the operation of pumps that you specify, and correct problems that may arise from inappropriately specified pump curves, poor rules controlling pumps, or incompatible rotational speed specifications in time dependent solutions.
EXAMPLE PROBLEM 4D

A possible alternative is to specify the head that a pump supplies to the flow in a pipe rather than giving the operating characteristics of a pump. The command DHEAD allow for the head to be specified in any pipe. Since the pressure or HGL elevation is not being controlled elsewhere in the network type #1 differential heads will be utilized for this purpose. In other words the magnitude of the differential head will be specified as the second item under the DHEAD command, and the first item will be the pipe that contains this differential head. The network in EXAMPLE 4 has had its input data modified so that the heads for the six pumps are specified. The following modifications are needed to input data for this example network. (1) Negative demands need to be placed at the ends of the pipes that contain the pumps equal to the amount of flow this pumps is supposed to supply to the network, and (2) The PUMPS command and the lines under it are replaced by the DHEAD command. The input data with this modification, as well as the solution results are listed below.

Input data using the DHEAD command to specify the amount of head the flow data pumps supply.

Simulation of Colorado Spring pumps
With DHEAD replacing pumps

| SPECIFY NFLOW=1, NGPM=1, NODESP=1, COEFRO=120 |
| NPRINT=-2 |

Reservoirs equal to -2625.000

Pipe data

- Title given to network
- Simulation of Colorado Spring pumps
- With DHEAD replacing pumps
- All demand flows are multiplied by 1.0000

<table>
<thead>
<tr>
<th>PIPE</th>
<th>NODES</th>
<th>SOURCE PUMPS</th>
<th>BOOSTER PUMPS</th>
<th>RESERVOIRS</th>
<th>MINOR LOSSES</th>
<th>PVE</th>
<th>NOZZLES</th>
<th>CHECK VALVE</th>
<th>BACK PRES. V.</th>
<th>DIP HEAD DEV</th>
<th>SPECIFIED PRES.</th>
</tr>
</thead>
<tbody>
<tr>
<td>PVES</td>
<td>11</td>
<td></td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RES.</td>
<td>(NOZZLE) PIPES &amp; THEIR ELEV. ARE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5500.0</td>
<td></td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FLOW FROM PIPES AND RESERVOIRS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EQUALS -2825.000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Pipe original flowrate incremental head total head c equivalent diameter

<table>
<thead>
<tr>
<th>NO. DIAMETER (CFS)</th>
<th>HEAD LOSS</th>
<th>LOSS</th>
<th>DIAMETER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8.00</td>
<td>.39</td>
<td>2.85</td>
</tr>
<tr>
<td>2</td>
<td>8.00</td>
<td>1.95</td>
<td>5.97</td>
</tr>
<tr>
<td>3</td>
<td>12.00</td>
<td>.39</td>
<td>2.85</td>
</tr>
<tr>
<td>4</td>
<td>8.00</td>
<td>1.95</td>
<td>5.97</td>
</tr>
<tr>
<td>5</td>
<td>8.00</td>
<td>1.95</td>
<td>5.97</td>
</tr>
<tr>
<td>6</td>
<td>8.00</td>
<td>1.95</td>
<td>5.97</td>
</tr>
</tbody>
</table>

Pipe data

<table>
<thead>
<tr>
<th>NO.</th>
<th>FROM</th>
<th>TO</th>
<th>LENGTH</th>
<th>DIAM</th>
<th>COEF</th>
<th>FLOW RATE</th>
<th>VELOCITY</th>
<th>LOSS</th>
<th>HEAD LOSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1670</td>
<td>8.0</td>
<td>120.0</td>
<td>175.00</td>
<td>1.12</td>
<td>1.40</td>
<td>84.80</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>5</td>
<td>1200</td>
<td>8.0</td>
<td>120.0</td>
<td>700.00</td>
<td>4.47</td>
<td>13.14</td>
<td>10.95</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>5</td>
<td>1200</td>
<td>8.0</td>
<td>120.0</td>
<td>700.00</td>
<td>4.47</td>
<td>13.14</td>
<td>10.95</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>7</td>
<td>1860</td>
<td>14.0</td>
<td>120.0</td>
<td>1050.00</td>
<td>2.19</td>
<td>2.83</td>
<td>1.52</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>7</td>
<td>1200</td>
<td>8.0</td>
<td>120.0</td>
<td>700.00</td>
<td>4.47</td>
<td>13.14</td>
<td>10.95</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>9</td>
<td>2340</td>
<td>16.0</td>
<td>120.0</td>
<td>1225.00</td>
<td>1.95</td>
<td>2.47</td>
<td>1.08</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>9</td>
<td>1200</td>
<td>8.0</td>
<td>120.0</td>
<td>700.00</td>
<td>4.47</td>
<td>13.14</td>
<td>10.95</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>11</td>
<td>1200</td>
<td>8.0</td>
<td>120.0</td>
<td>1925.00</td>
<td>2.43</td>
<td>2.53</td>
<td>1.37</td>
</tr>
<tr>
<td>9</td>
<td>11</td>
<td>12</td>
<td>2500</td>
<td>18.0</td>
<td>120.0</td>
<td>1925.00</td>
<td>2.43</td>
<td>2.53</td>
<td>1.37</td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td>12</td>
<td>4500</td>
<td>18.0</td>
<td>120.0</td>
<td>2625.00</td>
<td>3.31</td>
<td>10.97</td>
<td>2.44</td>
</tr>
</tbody>
</table>

Flow from pumps and reservoirs equals -2825.000

Units of solution are

- Diameters - inch
- Length - feet
- Heads - feet
- Elevations - feet
- Pressures - psi
- Flowrates - gpm

Hazen-Williams formula used for computing head loss

Pipe data

<table>
<thead>
<tr>
<th>NO.</th>
<th>DIAM.</th>
<th>HEAD LOSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.95</td>
<td>84.80</td>
</tr>
<tr>
<td>2</td>
<td>1.95</td>
<td>10.95</td>
</tr>
<tr>
<td>3</td>
<td>1.95</td>
<td>10.95</td>
</tr>
<tr>
<td>4</td>
<td>1.95</td>
<td>10.95</td>
</tr>
<tr>
<td>5</td>
<td>1.95</td>
<td>10.95</td>
</tr>
<tr>
<td>6</td>
<td>1.95</td>
<td>10.95</td>
</tr>
<tr>
<td>7</td>
<td>1.95</td>
<td>10.95</td>
</tr>
<tr>
<td>8</td>
<td>1.95</td>
<td>10.95</td>
</tr>
<tr>
<td>9</td>
<td>1.95</td>
<td>10.95</td>
</tr>
<tr>
<td>10</td>
<td>1.95</td>
<td>10.95</td>
</tr>
</tbody>
</table>

Node data

<table>
<thead>
<tr>
<th>NO.</th>
<th>(GPM)</th>
<th>(CFS)</th>
<th>ELEV</th>
<th>HEAD</th>
<th>PRESSURE</th>
<th>ELEV</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>870.0</td>
<td>-1.56</td>
<td>5340</td>
<td>-372.09</td>
<td>-161.24</td>
<td>4967.91</td>
</tr>
<tr>
<td>2</td>
<td>-700.0</td>
<td>-1.56</td>
<td>5340</td>
<td>-372.09</td>
<td>-161.24</td>
<td>4967.91</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>5340</td>
<td>-140.23</td>
<td>-60.71</td>
<td>5230.71</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>5340</td>
<td>-140.23</td>
<td>-60.71</td>
<td>5230.71</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>5340</td>
<td>-140.23</td>
<td>-60.71</td>
<td>5230.71</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>5340</td>
<td>-140.23</td>
<td>-60.71</td>
<td>5230.71</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>5340</td>
<td>-140.23</td>
<td>-60.71</td>
<td>5230.71</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>5340</td>
<td>-140.23</td>
<td>-60.71</td>
<td>5230.71</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>5340</td>
<td>-140.23</td>
<td>-60.71</td>
<td>5230.71</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>5340</td>
<td>-140.23</td>
<td>-60.71</td>
<td>5230.71</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
<td>5340</td>
<td>-140.23</td>
<td>-60.71</td>
<td>5230.71</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>0</td>
<td>5340</td>
<td>-140.23</td>
<td>-60.71</td>
<td>5230.71</td>
</tr>
</tbody>
</table>
EXAMPLE PROBLEM 5

This Example No. 5 might be a reduced version of the pipe system for a steam powered turbine involving the pipes between the condenser and boiler. Since a series of heaters increase the temperature of the water significantly (and thus its density) between nodes 1 and 8, a much better solution results from using weight or mass flowrates rather than volumetric flowrates. In addition to using this capability of the program comments are placed after some pipes and nodes cards to have them appear in the solution tables.

This simple example illustrates the following features:
1. Determine equivalent lengths for minor losses with larger Reynolds No.
2. Using more than 3 pipes to define pump characteristic curves.
3. Placing remarks by some pipes and nodes.
4. Using weight flow rates since large temperature changes occur.
5. Checking input data by redefinition of network from alternatw input.
6. Providing starting HGL for special network with all external flows given.

```
* SPECgamma=57.3, visc=.45e-5, NFlow=5, COEFrd=.0036, NUMP=1, PCHAR=0,
  REYNUS1=1.67, NLEV=0, NPRINT=3, NETCH=2, NSYMHT=5 $ END

Pipes
1  2  35.375/R LOW TEMP. WATER FROM CONDENSOR.
2  3  2000 23.5/
3  5  2 13.5/R HEAT EXCHANGERS INCR. FLUID TEMP.
4  5  3/5
4  6/5
4  6/5
5  7  180 22.625/
6  7/
7  8  20 35.2/R ARRIVES AT BOILER AT HIGH TEMP.

Nodes
1  -2280 0/R FLOW FROM CONDENSOR.
2  3  0 15
3  4
5  6  480 23/R MAKE UP WATER
6  480/
7  3  57
8  3240 57/R FEEDS BOILER

Bpump
2  800 1020 1139 1005 1005 1005 1005 1005 1005 800/3
3  800 1020 1139 1005 1005 1005 1005 1005 1005 1005 800/
4  10 1000 1900 1900 1900 1900 1900 1900 1900 1900 1900 1900 1900
5  11 1000 1900 1900 1900 1900 1900 1900 1900 1900 1900 1900 1900
6  12/
7  34.
RUN
-2280. 0. 1/
0. -3. -3 2 3/
0. 15. -2 4 5 6/
0. 15. -3 7 8 9/
-480. 33. -4. -5. -6. 10/
0. 57. -10. -11. 12/
3240. 57. -12/
1. 34.
CHANGE
OFRAC
.8
END
```

-130-
TITLE GIVEN TO NETWORK

THIS SIMPLE EXAMPLE ILLUSTRATES THE FOLLOWING FEATURES:
1. DETERMINE EQUIVALENT LENGTHS FOR MINOR LOSSES WITH LARGER REYNOLDS NO.
2. USING MORE THAN 3 PTS. TO DEFINE PUMP CHARACTERISTIC CURVES.
3. PLACING REMARKS BY SOME PIPES AND NODES.
4. USING WEIGHT FLOW RATES SINCE LARGE TEMPERATURE CHANGES OCCUR.
5. CHECKING INPUT DATA BY REDEFINITION OF NETWORK FROM ALTERNATIVE INPUT.
6. PROVIDING STARTING HGL FOR SPECIAL NETWORK WITH ALL EXTERNAL FLOWS GIVEN.

ALL DEMAND FLOWS ARE MULTIPLIED BY 1.0000

PIPES 13
NODES 8
SOURCE PUMPS 0
BOOSTER PUMPS 4
RESERVOIRS 1
MINOR LOSSES 2
PRVS 0
NOZZLES 0
CHECK VALVES 0
BACK PRES V. 0
DIF. HEAD DEV 0

PUMPS:
PIPE 2ND ORDER COEF LINEAR COEF SHUT-OFF HEAD SUMP ELEV
2 0.000 0.409 842.56 0.00
3 0.000 0.409 842.56 0.00
10 0.000 0.016 1939.64 0.00
11 0.000 0.016 1939.64 0.00

RESERVOIRS PLUS NOZZLES:
NO. PIPE NODE ELEVATION
1 13 0 2.84

MINOR LOSS PIPES AND THEIR COEFFICIENTS:
PIPE K
2 12.00
3 54.00

SOL. BY UTIL. SPARSE MATRIX METHODS

MINOR LOSSES:
PIPE MINOR LENGTH + EQUIVALENT LENGTH
NO. LOSS K LENGTH
2 12.00 1888.73
3 54.00 8249.31

FLOW FROM PUMPS AND RESERVOIRS EQUALS 0.000

LOSSES DUE TO FLUID FRICTION IN ALL PIPES
POWER LOSS= 106.523 H.P. = 79.466 KWATTS. ENERGY LOSS= 1907.190 KWHRS /DAY

PUMPS:
PIPE HEAD FLOW HORSEPOWER KILOWATTS KWATTS-HRS/DAY
2 988.97 1184.01 123225.01 91925.86 2205220.75
3 1009.88 1095.99 115310.49 86021.83 2064519.00
10 1812.71 1664.01 314249.72 234430.28 5626327.00
11 1827.08 1575.99 299987.78 223790.89 5370981.50

UNITS OF SOLUTION ARE:
DIAMETERS - INCH
LENGTH - FEET
HEADS - FEET
ELEVATIONS - FEET
PRESSURES - PSI
FLOWRATES - LB/S
DARCY-WEISBACH FORMULA USED FOR COMPUTING HEAD LOSS
### Pipe Data:

<table>
<thead>
<tr>
<th>Pipe No.</th>
<th>From</th>
<th>To</th>
<th>Length</th>
<th>Dia</th>
<th>Coef</th>
<th>Flow Rate</th>
<th>Velocity</th>
<th>Head Loss</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td>390.0</td>
<td></td>
<td></td>
<td>2880.0</td>
<td>5.83</td>
<td>0.04</td>
<td>2.23 Low Temp. Water</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>1905.0</td>
<td>21.5</td>
<td>0.036</td>
<td>1184.0</td>
<td>6.86</td>
<td>9.97</td>
<td>5.01 HEAT EXCHANGERS IN</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>4</td>
<td>8249.0</td>
<td>23.5</td>
<td>0.036</td>
<td>1095.9</td>
<td>6.25</td>
<td>25.52</td>
<td>4.21</td>
</tr>
</tbody>
</table>

### Node Data:

<table>
<thead>
<tr>
<th>Node</th>
<th>Demand</th>
<th>Vol/2</th>
<th>Elev</th>
<th>Head</th>
<th>Pressure</th>
<th>Elevation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1000.0</td>
<td>57</td>
<td>2744</td>
<td>972</td>
<td>973.55</td>
<td>973.55</td>
</tr>
</tbody>
</table>

### Changes Made to Network

- All previous demands are multiplied by 0.800

#### Flow from Pumps and Reservoirs equals 0.900

#### Losses due to fluid friction in all pipes

- Power Loss = 53.506 H.P. 
- 39.915 kilowatts 
- Energy Loss = 957.967 kilowatts/day

### Pumps

<table>
<thead>
<tr>
<th>No.</th>
<th>Head</th>
<th>Flow</th>
<th>Horsepower</th>
<th>Kwatts</th>
<th>Kwatts-HRS/Day</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1019.4</td>
<td>964.21</td>
<td>102405.36</td>
<td>76394.40</td>
<td>1833465.63</td>
</tr>
<tr>
<td>2</td>
<td>1021.25</td>
<td>859.79</td>
<td>91478.04</td>
<td>66243.21</td>
<td>1623783.13</td>
</tr>
</tbody>
</table>

### Units of Solution Are:

- Diameters - Inch
- Length - Feet
- Heads - Feet
- Elevations - Feet
- Pressures - psi
- Flow Rates - lb/s

### Water Formula

#### Pipe Data:

<table>
<thead>
<tr>
<th>Pipe No.</th>
<th>From</th>
<th>To</th>
<th>Length</th>
<th>Dia</th>
<th>Coef</th>
<th>Flow Rate</th>
<th>Velocity</th>
<th>Head Loss</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1905.0</td>
<td>21.5</td>
<td>0.036</td>
<td>1184.0</td>
<td>6.86</td>
<td>9.97</td>
<td>5.01 HEAT EXCHANGERS IN</td>
</tr>
</tbody>
</table>

### Node Data:

<table>
<thead>
<tr>
<th>Node</th>
<th>Demand</th>
<th>Vol/2</th>
<th>Elev</th>
<th>Head</th>
<th>Pressure</th>
<th>Elevation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1000.0</td>
<td>57</td>
<td>2744</td>
<td>972</td>
<td>973.55</td>
<td>973.55</td>
</tr>
</tbody>
</table>

---

**FLOW FROM CONDENSER**

**MAKE UP WATER**

**FEEDS BOILER**

**ARRIVES AT BOILER**

---

**-132-**
This example illustrates the use of the program to solve for pipe diameters in order to satisfy dual demands and pressures at all nodes of the network. For this example the pressures at all nodes are specified at 40 psi except at the 3 nodes 1, 5 and 8 running through the center of the network where the pressure is set at 70, 60 and 50 psi, respectively. These pressures must exceed 40 psi for a solution to be possible, since the ground elevations of some nodes are larger than the ground elevation of the center nodes. The diameter of the following pipes are specified: 7-10 inch, 11 through 18-8 inch. The remaining J = 9 diameters are given zero values in the input communicating that these 9 diameters are to be solved for as part of the solution. In addition to the demand shown, a fire demand of 1000 gpm exists at node 9.

From the solution it is worth noting the following:

1. The pump does not produce sufficient head for a solution and, therefore, the program added head to the pump. In this process it assumed a friction loss of 1 ft in pipe #1, resulting in the relatively large diameter of 25.75 inches.

2. The reservoir supplies a relatively small portion of the total flow.

3. After solving for diameters to satisfy the dual specification of pressure and demand an analysis solution is obtained using the nearest nominal diameters. This is done since NOMDIA = 1 is in the $SPECIF$ list.
<table>
<thead>
<tr>
<th>NODE</th>
<th>DIAM</th>
<th>AREA</th>
<th>LENGTH</th>
<th>COEF.</th>
<th>RATE</th>
<th>VEL</th>
<th>LOSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>23.51</td>
<td>3.016</td>
<td>24.0</td>
<td>.01020</td>
<td>18.47</td>
<td>6.12</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>4.07</td>
<td>0.090</td>
<td>4.0</td>
<td>.01020</td>
<td>3.743</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>3</td>
<td>4.38</td>
<td>0.155</td>
<td>4.0</td>
<td>.01020</td>
<td>0.072</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>5</td>
<td>16.26</td>
<td>1.443</td>
<td>15.0</td>
<td>.01020</td>
<td>9.490</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>6</td>
<td>4.29</td>
<td>0.100</td>
<td>4.0</td>
<td>.01020</td>
<td>0.628</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>5</td>
<td>3.08</td>
<td>0.052</td>
<td>4.0</td>
<td>.01020</td>
<td>0.253</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>8</td>
<td>10.00</td>
<td>0.546</td>
<td>4.0</td>
<td>.01020</td>
<td>2.267</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>7</td>
<td>4.58</td>
<td>0.115</td>
<td>4.0</td>
<td>.01020</td>
<td>0.599</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>9</td>
<td>9.26</td>
<td>0.444</td>
<td>10.0</td>
<td>.01020</td>
<td>3.246</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>8</td>
<td>8.32</td>
<td>0.378</td>
<td>8.0</td>
<td>.01020</td>
<td>1.060</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>6</td>
<td>8.00</td>
<td>0.349</td>
<td>8.0</td>
<td>.01020</td>
<td>1.356</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>4</td>
<td>8.00</td>
<td>0.349</td>
<td>8.0</td>
<td>.01020</td>
<td>1.485</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>5</td>
<td>8.00</td>
<td>0.349</td>
<td>8.0</td>
<td>.01020</td>
<td>1.783</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>8</td>
<td>5.80</td>
<td>0.349</td>
<td>8.0</td>
<td>.01020</td>
<td>2.591</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>4</td>
<td>8.00</td>
<td>0.349</td>
<td>8.0</td>
<td>.01020</td>
<td>2.598</td>
</tr>
<tr>
<td>16</td>
<td>2</td>
<td>3</td>
<td>8.00</td>
<td>0.349</td>
<td>8.0</td>
<td>.01020</td>
<td>2.598</td>
</tr>
<tr>
<td>17</td>
<td>7</td>
<td>5</td>
<td>8.00</td>
<td>0.349</td>
<td>8.0</td>
<td>.01020</td>
<td>2.124</td>
</tr>
<tr>
<td>18</td>
<td>8</td>
<td>9</td>
<td>8.00</td>
<td>0.349</td>
<td>8.0</td>
<td>.01020</td>
<td>1.991</td>
</tr>
</tbody>
</table>

**NODE DATA**

**NO. FROM TO** | **DIAM.** | **AREA** | **LENGTH** | **COEF.** | **RATE** | **VEL** | **LOSS**
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2</td>
<td>22.28</td>
<td>110.00</td>
<td>161.54</td>
<td>70.00</td>
<td>100.00</td>
<td>271.54</td>
<td></td>
</tr>
<tr>
<td>2 3</td>
<td>22.28</td>
<td>100.00</td>
<td>92.31</td>
<td>40.00</td>
<td>192.31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 4</td>
<td>22.28</td>
<td>105.00</td>
<td>92.31</td>
<td>40.00</td>
<td>197.31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 5</td>
<td>22.28</td>
<td>118.00</td>
<td>92.31</td>
<td>40.00</td>
<td>210.31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 6</td>
<td>22.28</td>
<td>113.00</td>
<td>138.46</td>
<td>60.00</td>
<td>253.46</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 7</td>
<td>22.28</td>
<td>120.00</td>
<td>92.31</td>
<td>40.00</td>
<td>212.31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7 8</td>
<td>22.28</td>
<td>120.00</td>
<td>92.31</td>
<td>40.00</td>
<td>212.31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8 9</td>
<td>22.28</td>
<td>115.00</td>
<td>118.46</td>
<td>50.00</td>
<td>240.38</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9 10</td>
<td>22.28</td>
<td>125.00</td>
<td>92.31</td>
<td>40.00</td>
<td>217.31</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**PUMPS**

**DOUBLE LOOPED SYSTEM DESIGNED.**

**ALL DEMAND FLOWS ARE MULTIPLIED BY 1.0000**

**PIES**

**NOISES**

**SOURCE PUMPS**

**BOOSTER PUMPS**

**NOZZLES**

**CHECK VELVES**

**BACK PRES K.**

**DIFF. HEAD DEV 0**

**TOTAL POWER LOSS = 126.228 H.P. = 94.164 KWATTS. ENERGY LOSS = 2259.944 KW/HRS/ DAY**
EXAMPLE PROBLEM 7

This example network has demands given in Newtons per second or a weight flowrate in the SI units. Four points instead of the usual 3 are used to define the pump characteristics. The sources of supply are numbered as nodes. The check option is also used in this network to help assure that the input is correct.

INPUT DATA FOR EXAMPLE 7

NETWORK USING SI UNITS WITH FLOWRATES IN NEWTONS PER SECOND (I.E. A WEIGHT FLOWRATE). THE AVERAGE TEMPERATURE OF THE WATER IS 80 DEGREES CELSIUS

S MENU NODES=1,NUNIT=2,NGPM=3,NFLOW=6,GAMMA=9530.,VISC=3.67E-7,NUMP=1,CHAR=0,COEF=2.59E-4,NEH=2,NELEV=0 $END

RESER
7 122
8 107
PUMPS
9 830 28.5 970 28.1 1110 27.5 1390 26 1525 25.1 107
VALVE
8 460 97.5
PIPE-NODES
.1 150. 7 118. 2
.2 370. 3 2 555. 91.5
.2 490. 3 4 1665. 61.
.25 610. 4 1 277.5 85.3
.25 550. 2 1
.2 670. 5 3 277.5 85.3
.2 305. 9 110. 5 277.5 82.5
.2 915. 5 4
.5 425. 5 6 832.5 61.
.15 610. 6 4
.1 130. 1 100. 8 100.
RUN
555. 91.5 -1 -2 5/
277.5 85.3 2 -6 3/
1665. 61. -3 -8 -10 4/
277.5 85.3 -5 -4 -11/
277.5 82.5 -7 6 8 9/
832.5 61. 10 -9

-135-
EXAMPLE PROBLEM 8

In this example the DHEAD command is used to design a pump. The nozzle at the end of pipe 14 is used to determine what fire flow might be obtained from a fire nozzle of .05 m dia. attached at node 7. The question being asked is, "If with this fire demand, what head and capacity must the pump have if the head at node 8 is to be maintained 11 meters above the ground elevation?" The first column of data below the sketch of the network shows this input and the following pages show the solution results.

In the second column of data below the sketch a fire demand of 0.063 m$^3$/s (1000 gpm) has been specified from pipe 14 and furthermore the pressure at node 1, which is too small in the solution for the first situation has now been specified equal to 275 kPa. For this situation the pump must produce a head of 110.2 m instead of only 40.5 m as for the first situation. The reservoir now supplies only 0.155 m$^3$/s and the pump 0.381 m$^3$/s.

EXAMPLE IN WHICH THE DHEAD COMMAND IS USED TO SIZE A PUMP.

/*
$SPECIF NFLOW=3,NUMPIP=1,NUNIT=2,
NPGPM=3,PEAKF=1.8,COEFRO=.00005 $END
PIPE
.25 300. 1 .04 430.
.2 1000. 1 2 .05 420.
.25 300. 2
.25 800. 2 3 .04 420.
.25 1500. 1 3
.2 900. 3 8 .03 405.
.2 1000. 4 3
.25 800. 1 4 .025 410.
.2 600. 4 6 .02 400.
.25 800. 4 5 .03 405.
.2 300. 5 7 400.
.2 300. 6 7
.2 300. 7 8
.2 10. 7
NOZZLE
14 .05 .6 400./
RESER
1 .04 430
3 460
1 400
DHEAD
1 80 8 3 419
RUN
*/

EXAMPLE IN WHICH THE DHEAD COMMAND IS USED TO SIZE A PUMP.

/*
$SPECIF NFLOW=3,NUMPIP=1,NUNIT=2,
NPGPM=3,PEAKF=1.8,COEFRO=0.00005 $END
PIPE
.25 300. 1 .04 430.
.2 1000. 1 2 .05 420.
.25 300. 2
.25 800. 2 3 .04 420.
.25 1500. 1 3
.2 900. 3 8 .03 405.
.2 1000. 4 3
.25 800. 1 4 .025 410.
.2 600. 4 6 .02 400.
.25 800. 4 5 .03 405.
.2 300. 5 7 400.
.2 300. 6 7
.2 300. 7 8
.2 10. 7 9 .063 400.
RESER
1 400
DHEAD
1 80 -1 3 275000
RUN
*/
Example in which the head command is used to size a pump.

LI demand flows are multiplied by 1.8000.

Pipes 14
Nodes 6
Source pumps 0
 Booster pumps 0
Reservoirs 2
Min. line loss 0
Pumps 0
Nozzles 1
Check valves 0
Back pres. 0
Diff. head dev 1

Reservoirs plus nozzles:

<table>
<thead>
<tr>
<th>No.</th>
<th>Pipe node elevation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>14</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

Nozzles:

Pipe ground elev. Coef.2 area

<table>
<thead>
<tr>
<th>No.</th>
<th>Flow from pumps and reservoirs equals</th>
<th>Iteration</th>
<th>1 sum of differences</th>
<th>0.575-02</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Iteration</td>
<td>2 sum of differences</td>
<td>0.914-01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Iteration</td>
<td>3 sum of differences</td>
<td>0.707-01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Iteration</td>
<td>4 sum of differences</td>
<td>0.170-01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Iteration</td>
<td>5 sum of differences</td>
<td>0.123-00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Iteration</td>
<td>6 sum of differences</td>
<td>0.746-03</td>
</tr>
</tbody>
</table>

Losses due to fluid friction in all pipes:

Power loss= 131.227 H.P. = 97.895 kWATTS

Energie loss= 2349.485 KWHRS /DAY

Devices with diff. and asscd. specified head node:

Device pipe head run. with spec. head

<table>
<thead>
<tr>
<th>No.</th>
<th>No. increment no.</th>
<th>Head head</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>40.40</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>8.40</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>14.00</td>
</tr>
<tr>
<td>4</td>
<td>9</td>
<td>2.00</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>0.00</td>
</tr>
<tr>
<td>6</td>
<td>11</td>
<td>0.00</td>
</tr>
<tr>
<td>7</td>
<td>12</td>
<td>0.00</td>
</tr>
<tr>
<td>8</td>
<td>13</td>
<td>0.00</td>
</tr>
<tr>
<td>9</td>
<td>14</td>
<td>0.00</td>
</tr>
<tr>
<td>10</td>
<td>15</td>
<td>0.00</td>
</tr>
<tr>
<td>11</td>
<td>16</td>
<td>0.00</td>
</tr>
<tr>
<td>12</td>
<td>17</td>
<td>0.00</td>
</tr>
<tr>
<td>13</td>
<td>18</td>
<td>0.00</td>
</tr>
<tr>
<td>14</td>
<td>19</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Pipe original flowrate incremental head head equivalent

<table>
<thead>
<tr>
<th>No.</th>
<th>Diameter (CMS)</th>
<th>Head loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1000.00</td>
<td>1.00</td>
</tr>
<tr>
<td>2</td>
<td>500.00</td>
<td>0.50</td>
</tr>
</tbody>
</table>

Units of Solution are:

Diameters - Meter
Length - Meter
Heads - Meter
Elevations - Meter
Pressures - K N/M

Flowrates - CYS

UHNT-WEISBACH FORMULA USED FOR COMPUTING HEAD LOSS

Pipe data:

<table>
<thead>
<tr>
<th>No.</th>
<th>Pipe nodes</th>
<th>Head loss</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Node data:

<table>
<thead>
<tr>
<th>No.</th>
<th>Demand head</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0720</td>
</tr>
<tr>
<td>2</td>
<td>0.0900</td>
</tr>
<tr>
<td>3</td>
<td>0.0720</td>
</tr>
<tr>
<td>4</td>
<td>0.0450</td>
</tr>
<tr>
<td>5</td>
<td>0.0600</td>
</tr>
<tr>
<td>6</td>
<td>0.0200</td>
</tr>
<tr>
<td>7</td>
<td>0.0000</td>
</tr>
<tr>
<td>8</td>
<td>0.0400</td>
</tr>
</tbody>
</table>

-137-
EXAMPLE PROBLEM 9

In this example the DREAD command is used to size a pump while simultaneously sizing pipes 12 and 5. Since for each such device that is to be sized a dual specification of pressure as well as demand must be given at another node, pressures must be specified at 3 nodes, i.e. 3 DHEAD devices of type 12 must be used in the input data. The criteria used for this design is that when the demand is 1.1 times the values given on the sketch, the HGL at node 5 just downstream from the reservoir should be 490 m, i.e. the head loss in pipe 60 from the reservoir should be 10 meters. This assumption establishes data for one DHEAD device. It is further decided that the HGL at nodes 7 and 1 should be 475 m and 465 m respectively. These assumptions establish the criteria for the DHEAD devices in pipes 12 and 5 respectively.

The solution to this example is given on the following pages. These results indicate that pipes 12 and 5 should be 35 cm and 30 cm in diameter respectively and the head the pump should produce should be 84.6 m.
### Balancing Flow Between a Single Reservoir and Pump

The head devices are used to fix pressure at nodes 5, 10, and 12 in addition to sizing the pump. All demand flows are multiplied by 1,100.

**Pipes**

- Node: 13
- Source Pumps: 0
- Booster Pumps: 0
- Reservoirs: 2
- Minor Losses: 0
- Nozzles: 0
- Check Valves: 0
- Back Pressure: 0
- Diff. Head Dev: 3

**Reservoirs plus nozzles**

<table>
<thead>
<tr>
<th>No.</th>
<th>Pipe Node Elevation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6 10 500.00</td>
</tr>
<tr>
<td>2</td>
<td>13 11 470.00</td>
</tr>
</tbody>
</table>

Flow from pumps and reservoirs equals 0.451.

**Losses due to fluid friction in all pipes**

- Power loss: $29,478$ H.P. = $394,990$ KWatts
- Energy loss: $9479.770$ KWhrs/day

**Devices with diff., head and assoc. specified head node**

**Device Pipe**

- Source Pipe: 2
- Head: 89.78
- Jun. with Spec. Head: 140.00

**Pipe original flow rate incremental head total head E equivalent diameter**

<table>
<thead>
<tr>
<th>No.</th>
<th>Diameter (CMS)</th>
<th>Head</th>
<th>Loss</th>
<th>Total Head</th>
<th>E equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>0.35</td>
<td>0.35</td>
<td>64.64</td>
<td>41.66</td>
<td>N.O equivalent dia. possible</td>
</tr>
<tr>
<td>12</td>
<td>0.25</td>
<td>0.14</td>
<td>49.35</td>
<td>61.51</td>
<td>12.19 0.02756 0.345</td>
</tr>
<tr>
<td>11</td>
<td>0.35</td>
<td>0.35</td>
<td>85.40</td>
<td>98.56</td>
<td>13.18 0.02657 0.299</td>
</tr>
</tbody>
</table>

**Units of solution are:**

- Diameters - Meter
- Length - Meter
- Heads - Meter
- Elevations - Meter
- Pressures - kV/SI
- Flowrates - CMS

**Darcy-Weisbach formula used for computing head loss**

**Pipe data**

<table>
<thead>
<tr>
<th>No.</th>
<th>From</th>
<th>To</th>
<th>Length</th>
<th>DIAM</th>
<th>COEF</th>
<th>Flow Rate</th>
<th>Velocity</th>
<th>Loss</th>
<th>/1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1500.</td>
<td>0.150</td>
<td>0.00050</td>
<td>0.0093</td>
<td>0.528</td>
<td>4.099</td>
<td>2.733</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>1500.</td>
<td>0.150</td>
<td>0.00050</td>
<td>0.0227</td>
<td>0.717</td>
<td>7.447</td>
<td>4.294</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>1</td>
<td>1000.</td>
<td>0.200</td>
<td>0.00050</td>
<td>0.0423</td>
<td>1.347</td>
<td>11.940</td>
<td>6.840</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>3</td>
<td>1000.</td>
<td>0.200</td>
<td>0.00050</td>
<td>0.0567</td>
<td>1.804</td>
<td>21.076</td>
<td>12.076</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>4</td>
<td>1500.</td>
<td>0.200</td>
<td>0.00050</td>
<td>0.1012</td>
<td>3.221</td>
<td>98.563</td>
<td>65.709</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>5</td>
<td>500.</td>
<td>0.250</td>
<td>0.00050</td>
<td>0.0995</td>
<td>2.027</td>
<td>10.000</td>
<td>20.000</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>6</td>
<td>1500.</td>
<td>0.250</td>
<td>0.00050</td>
<td>0.0431</td>
<td>1.267</td>
<td>3.676</td>
<td>1.149</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>6</td>
<td>1200.</td>
<td>0.200</td>
<td>0.00050</td>
<td>0.0385</td>
<td>1.171</td>
<td>10.783</td>
<td>8.866</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>5</td>
<td>1200.</td>
<td>0.250</td>
<td>0.00050</td>
<td>0.1429</td>
<td>2.912</td>
<td>61.513</td>
<td>41.010</td>
</tr>
<tr>
<td>10</td>
<td>9</td>
<td>6</td>
<td>1200.</td>
<td>0.250</td>
<td>0.00050</td>
<td>0.1429</td>
<td>2.912</td>
<td>61.513</td>
<td>41.010</td>
</tr>
<tr>
<td>11</td>
<td>8</td>
<td>6</td>
<td>1500.</td>
<td>0.250</td>
<td>0.00050</td>
<td>0.1429</td>
<td>2.912</td>
<td>61.513</td>
<td>41.010</td>
</tr>
<tr>
<td>12</td>
<td>9</td>
<td>7</td>
<td>1000.</td>
<td>0.350</td>
<td>0.00050</td>
<td>0.3515</td>
<td>3.653</td>
<td>41.662</td>
<td>41.662</td>
</tr>
</tbody>
</table>

**Node data**

<table>
<thead>
<tr>
<th>No.</th>
<th>Flow</th>
<th>DIAM</th>
<th>ELEV</th>
<th>HEAD</th>
<th>PRESSURE</th>
<th>ELEV</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0330</td>
<td>1.17</td>
<td>420.</td>
<td>45.00</td>
<td>441.00</td>
<td>465.00</td>
</tr>
<tr>
<td>2</td>
<td>0.0220</td>
<td>0.78</td>
<td>420.</td>
<td>40.90</td>
<td>409.83</td>
<td>460.90</td>
</tr>
<tr>
<td>3</td>
<td>0.0440</td>
<td>1.55</td>
<td>420.</td>
<td>48.35</td>
<td>473.80</td>
<td>468.35</td>
</tr>
<tr>
<td>4</td>
<td>0.0440</td>
<td>1.55</td>
<td>410.</td>
<td>66.84</td>
<td>655.03</td>
<td>476.84</td>
</tr>
<tr>
<td>5</td>
<td>0.0220</td>
<td>0.78</td>
<td>450.</td>
<td>40.00</td>
<td>392.00</td>
<td>490.00</td>
</tr>
<tr>
<td>6</td>
<td>0.0550</td>
<td>1.34</td>
<td>480.</td>
<td>29.42</td>
<td>288.35</td>
<td>489.42</td>
</tr>
<tr>
<td>7</td>
<td>0.0660</td>
<td>2.33</td>
<td>430.</td>
<td>45.00</td>
<td>441.00</td>
<td>475.00</td>
</tr>
<tr>
<td>8</td>
<td>0.0550</td>
<td>1.34</td>
<td>460.</td>
<td>40.78</td>
<td>399.48</td>
<td>500.78</td>
</tr>
<tr>
<td>9</td>
<td>0.1100</td>
<td>3.88</td>
<td>470.</td>
<td>42.98</td>
<td>421.17</td>
<td>512.98</td>
</tr>
<tr>
<td>10</td>
<td>-0.0995</td>
<td>3.51</td>
<td>480.</td>
<td>20.00</td>
<td>196.00</td>
<td>500.00</td>
</tr>
<tr>
<td>11</td>
<td>-0.1515</td>
<td>1.21</td>
<td>450.</td>
<td>20.00</td>
<td>196.00</td>
<td>470.00</td>
</tr>
</tbody>
</table>
EXAMPLE PROBLEM 10

Based on the analysis in Example 9, select a suitable pump (several may exist in parallel or series) and obtain a time dependent solution over 24 hours of time in 2 hour increments using a typical demand function. Develop an appropriate rule for the operation of the pumps at the station that supplies the network through pipe 13 and verify the appropriateness of the rule and the rest of the network design by determining how close the water surface elevation in the reservoir is after 24 hours to where it began at time zero.

```
EXAMPLE 10 THAT IS A TIME DEPEND. SOL. OF PROB. 9
/*
 $SPECIF NFLOW=3,NPGPM=3,
  NUNIT=3,NODESP=1,NPLENG=1
  ISIML=1 $END
PIPE-
  1 15. 1.5 1 .03 420. 2 .02 420.
  2 15. 1.5 3 .04 420. 2
  3 20. 1. 4 1
  4 20. 1. 6 .04 420. 3
  5 30. 1.5 5 .04 420.
  6 25. .5 10 480. 5 .02 450.
  7 25. 1.5 6 5
  8 20. 1.2 7 4
  9 20. 1.2 8 5
 10 25. 1.2 9 6 .05 460.
 11 20. 1.5 8 7 .06 430.
 12 35. 1.5 9 8 .05 460.
 13 35. 1.1 11 490. 9 .1 470.
RESER
  10 500
PUMPS
  11 .15 102 .35 97 .45 92 470
RUN
  $TDATA HTIME=24,INCHR=2,PRINTT=3,
  ISUNIT=4,NPUNOD=0,NPNRES=1 $END
PIPE TABLE
  ALL
NODE TABLE
  ALL
RESER. TABLE
  1
END TABLES
DEMAND FUNCTION
  1 2 .9 6 .05 8 .6 12 1 16 1.5 18 1.3 24 1/<
  1-9/
STORAGE FUNCTION
  1 470 300 500 2500 520 7500 530 10000
  1/
PUMP RULES
  1 2 1 2 480 3 500 2 520 1 530 0/
END SIML
```

The solution output for this simulation is not given herein. However, much of the information of the solution is provided by the special tables requested by the parameter PRINTT=3 in the $TDATA list and the command that follows this. These special tables are given on the following page.
### SPECIAL TABLE GIVING PRESSURES AT DESIGNATED NODES AS A FUNCTION OF TIME

<table>
<thead>
<tr>
<th>TIME (HOURS)</th>
<th>NODE NO. 1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>652.21</td>
<td>625.30</td>
<td>670.67</td>
<td>723.77</td>
<td>402.41</td>
<td>486.40</td>
<td>625.40</td>
<td>566.03</td>
<td>570.07</td>
</tr>
<tr>
<td>2.00</td>
<td>687.09</td>
<td>668.70</td>
<td>711.20</td>
<td>743.29</td>
<td>423.45</td>
<td>485.40</td>
<td>648.41</td>
<td>608.10</td>
<td>617.12</td>
</tr>
<tr>
<td>4.00</td>
<td>749.80</td>
<td>741.18</td>
<td>779.30</td>
<td>786.45</td>
<td>466.28</td>
<td>510.80</td>
<td>699.30</td>
<td>637.94</td>
<td>635.66</td>
</tr>
<tr>
<td>6.00</td>
<td>806.32</td>
<td>803.37</td>
<td>842.00</td>
<td>831.14</td>
<td>514.17</td>
<td>546.33</td>
<td>752.48</td>
<td>681.06</td>
<td>672.85</td>
</tr>
<tr>
<td>8.00</td>
<td>845.33</td>
<td>844.17</td>
<td>883.08</td>
<td>865.08</td>
<td>548.86</td>
<td>577.09</td>
<td>790.12</td>
<td>709.15</td>
<td>696.34</td>
</tr>
<tr>
<td>10.00</td>
<td>783.75</td>
<td>771.00</td>
<td>806.71</td>
<td>826.31</td>
<td>496.94</td>
<td>557.49</td>
<td>733.31</td>
<td>655.44</td>
<td>647.77</td>
</tr>
<tr>
<td>12.00</td>
<td>711.01</td>
<td>683.40</td>
<td>727.89</td>
<td>783.00</td>
<td>465.28</td>
<td>510.80</td>
<td>699.30</td>
<td>637.94</td>
<td>635.66</td>
</tr>
<tr>
<td>14.00</td>
<td>431.65</td>
<td>379.92</td>
<td>463.43</td>
<td>568.81</td>
<td>307.06</td>
<td>417.63</td>
<td>445.21</td>
<td>431.58</td>
<td>452.44</td>
</tr>
<tr>
<td>16.00</td>
<td>314.89</td>
<td>255.46</td>
<td>349.87</td>
<td>471.88</td>
<td>226.76</td>
<td>343.67</td>
<td>341.94</td>
<td>358.37</td>
<td>390.55</td>
</tr>
<tr>
<td>18.00</td>
<td>462.12</td>
<td>417.10</td>
<td>489.15</td>
<td>581.05</td>
<td>301.94</td>
<td>399.94</td>
<td>471.61</td>
<td>457.81</td>
<td>495.82</td>
</tr>
<tr>
<td>20.00</td>
<td>519.94</td>
<td>484.05</td>
<td>541.10</td>
<td>614.01</td>
<td>310.81</td>
<td>398.76</td>
<td>513.90</td>
<td>501.34</td>
<td>526.56</td>
</tr>
<tr>
<td>22.00</td>
<td>545.20</td>
<td>517.05</td>
<td>568.46</td>
<td>622.94</td>
<td>313.42</td>
<td>386.12</td>
<td>525.14</td>
<td>511.40</td>
<td>542.45</td>
</tr>
<tr>
<td>24.00</td>
<td>560.65</td>
<td>536.64</td>
<td>585.91</td>
<td>630.41</td>
<td>320.04</td>
<td>384.02</td>
<td>534.08</td>
<td>528.15</td>
<td>552.62</td>
</tr>
</tbody>
</table>

### SPECIAL TABLE GIVING FLOWRATES IN DESIGNATED PIPES AS A FUNCTION OF TIME

<table>
<thead>
<tr>
<th>TIME (HOURS)</th>
<th>PIPE NO. 1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>0.06</td>
<td>0.16</td>
<td>0.39</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.00</td>
<td>0.07</td>
<td>0.17</td>
<td>0.39</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.00</td>
<td>0.06</td>
<td>0.16</td>
<td>0.36</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.00</td>
<td>0.06</td>
<td>0.16</td>
<td>0.34</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.00</td>
<td>0.06</td>
<td>0.15</td>
<td>0.33</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10.00</td>
<td>0.06</td>
<td>0.15</td>
<td>0.35</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12.00</td>
<td>0.06</td>
<td>0.16</td>
<td>0.38</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14.00</td>
<td>0.06</td>
<td>0.18</td>
<td>0.44</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16.00</td>
<td>0.07</td>
<td>0.19</td>
<td>0.46</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18.00</td>
<td>0.08</td>
<td>0.18</td>
<td>0.45</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20.00</td>
<td>0.06</td>
<td>0.18</td>
<td>0.43</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22.00</td>
<td>0.06</td>
<td>0.18</td>
<td>0.43</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24.00</td>
<td>0.06</td>
<td>0.18</td>
<td>0.42</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### SPECIAL TABLE GIVING WATER SURFACE ELEV. IN RESERVOIRS

<table>
<thead>
<tr>
<th>TIME (HOURS)</th>
<th>DESIGNATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>500.00</td>
</tr>
<tr>
<td>2.00</td>
<td>499.16</td>
</tr>
<tr>
<td>4.00</td>
<td>500.01</td>
</tr>
<tr>
<td>6.00</td>
<td>501.51</td>
</tr>
<tr>
<td>8.00</td>
<td>502.68</td>
</tr>
<tr>
<td>10.00</td>
<td>506.09</td>
</tr>
<tr>
<td>12.00</td>
<td>506.99</td>
</tr>
<tr>
<td>14.00</td>
<td>506.01</td>
</tr>
<tr>
<td>16.00</td>
<td>502.08</td>
</tr>
<tr>
<td>18.00</td>
<td>496.45</td>
</tr>
<tr>
<td>20.00</td>
<td>491.90</td>
</tr>
<tr>
<td>22.00</td>
<td>489.43</td>
</tr>
<tr>
<td>24.00</td>
<td>489.07</td>
</tr>
</tbody>
</table>

It should be noted from these water surface elevations in the reservoir that the conditions specified in the problem statement have not been met, since at the end of the 24 hour simulation the w.s. elevation is well below 500 ft. Either more pumps need to be turned on at lower w.s. elevations, or the head produced by the pumps must be increased. Note that the pump char. given in the original data are for 2 pumps operating in parallel.
EXAMPLE PROBLEM 10A

In this variation of Example 10 the effect of placing a valve in pipe 5 in an attempt to keep the reservoir water level higher is attempted. In this application the valve is requested to close, and thus increase its minor loss as a rule that is governed by the water surface elevation in the reservoir. Therefore the RMINOR command is used within the simulation data. The summary tables requested in the input data are provided below. It can be noted by comparing these with the same summary tables given for Example 10, that the valve does help in reducing the flowrate in pipe 5 as the water surface in the reservoir falls below 500 feet, and therefore at the end of the 24 hours simulation the reservoir level is not quite as low as in the previous example. However, the single valve is not the entire solution, since the water surface elevation is still too low, but in addition negative pressures exist.

EXAMPLE 10 THAT IS A TIME DEPEND. SOL. OF PROB. 9
/*
$SPECIF AFL0W=3,NPGPM=3,
NUUNIT=3,NODESP=1,NPLENG=1
ISIML=1 SEND
PIPE
1 15.1 .03 420. 2 .02 420.
2 15.1 3 .04 420. 2
3 20.1 4 1
4 20.1 6 .04 420. 3
5 30.1 5 4 .04 420.
6 25.1 10 480. 5 .02 450.
7 25.1 6 5
8 20.1 2 7 4
9 20.1 2 8 5
10 25.1 2 9 6 .05 460.
11 20.1 5 8 7 .06 430.
12 35.1 5 8 8 .05 460.
13 35.1 11 450. 9 .1470.
MINOR
5 10.
RESER
10 500
PUMPS
11 .15 102 .35 97 .45 92 470
RUN
$STDATA HTIME=24,INCHR=2,PRINTT=3,
ISUNIT=4,NPUNOD=0,NPNRES=1 SEND
PIPE TABLE
ALL
NODE TABLE
ALL
RESER. TABLE
1
END TABLES
DEMAND FUNCTION
1 2 .9 6 .65 8 .6 12 1 16 1.5 18 1.3 24 1/
1-9/
STORAGE FUNCTION
1 470 300 500 2500 520 7500 530 10000
1/
PUMP RULES
1 2 1 2 480 3 500 2 520 1 530 0/
RMINOR
5 1 490 3500 495 2500 498 800 500 10 510 0/
END SIML

-142-
### SPECIAL TABLE GIVING PRESSURES AT DESIGNATED NODES AS A FUNCTION OF TIME

<table>
<thead>
<tr>
<th>TIME (HOURS)</th>
<th>NODE NO.</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td></td>
<td>647.52</td>
<td>629.02</td>
<td>669.25</td>
<td>718.22</td>
<td>402.13</td>
<td>486.49</td>
<td>620.01</td>
<td>564.96</td>
<td>569.43</td>
</tr>
<tr>
<td>2.00</td>
<td></td>
<td>614.61</td>
<td>610.79</td>
<td>687.29</td>
<td>658.79</td>
<td>417.47</td>
<td>487.01</td>
<td>572.94</td>
<td>593.79</td>
<td>608.69</td>
</tr>
<tr>
<td>4.00</td>
<td></td>
<td>749.36</td>
<td>741.03</td>
<td>779.74</td>
<td>785.79</td>
<td>466.03</td>
<td>511.96</td>
<td>698.81</td>
<td>638.25</td>
<td>635.94</td>
</tr>
<tr>
<td>6.00</td>
<td></td>
<td>806.64</td>
<td>803.76</td>
<td>842.62</td>
<td>831.39</td>
<td>514.91</td>
<td>547.37</td>
<td>752.77</td>
<td>681.48</td>
<td>673.18</td>
</tr>
<tr>
<td>8.00</td>
<td></td>
<td>845.93</td>
<td>844.78</td>
<td>883.77</td>
<td>865.66</td>
<td>549.58</td>
<td>578.05</td>
<td>790.67</td>
<td>709.59</td>
<td>696.69</td>
</tr>
<tr>
<td>10.00</td>
<td></td>
<td>784.02</td>
<td>771.42</td>
<td>807.36</td>
<td>824.68</td>
<td>497.73</td>
<td>538.53</td>
<td>655.87</td>
<td>648.13</td>
<td></td>
</tr>
<tr>
<td>12.00</td>
<td></td>
<td>710.61</td>
<td>683.40</td>
<td>728.40</td>
<td>782.35</td>
<td>459.91</td>
<td>500.31</td>
<td>682.78</td>
<td>604.55</td>
<td>600.61</td>
</tr>
<tr>
<td>14.00</td>
<td></td>
<td>428.28</td>
<td>377.86</td>
<td>463.03</td>
<td>564.63</td>
<td>307.72</td>
<td>418.63</td>
<td>441.90</td>
<td>431.45</td>
<td>452.48</td>
</tr>
<tr>
<td>16.00</td>
<td></td>
<td>305.80</td>
<td>249.49</td>
<td>347.90</td>
<td>307.72</td>
<td>418.63</td>
<td>441.90</td>
<td>431.45</td>
<td>452.48</td>
<td></td>
</tr>
<tr>
<td>18.00</td>
<td></td>
<td>12.20</td>
<td>29.93</td>
<td>368.21</td>
<td>62.73</td>
<td>307.72</td>
<td>418.63</td>
<td>441.90</td>
<td>431.45</td>
<td></td>
</tr>
<tr>
<td>20.00</td>
<td></td>
<td>620.02</td>
<td>669.25</td>
<td>718.22</td>
<td>402.13</td>
<td>486.49</td>
<td>620.01</td>
<td>564.96</td>
<td>569.43</td>
<td></td>
</tr>
<tr>
<td>22.00</td>
<td></td>
<td>669.25</td>
<td>718.22</td>
<td>402.13</td>
<td>486.49</td>
<td>620.01</td>
<td>564.96</td>
<td>569.43</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24.00</td>
<td></td>
<td>718.22</td>
<td>402.13</td>
<td>486.49</td>
<td>620.01</td>
<td>564.96</td>
<td>569.43</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### SPECIAL TABLE GIVING FLOWRATES IN DESIGNATED PIPES AS A FUNCTION OF TIME

<table>
<thead>
<tr>
<th>TIME (HOURS)</th>
<th>PIPE NO.</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td></td>
<td>0.01</td>
<td>0.01</td>
<td>0.04</td>
<td>0.05</td>
<td>0.08</td>
<td>0.02</td>
<td>0.02</td>
<td>0.00</td>
<td>0.06</td>
<td>0.12</td>
</tr>
<tr>
<td>2.00</td>
<td></td>
<td>0.00</td>
<td>0.02</td>
<td>0.03</td>
<td>0.05</td>
<td>0.05</td>
<td>-0.02</td>
<td>0.03</td>
<td>0.01</td>
<td>0.06</td>
<td>0.13</td>
</tr>
<tr>
<td>4.00</td>
<td></td>
<td>0.00</td>
<td>0.01</td>
<td>0.03</td>
<td>0.04</td>
<td>0.04</td>
<td>-0.05</td>
<td>0.05</td>
<td>0.01</td>
<td>0.06</td>
<td>0.13</td>
</tr>
<tr>
<td>6.00</td>
<td></td>
<td>0.00</td>
<td>0.01</td>
<td>0.02</td>
<td>0.04</td>
<td>0.03</td>
<td>-0.08</td>
<td>0.05</td>
<td>0.02</td>
<td>0.06</td>
<td>0.12</td>
</tr>
<tr>
<td>8.00</td>
<td></td>
<td>0.00</td>
<td>0.01</td>
<td>0.02</td>
<td>0.03</td>
<td>0.02</td>
<td>-0.08</td>
<td>0.06</td>
<td>0.02</td>
<td>0.06</td>
<td>0.12</td>
</tr>
<tr>
<td>10.00</td>
<td></td>
<td>0.01</td>
<td>0.01</td>
<td>0.03</td>
<td>0.04</td>
<td>0.05</td>
<td>-0.03</td>
<td>0.04</td>
<td>0.01</td>
<td>0.06</td>
<td>0.12</td>
</tr>
<tr>
<td>12.00</td>
<td></td>
<td>0.01</td>
<td>0.01</td>
<td>0.04</td>
<td>0.05</td>
<td>0.08</td>
<td>0.03</td>
<td>0.02</td>
<td>0.00</td>
<td>0.05</td>
<td>0.12</td>
</tr>
<tr>
<td>14.00</td>
<td></td>
<td>0.01</td>
<td>0.02</td>
<td>0.05</td>
<td>0.07</td>
<td>0.13</td>
<td>0.14</td>
<td>-0.02</td>
<td>-0.02</td>
<td>0.04</td>
<td>0.12</td>
</tr>
<tr>
<td>16.00</td>
<td></td>
<td>0.01</td>
<td>0.02</td>
<td>0.06</td>
<td>0.08</td>
<td>0.14</td>
<td>0.15</td>
<td>-0.03</td>
<td>-0.02</td>
<td>0.04</td>
<td>0.12</td>
</tr>
<tr>
<td>18.00</td>
<td></td>
<td>-0.01</td>
<td>0.03</td>
<td>0.03</td>
<td>0.08</td>
<td>0.06</td>
<td>0.07</td>
<td>-0.03</td>
<td>0.02</td>
<td>0.04</td>
<td>0.12</td>
</tr>
<tr>
<td>20.00</td>
<td></td>
<td>-0.01</td>
<td>0.03</td>
<td>0.03</td>
<td>0.08</td>
<td>0.04</td>
<td>0.03</td>
<td>-0.01</td>
<td>0.03</td>
<td>0.05</td>
<td>0.13</td>
</tr>
<tr>
<td>22.00</td>
<td></td>
<td>-0.01</td>
<td>0.03</td>
<td>0.02</td>
<td>0.07</td>
<td>0.04</td>
<td>-0.01</td>
<td>0.01</td>
<td>0.03</td>
<td>0.05</td>
<td>0.13</td>
</tr>
<tr>
<td>24.00</td>
<td></td>
<td>-0.01</td>
<td>0.03</td>
<td>0.02</td>
<td>0.07</td>
<td>0.04</td>
<td>-0.02</td>
<td>0.02</td>
<td>0.03</td>
<td>0.06</td>
<td>0.13</td>
</tr>
</tbody>
</table>

### SPECIAL TABLE GIVING FLOWRATES IN DESIGNATED PIPES AS A FUNCTION OF TIME

<table>
<thead>
<tr>
<th>TIME (HOURS)</th>
<th>PIPE NO.</th>
<th>11</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td></td>
<td>0.06</td>
<td>0.16</td>
<td>0.39</td>
</tr>
<tr>
<td>2.00</td>
<td></td>
<td>0.07</td>
<td>0.17</td>
<td>0.39</td>
</tr>
<tr>
<td>4.00</td>
<td></td>
<td>0.06</td>
<td>0.16</td>
<td>0.36</td>
</tr>
<tr>
<td>6.00</td>
<td></td>
<td>0.06</td>
<td>0.15</td>
<td>0.34</td>
</tr>
<tr>
<td>8.00</td>
<td></td>
<td>0.06</td>
<td>0.15</td>
<td>0.33</td>
</tr>
<tr>
<td>10.00</td>
<td></td>
<td>0.06</td>
<td>0.15</td>
<td>0.35</td>
</tr>
<tr>
<td>12.00</td>
<td></td>
<td>0.06</td>
<td>0.16</td>
<td>0.38</td>
</tr>
<tr>
<td>14.00</td>
<td></td>
<td>0.06</td>
<td>0.18</td>
<td>0.44</td>
</tr>
<tr>
<td>16.00</td>
<td></td>
<td>0.07</td>
<td>0.19</td>
<td>0.46</td>
</tr>
<tr>
<td>18.00</td>
<td></td>
<td>0.10</td>
<td>0.21</td>
<td>0.46</td>
</tr>
<tr>
<td>20.00</td>
<td></td>
<td>0.10</td>
<td>0.20</td>
<td>0.45</td>
</tr>
<tr>
<td>22.00</td>
<td></td>
<td>0.09</td>
<td>0.20</td>
<td>0.44</td>
</tr>
<tr>
<td>24.00</td>
<td></td>
<td>0.09</td>
<td>0.19</td>
<td>0.43</td>
</tr>
</tbody>
</table>

### SPECIAL TABLE GIVING WATER SURFACE ELEV. IN RESERVOIRS

<table>
<thead>
<tr>
<th>TIME (HOURS)</th>
<th>DESIGNATION</th>
<th>ELEV.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>500.00</td>
<td></td>
</tr>
<tr>
<td>2.00</td>
<td>499.17</td>
<td></td>
</tr>
<tr>
<td>4.00</td>
<td>500.14</td>
<td></td>
</tr>
<tr>
<td>6.00</td>
<td>501.64</td>
<td></td>
</tr>
<tr>
<td>8.00</td>
<td>503.80</td>
<td></td>
</tr>
<tr>
<td>10.00</td>
<td>506.20</td>
<td></td>
</tr>
<tr>
<td>12.00</td>
<td>507.10</td>
<td></td>
</tr>
<tr>
<td>14.00</td>
<td>506.12</td>
<td></td>
</tr>
<tr>
<td>16.00</td>
<td>502.18</td>
<td></td>
</tr>
<tr>
<td>18.00</td>
<td>496.63</td>
<td></td>
</tr>
<tr>
<td>20.00</td>
<td>492.91</td>
<td></td>
</tr>
<tr>
<td>22.00</td>
<td>491.42</td>
<td></td>
</tr>
<tr>
<td>24.00</td>
<td>491.80</td>
<td></td>
</tr>
</tbody>
</table>
EXAMPLE PROBLEM 11

In this example, the demands are to be held constant as specified on the sketch, but storage in the reservoirs accounted for in the simulation. In the input data listed, the demand function that is given contains a multiplier of 1 at 1, 12, and 24 hours. An alternative would be to completely remove the command DEMAND function and the data it enters. In the input data for this problem NTRAND=2, which indicates that a direct access file will be created from which the auxiliary program PRINT that is described at the end of this users manual, can be used to recreate any portions of the solution that the user desires. In the solution that follows, selected bits and pieces of the output are given, including using the auxiliary program PRINT.

EXAMPLE OF SIMULATION NETWORK
/*
$SPECIF NPRINT=2, NFLOW=0, NPCPM=0, ISIML=1, NTRAND=2, $END
PIPEG
1 7 2 500 12 .0102
2 3 1200 8/
3 4 1600 8/
4 1 4 2000 10/
5 2 1 1800 10/
6 3 5 2200/
7 9 5 1000 12/
8 5 4 3000 8/
9 5 6 1400 6/
10 4 6 2000 10./
11 8 1 500 12/

NODGES
1 1. 280
2 2 300
3 1.5 230
4 1 6 200
5 1 270
6 3 200
7 0 390
6 0 240
9 0 360

RESER
7 408
8 350

PUMPS
9 3 80 4 77 5 72 350

VALVE
8 1500 320
MINOR
4 10/

RUN
$DATA PRINTT=3, INCHR=12, DTIME=1, ISUNIT=1, $END
PIPE TABLE
1-3/
NODE TABLE
7 8 9/
END TABLES
DEMAND FUNCTION
1 1 1 12 1. 24 1.0
1-6/
STORAGE FUNCTION
1 300 2800 370 2900 460 3000/
7 8/
PUMP SCHEDULE
9 2 0 2 12 2 24 2/
END SIMULATION
The solution to Example 11 is provided in several forms: (1) The usual tables for each time step are available in a file called "TABLE." Special tables giving flow rates in pipes 1, 2, 3 and the pressures at nodes 7, 8, and 9. (3) A direct access file is created which contains the data needed to completely regenerate the simulation solution. With the auxiliary program PRINT it is possible to have any information at any time provided by simple requests. The use of the auxiliary program is also described in this manual. The usual tables given as (1) above are listed immediately below.

### Tables for Time 12 (1200 hours)

<table>
<thead>
<tr>
<th>NODE</th>
<th>DEMAND</th>
<th>ELEV</th>
<th>PRESSURE</th>
<th>NODE DATA (TIME=12)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7.815</td>
<td>3507.74</td>
<td>0.00000</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3.967</td>
<td>332.57</td>
<td>0.00000</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3.435</td>
<td>318.02</td>
<td>0.00000</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3.500</td>
<td>320.00</td>
<td>0.00000</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>3.500</td>
<td>320.00</td>
<td>0.00000</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>3.500</td>
<td>320.00</td>
<td>0.00000</td>
<td></td>
</tr>
</tbody>
</table>

The following solution results were obtained from the direct access file (form (3) above). These results were obtained by requesting (a) the flow conditions in pipes 3 through 6 at 12 hours; (b) the conditions at nodes 2 through 4 at time 12 hours; (c) the table of pipe data at time 12 hours; (d) the node table at time 12 hours; (e) the complete solutions for times 12 and 24 hours.

### Tables for Time 24 (2400 hours)

<table>
<thead>
<tr>
<th>NODE</th>
<th>DEMAND</th>
<th>ELEV</th>
<th>PRESSURE</th>
<th>NODE DATA (TIME=24)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7.815</td>
<td>3507.74</td>
<td>0.00000</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3.967</td>
<td>332.57</td>
<td>0.00000</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3.435</td>
<td>318.02</td>
<td>0.00000</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3.500</td>
<td>320.00</td>
<td>0.00000</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>3.500</td>
<td>320.00</td>
<td>0.00000</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>3.500</td>
<td>320.00</td>
<td>0.00000</td>
<td></td>
</tr>
</tbody>
</table>

The special tables given as (2) previously, follow.

### Special Table Giving Pressures at Designated Nodes as a Function of Time

<table>
<thead>
<tr>
<th>TIME (HOURS)</th>
<th>NODE NO</th>
<th>PRESSURE</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.00</td>
<td>7</td>
<td>1.77</td>
</tr>
<tr>
<td>24.00</td>
<td>7</td>
<td>1.65</td>
</tr>
</tbody>
</table>

### Special Table Giving Flow Rates in Designated Pipes as a Function of Time

<table>
<thead>
<tr>
<th>TIME (HOURS)</th>
<th>PIPE NO</th>
<th>FLOW RATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.00</td>
<td>1</td>
<td>1.68</td>
</tr>
<tr>
<td>24.00</td>
<td>1</td>
<td>1.48</td>
</tr>
</tbody>
</table>

The special tables given as (2) previously, follow.
EXAMPLE PROBLEM 12
(Simulation with 2 demand functions and storage as function of elevation)

All diameters in inches and lengths in feet. All cast iron pipe.

Pump Characteristics

<table>
<thead>
<tr>
<th>Q (cfs)</th>
<th>h (ft)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.5</td>
<td>105.</td>
</tr>
<tr>
<td>3.0</td>
<td>98.</td>
</tr>
<tr>
<td>4.0</td>
<td>85.</td>
</tr>
</tbody>
</table>

INPUT DATA

THIRD EXAMPLE OF NETWORK WHOSE PERFORMANCE IS SIMULATED THROUGH TIME (in 2 HOUR INTERVALS). (See: SPEC IF SIM = 1, FLOW = 1, NODES = 1, NPIPES = 2, SEND PIPES 1 1 2 2000 12 .0102 2 1 3/4 3 2 2000 10/ 4 2 4 2500 8/ 5 5 1000 10/ 6 6 6 1500 6/ 7 7 1000/ 8 8 6 2500/ 9 9 4 1500/ 10 10 8 500/ 11 11 1 1000 12/ 12 12 10 2 1000 10/ NODES 1 1 330. 2 1.5 3 3/4 4 4/5 5 5/6 6 6/7 7 7/8 8 8/9 9 9 4 10. 10 0 410. PUMPS 10 2.5 105. 3 98. 4 85. 410. METER 5 500. VALVE 7 2600 485. RUN

Storage capacity elevation curve for reservoir for example 12. (Data from this curve defines the three points (500, 3000), (400, 2400), and (300, 2000) given below STORAGE FUNCTION of the input data).

Demand function No. 1 which applies to nodes 1, 2, and 6 through 8. (Data from this curve provide the 3 values of time versus dimensionless demand as shown in the set of data immediately below DEMAND. The program fits a 2nd degree polynomial through each consecutive three values. The data are (2, .8), (6, 1.1), (12, 1.5), (.14, 1.), (24, .8).
EXAMPLE PROBLEM 13

The network of example 13 contains a back pressure valve, a pressure reducing valve and two nozzles. The reservoir at node 15 of the network receives a constant supply of water at a rate of 4.0 cfs. It is a circular tank with a diameter of 80 feet and its bottom and top are at elevations 115 and 135 respectively. The demand at node 2 is constant, and the demand functions for the other nodes are defined in the tables below the network. The network's performance is to be simulated over a 24 hour period of increments of 2 hours. The operation of the pumping station is shown by the time schedule data in the table below the network.

The input data for this network has been prepared below using the PIPE-NODES command. It is worth noting that a convenient method for handling the constant inflow into the reservoir is to add the additional node 17 very close to the reservoir and specify that it has a negative demand of 4 cfs. Note also that since four pumps were operating in parallel at time zero, the flow rate was multiplied by 4 from that of each pump in preparing the input data. This is an alternative to adding the command PARALLEL and giving data to indicate that the pump station at node 1 has four pumps operating.
Pump Characteristics (Q's in cfs, h_p in feet)

<table>
<thead>
<tr>
<th>no.</th>
<th>Q_1</th>
<th>h_p1</th>
<th>Q_2</th>
<th>h_p2</th>
<th>Q_3</th>
<th>h_p3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.75</td>
<td>115</td>
<td>1.5</td>
<td>100</td>
<td>2.25</td>
<td>80</td>
</tr>
<tr>
<td>2</td>
<td>.2</td>
<td>77</td>
<td>.5</td>
<td>72</td>
<td>.8</td>
<td>65</td>
</tr>
</tbody>
</table>

Demand functions for nodes 3,4,6 and 7

<table>
<thead>
<tr>
<th>Time (hrs)</th>
<th>0</th>
<th>2</th>
<th>6</th>
<th>8</th>
<th>12</th>
<th>15</th>
<th>24</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>1.2</td>
<td>1.1</td>
<td>1.0</td>
<td>0.9</td>
<td>0.8</td>
<td>0.9</td>
<td>1.0</td>
</tr>
<tr>
<td>Value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Demand functions for nodes 10,11,12,13 and 14

<table>
<thead>
<tr>
<th>Time (hrs)</th>
<th>0</th>
<th>3</th>
<th>6</th>
<th>12</th>
<th>20</th>
<th>24</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>1.2</td>
<td>0.9</td>
<td>0.8</td>
<td>0.6</td>
<td>0.7</td>
<td>1.0</td>
</tr>
<tr>
<td>Value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Pump operation schedule (station no. 1)

<table>
<thead>
<tr>
<th>Time (hrs)</th>
<th>0</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>no. on</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

EXAMPLE NO. 13

$SPECIF NODESP=1,NUMPIP=1,NPLENG=1,NPSERI=0,NPRINT=10,NFLOW=0,NPGPM=0,PEAKF=1.2,ISIML=1,NCONTI=1$END

PIPEC
10. 2. 1 2 .7 50.
8. 1.5 13 2
10. 2. 2 3 .4 0.
6. 8 3 4 .5 0.
4. 1.5 4 5 0. 0.
4. 1.5 3 6 .3 0.
6. .8 3 8 10.
6. .8 8 9 -5.
4. .6 8 16 0.
6. 2. 9 10 1.1 60.
4. 1.7 9.2 -12.
4. 1.7 6.
6. 1.1 11 10.
6. 1.2 12 10.
6. .8 12 11 .2 60.
6. .5 13 12 1. 60.
8. 1. 14 13 .8 60.
10. .79 18 14 .5 60.
10. .05 17 -3.3333 60. 18.
10. .01 15 18.
RESER
15 120.
PUMPS
1 3 115 6 100 9 80 -20.
BOOSTER
10. 2 77 .5 72 .8 65.
VALVE
3 800 40.
BPVAL
2 1000 85.
NOZZLE
5 1 .6 16./
16 1.1 .6 5./
RUN
$DATA HTIME=24,INCHR=2,ISUNIT=0,NPUNOD=0$END

DEMAND FUNCTION
1 2 .917 6 .833 8 .750 12 .667 15 .75 24 .833/3.4,6,7/2 3 .75 6 .667 12 .5 20 .593 24 .833/10-14/

STORAGE FUNCTION
1 115 0 135 100531./1/PUMP SCHEDULE
1 2 0 1 4 3 6 2 12 1/END SIMULATION -148-
EXAMPLE PROBLEM 14

Two variations of the small example network below are given.

(1) The network is analyzed as shown assuming all pipes have an equivalent sand roughness of 0.02 inches for use with the Darcy-Weisbach formula. The input immediately below the sketch is applicable for this problem. This input utilizes the CHANGE capability to request that USU-NETWK do a follow-on alternative analysis in which all equivalent sand roughnesses will be 0.00008 inches.

(2) The analysis is intended to answer the question: "How much additional demand can be discharged from a fire hydrant at node 35?" This question is answered by assuming the fire hydrant is attached to node 35 through a 10 ft long 6-inch diameter pipe with a 4 inch orifice at its end.
### Reduced Pressure Zone Created by Back Pressure Valve and Pressure Reducing Valve

- **Data:**
  - LPV from node 36 is determined by placing a nozzle with a 4-inch dia. here at the end of a 10 ft long pipe.
  - Two analyses are done to study the effect of smoother pipe.
  - Units of solution are:
    - Pumps: m³/s
    - Elevation: ft
    - Head: ft
    - Pressure: psi
    - Numbers of pseudo loops: 0
    - Numbers of real loops: 3

### Pipe Data

<table>
<thead>
<tr>
<th>Pipe</th>
<th>Nodes</th>
<th>Length</th>
<th>Diam</th>
<th>Coef</th>
<th>Flow Rate</th>
<th>Velocity</th>
<th>Head</th>
<th>Pumps</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>10</td>
<td>4.00</td>
<td>4.20</td>
<td>9.00</td>
<td>2.30</td>
<td>1.00</td>
<td>1000</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>20</td>
<td>4.00</td>
<td>4.20</td>
<td>9.00</td>
<td>2.30</td>
<td>1.00</td>
<td>1000</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>30</td>
<td>4.00</td>
<td>4.20</td>
<td>9.00</td>
<td>2.30</td>
<td>1.00</td>
<td>1000</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>40</td>
<td>4.00</td>
<td>4.20</td>
<td>9.00</td>
<td>2.30</td>
<td>1.00</td>
<td>1000</td>
</tr>
</tbody>
</table>

### Losses Due to Elevation Changes

- **Losses:**
  - 0.04 psi per 1000 ft elevation change
  - 0 psi for elevation change

### Elevation Correction Formula Used for Computing Head Losses

- **Formula:**
  - \( h = kh \times \left( \frac{v^2}{2g} \right) \)

### Summary of Analysis

- **Note:**
  - All previous coef s have been changed to .000005-04
  - Solution to the 1 variation of the basic network is given below:
  - Flow from pumps and reservoirs equals 4650.00
  - During iter. no. 3 Prys given 2 in pipe 100 psi shut-off flow.
  - Losses due to pe. avg. in all pipes
  - Power loss = 157.060 ft-lbf = 171.744 kwatts. Energy loss = 2821.163 kwhr/day
  - Elevation correction: downstream and upstream of pumps
  - Elevation correction formula used for computing head losses

---

*Note: The document contains detailed engineering analysis and calculations related to water or fluid dynamics, focusing on pressure zones and flow rates.*
SOLUTION TO THE VARIATION OF THE BASIC NETWORK IS GIVEN BELOW:

All previous coef's have been changed to 1.000000-04.

In order to reach device 2, 10 pipe 150 bas must be rebuilt.

During iter. No. 3 Prv number 1 in pipe 180 has shut off flow. Q = -9.55
During iter. No. 3 Prv number 2 in pipe 130 has shut off flow. Q = -1.13

Losses due flow in 4 370 fr. in 4 11.5 f. watts. Energy loss = 1.08.107 base/d.

Pipe: 6 head flow horsepower kilowatts horse/day
59 91.99 364.89 52.33 24.22 378.36

Elevation of fl. upstream and downstream of prvs:

Pipe: 6133.71 345.99

Cost: 365.773 gives horse 415.000 bpv trim.

Solution cost valid.

Moments and inertia from bpv
160 365.99 170.60

Pipe data:

Pipe: 60 length diam. cost fr. rate vel. perf. perf.
40 40 40 40 40 40 40 40 40 40 40
50 50 50 50 50 50 50 50 50 50 50
60 60 60 60 60 60 60 60 60 60 60

AV. vel = 5.12 AV. HR/1000 = 10.73 MAX vel = 10.73 MIN vel = 0.0

Node data:

Node (eas. (deg) elev head pressure perf.
4 1 100.00 100.00 44 100.00 44
5 1 100.00 100.00 44 100.00 44
6 1 100.00 100.00 44 100.00 44

AV. head = 100.14 AV. HSL = 328.59 MAX head = 137.32 MIN head = 65.21

Changes made to network:

Reduction in Q due to 4 nd prv.

775.337 to 415.000

Pipe: 6 head flow horsepower kilowatts horse/day
59 91.99 364.89 52.33 24.22 378.36

Elevation of fl. upstream and downstream of prvs:

Pipe: 6133.71 345.99

Cost: 365.773 gives horse 415.000 bpv trim.

Solution cost valid.

Moments and inertia from bpv
160 365.99 170.60

Pipe data:

Pipe: 60 length diam. cost fr. rate vel. perf. perf.
40 40 40 40 40 40 40 40 40 40 40
50 50 50 50 50 50 50 50 50 50 50
60 60 60 60 60 60 60 60 60 60 60

AV. vel = 5.12 AV. HR/1000 = 10.73 MAX vel = 10.73 MIN vel = 0.0

Node data:

Node (eas. (deg) elev head pressure perf.
4 1 100.00 100.00 44 100.00 44
5 1 100.00 100.00 44 100.00 44
6 1 100.00 100.00 44 100.00 44

AV. head = 100.14 AV. HSL = 328.59 MAX head = 137.32 MIN head = 65.21

Changes made to network:

Reduction in Q due to 4 nd prv.
It should be noticed that in solving the alternate problem 14 for which all of the pipe wall roughness coefficient are set to 0.00008 inches and for which the fire hydrant is not discharging, that a message occurs that reads:

THE HGL YOU GAVE FOR BPV 100 IS TOO LOW FOR DOWNS. PRES.
DOWNST. HGL= 563.773 GIVEN HGL= 470.000 BPV INOPER.

SOLUTION NOT VALID

This message indicates that the version of USU-NETWK that was used to solve this problem does not replace a BPV with a minor losses device, and then solve the newly formulated problem. It would have done this if a PRV would have had a pressure setting greater than at the upstream node rather than a BPV. However, to get a valid solution for this problem the burden is upon the user to change the input data and rerun the problem. The input data should be modified as shown below. The solution for this revised problem is also shown below. One should note that considerable difference occurs between the solutions with e = 0.02 and e = 0.00008 inches.
The supply for the network in Example No. 15 comes from large conduits that deliver water at a pressure of 50 psi at two points, nodes 18 and 16, plus a local storage tank at node 17. In this example instead of converting the two 50 psi pressure to HGL elevations, the program option that allows pressures for pumps and reservoirs is used. The network is to be solved over a 12 hour time period in 2 hour increments. The tables below give the needed data. Source pump #1 has its operation controlled by the water surface elevation of the reservoir at node 17. It is a variable speed pump whose operation can be duplicated by assuming up to 5 pumps are operating in parallel as shown by the data table below.

### Pump Characteristics

<table>
<thead>
<tr>
<th>Source Pump #1</th>
<th>Booster Pump</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q (gpm)</td>
<td>H (ft)</td>
</tr>
<tr>
<td>1000</td>
<td>70</td>
</tr>
<tr>
<td>2000</td>
<td>60</td>
</tr>
<tr>
<td>3000</td>
<td>40</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Q (gpm)</th>
<th>H (ft)</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>30</td>
</tr>
<tr>
<td>1000</td>
<td>25</td>
</tr>
<tr>
<td>2000</td>
<td>12</td>
</tr>
</tbody>
</table>

### Storage Capacity for Reservoir at Node 17

<table>
<thead>
<tr>
<th>W. S. elev. (ft)</th>
<th>Storage (ac. ft.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>280</td>
<td>.5</td>
</tr>
<tr>
<td>300</td>
<td>4</td>
</tr>
<tr>
<td>320</td>
<td>6</td>
</tr>
</tbody>
</table>

### Demand Functions

<table>
<thead>
<tr>
<th>At nodes 1-8</th>
<th>At nodes 9-15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time (hrs)</td>
<td>Value</td>
</tr>
<tr>
<td>4</td>
<td>1.5</td>
</tr>
<tr>
<td>6</td>
<td>2.0</td>
</tr>
<tr>
<td>8</td>
<td>1.4</td>
</tr>
<tr>
<td>10</td>
<td>1.8</td>
</tr>
<tr>
<td>12</td>
<td>1.4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>W. S. elev. at #17</th>
<th>No. of pumps on</th>
</tr>
</thead>
<tbody>
<tr>
<td>280</td>
<td>5</td>
</tr>
<tr>
<td>290</td>
<td>4</td>
</tr>
<tr>
<td>300</td>
<td>3</td>
</tr>
<tr>
<td>305</td>
<td>2</td>
</tr>
<tr>
<td>308</td>
<td>1</td>
</tr>
<tr>
<td>312</td>
<td>0</td>
</tr>
</tbody>
</table>

### Operation of Pump #1

<table>
<thead>
<tr>
<th>W. S. elev. at #17</th>
<th>No. of pumps on</th>
</tr>
</thead>
<tbody>
<tr>
<td>280</td>
<td>5</td>
</tr>
<tr>
<td>290</td>
<td>4</td>
</tr>
<tr>
<td>300</td>
<td>3</td>
</tr>
<tr>
<td>305</td>
<td>2</td>
</tr>
<tr>
<td>308</td>
<td>1</td>
</tr>
<tr>
<td>312</td>
<td>0</td>
</tr>
</tbody>
</table>

-153-
EXAMPLE OF NETWORK THAT SPECIFIES PRESSURES INSTEAD OF HGL FOR SOURCE PUMPS AND RESERVOIRS. NETWORK IS ALSO CHECKED BY DUPLICATE DATA.

&SPECIF NFLOW=1,NPGFM=1,NETCHK=1,NEPRES=1, 
NPRPUN=1,NODESP=1,NUMP1=1,NPLENG=1,ISIML=1, 
VELMAX=10.,VELMIN=.5,PRESMIN=40.,PRESMA=180., 
COEFRO=.00006,NPRES=1,INITAL=0,NSYM=0 &END 

PIPES
18 5 2 36/ 
5 1 2 12/ 
1 2 3/
2 3 .8 10 
4 3 .7/ 
5 4 3 10/ 
5 4 3 8/ 
5 4 3 16/ 
4 7 .7 10/ 
8 7 1.8 12/ 
6 8 3 8/ 
5 6 2 12/ 
8 9 2.4 10/ 
7 10 1.2 12/ 
10 9 .5 10/ 
10 12 .7/ 
11 9 1.2/ 
17 11 1.5 12/ 
11 12 1.5 10/ 
13 12 .7/ 
3 13 2.4 10/ 
14 13 1/ 
15 14 1.2/ 
15 12 2.5/ 
16 15 2 14/ 
RESER
18 50. 
17 43.3 
PUMPS
16 1000. 70. 2000 60 3000 40 50 
BOOSTER
14 500 30 1000 25 2000 12/ 
NODES
1 300 180 2 3/ 
2 500 190 3 4/ 
3 200 190 4 21 5/ 
4 600 185 5 6 7 8 9/ 
5 0 185 1 2 6 7 8 12/ 
6 400 180 12 11/ 
7 400 205 9 10 14/ 
8 200 200 11 10 13/ 
9 500 208 13 15 17/ 
10 100 210 14 16 15/ 
11 300 200 19 17 18/ 
12 500 200 16 19 20 24/ 
13 300 201 21 20 22/ 
14 300 195 22 23/ 
15 300 193 25 24 23/ 
16 0 195 25/ 
17 0 200 18/ 
18 0 185 1/ 
RUN 
&DATA NPUNOD=1,LINVAR=1,INCHR=2,NLTIME=12,ALTV=0 &END 
DEMAND FUNCTION 
1 4 1.5 6 2. 8 1.4 10 .8 12 .4/ 
1-8/ 
2 2 3.1 4 1.6 6 1.9 8 1.3 10 .9 12 .5/ 
9-15/ 
STORAGE FUNCTION 
1 280 .5 300 4. 320 6./ 
17/ 
PUMP RULES 
14 2 17 3 280 5 290 4 300 3 305 2 308 1 312 0/ 
END SIMULATION
EXAMPLE PROBLEM 15A

In example No. 15 assume a much smaller tank exists that can be supplied by an auxiliary supply. The rate at which the auxiliary supply fills the tank is controlled by the tank elevation. The storage capacity of this smaller tank is given in a table below, and the rule that governs the filling of the tank is shown in the other table below. To include this auxiliary tank supply in the analysis the following can be done: (1) Add an additional node No. 19 in pipe 18 and call the pipe between this node and the reservoir pipe No. 26. (2) Utilize the FLOW RULE command to control the negative demand at the near node 19 according to the manner in which the tank is filled by the auxiliary supply. The input data utilizing the PIPE-NODES command for this revised Problem 11A is shown below.

### Storage Capacity for Reser. at Node 17

<table>
<thead>
<tr>
<th>W. S. elev (ft)</th>
<th>Storage Millions gal.</th>
</tr>
</thead>
<tbody>
<tr>
<td>280</td>
<td>1.0</td>
</tr>
<tr>
<td>300</td>
<td>1.5</td>
</tr>
<tr>
<td>320</td>
<td>2.0</td>
</tr>
</tbody>
</table>

### Auxiliary Tank Supply

<table>
<thead>
<tr>
<th>W. S. elev (ft)</th>
<th>Q (gpm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>280</td>
<td>1500</td>
</tr>
<tr>
<td>290</td>
<td>800</td>
</tr>
<tr>
<td>300</td>
<td>0</td>
</tr>
<tr>
<td>310</td>
<td>10</td>
</tr>
</tbody>
</table>

**EXAMPLE OF NETWORK THAT SPECIFIES PRESSURES INSTEAD OF HGL FOR SOURCE PUMPS AND RESERVOIRS. NETWORK IS ALSO CHECKED BY DUPLICATE DATA.**

```plaintext
*ASPECIF NFLOW=1,NPCPM=1,NPRRES=1, NPRPUMP=1,NODES=1,NUMPIE=1,ISIML=1, VELMAX=10, VELMIN=.5, PRESM=40., PRESMA=180., COEP=0.0008, NPRRES=1,INITIAL=0,NSYMMT=0 &END
PIPE-
36. 2. 18.185. 5 185.
12. 2. 5 1 300. 180.
12. 3. 1 2 300. 190.
10. .8 2 3 300. 190.
10. .7 4 3
10. 3. 5 4 600. 185.
8. 3. 5 4
16. 3. 5 4
10. .7 4 7 400. 205.
12. 1.8 8 7
8. 3. 6 8 200. 200.
12. 2. 5 6 400. 180.
10. 2.6 8 9 500. 208.
10. 1.2 7 10 100. 210.
10. 1.5 11 12
10. .7 13 12
10. 1.2 11 9
12. 1.5 19 11 300. 200.
10. 1.5 11 12
10. .7 13 12
10. 2.4 3 13 300. 200.
10. 1. 14 13
10. 1.2 15 14 300. 195.
10. 2.5 15 12
14. 2. 16 195. 15 300.193.
12. .001 17 200. 19
RESER
18 50.
17 43.3
PUMPS
16 1000. 70. 2000 60 3000 40 50
BOOSTER
14 500 30 1000 25 2000 12/
RUN
&TDATA NPUNOD=1,LINCR=1,INCHR=2,HTIME=12,ALTV=0 &END
DEMAND FUNCTION
1 4 1.5 6 2. 8 1.4 10 .8 12 .4/
1-8/
2 2 1.1 4 1.6 6 1.9 8 1.3 10 .9 12 .5/
9-15/
STORAGE FUNCTION
1 280 1.0 300 1.5 320 2.0/
17/
PUMP RULES
14 2 17 3 280 5 290 4 300 3 305 2 308 1 312 0/
FLOW RULE
-19 -17 0. 300. -800. 290. -1500. 280./
END SIMULATION
```
EXAMPLE PROBLEM 16

This problem illustrates the special input that can be used for a network that consists of a tree like structure. The problem consists of a large well field collection piping system. The wells are spaced at one-half mile intervals along three laterals. The flow from the wells is discharged at the final collection point at a needed hydraulic grade line elevation of 380 feet, and other dimensions of the collection system are as shown on the sketch below. The problem calls for: 1. Determining the size of each pipe in the system so that the gradient of the HGL is .002 (2 feet per 1000 feet of pipe length), and to follow this design solution with an analysis of the system that consists of the nominal pipe sizes closest to those computed. An economic analysis of the system is to be done based on: (a) an interest rate of 12 percent for all components, (b) a life of 35 years for all components, (c) the default costs of pipe per foot, and (d) electrical costs of $0.05 per kilowatt hour plus an initial capital investment of $800,000 to get this electrical energy. The combined efficiency of the pumps and electrical motors is 80 percent. Note in the input that after the RUN command, that a line gives the node and an HGL elevation. This is needed since for the analysis solution no source of supply exist, and therefore this is a special network whose input data must specify a node where the HGL is known.

```
LARGE WELL FIELD COLLECTION PIPING SYSTEM
/*
$SPECIF IHGL=-2,DESIGN=1,
NOMSOL=1,ICOST=1 $END
380.302.240. -2 .005
1 52 .002 5280.10560.2640./
2 102 .002 2640./
2 152 .002 5280.2640./
END
RUN
1 380.
INTEREST=.12
LIFE=35
EFFIC
.8
PUMP
CAPI=800000.
UNIT=.05
END
```
EXAMPLE PROBLEM 17

This problem also uses the special input for a tree shaped network as does the previous problem. In this example a large irrigation system is involved. At each of 120 discharge points a flowrate of 1500 gpm is needed. These discharge points are along 4 laterals at a spacing of 1000 feet. The other features of the design are shown on the sketch. The problem specifications is based on the assumption that the HGL will have the specified gradient. The solution gives information about how each discharge point should be designed in order to have the correct flowrate leave from that point, since the head at each node is given. Also the head that the pump must supply is given, as well as all pipe diameters. In the economic analysis it should be noted that the interest rate associated with the electrical energy for the pump is 12 percent, whereas the interest rate associated with the costs of pipe is only 8 percent.

IRRIGATION SYSTEM
/*
$SPECIF IHGL=-2,DESIGN=1,NFLOW=1,
NPGPM=1,NOMSOL=1,ICOST=1 $END
500 -180000 310 1500 .008
1 31 -.0005 1500 1000./
2 61 -.0005 3000 1000./
32 91 -.0005 3000 1000./
62 121 -.0005 3000 1000./
END
RUN
1 500
INTEREST=.12
LIFE=80
PUMP
UNIT=.1
INTEREST=.08
END

-157-
EXAMPLE PROBLEM 18

In Problem 17, the supply at node 1 is actually obtained from 3 pumps and the piping configuration shown in the sketch below. In this problem also increase the slope of the HGL from .0005 to .0015. The input data now can be as given below; or if one desires the data for the 120 pipe system of problem 17 could be entered in the usual manner with the PIPES and NODES commands, or the PIPE- command. Note the following changes to the input of Problem 17 have been made: 1. the added portion of the network has been included by the usual demands, 2. the line after the RUN command that gave a starting HGL has been removed, 3. The demand at node 1 has been changed to zero, and 4. to satisfy the different HGL specification the -.0005's have been changed to -.0015's.

IRRIGATION SYSTEM
/*
 * $SPECIF IHGL=-2,DESIGN=1,NFLOW=1,
 *  NPGPM=1,NOMSOL=1,ICOST=1 $END
500 0 310 1500 .008
1 31 -.0015 1500. 1000./
2 61 -.0015 3000 1000./
32 91 -.0015 3000 1000./
62 121 -.0015 3000 1000./
END
PIPES
122 0 1 500 60 .008
123 0 1/
124 122 1 2000 0/
125 0 122 1000/
NODES
122 0 320. 503
PUMPS
122 40000 130 60000 105 90000 85 400
123 40000 130 60000 105 90000 85 390
125 40000 160 60000 135 90000 115 370
RUN
INTEREST=.12
LIFE=80
PUMP
UNIT=.1
INTEREST=.08
END
-158-
EXAMPLE PROBLEM 19

Example Problem 19. This example illustrates the use of level and pressure control algorithms.

CONTAINS LEVEL AND PRESSURE CONTROLLERS

```
*/
SPECIFIC NPLCM=1,NPGM=1,NODESF=1,ISIML=1,NTRAND=1 SEND
PIPPES

```

## EXAMPLE PROBLEM 19

<table>
<thead>
<tr>
<th>NODES</th>
<th>10 1.78 8.8</th>
<th>2 0.78 9.6</th>
<th>0 140.7</th>
<th>5 0.39 9.9</th>
<th>6 1 140.9</th>
<th>7 1368 36</th>
<th>8 0 18.2</th>
<th>9 0 27.6</th>
<th>10 0 29.6</th>
<th>11 0 29.9</th>
<th>12 100000 275.75</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONTAINS</td>
<td>19 15 11</td>
<td>14 0 140.5</td>
<td>15 0 140.5</td>
<td>16 0 28.4</td>
<td>17 0 1355</td>
<td>18 0 74.08</td>
<td>19 0 140.5</td>
<td>20 0 140.5</td>
<td>21 0 140.5</td>
<td>22 0 140.5</td>
<td>23 0 140.5</td>
</tr>
<tr>
<td>PROBLEMS</td>
<td>1 52</td>
<td>2 3 0.3075</td>
<td>4 5 0.3075</td>
<td>6 7 0.3075</td>
<td>8 9 0.3075</td>
<td>10 11 0.3075</td>
<td>12 13 0.3075</td>
<td>14 15 0.3075</td>
<td>16 17 0.3075</td>
<td>18 19 0.3075</td>
<td>20 21 0.3075</td>
</tr>
<tr>
<td>NODE</td>
<td>10 15 1.78 8.8</td>
<td>2 0.78 9.6</td>
<td>0 140.7</td>
<td>5 0.39 9.9</td>
<td>6 1 140.9</td>
<td>7 1368 36</td>
<td>8 0 18.2</td>
<td>9 0 27.6</td>
<td>10 0 29.6</td>
<td>11 0 29.9</td>
<td>12 100000 275.75</td>
</tr>
<tr>
<td>PROBLEM</td>
<td>1 23 1 2</td>
<td>3 0.3075</td>
<td>4 5 0.3075</td>
<td>6 7 0.3075</td>
<td>8 9 0.3075</td>
<td>10 11 0.3075</td>
<td>12 13 0.3075</td>
<td>14 15 0.3075</td>
<td>16 17 0.3075</td>
<td>18 19 0.3075</td>
<td>20 21 0.3075</td>
</tr>
</tbody>
</table>

-159-
EXAMPLE PROBLEM 20

This example problem illustrates the use of valves being controlled by the commands SMINOR, RMINOR, SVALVE and RVALVE. There are several variations of the input data that are shown below that use these command. The valve in pipe 11 can be closed to force more of the flow from the pump into pipe 2 and thus into the reservoir more directly so that the reservoir attached to the network by pipe 4 will not supply as much flow or will fill faster during periods of lower demands. The valve in pipe 4 can be closed also if less flow is to come from the reservoir at the end of this pipe.

In studying the various alternative input data files that are given associated with this example, you should note that in the first file, the minus values for time after the command SMINOR indicate that USU-NETWK should pause at these times and give a prompt that the user has selected to give the minor loss coefficient for the next solution(s). Whenever such pauses occur it is possible to request USU-NETWK to provide the flowrate(s) that currently exist in any pipe(s), or the pressure(s) that exists at any node(s), or the water surface elevation(s) in any reservoir(s) to help in making the decision regarding the loss coefficient that should be given for the next time increment solution. The same type of interaction in supplying values for the next time solution(s) is possible for: (1) demands at nodes, (2) pump characteristics, or turning pumps on or off, and (3) specifying flowrates in selected pipes. In all cases it is possible to ask to have portions of the solution displayed on the screen to assist in making decisions regarding the values to supply.

In the last input data file given below for this example network the flowrate has been specified as a function of time with the command SQGIVE rather than controlling the flow with a valve. The advantage of specifying flowrates in pipes is that the solution provided values for head loss, or head increase which are needed to achieve the specified flowrates.
CONTROL OF PUMP FLOW BY VALVE
/* $SPECIF
NODES=0, NPSERI=0, ISIML=1 $END
PIPS
1 1 2000 10 .005
2 1 1800/
3 3 9000/
4 3 0 1000 8/
5 8 4 1300/
6 5 2000/
7 6 6/
8 9 1000/
9 8 9 700/
10 8 9 700/
11 8 3000 10/
12 8 11 2000 8/
13 11 12 3000 6/
14 13 16/
15 11 7/
16 12 9 2000 8/
17 12 9 16 2300/
19 9 10 2000/
20 11 12 2000 6/
21 13 14/
22 16 15/
23 10 12/
24 12 14/
25 14 15/
NODES
1 .5 3080
2 8 3080
3 0 3150
4 .3 3100
5 .3 3100
6 .5 3120
7 .3 3140
8 .2 3060
9 .4 3070
10 .3 3050
11 .4 3090
12 .35 3055
13 .4 3100
14 .3 3070
15 .3 3070
16 .5 3130
RESER
4 3200
18 3180
PUMPS
1 1 310 2.25 270 4 190 3000 PARALLEL
1 2/ MINOR
11 10 4 6
RUN $STDATA HTIME=24, INCHR=2, PRINTT=3, ISUNIT=0, LINEAR=1 $END
DEMAND FUNCTION
1 2 1 2 6 2 8 9 10 .6 15 .3 20 .8 24 1. / 1-16/
PUMP SCHEDULE
1 2 0 2 2 3 8 2 12 1 18 2/ SIMNOR
11 -2 10 -4 10 -6 10 -8 10 -10 10 -12 10 -16 10 -20 10/ RMINOR
4 1 3100 2 3200 10 3250 50 3300 2000 /
STORAGE FUNCTION
2 3120 0 3310 134050/
18/ 1 3100 0 3302 294910/
4/ END SIMULATION
In this variation of input the command PIPE:: replaces the commands PIPES & NODES.
CONTROL OF PUMP FLOW BY VALVE/* $SPECIF
NODES=1, LENGON=0, ISIML=1 $END
PIPS
1 19 1 10 .005
2 12/
In this variation the coordinates are given following the command PIPE.

CONTROL OF PUMP FLOW BY VALVE

$SPECIF NPSERI=0, NODESP=0, ISIML=1
END PIPE

PUMPS
1 310 2.25 270 4 190 3000
PARALLEL
12/ MINOR
11
4 6
RUN
$DATA HTIME=24, INCHR=2, NPUNOD=0, ISUNIT=0, LINEAR=1
SEND
DEMAND FUNCTION
-1 12 4 1.4 6 1.2 8 .9 10 .6 15 .3 20 .8 24 1./
1-16/
PUMP SCHEDULE
2 0 2 2 3 8 2 12 1 18 2/
VALVE
11 4800 .112 0 0 1 3100 0 3200 55 3300 90.0
STORAGE FUNCTION
2 3120 0 3310 134050/
1/
3100 0 3302 294910/
END SIMULATION

In this variation the flow rate in pipe 11 is specified as a function of time, rather than specifying losses with a valve.

CONTROL OF PUMP FLOW BY SPECIFYING FLOW IN ONE LINE OF DOWNSTREAM BRANCH

$SPECIF NPSERI=0, NODESP=0, ISIML=1
END PIPE

PUMPS
1 310 2.25 270 4 190 3000
PARALLEL
12/ MINOR
11
4 6
RUN
$DATA HTIME=24, INCHR=2, NPUNOD=0, ISUNIT=0, LINEAR=1
SEND
DEMAND FUNCTION
-1 12 4 1.4 6 1.2 8 .9 10 .6 15 .3 20 .8 24 1./
1-16/
PUMP SCHEDULE
2 0 2 2 3 8 2 12 1 18 2/
VALVE
11 4800 .112 0 0 1 3100 0 3200 55 3300 90.0
STORAGE FUNCTION
2 3120 0 3310 134050/
1/
3100 0 3302 294910/
END SIMULATION
The special tables requested from the last alternative input data file are provided below. Since this solution specified flowrates in pipe 11 either a head loss or positive head may be required. The table below summarizes these values. You might want to run this problem and verify these results. Notice a booster pump is needed in pipe 11 for most of the time. The head from the booster pump are needed in part of off set the high friction losses in pipe 11, which are also given in this table.

### Table giving differential heads in pipe 11 to achieve specified flowrates

<table>
<thead>
<tr>
<th>Time (HOURS)</th>
<th>0</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>12</th>
<th>14</th>
<th>16</th>
<th>18</th>
<th>20</th>
<th>22</th>
<th>24</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P head</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spec. Fric. L.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Special Table giving pressures at designated modes as a function of time

<table>
<thead>
<tr>
<th>TIME (HOURS)</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>NODE NO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Special Table giving flowrates in designated pipes as a function of time

<table>
<thead>
<tr>
<th>TIME (HOURS)</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIPE NO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Special Table giving water surface elev. in reservoirs

<table>
<thead>
<tr>
<th>TIME (HOURS)</th>
<th>4</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>DESIGNATION</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
EXAMPLE PROBLEM 21

The analysis illustrated in this example represents a time dependent solution to determine how well the four reservoir shown might supply the demands over a 24 hour period if no other sources of make-up water existed during this period of time. Valves exist in a couple of pipes to control the flows. The first input data uses the commands SVALVE and RVALVE for schedules and rules, respectively, that determine the openings of the valves. In the second input data the commands SVALVC and RVALVC are used in which the given data control the incremental change in the valves openings in which one of the parameters in the equation is changed as a function of time and as a rule governed by a reservoir water surface elevation, respectively.

FLOW CONTROL BY VALVES
/*
  $SPECIP NPRINT=-1,NODESP=0,ISIML=1,NFLOW=1
  NGPM=1,COEFRO=.0005 SEND
PIPE-
  1 14. 800. 1 500. 450.
  2 14. 800. 2 800. 450.
  3 14. 800. 3 1200. 430.
  7 14. 1500. 1 2
  8 14. 1800. 2 3
  9 12. 1300. 4 5
  10 12. 1300. 5 6
  11 12. 2000. 4 7 300. 370.
  12 12. 2000. 5 8 300. 370.
  13 12. 3000. 6 11 1200. 350.
  14 12. 1300. 7 8
  15 12. 1400. 9
  16 12. 1000. 8 10 800. 360.
  17 8. 1500. 10 9 600. 350.
  18 12. 1800. 10 11 1200. 350.
  19 12. 1500. 11 12 500. 340.

-164-
Note: the lines after SVALVE, RVALVE, SVALVC & RVALVC should be on a single line. These lines have been split here to show the input files in two columns across a page.
EXAMPLE PROBLEM 22

A new water distribution system is to be designed that will receive its supply from a well. The points of demand are shown on the sketch below, and it is estimated that the demand will equal 0.5 cfs at each of these 15 points. It is proposed that a storage tank with 3.6 million gallon capacity be placed at node 17 which is 1000 feet away from node 16. Determine the following: (a) the locations and pipe sizes throughout the network, (b) the water surface elevations in the tank so the tank can be properly elevated, and (c) the pump to use.

The design process will begin by assuming a branched system such as shown in the sketch below.

For this network the special input allowed with the option IHGL negative can be used, and is:

START DESIGN WITH BRANCHED SYSTEM
/*
$SPECIF IHGL=-2,DESIGN=1,NETPLT=10,NOMSOL=1 $END
650 -8.5 500 .5 .004
ELEV
480
1 5 .002 3000 2100 1500 1800/
ELEV
475
2 7 .002 1500 1800/
ELEV
470
2 10 .002 2100 1500 1800/
ELEV
460
8 13 .002 2100 1500 1800/
ELEV
450
11 16 .002 2100 1500 1800/
DEMAND
1.
16 17 .002 1000/
END
RUN
1 650

-166-
Note the option NETPLT-10 in this input tells USU-NETWK that a file should be written with the description of this network that can be used as input for subsequent analyses after it has been edited. It preparing the above input the decision was that a slope of HGL equal to about 0.002 is desirable. Furthermore, it is decided that under the average demands used in this analysis that the pump should fill the reservoir at a rate of 1 cfs; therefore the command DEMAND with a 1 following it is given before the pipe to node 17 is given. The commands ELEV in the above input file give the new ground elevations for the last node in the the series of nodes defined by the next line. USU-NETWK will determine the elevations of the intermediate nodes by linearly interpolating these elevations. The solution, which is not given here gives the pipe sizes that can be used. This solution also indicates the water surface elevation of 620.77' is desirable under average demand; so the bottom and top of the tank are placed at 615' and 625' respectively.

The file written by USU-NETWK has been edited by adding minimum diameter pipes of 6 inches to give the looped system shown below.

This input data file, which includes the option to do a time-dependent analysis is as follows. (In this file you can see the additional 6 inch pipes that have been added, but these lines of added data have been placed in the appropriate columns so the option NETPLT=13 can be used to read a formatted input data file. Also the time-dependent data has been added.)

USE OF DHEAD DURING SIMULATION TO DETERMINE HOW PUMPS SHOULD OPERATE

/*
$SPECIF NETPLT=13,ISIML=1,NODESP=1 $END
PIPEC

<table>
<thead>
<tr>
<th>Pipe</th>
<th>Node 1</th>
<th>Node 2</th>
<th>Length</th>
<th>Diameter</th>
<th>Slope</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3000.0</td>
<td>20.00</td>
<td>.00400</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>2100.0</td>
<td>10.00</td>
<td>.00400</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>4</td>
<td>1500.0</td>
<td>10.00</td>
<td>.00400</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>5</td>
<td>1800.0</td>
<td>6.00</td>
<td>.00400</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>6</td>
<td>1500.0</td>
<td>10.00</td>
<td>.00400</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>7</td>
<td>1800.0</td>
<td>6.00</td>
<td>.00400</td>
</tr>
</tbody>
</table>
7 2 8 2100.0 18.00 .00400
8 8 9 1500.0 10.00 .00400
9 9 10 1800.0 6.00 .00400
10 8 11 2100.0 15.00 .00400
11 11 12 1500.0 10.00 .00400
12 12 13 1800.0 6.00 .00400
13 11 14 2100.0 12.00 .00400
14 14 15 1500.0 12.00 .00400
15 15 16 1800.0 10.00 .00400
16 16 17 1000.0 10.00 .00400
17 4 6 2100.0 6.00 .004
18 6 9 2100.0 6.00 .004
19 9 12 2100.0 6.00 .004
20 12 15 2100.0 6.00 .004
21 5 7 2100.0 6.00 .004
22 7 10 2100.0 6.00 .004
23 10 13 2100.0 6.00 .004
24 13 16 2100.0 6.00 .004

RESER
    1 520.
    17 620.77
DHEAD
    1 130 16 1 622
RUN
$DATA DTIME=1,INCHR=1,ISUNIT=2,NPUNOD=1,LINR=1,PRINTT=2 $END
PIPE TABLE
    1 2 15 16 24/
END TABLES
STORAGE FUNCTION
    1.615 0 620 1.8 625 3.6/
    17/
DEMAND FUNCTIONS
    1 0 1.4 1.5 8 2.2 12 1.4 16 1.0 20 .5 24 1.0/
    2-10/
    1 0 1.6 1.5 12 1.1 16 .7 20 1.2 4 1.0/
    11-16/

-168-
In preparing the time-dependent input data a variation of the HGL elevation at node 16 was decided upon as a function of time, and is entered as data after the command HGLSET. The solution will indicate what differential head is needed in pipe 1, i.e. the pump for which the DHEAD command is used in the network definition above. This variation of the HGL at node 16 was based on the fact that as the demands increase this elevation should decreased thus causing flow to come from the reservoir rather than filling it.

The special table of output requested with PRINTT=2 is given below. Also from the individual solutions the differential heads have been recorded in the last column of this table to show what heads the pump should produce as a function of time. (You should duplicate this run studying the solution, and decide how you can improve upon the given design.)

<table>
<thead>
<tr>
<th>TIME (HOURS)</th>
<th>PIPE NO.</th>
<th>1</th>
<th>2</th>
<th>15</th>
<th>16</th>
<th>24</th>
<th>Differential Head (ft) pump in pipe 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>8.50</td>
<td>1.32</td>
<td>1.37</td>
<td>-1.00</td>
<td>0.13</td>
<td></td>
<td>125.07</td>
</tr>
<tr>
<td>1.00</td>
<td>8.64</td>
<td>1.41</td>
<td>1.02</td>
<td>-0.32</td>
<td>-0.16</td>
<td></td>
<td>120.94</td>
</tr>
<tr>
<td>2.00</td>
<td>8.30</td>
<td>1.51</td>
<td>0.14</td>
<td>0.83</td>
<td>-0.38</td>
<td></td>
<td>112.90</td>
</tr>
<tr>
<td>3.00</td>
<td>8.71</td>
<td>1.65</td>
<td>-0.17</td>
<td>1.23</td>
<td>-0.43</td>
<td></td>
<td>112.01</td>
</tr>
<tr>
<td>4.00</td>
<td>9.23</td>
<td>1.79</td>
<td>-0.37</td>
<td>1.52</td>
<td>-0.49</td>
<td></td>
<td>111.56</td>
</tr>
<tr>
<td>5.00</td>
<td>9.89</td>
<td>1.98</td>
<td>-0.63</td>
<td>1.89</td>
<td>-0.55</td>
<td></td>
<td>110.48</td>
</tr>
<tr>
<td>6.00</td>
<td>10.62</td>
<td>2.18</td>
<td>-0.84</td>
<td>2.20</td>
<td>-0.62</td>
<td></td>
<td>109.88</td>
</tr>
<tr>
<td>7.00</td>
<td>10.94</td>
<td>2.36</td>
<td>-1.09</td>
<td>2.47</td>
<td>-0.66</td>
<td></td>
<td>107.04</td>
</tr>
<tr>
<td>8.00</td>
<td>11.29</td>
<td>2.55</td>
<td>-1.32</td>
<td>2.71</td>
<td>-0.70</td>
<td></td>
<td>104.24</td>
</tr>
<tr>
<td>9.00</td>
<td>10.32</td>
<td>2.32</td>
<td>-1.28</td>
<td>2.58</td>
<td>-0.66</td>
<td></td>
<td>102.47</td>
</tr>
<tr>
<td>10.00</td>
<td>9.35</td>
<td>2.09</td>
<td>-1.23</td>
<td>2.45</td>
<td>-0.61</td>
<td></td>
<td>100.86</td>
</tr>
<tr>
<td>11.00</td>
<td>8.38</td>
<td>1.86</td>
<td>-1.18</td>
<td>2.32</td>
<td>-0.56</td>
<td></td>
<td>99.63</td>
</tr>
<tr>
<td>12.00</td>
<td>7.42</td>
<td>1.63</td>
<td>-1.12</td>
<td>2.18</td>
<td>-0.51</td>
<td></td>
<td>98.56</td>
</tr>
<tr>
<td>13.00</td>
<td>7.06</td>
<td>1.52</td>
<td>-0.84</td>
<td>1.79</td>
<td>-0.45</td>
<td></td>
<td>100.86</td>
</tr>
<tr>
<td>14.00</td>
<td>6.78</td>
<td>1.41</td>
<td>-0.48</td>
<td>1.32</td>
<td>-0.38</td>
<td></td>
<td>103.39</td>
</tr>
<tr>
<td>15.00</td>
<td>6.76</td>
<td>1.31</td>
<td>0.12</td>
<td>0.59</td>
<td>-0.31</td>
<td></td>
<td>106.61</td>
</tr>
<tr>
<td>16.00</td>
<td>7.53</td>
<td>1.26</td>
<td>1.22</td>
<td>-0.93</td>
<td>0.06</td>
<td></td>
<td>117.14</td>
</tr>
<tr>
<td>17.00</td>
<td>7.56</td>
<td>1.16</td>
<td>1.46</td>
<td>-1.30</td>
<td>0.23</td>
<td></td>
<td>121.54</td>
</tr>
<tr>
<td>18.00</td>
<td>7.51</td>
<td>1.04</td>
<td>1.68</td>
<td>-1.58</td>
<td>0.33</td>
<td></td>
<td>126.05</td>
</tr>
<tr>
<td>19.00</td>
<td>7.41</td>
<td>0.92</td>
<td>1.88</td>
<td>-1.82</td>
<td>0.41</td>
<td></td>
<td>130.54</td>
</tr>
<tr>
<td>20.00</td>
<td>7.28</td>
<td>0.78</td>
<td>2.06</td>
<td>-2.03</td>
<td>0.47</td>
<td></td>
<td>135.04</td>
</tr>
<tr>
<td>21.00</td>
<td>7.68</td>
<td>0.93</td>
<td>1.95</td>
<td>-1.87</td>
<td>0.42</td>
<td></td>
<td>133.37</td>
</tr>
<tr>
<td>22.00</td>
<td>8.08</td>
<td>1.07</td>
<td>1.84</td>
<td>-1.71</td>
<td>0.37</td>
<td></td>
<td>131.99</td>
</tr>
<tr>
<td>23.00</td>
<td>8.47</td>
<td>1.20</td>
<td>1.72</td>
<td>-1.53</td>
<td>0.31</td>
<td></td>
<td>130.51</td>
</tr>
<tr>
<td>24.00</td>
<td>8.84</td>
<td>1.34</td>
<td>1.61</td>
<td>-1.34</td>
<td>0.23</td>
<td></td>
<td>129.04</td>
</tr>
</tbody>
</table>

A curve fit of the data giving the flowrate in pipe 1 as the absissa, and the pump head provides a desired pump curve for this network.
EXAMPLE PROBLEM 23

For this example network assume that an unknown discharge is occurring from node 6, and that the pressure is being periodically measured at node 4. The performance of the network is to be simulated over a 24 hours period of time in 1 hour increments. The pressure at node 4 produces a HGL elevation of 92.31 feet at the beginning of the period, and then decreased to an HGL elevation of 60.2 feet at 12 hours, and then increases again to 92.0 feet at 24 hours.

Two alternatives of input data files are given below the sketch of the network. The second uses x y coordinates rather than pipe lengths, and special tables are requested, and these are given on the following pages.

```
Determination of flow from
node 6 from pressure at
node 4

/*
$SPECIF NFLOW-O.ISIML-1
NODESP-O $END

Pipes
4 3 500 10 .001/
5 3 4 1200 8/
6 2 5/
7 1 6/
8 6 5 1000/
1 0 1 500 10 .001
2 1 2 1000 8/
3 2 1200/
9 5 4 1200/

Nodes
1 .3
2 .5/
3 .4/
4 .5/
5 .5/
6 .5/

Reser
1 100
4 95

SETHCL
4 6 1 92.31

RUN
$DATA HTIME=24, INCHR-1,
PRINT=3, DPUMP=0 $END

Pipe Table
All
Node Table
All
Reser. Table
All
End Tables
Demand Function
1 1 1.1 4 1.3 8 1.4 12 1.25 15 .8 20 .6 24 1./
1.5 /

Storage Function
1 80 1 100 2500 120 5000/
1/
Settings
4 0 92.31 3 70.4 6 12 40.2 16 60.3 24 85.0 /

End Simulation
```
### SPECIAL TABLE GIVING PRESSURES AT DESIGNATED NODES AS A FUNCTION OF TIME

<table>
<thead>
<tr>
<th>TIME (HOURS)</th>
<th>NODE NO</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>42.15</td>
<td>40.79</td>
<td>40.80</td>
<td>36.84</td>
<td>36.39</td>
<td>34.24</td>
<td></td>
</tr>
<tr>
<td>1.00</td>
<td>40.87</td>
<td>39.12</td>
<td>40.11</td>
<td>36.84</td>
<td>36.39</td>
<td>34.24</td>
<td></td>
</tr>
<tr>
<td>2.00</td>
<td>37.58</td>
<td>37.58</td>
<td>36.56</td>
<td>33.39</td>
<td>32.84</td>
<td>30.79</td>
<td></td>
</tr>
<tr>
<td>3.00</td>
<td>37.86</td>
<td>37.17</td>
<td>37.46</td>
<td>33.02</td>
<td>32.69</td>
<td>30.72</td>
<td></td>
</tr>
<tr>
<td>4.00</td>
<td>37.86</td>
<td>37.17</td>
<td>37.46</td>
<td>33.02</td>
<td>32.69</td>
<td>30.72</td>
<td></td>
</tr>
<tr>
<td>5.00</td>
<td>37.86</td>
<td>37.17</td>
<td>37.46</td>
<td>33.02</td>
<td>32.69</td>
<td>30.72</td>
<td></td>
</tr>
<tr>
<td>6.00</td>
<td>37.86</td>
<td>37.17</td>
<td>37.46</td>
<td>33.02</td>
<td>32.69</td>
<td>30.72</td>
<td></td>
</tr>
<tr>
<td>7.00</td>
<td>37.86</td>
<td>37.17</td>
<td>37.46</td>
<td>33.02</td>
<td>32.69</td>
<td>30.72</td>
<td></td>
</tr>
<tr>
<td>8.00</td>
<td>37.86</td>
<td>37.17</td>
<td>37.46</td>
<td>33.02</td>
<td>32.69</td>
<td>30.72</td>
<td></td>
</tr>
<tr>
<td>9.00</td>
<td>37.86</td>
<td>37.17</td>
<td>37.46</td>
<td>33.02</td>
<td>32.69</td>
<td>30.72</td>
<td></td>
</tr>
<tr>
<td>10.00</td>
<td>37.86</td>
<td>37.17</td>
<td>37.46</td>
<td>33.02</td>
<td>32.69</td>
<td>30.72</td>
<td></td>
</tr>
</tbody>
</table>

### SPECIAL TABLE GIVING DEMANDS AT DESIGNATED NODES AS A FUNCTION OF TIME

<table>
<thead>
<tr>
<th>TIME (HOURS)</th>
<th>NODE NO</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>0.38</td>
<td>0.38</td>
<td>0.50</td>
<td>0.50</td>
<td>0.28</td>
<td>0.28</td>
<td></td>
</tr>
<tr>
<td>1.00</td>
<td>0.33</td>
<td>0.33</td>
<td>0.55</td>
<td>0.55</td>
<td>0.27</td>
<td>0.27</td>
<td></td>
</tr>
<tr>
<td>2.00</td>
<td>0.37</td>
<td>0.37</td>
<td>0.55</td>
<td>0.55</td>
<td>0.27</td>
<td>0.27</td>
<td></td>
</tr>
<tr>
<td>3.00</td>
<td>0.39</td>
<td>0.39</td>
<td>0.55</td>
<td>0.55</td>
<td>0.27</td>
<td>0.27</td>
<td></td>
</tr>
<tr>
<td>4.00</td>
<td>0.41</td>
<td>0.41</td>
<td>0.55</td>
<td>0.55</td>
<td>0.27</td>
<td>0.27</td>
<td></td>
</tr>
<tr>
<td>5.00</td>
<td>0.42</td>
<td>0.42</td>
<td>0.55</td>
<td>0.55</td>
<td>0.27</td>
<td>0.27</td>
<td></td>
</tr>
<tr>
<td>6.00</td>
<td>0.45</td>
<td>0.45</td>
<td>0.55</td>
<td>0.55</td>
<td>0.27</td>
<td>0.27</td>
<td></td>
</tr>
<tr>
<td>7.00</td>
<td>0.47</td>
<td>0.47</td>
<td>0.55</td>
<td>0.55</td>
<td>0.27</td>
<td>0.27</td>
<td></td>
</tr>
<tr>
<td>8.00</td>
<td>0.47</td>
<td>0.47</td>
<td>0.55</td>
<td>0.55</td>
<td>0.27</td>
<td>0.27</td>
<td></td>
</tr>
<tr>
<td>9.00</td>
<td>0.49</td>
<td>0.49</td>
<td>0.55</td>
<td>0.55</td>
<td>0.27</td>
<td>0.27</td>
<td></td>
</tr>
<tr>
<td>10.00</td>
<td>0.50</td>
<td>0.50</td>
<td>0.55</td>
<td>0.55</td>
<td>0.27</td>
<td>0.27</td>
<td></td>
</tr>
</tbody>
</table>

### SPECIAL TABLE GIVING FLOWRATES IN DESIGNATED PIPES AS A FUNCTION OF TIME

<table>
<thead>
<tr>
<th>TIME (HOURS)</th>
<th>PIPE NO</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>1.09</td>
<td>0.50</td>
<td>0.50</td>
<td>0.15</td>
<td>3.37</td>
<td>0.97</td>
<td>0.97</td>
<td>0.19</td>
<td>0.08</td>
<td></td>
</tr>
<tr>
<td>1.00</td>
<td>0.96</td>
<td>1.00</td>
<td>0.96</td>
<td>1.29</td>
<td>3.69</td>
<td>0.95</td>
<td>0.97</td>
<td>0.21</td>
<td>0.06</td>
<td></td>
</tr>
<tr>
<td>2.00</td>
<td>0.96</td>
<td>1.00</td>
<td>0.96</td>
<td>1.29</td>
<td>3.69</td>
<td>0.95</td>
<td>0.97</td>
<td>0.21</td>
<td>0.06</td>
<td></td>
</tr>
<tr>
<td>3.00</td>
<td>0.96</td>
<td>1.00</td>
<td>0.96</td>
<td>1.29</td>
<td>3.69</td>
<td>0.95</td>
<td>0.97</td>
<td>0.21</td>
<td>0.06</td>
<td></td>
</tr>
<tr>
<td>4.00</td>
<td>0.96</td>
<td>1.00</td>
<td>0.96</td>
<td>1.29</td>
<td>3.69</td>
<td>0.95</td>
<td>0.97</td>
<td>0.21</td>
<td>0.06</td>
<td></td>
</tr>
<tr>
<td>5.00</td>
<td>0.96</td>
<td>1.00</td>
<td>0.96</td>
<td>1.29</td>
<td>3.69</td>
<td>0.95</td>
<td>0.97</td>
<td>0.21</td>
<td>0.06</td>
<td></td>
</tr>
<tr>
<td>6.00</td>
<td>0.96</td>
<td>1.00</td>
<td>0.96</td>
<td>1.29</td>
<td>3.69</td>
<td>0.95</td>
<td>0.97</td>
<td>0.21</td>
<td>0.06</td>
<td></td>
</tr>
<tr>
<td>7.00</td>
<td>0.96</td>
<td>1.00</td>
<td>0.96</td>
<td>1.29</td>
<td>3.69</td>
<td>0.95</td>
<td>0.97</td>
<td>0.21</td>
<td>0.06</td>
<td></td>
</tr>
<tr>
<td>8.00</td>
<td>0.96</td>
<td>1.00</td>
<td>0.96</td>
<td>1.29</td>
<td>3.69</td>
<td>0.95</td>
<td>0.97</td>
<td>0.21</td>
<td>0.06</td>
<td></td>
</tr>
<tr>
<td>9.00</td>
<td>0.96</td>
<td>1.00</td>
<td>0.96</td>
<td>1.29</td>
<td>3.69</td>
<td>0.95</td>
<td>0.97</td>
<td>0.21</td>
<td>0.06</td>
<td></td>
</tr>
</tbody>
</table>
EXAMPLE PROBLEM 24

Three variations of this example problem are given. In the first a piping system has developed a leak in the vicinity of node 3. A pressure measurement at node 2 indicates a pressure of 30 psi. How much flow is discharging from the leak? It is assume that this system has been previously modelled and verified so that the computer does accurately describe its performance.

Example of determining loss of flow from a leak based on a pressure measurement, and known conditions in the network.

/*
$SPECIF NPRINT=10, COEPRO=.004 $END
PIPEC
1 10. 1000. 1 3 4950.
2 8. 3000. 1 2 1 4920.
3 8. 2000. 1 5 1.2 4950.
4 8. 2000. 5 6 .55 4970.
5 6. 2000. 5 2
6 6. 3000. 2 3
7 6. 1000. 3 4850.
8 6. 3000. 2 4 .85 4900.
9 6. 1500. 4 3
10 8. 4000. 6 4
11 8. 600. 6
SETRES
2 3 7 30.
PUMPS
1 2 50 3 45 4 30 5000
RESER
7 4900
11 5100
VALVE
9 750 4900
RUN

$SPECIF NPRINT=-1 $END

PIPEC
1 0 1 1000 10 .004
2 1 2 3000 8/
3 1 5 2000 8/
4 5 6 2000 8/
5 5 2 2000 6/
6 2 3 3000/.
7 0 3 1000/.
8 2 4 3000/.
9 4 3 1500/.
10 6 4 4000 8/
11 0 6 600/.
NODES
1 .3 4950
2 1 4920
3 0 4850
4 .85 4900
5 1.2 4920
6 .55 4970

-172-
In this variation of the problem it is desirable to determine what head and flowrate the pump should supply so that under the demands given in the sketch below the reservoir attached to the network by pipe 7 is neither filling or supplying any flow to the network. The differential head command DHEAD is an effective means for determining the required pump output because it allows the pressure, or head, at a node to be specified, and the solution provides the head required somewhere else in the network. For this problem the HGL elevation at node 3 is specified equal to the water surface elevation of the reservoir attached to the network by pipe 7, and therefore no flow will pass through pipe 7. The pump has been replaced in the input data by a reservoir, so that the reported differential head will be the head that the pump must supply. If the pump were retained it would be necessary to add the head produce by the pump to the reported differential head for the DHEAD device.

EXAMPLE OF DETERMINING THE HEAD A PUMP MUST PRODUCE SO THAT A RESERVOIR NEITHER SUPPLIES OR RECEIVES WATER.

/*
 $SPECIF NPRINT=-1,COEFRO=.004 $END

PIPE-
  1 10. 1000. 1 .3 4950.
  2 8. 3000. 1 2 1. 4920.
  3 8. 2000. 1 5 1.2 4950.
  4 8. 2000. 5 6 .55 4970.
  5 6. 2000. 5 2
  6 6. 3000. 2 3
  7 6. 1000. 3 1.5 4850.
  8 6. 3000. 2 4 .85 4900.
  9 6. 1500. 4 3
 10 8. 4000. 6 4
 11 8. 600. 6

RESER
  1 5000
  7 4900
 11 5100

VALVE
  9 750 4900

DHEAD
  1 50 3 1 4900

RUN

$SPECIF NPRINT=-1 $END

PIPES
  1 0 1 1000 10 .004
  2 1 2 3000 8/
  3 1 5 2000 8/
  4 5 6 2000 8/
  5 5 2 2000 6/
  6 2 3 3000/
  7 0 3 1000/
  8 2 4 3000/
  9 4 3 1500/
 10 6 4 4000 8/
 11 0 6 600/

NODES
  1 .3 4950
  2 1 4920
  3 1.5 4850
  4 .85 4900
  5 1.2 4920
  6 .55 4970
The third variation of this problem illustrates how the pressure setting of a PRV might be determined to limit the flow into the reservoir attached to the network by pipe 7 to 0.8 cfs under the demands and pump given in the sketch below. The differential head command DHEAD is well suited for this type of problem also. With its use the flow at the end of pipe 7 is specified equal to 0.8 cfs, the reservoir is removed and the head at node 7 at the end of pipe 7 is specified equal to 4900 feet, the elevation of the water surface in the reservoir.

EXAMPLE OF DETERMINING THE SETTING NEEDED ON A PRV TO LIMIT THE AMOUNT OF FLOW INTO A RESERVOIR
/* $SPECIF NPRINT=-1,COEFO=.004 $END $SPECIF NPRINT=-1 $END
PIPE-
PUMPS
1 10. 1000. 1 .3 4950.
2 8. 3000. 1 2 1. 4920.
3 8. 2000. 1 5 1.2 4950.
4 8. 2000. 5 6 .55 4970.
5 6. 2000. 5 2
6 6. 3000. 2 3 1.5 4850.
7 6. 1000. 3 7 .8 4900.
8 6. 3000. 2 4 .85 4900.
9 6. 1500. 4 3
10 8. 4000. 6 4
11 8. 600. 6
PUMPS
1 2 50 3 45 4 30 5000
RESER
11 5100
DHEAD
9 -20 7 1 4900
RUN

-174-
In this example problem it is desired that pipes that do not have loops through them be sized to give a velocity of 5 fps in them. The data for the initial "run" will contain the option DESIGN-5, requesting that USU-NETWK indicate which pipe diameters should be given in the input data, and which pipes should have the velocity specified. The first column given below the figure is for this initial "run." After obtaining the list of pipe through which no loops pass it is decided that pipes 1, 14 and 15 should have diameters of 14, 4 and 4 inches, respectively. Therefore in the input data in the second column given below these diameters have a minus sign in front of them to tell USU-NETWK that these are diameters, and not flow velocities. Note in the second column of input data that the option NOMSOL-10 is given indicating that a follow on solution should be obtained in which the computed diameters should be replaced by the next larger standard pipe size that are in USU-NETWK default list of standard pipe sizes, since NOMDIA is not included in the $SPEGIF list.

SIZING PIPES TO GIVE SPECIFIED VELOCITIES
"$SPECIF DESIGN=5,NODESP=1 SEND
PIPES
 1 2 2200 5 .005
 2 3 2000
 3 4 1500/
 4 5 1500/
 5 6 1500/
 6 7 3000/
 7 8 2500/
 8 7 2500/
 9 8 9 1500/
10 9 10 1500/
11 10 11 1500/
12 7 12 1000/
13 12 14 1000/
14 8 13 1000/
15 13 15 1000/
NODES
 1 0 600
 3 2 550
 3 4 530
 4 3 520
 5 3 510
 8 3 500
 2 3 500
 6 4 450
 6 5 450
 9 3 430
10 3 520
11 3 510
12 3 500
13 3 500
14 2 500
15 2 500
PUMPS
 1 4 100 6 8 04 8 85 590
RUN
-175-

SIZING PIPES TO GIVE SPECIFIED VELOCITIES
"$SPECIF NOMSOL=10,DESIGN=4,NODESP=1 SEND
PIPES
 1 2 2200 -14 .005
 2 3 2000 8/
 3 4 1500 5/
 4 5 1500 5/
 5 6 1500 5/
 6 7 3000 8/
 7 8 2500 8/
 8 7 2500 8/
 9 8 9 1500 5/
10 9 10 1500 5/
11 10 11 1500 5/
12 7 12 1000 5/
13 12 14 1000 5/
14 8 13 1000 -4/
15 13 15 1000 -4/
NODES
 1 0 600
 3 2 550
 3 4 530
 4 3 520
 5 3 510
 6 3 500
 7 6 540
 8 5 450
 9 3 430
10 3 520
11 3 510
12 3 500
13 3 500
16 2 500
PUMPS
 1 4 100 6 8 04 8 85 590
RUN
The output from the problem on the previous page is given below. The first run represents the results from a preliminary "run" in which USU-NETWK is told to determine which pipe have loops passing through them. Based on this information the data in the second column of the previous page was prepared, and the other output results from this execution of USU-NETWK.

**Output from input with DESIGN=4**

**Title Given to Remark**

**Preliminary Problem to Find Pipes with Loops**

All demand flows are multiplied by 1.0000

<table>
<thead>
<tr>
<th>PIPES</th>
<th>NODES</th>
<th>SOURCE PUMPS</th>
<th>BACK PUMPS</th>
<th>RESERVOIRS</th>
<th>MUX LOSS</th>
<th>RISERS</th>
<th>FREE</th>
<th>NOSE</th>
<th>MUX VALUES</th>
<th>BACK FREE</th>
<th>DIE FREE</th>
<th>OPERATIONS</th>
<th>SPECIFIED</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Flow fromumps and reservoirs equals 4.80

The following pipes have loops thru them

(Give diameters for them)

The following pipes do not have loops thru them

(Give velocities rather than diameters for them & set DESIGN=4)

**Output from input with DESIGN=4 and NUSEND=10**

**Title Given to Remark**

Sizing pipes to give specified velocities

All other flows are multiplied by 1.0000

<table>
<thead>
<tr>
<th>PIPES</th>
<th>NODES</th>
<th>SOURCE PUMPS</th>
<th>BACK PUMPS</th>
<th>RESERVOIRS</th>
<th>MUX LOSS</th>
<th>RISERS</th>
<th>FREE</th>
<th>NOSE</th>
<th>MUX VALUES</th>
<th>BACK FREE</th>
<th>DIE FREE</th>
<th>OPERATIONS</th>
<th>SPECIFIED</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Flow fromumps and reservoirs equals 4.80

Lost due to fl. fric. in all pipes

Power loss= 33.50 ft. of loops. Energy loss= 781.325 hors/day

Pipe head flow horsepower kilowatts kwatt-hrs/day

**MODE DATA**

<table>
<thead>
<tr>
<th>NO. (inches)</th>
<th>(gpm)</th>
<th>ELEV</th>
<th>HEAD</th>
<th>PRESSURE</th>
<th>ELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>4</td>
<td>10</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>4</td>
<td>10</td>
<td>20</td>
<td></td>
</tr>
</tbody>
</table>

**Lost due to fl. fric. in all pipes**

Pipe head flow horsepower kilowatts kwatt-hrs/day

Pipe head flow horsepower kilowatts kwatt-hrs/day

The output from the problem on the previous page is given below. The first run represents the results from a preliminary "run" in which USU-NETWK is told to determine which pipe have loops passing through them. Based on this information the data in the second column of the previous page was prepared, and the other output results from this execution of USU-NETWK.

**Output from input with DESIGN=4**

**Title Given to Remark**

Preliminary Problem to Find Pipes with Loops

All demand flows are multiplied by 1.0000

<table>
<thead>
<tr>
<th>PIPES</th>
<th>NODES</th>
<th>SOURCE PUMPS</th>
<th>BACK PUMPS</th>
<th>RESERVOIRS</th>
<th>MUX LOSS</th>
<th>RISERS</th>
<th>FREE</th>
<th>NOSE</th>
<th>MUX VALUES</th>
<th>BACK FREE</th>
<th>DIE FREE</th>
<th>OPERATIONS</th>
<th>SPECIFIED</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Flow fromumps and reservoirs equals 4.80

The following pipes have loops thru them

(Give diameters for them)

The following pipes do not have loops thru them

(Give velocities rather than diameters for them & set DESIGN=4)

**Output from input with DESIGN=4 and NUSEND=10**

**Title Given to Remark**

Sizing pipes to give specified velocities

All other flows are multiplied by 1.0000

<table>
<thead>
<tr>
<th>PIPES</th>
<th>NODES</th>
<th>SOURCE PUMPS</th>
<th>BACK PUMPS</th>
<th>RESERVOIRS</th>
<th>MUX LOSS</th>
<th>RISERS</th>
<th>FREE</th>
<th>NOSE</th>
<th>MUX VALUES</th>
<th>BACK FREE</th>
<th>DIE FREE</th>
<th>OPERATIONS</th>
<th>SPECIFIED</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Flow fromumps and reservoirs equals 4.80

Lost due to fl. fric. in all pipes

Power loss= 33.50 ft. of loops. Energy loss= 781.325 hors/day

Pipe head flow horsepower kilowatts kwatt-hrs/day

**MODE DATA**

<table>
<thead>
<tr>
<th>NO. (inches)</th>
<th>(gpm)</th>
<th>ELEV</th>
<th>HEAD</th>
<th>PRESSURE</th>
<th>ELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>4</td>
<td>10</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>4</td>
<td>10</td>
<td>20</td>
<td></td>
</tr>
</tbody>
</table>

**Lost due to fl. fric. in all pipes**

Pipe head flow horsepower kilowatts kwatt-hrs/day

Pipe head flow horsepower kilowatts kwatt-hrs/day

The output from the problem on the previous page is given below. The first run represents the results from a preliminary "run" in which USU-NETWK is told to determine which pipe have loops passing through them. Based on this information the data in the second column of the previous page was prepared, and the other output results from this execution of USU-NETWK.
APPENDIX A

An alternate form of data input is described in this appendix. Originally this form of input was described in a Users Manual for the computer programs NETNEW and NETLIN. The program NEWTH was later expanded to provide the capability to simulate a network through time and was also given the capability to handle the data initially required by NETNEW as well. Now the version of NEWTH without time simulation capabilities will also allow this form of input.

The change data and the time simulation data are identical regardless of whether this alternate form or the form described in the body of the manual is used. Only the original network data is in a different form. Instead of entering data of a given type by a command name this form of data must be entered in the order listed below. All records are free format. In describing the input and its sequence separate card numbers will be listed. Many of these so-called cards, and for large networks most of these so-called cards, will consist of several records of data. Also card numbers preceded by an * must be omitted if the network has no device of this type.

To select this form of input instead of the form described in the body of the manual. The parameter INPUTA in the $SPECIF list must be set equal to 1.

Card No. 1

Is identical to that described in the body of the manual; namely, it is the title card.

Card No. 2

Is identical to card no. 2 described in the body of the manual; namely, it is the $SPECIF card containing the parameter names to override any of the default option of the program. Among other names in the list the following one must be given, INPUTA = 1.

Card No. 3

This record or card sequence must begin with $ASPEC1 starting in column "2" and going through column "8", and must end with $END. This record is a FORTRAN NAMELIST just as Card No. 2 is, and may contain any or all of the following:

**NPIPE**: The number of pipes in the network. The number is absolutely necessary in using the alternate form of input described in Appendix B.

**NODES**: The number of junctions in the network. This value is not provided when the form of input described in this appendix is used, but is necessary when using the alternate input described in Appendix B is used.

**LOOPS**: The number of real loops in the network. This value is not provided generally when the form of input described in this appendix is used but is necessary when the alternate input described in Appendix B is used.

**LOOPSS**: The number of pseudo loops in network. This value is not provided generally when the form of input described in this appendix is used, but is necessary when the alternate input described in Appendix B is used.

**MINOR**: The number of pipes containing devices which result in significant minor losses. (default, MINOR = 0).

**NPUMP**: The number of pumps in the network. This number includes the source pumps as well as the booster pumps (default, NPUMP = 0).

**NBPUMP**: The number of these pumps (i.e., these given by NPUMP) which one booster pumps, i.e., do not supply the network, but rather increase the pressure in a line. (default NBPUMP = 0).

**NRES**: The number of reservoirs (or tanks) without pumps which supply the network. (default, NRES = 0).

**NVALVE**: The number of pressure-reducing valves, PRV's, or pressure regulating valves in the network. (default NVALVE = 0).

**NCKVAL**: The number of check valves. (default NCKVAL = 0).

**NOZZLE**: The number of nozzles or orifices which discharge fluid from the network. These orifices, like reservoirs, are at the ends of pipes connected to the network and are not numbered as nodes unless NODESP = 1 is set in $SPECIF of card no. 2. (default NOZZLE = 0).

**NBPV**: The number of back pressure valves (default NBPV = 0).

**NPSEI**: The number of pumping stations given by NPUMP which contain 2 or more identical pumps operating in series (default NPSEI = 0).
NPARAL: The number of pumping stations given by NPUMP which contain 2 or more identical pumps operating in parallel (default NPARAL = 0).

NPPOWE: Is a parameter which must be set to 1 if pump characteristics are specified by giving the power they supply to the fluid rather than giving three points of Q versus H along the pump curves (default NPPOWE = 0).

NPSERR: Allows pumping stations to be referenced by the pump number (i.e., sequence number of the pump input data), or the pipe or node number used to identify that station for purposes of denoting pumps operating in series or parallel. If NPSERR = 0 (default) then such reference is by pump number; if NPSERR = 1, then such reference is by other identification.

NOZDIA: Allows the size of the orifice or nozzle outlets to be specified according to NOZDIA:

= 0, then outlet area in square feet or square meters is given respectively whether ES or SI units are used.

= 1, then outlet area is given as a diameter in inches (cannot be used with SI units).

= 2, then outlet area is given in feet or meters depending upon whether ES or SI units are used respectively (default NOZDIA = 0).

NOZHGL: Allows an estimate of the HGL immediately upstream from nozzles or orifices to be supplied. If NOZHGL = 0, such estimates are required;

= 1, no such estimates are needed, and the program generates an estimate internally to start the solution (default NOZHGL = 0).

LOOPRD If set to 1 loops will be determined internally when using input data form of Appendix B.

The following cards give the input data describing the detailed information for the computer program to determine precisely what the network is composed of. This information under this alternate form of input is not entered by a description command name as is done by the input data described in the body of the manual. Rather, the data must be entered in the sequence given. Any redundant record, or a missing record will provide erroneous data for subsequent network components and will most generally result in termination because the number of data given will not correspond to the number required. The sequential order of input data is identified below by card numbers. It is helpful to keep the following order of network devices in mind in following this sequence: (1) reservoirs, (2) nozzles, (3) pumps, (4) pressure reduction valves, (5) minor losses, (6) check valves, and (7) pipe and node data. Several records are required for most of these devices. If none of that particular device is present in the network, all such records (or cards) are not given. The program expects to read the data for that device or not read the data for that device based upon whether any of them exist or not are given in the list under $ASPEC1 (card no. 3). The exception to this rule is that pipe and node data will always be expected whether NPipe is given a value or not. The data requested under any card (with the exception of the pipe and node data) can spill over onto as many cards as needed. One or more blanks, or a comma which may be followed by blanks, separate the individual data items. If one too few data items are given, the next card will be read with a message that too many data items are provided if more than one item exists on that card.

*Card No. 4—Reservoir Information

(This card does not exist if default NRES = 0 is used.) The pipe numbers (integers without decimal points) which connect each reservoir (or tank) to the network. If NODESP = 1 is set in $SPEC1, these numbers are the node numbers defining the source of reservoir supply instead of the pipe numbers.

*Card No. 5—Reservoir Information

(Does not exist if default NRES = 0 is used.) The elevations of the liquid surface in the reservoirs (or tanks) in the same sequence as listed on the previous card. If ES units are used (NUNIT = 0, or = 1), these elevations are in feet, if SI units are used (NUNIT = 2, or = 3), these elevations are in meters.

*Card No. 6—This and the next four cards are for nozzles.

(Does not exist if NOZZLE = 0 is used.) The pipe numbers which contain the nozzles at the end. These nozzles discharge fluid to the atmosphere. If NODESP = 1, then these numbers are the nodes rather than the pipe numbers.

*Card No. 7

(Does not exist if NOZZLE = 0 is used.) An estimate of the HGL in the pipes immediately
upstream from the nozzle. This HGL equals the elevation of the nozzle plus the pressure head, \( p / g \), in the pipe at the nozzle. If \( \text{NOZDIGL} = 1 \) is set in the namelist, then this record or card is not given.

*Card No. 8

(Does not exist if \( \text{NOZZLE} = 0 \) is used.) The discharge coefficients of the nozzles or orifices. For an orifice, this value is approximately equal to 0.6.

*Card No. 9

(Does not exist if \( \text{NOZZLE} = 0 \) is used.) The cross-sectional areas of the nozzles or these diameters according to NOZDIA in the NAMELIST. These areas are given in the basic units, i.e., square feet if ES units are used or square meters if SI units are used, or if diameters are selected according to the description under NOZDIA.

*Card No. 10

(Does not exist of \( \text{NOZZLE} = 0 \) is used.) The elevations of the nozzles. If NODESP = 1 this card does not exist; rather the elevations are given on the pipe-node cards given below.

*Card No. 11—Pump Information

(Does not exist if default \( \text{NPUMP} = 0 \) is used.) The pipe numbers which contain pumps. The pipe number in which booster pumps exist must be listed after those which supply the network from wells or reservoirs, i.e., booster pumps come after the last source pumps. If NODESP = 1, then the node numbers instead of pipe numbers are given.

*Card No. 12—Pump Information

(Does not exist if default \( \text{NPUMP} = 0 \) is used or if none of the pumping stations (either source pumps or booster pumps) have a second or more pumps in series, i.e., \( \text{NPSERI} = 0 \).)

The pipe number (or station designation—see NPSERR) and the number of pumps in series as pairs of numbers. The number of such pairs equals \( \text{NPARAL} \). If only one pump exists or if the given characteristics reflect the pumps in parallel, this card should not be included, but may be if the number of parallel pumps is specified as 1.

*Card No. 13—Pump Information

(Does not exist if default \( \text{NPUMP} = 0 \) is used or if none of the pumping stations (either source pumps or booster pumps) have a second, or more, pumps in parallel, i.e., \( \text{NPARAL} = 0 \).)

The pipe number (or station designation—see NPSERR) and the number of pumps in parallel as pairs of numbers. The number of such pairs equals \( \text{NPARAL} \). If only one pump exists or if the given characteristics reflect the pumps in parallel, this card should not be included, but may be if the number of parallel pumps is specified as 1.

*Card No. 14—Pump Information (one card for each pump)

(Does not exist if default \( \text{NPUMP} = 0 \) is used.) For each pump in the network, a separate card is required defining the pump characteristic curve and the elevation of the liquid surface from which the supply is obtained. For booster pumps the elevation must be punched as 0.0. The pump characteristics are defined by one of two methods according to whether \( \text{NPPPOWE} = 0 \), or \( = 1 \). If \( \text{NPPPOWE} = 0 \), then these characteristics are defined by three pair of flow rate versus head produced by the pump. These pair of values can be taken from a pump curve giving the head the pump produces as a function of the flow rate. These points cannot all lie on a straight line, and the middle point should be taken as the normal capacity (highest efficiency) of the pump unless knowledge exists that the pump is not operating near its normal capacity.

If \( \text{NPPPOWE} = 1 \), then these characteristics are defined by the power supplied the fluid by the pump in horsepower or kilowatts, depending upon whether ES or SI units are used respectively, and the normal capacity (flow rate of maximum efficiency) of the pump. The power supplied the fluid is the pump brake power times the pump efficiency. The normal capacity is in cfs or \( m^3/s \) depending upon whether ES or SI units are used. If \( \text{NPGPM} = 1 \), the flow rates are in gpm for either option if ES units are used. In summary, if \( \text{NPPPOWE} = 0 \), (the default), then each such pump card contains:

\[
Q_1, H_1, Q_2, H_2, Q_3, H_3 \\
\text{elevation of supply surface}
\]

If \( \text{NPPPOWE} = 1 \), then each such pump card contains:

Power (HP or watts), \( Q_n \) (normal capacity), elevation of supply surface

*Card No. 15—PRV Information

(Does not exist if default \( \text{NVALVE} = 0 \) is used.) The pipe numbers which contains the pressure reducing valves, PRV's.

*Card No. 16—PRV Information

(Does not exist if default \( \text{NVALUE} = 0 \) is used.) The distances from the beginning of the pipe to the PRV's.

*Card No. 17—PRV Information

(Does not exist if default \( \text{NVALUE} = 0 \) is used.) The elevations of the HGL immediately downstream from the PRV's.
*Card No. 18—BPV Information
(Does not exist if default NBPV = 0 is used.) The pipe numbers which contain the back pressure values, BPV's.

*Card No. 19—BPV Information
(Does not exist if default NBPV = 0 is used.) The distances from the BPV's to the downstream node of the pipe which contains the BPV.

*Card No. 20—BPV Information
(Does not exist if default NBPV = 0 is used.) The elevation of the HGL immediately upstream from the BPV's that constitute the values setting. If NPRERV = 1 then in place of the HGL elevation the pressure is given.

*Card No. 21—Minor Loss Information
(Does not exist if MINOR = 0 is used.) The number of the pipes which contain the minor loss devices.

*Card No. 22—Minor Loss Information
(Does not exist if MINOR = 0 is used.) The minor loss coefficient $K_L$ in $h_L = K_L(V^2/2g)$ in the same order as the pipe numbers are given on the previous card.

*Card No. 23—Check Value Information
(Does not exist if NCKVAL = 0 is used.) The pipe numbers containing the check values.

*Card No. 24 (pipe-node data)
The pipe and node data are given hereunder. It will consist of a separate card or record for each pipe of the network. The order of the individual data items on this card are described in the body of the report under the command name PIPE-node used to enter data. That is all items are as described there with the exception that the name PIPE- is not given. However, this sequence of data records is terminated with the word END.

The order of each pipe node data record is: (1) pipe number, (2) pipe diameter, (3) pipe length, (4) pipe roughness, (5) upstream node number, (6) demand at upstream node, (7) downstream node number, (8) demand at downstream node, and (9) the elevation of the downstream node. Many of these items can be omitted from any record as described earlier in the paragraphs under PIPE-node.

For special networks data may follow card number 24. For instance a network with no source of supply require the elevation of the HGL at a beginning node as described in body of report. Also if options LOOPSE = 1, or NPERCT = 1, the data required by these options must follow card number 21. In addition changes to basic network follow for separate alternate analyses.

**EXAMPLES OF INPUT DATA**

In the following example of input data written notes are provided after $J$ marks to assist in identifying data of a given type, but these notes are not part of what is actually contained on the record.

Example A-1

The following data is the input data required for the network illustrated in Figure A-1.
Example A-2 and Example A-3.

Another small network, consisting of 10 pipes is given in Figure A-1, and the input utilizing that described in this Appendix A is given to its left. A slightly larger example is given in Figure A-3 with the accompanying input data. In Appendix B there are other examples in which the input described in this Appendix A is utilized.

**Input data example problem in Fig. A-2**

```plaintext
$SPECIFIC INPUTA=1 $END
$SPECIFIC NPUMP=2,NBPUMP=1,NRES=1 $END
9
500.
10 3
1.5 120. 2.5 100. 3.5 75. 400.
1.0 40. 1.5 35. 2.0 28. 0.
1 6. 2000. .0.02 1 7 2.0 400.
2 6. 2000. .0.02 2 3 1.0 100.
3 6. 2500. .0.02 3 4 3.5 150.
4 6. 2000. .0.02 4 4 .5 100.
5 6. 800. .0.02 3 4
6 6. 1000. .0.02 3 2
7 6. 1800. .0.02 5 6 .5 100.
8 6. 1500. .0.02 4 6
9 8. 1000. .0.02 5 6
10 8. 1000. .0.02 1 2
END
```

**Third example of data for network (Figure A-3)**

```plaintext
$SPECIFIC NFLOW=1,NFLENG=1,INPUTA=1 $END
$SPECIFIC NPUMP=2,NPUMP=3, NBPUMP=1 $END
26 27 25
5.568 453.97 7.0156 455.81 7.796 455.81 910.
2.004 285.5 2.385 284.25 3.341 277. 1080.
4.009 182. 4.454 180. 4.9 177. 0.
1 12. 1.5 130. 1 2 1022.
2 8. 1. 130. 3 2
3 10. 1.2 120. 3 4 500. 1057.
4 10. 2.0 120. 5 4
5 8. 2.8 120. 5 6 400. 1047.
6 8. 1.1 120. 7 6
7 8. 1. 120. 8 7 1025.
8 8. 2.5 120. 9 600. 8 1011.
9 8. 811. 100. 1 9 987.
10 6. 1.3 100. 9 10 1015.
11 6. 1. 100. 10 11 700. 1030.
12 6. 1.1 130. 12 11
13 6. 1. 130. 5 12 1051.
14 6. 1.8 120. 10 8
15 6. 1.1 120. 2 10
16 6. 1.8 120. 11 7
17 10. 1.2 130. 11 3 1038.
18 6. 1.8 120. 6 12
19 6. 1.3 120. 4 12
20 8. 1. 130. 13 500. 14 1085.
21 8. 1.2 120. 13 15 400. 1075.
22 8. 1.1 120. 14 16 500. 1110.
23 8. 1.1 120. 15 16
24 6. 1.8 120. 14 600. 15
25 12. 1.5 130. 8 13 1100.
26 20. .5 110. 1 990.
27 20. .5 110. 5 1080.
28 12. 2.3 120. 6 14
END
```
APPENDIX B

This appendix describes a possible form of input data that is node oriented rather than pipe oriented. This node oriented form is also referred to as the third form of input with the form in Appendix A called the second form. Communication is provided to the program that this form of input data is selected by setting INPUTA = 2 or 3 in the $SPECIF list of parameters. This form of input is a variation of that described in Appendix A; that is all data is identical to that described in Appendix A with the exception that the information described under Card No. 24 which contains the pipe-node data is different. That is auxiliary items to the pipes such as reservoirs, pumps, etc. are entered as in Appendix A. This node oriented alternate form of input was originally the form described as the alternate form in the users manual for the computer programs—NETNEW and NETLTIN.

In using this third possible form of input which is basically oriented around the nodes of the network rather than the pipes of the network it is possible to either let the program find the loops (both corrective flow rate and the different set of energy loops which are necessary if PRV’s or BPV are present) internally or to provide data which define these loops. If INPUTA = 3 the loops will be determined internally. If INPUTA = 2 it is necessary to provide data defining these loops as described in card group No. 28 and 30 below. That is by reading in loop data the algorithms used in the program to define loops of the network are by-passed. Consequently using INPUTA = 2 will result in a slight reduction in the amount of computer time needed for a solution, at the expense of additional data on your part. Use of this input does allow you as a user to exercise your best judgment in defining the network so that loops contain the fewest possible number of pipes which may allow the Jacobian to be placed in a smaller band width. (Some control of this is permitted in the other modes of input by ordering source pumps and reservoirs properly, or using the option LOOPSE = 1.) The main advantage of using INPUTA = 2 occurs only if the program for some reason is unable to properly define the network’s loops.

In this third form of data the pipe numbers are taken consecutively in the order in which the data is entered, i.e. the same as if NUMPIP = 1 (number of pipes is omitted) is set in main form of input, unless this default is overridden. Since, however, pipe numbers are required to define loops it is necessary that pipes be numbered consecutively without pipe numbers omitted in the default node. If pipe numbers are not omitted i.e., all consecutive numbers are not included, it is necessary to add the parameter NOSEQP = 1 to the $ASPECI list. If the option NOSEQP = 1 is set, then before Card No. 24 listed below the pipe numbers must be listed on a record.

If nodes are not numbered consecutively then the parameter NOSEQN = 1 must be set in the $ASPECI list. When NOSEQN = 1, then the node numbers are read in before Card No. 24 listed below. If NOSEQP = 1 is also set these node numbers follow the pipe numbers. The only effect of setting NOSEQN = 1 is in the printed numbers in the output tables.

Input Data

Input data is identical to that described in Appendix A from Card No. 1 through Card No. 23. The only exception is Card No. 2 should contain INPUTA = 2, or 3.

In place of the pipe-node data, Card No. 24, the following data is required.

Card No. 24

The diameters of the pipes listed in the same order as they are numbered throughout the network. No numbers may be omitted in the pipe numbering sequence.

Card No. 25

The lengths of the pipes listed in the same order as they are numbered throughout the network.

Card No. 26

The wall roughness, e, of the pipes listed in the same order as they are numbered throughout the network, or the Hazen-Williams c or Manning n equations. If Hazen-Williams or Manning’s equation is used, this must be communicated in the $SPECIF list by NHAZEN = 1 or 5 respectively.

Card No. 27

(In this group there must be a card for each junction of the network given in the same order as the junctions are numbered.) Each of these cards contains the external flow at the junction first, and thereafter the pipe numbers, which meet at that junction. If the assumed direction of flow is into the junction, a minus sign precedes the pipe number. Blanks are not permitted between the minus sign and the number. If NELEV = 0 is set in the $SPECIF list, then the format of this card contains the elevation of the function immediately following the external flow or demand at the junction. That is the format of the card becomes demand, elevation, list of pipes that join at junction.

If NELEV = 0 Card No. 29 does not exist since the junction elevations have already been given.

These records are the basic data from which the program is able to determine the topology and connectivity of the network. In preparing the data the user goes to each node of the network and notes the pipe numbers that join at that node. A node number is not actually given here however, under the assumption that the nodes are numbered consecutive in the order in which the data is entered without any missing

B-1
node numbers. Should this not be the case NOSEQN = 1 must occur in the $ASPECTI list and the number actual node number entered as data before Card No. 24 above as described earlier.

Card Group No. 23 (Does not exist if INPUTA = 3)

(In this group there must be a card for each loop of the network. Cards for pseudo loops precede cards for real loops.)

Each of these cards contains the pipe numbers around the given loop. If the direction of the assumed flow in the pipe is opposite to the positive direction around the loop, this number is preceded by a minus sign (with no blank between it and its number).

Card No. 29 (Does not exist if NELEV = 0)

Contains the elevations of the junctions in the same order as these junctions are numbered throughout the network.

*Card Group No. 30 (Does not exist if INPUTA = 3)

(Does not exist if there are no PRV's in the network.) The first card of this group contains two integers: the number of pseudo energy loops, and the number of real energy loops.

The following cards in this group contain the pipe numbers around the energy loops of the network, with the pseudo energy loops first before the real energy loops. That is, this group of loop data cards are like number 28 with the exception that they define the energy loops which are formed by replacing PRV's with artificial reservoirs.

As an illustration of the use of this alternate form of input, the simple illustration network in Figure B-1 is given first.

*See book "Analysis of Flow in Pipe Network" for explanation of difference between corrective flow rate loops, card 25, and energy loops, card 27.

---

**Figure B-1.**

**Table B-1.**

<table>
<thead>
<tr>
<th>Node</th>
<th>1.2 cfs K_\text{L}=10</th>
<th>25 cfs</th>
<th>8 - 500</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12 - 2500</td>
<td>10 - 2000</td>
<td>(1)</td>
</tr>
<tr>
<td>2</td>
<td>150'</td>
<td>50'</td>
<td>2.0 cfs</td>
</tr>
<tr>
<td>3</td>
<td>HP=16, \eta_0=2 cfs</td>
<td>50'</td>
<td>NGL=65'</td>
</tr>
<tr>
<td>4</td>
<td>HP=16, \eta_0=1.45 cfs</td>
<td>8 - 500</td>
<td>(5)</td>
</tr>
<tr>
<td>5</td>
<td>100'</td>
<td>50'</td>
<td>2.0 cfs</td>
</tr>
<tr>
<td>6</td>
<td>150'</td>
<td>1.5 cfs</td>
<td></td>
</tr>
</tbody>
</table>

**SIMPLE NETWORK USED TO ILLUSTRATE ALTERNATE INPUT.**

- \text{SPECIFIC INPUTA=2,NLEVEL}=1 \text{ SEND}
- \text{ASPECTI NPIPE}=6, NMODES=4, NFREE=1, NFUP=2, NFPUMP=1, NVALVE=1, NPSWELL=1, NHIBOR=1, LOOPSR=1, LOOPPS=1, NPARAL=1 \text{ SEND}
- \text{reservoir}
  - 100.
  - 1
  - 2
  - 16.
  - 2. 50. \{ \text{pumps} \}
  - 16.
  - 1.45 0. \{ \}
- \text{4.800.} \{ \text{PRV} \}
- 63.
  - 2 \{ \text{minor loss} \}
- 10.
  - 12. 10. 6. 10. 8. 8. \} \text{diameters}
  - 2.5 2. 1.2 1.3 1.3 \} \text{lengths (1000 ft)}
  - .0102 .0102 .0102 .0102 .0102 \} \text{coefficients}
  - 1.2 -1 3 2
  - 95 -2 4 6 -5 \} \text{junction data}
  - 1.5 -6
  - 2 -3 -4
  - 5 -2 -1 \} \text{corrective flow rate loops}
  - 2 4 -3 \} \text{corrective flow rate loops}
- 70.
  - 60. 150. .50 \} \text{elev. of nodes}
  - 2 0 \} \text{number of pseudo and real energy loops}
  - 5 -2 -1
  - 4 -3 2 -5 \} \text{energy loops}
If this same 6 pipe network were to be analyzed using the data input described in Appendix A the input data would be:

SIMPLE NETWORK USED TO ILLUSTRATE ALTERNATE INPUT A.

\[
\begin{align*}
\text{SPECIF} & \text{ INPUTA} = 1, \text{ NLPE} = 6, \text{ NRES} = 1, \text{ NVALVE} = 1, \\
\text{NPIPE} & = 4, \text{ NREAL} = 1, \text{ NPAR} = 1 \text{ SEND} \\
5 & \text{ reservoir} \\
1 & \text{ 120.} \\
1 & \text{ 90.} \\
16. & \text{ 50.} \\
4 & \text{ 0.} \\
800. & \text{ PRV} \\
65. & \text{ 100.} \\
2 & \text{ minor} \\
1 & \text{ 12.} \\
2 & \text{ 20.} \\
3 & \text{ 12.} \\
4 & \text{ 12.} \\
5 & \text{ 12.} \\
6 & \text{ 12.} \\
\text{ END} \\
\end{align*}
\]

In using the data input described in the body of this user's manual the network of Figure B-1 would be analyzed by:

SIMPLE NETWORK USED TO ILLUSTRATE STANDARD INPUT.

\[
\begin{align*}
\text{SPECIF} & \text{ INPUTA} = 1, \text{ NLPE} = 6, \text{ NRES} = 1, \text{ NVALVE} = 1, \\
\text{NPIPE} & = 4, \text{ NREAL} = 1, \text{ NPAR} = 1 \text{ SEND} \\
5 & \text{ reservoir} \\
1 & \text{ 120.} \\
1 & \text{ 90.} \\
16. & \text{ 50.} \\
4 & \text{ 0.} \\
800. & \text{ PRV} \\
65. & \text{ 100.} \\
2 & \text{ minor} \\
1 & \text{ 12.} \\
2 & \text{ 20.} \\
3 & \text{ 12.} \\
4 & \text{ 12.} \\
5 & \text{ 12.} \\
6 & \text{ 12.} \\
\text{ END} \\
\end{align*}
\]

As a second example of the third form of input described in this appendix, input data for the network in the main body of this users manual example number 1 is given below:

SIMPLE NETWORK USED TO ILLUSTRATE INPUT B.

\[
\begin{align*}
\text{SPECIF} & \text{ INPUTA} = 2, \text{ NLPE} = 6, \text{ NRES} = 1, \text{ NVALVE} = 1, \\
\text{NPIPE} & = 4, \text{ NREAL} = 1, \text{ NPAR} = 1 \text{ SEND} \\
5 & \text{ reservoir} \\
1 & \text{ 120.} \\
1 & \text{ 90.} \\
16. & \text{ 50.} \\
4 & \text{ 0.} \\
800. & \text{ PRV} \\
65. & \text{ 100.} \\
2 & \text{ minor loss} \\
1 & \text{ 12.} \\
2 & \text{ 20.} \\
3 & \text{ 12.} \\
4 & \text{ 12.} \\
5 & \text{ 12.} \\
6 & \text{ 12.} \\
\text{ END} \\
\end{align*}
\]

The final summary tables of the solution for the network of Figure B-1 are given below:

<table>
<thead>
<tr>
<th>PIPE</th>
<th>NO. FROM TO LENGTH DEAN</th>
<th>COST</th>
<th>FLOW RATE VELOCITY</th>
<th>HEAD LOSSS /1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0 1 2500. 12.0 .012000 3.97% 5.06 19.5% 7.42</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1 2 2523. 10.0 .012000 1.43% 2.74 7.15 2.98</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3 4 1200. 10.0 .012000 1.23% 5.81 36.82 10.68</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2 4 1200. 10.0 .012000 1.23% 5.81 36.82 10.68</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>5 2 1200. 10.0 .012000 1.23% 5.81 36.82 10.68</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>3 5 1200. 10.0 .012000 1.23% 5.81 36.82 10.68</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>6 3 1200. 10.0 .012000 1.23% 5.81 36.82 10.68</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The node data is:

<table>
<thead>
<tr>
<th>NODE</th>
<th>NO.</th>
<th>(SF)</th>
<th>ELEV</th>
<th>HEAD PRESSURE</th>
<th>ELEV</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>1.2000 538.80 70. 31.31 13.57 10.31</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>.9500 426.55 60. 34.15 14.60 9.16</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>1.5000 673.50 150. 24.79 10.74 17.79</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>2.0000 898.00 50. 14.49 6.29 64.49</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In solving this network it would not have been necessary to set NOSEQP = 1 and NOSEQP = 0, but if this had not been done the data starting with pipe diameters would have needed to be entered in the same order as the pipe are numbered. Likewise the node data would have to be entered in the order to the node numbering.

As a third example of input data for example number 2 of the body of the report is given below:

EXAMPLE NO. 2 USING INPUT ALTERNATIVE B.

\[
\begin{align*}
\text{SPECIF} & \text{ INPUTA} = 2, \text{ NLPE} = 6, \text{ NRES} = 1, \text{ NVALVE} = 1, \\
\text{NPIPE} & = 4, \text{ NREAL} = 1, \text{ NPAR} = 1 \text{ SEND} \\
5 & \text{ reservoir} \\
1 & \text{ 120.} \\
1 & \text{ 90.} \\
16. & \text{ 50.} \\
4 & \text{ 0.} \\
800. & \text{ PRV} \\
65. & \text{ 100.} \\
2 & \text{ minor loss} \\
1 & \text{ 12.} \\
2 & \text{ 20.} \\
3 & \text{ 12.} \\
4 & \text{ 12.} \\
5 & \text{ 12.} \\
6 & \text{ 12.} \\
\text{ END} \\
\end{align*}
\]
Simple network solved using various options of input (steady state). To help in comparing the various forms of input available the network in Figure B-2 which uses SI units will be solved using the various form of input available. The inputs forms used are:

(1) The input described in this Appendix B in which the loops are determined internally by the program.

(2) The input described in this Appendix B, but in which the loops are defined by additional input data.

(3) The same as (2) except that node elevation data is given on a separate record instead of on individual node cards.

(4) The input described in Appendix A, loops are determined internally by the program.

(5) The same as No. 4 except that loops are defined by input data.

(6) The input as described in the main body of the report, or standard form of input.

(7) The standard form of input using the PIPE-NODES command in place of the PIPES and NODES command used in No. 6.

(8) The same as No. 6 with the exception that data is supplied to define the loops of the network.

While these eight separate input data files for the network B-2 do not represent all the combinations of input available they illustrate the versatility of input allowed. A new user might try all and decide which form serves his needs best. The forms described in Appendixes A and B are accommodated in the subroutine AINPUT. Should these forms never be used this subroutine should be deleted with the call to it in the main program removed, or replaced by a subroutine with only a return statement. If in addition the PIPE-NODES command is not to be used the subroutine PNREAD might be removed along with removal of statements in RDATA that call on this subroutine and identify the command PIPE. The last few statements in AINPUT read in loop defining data, and therefore this portion of subroutine AINPUT should be retained if this option is to be used even if only the standard form of input is to be retained.

<table>
<thead>
<tr>
<th>Pump No. 1</th>
<th>Pump No. 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flowrate (m³/s)</td>
<td>Head (m)</td>
</tr>
<tr>
<td>0.0425</td>
<td>36.6</td>
</tr>
<tr>
<td>0.0708</td>
<td>30.5</td>
</tr>
<tr>
<td>0.0991</td>
<td>22.9</td>
</tr>
</tbody>
</table>

Figure B-2.
INPUT DATA POSSIBILITY # 1

Fig B-2 network, using input data format of this appendix B, program finds loops loops internally (inputA=3)

/*
$SPEC IF NUNIT=3.INPUTA=3.COEFRO=.0259,NGPMP=3, NFLOW=3, SEND
$SPEC IF NPipe=10,NPUMP=1,NRES=3,NVALVE=3, NODES=6, NPUMP=2 SEND
9
150.
10 3
.0425 36.6 .0708 30.5 .0991 22.0 120.
.0283 12.2 .0425 10.7 .0568 8.5
5 4 7
120. 300. 210.
45. 45 45.
15. 15. 15. 15. 15. 15. 15. 20. 20.
600. 600. 760. 600.
240. 300. 550. 400. 300. 300.
.0259 .0259 .0259 .0259 .0259 .0259 .0259 .0259 .0259 .0259
0. 1 3 -10/
.057 -1 2 4 -6/
.0285 -3 -2 5 /
.01425 -4 -5 8 /
0. 6 7 -9/
.01425 -7 -8 /
120. 120. 106. 30.5 120. 30.5
*/

INPUT DATA POSSIBILITY # 2

Fig B-2 network, using input data format of this appendix B with node elevations on node cards instead of separate card

/*
$SPEC IF NUNIT=3.INPUTA=2.COEFRO=.0259,NELEVE=0, NFLOW=3, SEND
$SPEC IF NPipe=10,NBPUMP=1,NRES=1,NVALVE=3,NODES=6, NPUMP=2, LOOPSR=3, LOOPSS=1 SEND
9
150.
10 3
.0425 36.6 .0708 30.5 .0991 22.0 120.
.0283 12.2 .0425 10.7 .0568 8.5
5 4 7
120. 300. 210.
45. 45 45.
15. 15. 15. 15. 15. 15. 15. 20. 20.
600. 600. 760. 600.
240. 300. 550. 400. 300. 300.
.0259 .0259 .0259 .0259 .0259 .0259 .0259 .0259 .0259 .0259
0. 1 3 -10/
.057 -1 2 4 -6/
.0285 -3 -2 5 /
.01425 -4 -5 8 /
0. 6 7 -9/
.01425 -7 -8 /
120. 120. 106. 30.5 120. 30.5
*/

INPUT POSSIBILITY # 4

Fig B-2 network using input data format of appendix I, loops defined internally

/*
$SPEC IF NUNIT=3,INPUTA=1.COEFRO=.0259,NGPMP=3, NFLOW=3, SEND
$SPEC IF NPipe=10,NBPUMP=1,NRES=1,NVALVE=3, NPUMP=2 SEND
9
150.
10 3
.0425 36.6 .0708 30.5 .0991 22.0 120.
.0283 12.2 .0425 10.7 .0568 8.5 0.
5 4 7
120. 300. 210.
45. 45 45.
15. 15. 15. 15. 15. 15. 15. 20. 20.
600. 600. 760. 600.
240. 300. 550. 400. 300. 300.
.0259 .0259 .0259 .0259 .0259 .0259 .0259 .0259 .0259 .0259
0. 1 3 -10/
.057 -1 2 4 -6/
.0285 -3 -2 5 /
.01425 -4 -5 8 /
0. 6 7 -9/
.01425 -7 -8 /
120. 120. 106. 30.5 120. 30.5
*/
APPENDIX C - GRAPHICS
ASSOCIATED WITH USU-NETWK

There are several graphics packages associated with USU-NETWK that assist in displaying the networks it processes and present the results from the solutions obtained. The options NETPLT in the $SPECIF list controls what type of graphics is to be used, and in the main body of this manual a summary of some of the available graphics is given in the description under this option. This appendix provides more information related to using the graphics that are implemented. The available graphics, and how these programs are implemented will vary depending upon whether you are using USU-NETWK on a PC, a larger computer such as a VAX, the version of the program you have, and the graphic capabilities and/or graphic devices attached to the computer you are using. For example, if you are using a PC on a stand alone basis and the only graphics capability it has access to is its color monitor through a CGA, EGA or VGA graphics card, then the graphics you can use will be limited to those packages that place the displays on the PC's monitor. By issuing the command GRAPHICS to DOS prior to running this graphics program you can get a hard copy of the display on a dot matrix printer by pressing the shift and PrtSc keys simultaneously, provided your PC is attached to a printer with graphics capabilities. If you want this display printed on a laser printer it will be necessary to use other software and/or import the screen graphics into a word processor such as WordPerfect 5.0. The graphic capabilities of spread sheets such as EXCEL, LOTUS 123, or QUATTRO might also be utilized. General graphics software packages such as SIGMAPLOT, SLIDEWRITE or GOLDEN Graphics might also be utilized.

The graphics designed specifically for use with USU-NETWK consist of the following:

I. Calcomp Plots of the Network and its Solution

This graphics consists of having the network consisting of the pipes and the nodes, and selected items from a solution plotted using a drum, or flat table plotter such as a Calcomp plotter. In the VAX version of USU-NETWK, for example, this "Calcomp Plotting" capability is part of the program, and to activate it the option NETPLT is given a value with an absolute value of 1 or 2. If this is done, then upon completing the analysis solution the user will be prompted to provide information about the nature of the plot he wants. This "Calcomp plotting" capability is not part of the DOS PC version of USU-NETWK, but the same capability is available through a post processor program OFPLOT (which stands for off line plot). OFPLOT utilizes the general purpose graphics file (with the default name PLOT10.DAT) that USU-NETWK can be requested to write by giving the option NETPLT a value of 3. Of course the use of OFPLOT is not restricted to use on PCs. For example a
network could be solved using a PC, and the general purpose graphics file written. This graphics file could next be transferred to a larger computer that drives a Calcomp plotter, or other graphics device, and OFPLOT used on this larger computer to make the plot. Alternatively the larger computer could solve the network, write the graphics file, and a PC could utilize information in this output file intended to use to produce graphics associated with the network, and its solution.

This calcomp graphics capability calls on pretty standard "calcomp routines" such as PLOT, SYMBOL, NUMBER, etc. Depending upon the installation the graphics file written with the calcomp information in it can be sent to graphics devices other than just a Calcomp plotter. It may even be printed on a line printer such as a PRINTRONIX printer or TEKTRONIX video terminal, or even on a PC's monitor that is communicating to the host computer with a software package such as KERMIT with graphics capabilities.

II. PC Monitor Display with Post Processor Program PLTNET

This graphics is limited to PCs, and is described in an early part of this manual under the post processor PLTNET. If the option NETPLT=1 when using the PC version of the program then a file PLTNET.DAT is written that this program PLTNET.COM or PLTNET.EXE utilizes. When using the version of USU-NETWK with time dependent solution capabilities on a larger computer like a VAX, then the file PLTNET.DAT can be written by setting the option NETPLT=14. This file will be read by the post processor PLTNET just like the file under the same name that a PC writes with NETPLT=1. However, the larger program designed for larger computers than PC's has the plot routines built into it, and the option NETPLT=1 requests that a plot be made directly by USU-NETWK.

It is also worth noting here that if you wish to have USU-NETWK (either the PC version or the full program) write out a data file that can be subsequently read in as an input file after it has been changed for new conditions, then NETPLT must be given a value of 10.

III. Utilization of Evans & Sutherland 300 Graphics System

This capability of utilizing an Evans & Sutherland 300 graphics system was developed some years ago, and only the full version of USU-NETWK that also has the time dependent solution capabilities still retains this graphics capability. Its use is described in a subsequent section in this appendix, and its use is restricted to installations that have this special hardware.

IV. Three Dimensional Graphics Displays and Contouring of Solutions

This graphics capability allow any of the following types of displays or hard copies of them to be created: 1. A three dimensional perspective view of the hydraulic grade line from the solution superimposed over the ground surface containing the pipes of the network from any viewpoint, 2. A contour type map of the hydraulic grade line surface, the pressure (or pressure head) surface, or just the ground elevation, and 3. A three dimensional perspective view as in # 1 superimposed above a contour type map such as in # 2. This graphics capability is by means of a post processor program PLTNET3. Program PLTNET3 utilizes the general purpose graphics file that USU-NETWK is requested to write when the option NETPLT=3 or NETPLT=4 is in the $SPECIF list. PLTNET3 interfaces with the DISSPLA software package, and therefore can only be utilized at installations where DISSPLA is licensed. Since DISSPLA will do the graphics on any graphics device, including laser jet, Printronix printers, and Calcomp plotters hard copies of this graphics should be obtainable from any graphics devices.

V. Plotting Pressure Profiles, Displaying Pressure Bars, or Contour Mappings

A post processor program PROFILM has three separate graphic capabilities, i.e. 1. Profiles of pressures, or pressure heads can be shown as profiles over the pipe line elevations through selected sequences of pipes within the network. (Up to 8 such profiles can be place on any single
graph and as may graphs made as desirable.) 2. The pressure, pressure head, or hydraulic grade line elevations can be shown as bars at the nodes of a display of the network, or any rectangular portion of the network. 3. A contour type map of the pressure surface, the pressure head surface, or the hydraulic grade line surface can be created.

Several versions of program PROFILM are available. One version of PROFILM is designed to be utilized on a computer that has Calcomp Calls available on it to make a "Calcomp plot" as I above. This program is written in FORTRAN and calls on libraries such as PLOT, SYMBOL, and NUMBER. Another version, which is also in FORTRAN, will do the profiles and bars (capabilities # 1 and # 2) on a PC that contains an IBM compatible graphic card such as a CGA, EGA, or VGA. This version of PROFILM does font mapped lettering, the size of which is fixed by the monitor, to define axes, etc, but its main functions are to show pressure profiles, and the network with bars for pressure at its nodes. Another version of PROFILM is written in PASCAL. This latter program is also designed to run on IBM compatible PCs with graphics cards. It identifies nodes and pipes with their numbers, etc, and gives scales to the graph's axes by lettering the abscissa and ordinate. Options allow the size and spacing of this lettering to be changed. It allows a little more flexibility than the FORTRAN version, permitting selection of different foreground and background colors, etc.

VI. Graphing Time Variations of Pressures, Flow rates, or Water Surface Elevations

This graphics capability is for use in displaying variations in time that are defined from a time-dependent solution that USU-NETWK can be requested to obtain by setting the option ISIML=1 in the $SPECIF list or requesting auxiliary program PRINT (see Appendix E) to write such a special table. The graphics is done by a post processor program PLTTIM, that utilizes the information given in the special tables that will be written by the time dependent version of USU-NETWK if the option PRINTT >0 in the $DATA list of options. Also commands such as PIPE F, PIPE H, NODE P, NODE H and NODE E to program PRINT will write these special tables. Program PLTTIM reads the information directly from these special tables and will display any column (or several columns) that is in one of these special tables on the vertical axis with time as the horizontal axis of the graph.

There are three versions of PLTTIM available. Two are written in FORTRAN. One of these calls on standard Calcomp libraries such as PLOT, SYMBOL and NUMBER to make the graphics. This version of PLTTIM is designed to be used on a computer that either write a file for a plotter, or drives a plotter directly as # I above, and that has a library of Calcomp Plotting routines. The other FORTRAN version is designed just for use on PCs, and places the plot on its monitor, but this version does no lettering on the graph. The third version of PLTTIM is in PASCAL and it is also designed for use on PCs with an IBM compatible graphics card, such as an CGA, EGA or VGA, and the plot of the information from selected columns in the special tables written by USU-NETWK against time as the abscissa are displayed on the PCs monitor. Lettering, etc is included on the graphical display, an option menus can be opened to alter background, and foreground colors, and the characteristics of the plot. This program can also be executed on PCs with Hercules, and other graphics cards. By giving the DOS command GRAPHICS prior to running PLTTIM, the displays on the PC monitor can be send to a dot matrix line printer driven by the PC, or the graphics captured in a file with such utility program as FRIEZE in Microsoft's PAINT BRUSH, or GRAB supplied with WordPerfect 5, and incorporated as part of a WordPerfect document, and then printed to a HP Laserjet printer.

Each of these graphics capabilities are described in the following sections. In reading the remainder of this appendix you need read only that section, or those sections, for which you have the graphics programs, etc. In reading about how to use the graphics it will also be helpful to run that program while reading the section, especially for the graphics packages that are designed to run on PC's.
Calcomp Type Plots

Calcomp type plots consist of displaying the network on a plane surface in which pipes are represented by lines, their numbers can be shown along the side of these lines, and the nodes show arrows outward from them by which the demands can be shown etc. Figure C-1 gives an example of a calcomp plot that gives the following information by the values written on the plot besides the layout of the network (The input data file for this network is given as Table C-1):

1. The pipe numbers within ( ), 2. the diameter and length of the pipes, 3. the node number within the rectangular boxes of the nodes, 4. the ground elevations at the nodes, 5. the demand at the nodes, 6. the location of sources of supply, 7. the pipes that contain the pumps and 8. the water surface elevations in the reservoirs. A menu will be displayed that allows some of these items to be turned off, i.e. not displayed on the plot. Other menu items allow are: The pressure in psi at each node, and The flow rate in each pipe along with an arrow that shows the direction of the flow in this pipe.

Table C-1. Input data file for the Network in Figure C-1.
EXAMPLE NETWORK SUPPLIED BY TWO RESERVOIRS AND A LOWER PRESSURE ZONE
$SPECIF NETPLT=3,NODESP=1,LENGON=0,NUMPIP=1 SEND PIPES 26 1 .005
1 2 6/
1 3 12/
3 4/
4 5/
5 6/
6 7/
27 7 14/
1 8 12/
3 9/
4 10/
5 11/
6 12 10/
7 13/
8 12/
9 10/
10 11/
11 12 10/
12 13/
8 14 /
9 15/
10 18/
11 17 6/
12 18/
13 19 10/
14 15 6/
15 16/
16 17 6/
17 18/
18 19/

Table C-1. Input data file for the Network in Figure C-1.
EXAMPLE NETWORK SUPPLIED BY TWO RESERVOIRS AND A LOWER PRESSURE ZONE
$SPECIF NETPLT=3,NODESP=1,LENGON=0,NUMPIP=1 SEND PIPES 26 1 .005
1 2 6/
1 3 12/
3 4/
4 5/
5 6/
6 7/
27 7 14/
1 8 12/
3 9/
4 10/
5 11/
6 12 10/
7 13/
8 12/
9 10/
10 11/
11 12 10/
12 13/
8 14 /
9 15/
10 18/
11 17 6/
12 18/
13 19 10/
14 15 6/
15 16/
16 17 6/
17 18/
18 19/
14 20 8/
15 21/
16 22 6/
17 23/
18 24/
19 25/
20 21 8/
21 22/
22 23 6/
23 24/
24 25/

Reservoirs
26
27

Valves
25
26
27

Nodal Coordinates
1 1 350 0 4500
2 3 355 -1500 4500
3 5 340 1000 4500
4 3 355 2100 4500
5 4 330 3800 4500
6 6 340 5600 4500
7 8 345 7500 4500
8 1 345 0 3500
9 8 335 1000 3500
10 6 330 2100 3500
11 .2 300 3800 3500
12 5 270 5600 3500
13 .7 250 7500 3500
14 .8 300 0 1800
15 .6 280 1000 1800
16 .5 260 2100 1800
17 .6 250 3800 1800
18 .5 240 5600 1700
19 .5 200 7500 1500
20 .7 270 0 500
21 .4 220 1000 500
22 .3 230 2100 500
23 .5 210 3800 400
24 .7 190 5600 200
25 1.5 180 7500 0
26 0 480 0 6500
27 0 480 7500 6500

Calcomp Routines Incorporated in USU-NETWK

The Calcomp plotting capability is part of the USU-NETWK program when it is on a larger computer such as a VAX. When using such a version of USU-NETWK, and with the option NETPLT= -2, -1, 1 or 2 in the $SPECIF list, a calcomp plot file will be written or may be used directly to drive the plotter, depending upon the computer installation. If NETPLT=1 or -1 then it is necessary to provide the x and y coordinates with the original input data. These x and y coordinates are entered after the nodal elevations when the NODES command is used to enter the data, or after the nodal elevations with data entered through the PIPE- command. When giving these x and y coordinates as part of the original input data, it is also possible, and likely preferable, to set the option LENGON=0,
Figure C-1. Calcomp plot of network whose input data file is in Table C-1.
indicating that pipe lengths will not be given, but that USU-NETWK is to compute these length from nodal coordinate data. If LENGN=1 (the default) and the absolute value of NETPLT equals 1 then the x and y coordinates given in the input data will be used to plot the positions of the nodes of the network, but the given pipe lengths will be used in the solution of the flow rates and pressure throughout the network. When the absolute value of NETPLT equal 1 it is best to also include NODESP=1 (i.e. designate sources of supply as nodes), for otherwise the positions of reservoirs and source pumps are not provided in any of the input data. If NODESP=0 then the plotting of sources of supply will be based on internal code within NETWK, but their locations will not generally agree with their actual locations.

The option NETPLT=2 or NETPLT=-2 must be used if the x and y coordinates of the nodes of the network are not provided with the input entered by either the NODES or the PIPE-command. When the absolute value of NETPLT equals 2, then the x and y coordinates are provided after the RUN or END command. Since the writing of the plot instructions takes place after a solution has been obtained, these coordinates will occur after any other special data that may be called for by other options, but before any changes that are included by the CHANG command. These coordinates also follow time dependent analysis data if the option ISIML=1 in the $SPECIF list.

It is important that these coordinates (as pairs) be given in the correct order. This order consists of the same order in which the real nodes are entered in the original input data. When the NODES command is used exclusively to enter data about the nodes of the network, then the order is the same as the sequential order of these data with any sources of supply being ignored. In other words, should the option NODESP=1, then any source nodes for reservoirs, or source pumps must be skipped in establishing the order of the x and y coordinate pairs. Consequently, it is best when abs(NETPLT)=2 and NODESP=1 to place nodes at sources of supply after the real nodes of the network, when using the NODES command in the original data.

When the command PIPE- is used to enter data, then the internal order of the nodes is established by the order in which a node first appears in the data. Again source nodes are not counted even though NODESP=1.

After the x and y coordinates for the real nodes are given, then the x and y coordinates, as pairs of values, for reservoirs, and then for source pumps must be provided. These sources of supply coordinates are given regardless of whether NODESP=1 or not. The order in which reservoir coordinates must be given corresponds to the order in which reservoirs occur in the original data. Likewise, the x and y coordinates for source pumps follow the reservoir coordinates in the order in which source pumps are given after the PUMPS command. No coordinates are given for booster pumps. Thus whether NODESP=0 or NODESP=1 does not matter if abs(NETPLT)=2 because the coordinates of sources are supplied with the other coordinate data after the RUN or END command.

When NETPLT is negative, i.e. = -1 or -2, then the user is provided an opportunity just before the plot is made to give a message to the Calcomp Plotter Operator. The operation of making Calcomp plots will vary at different installation, and if the installation that you are using the Calcomp plot capabilities at does not provide for such messages to be written to the operator you may call on a non existent library with NETPLT negative. However, at the USU installation, the user will be requested to provide up to a 70 character message that the operator will see before the plot is made. This message may be a request such as: PLEASE USE LIQUID INK, #3 SIZE PEN. Another message may be: THIS PLOT REQUIRES THE 30" WIDE PAPER.

The options PLOTH and NLETTE in the $SPECIF list can also be utilized in the version of USU-NETWK that contains the Calcomp plotting subroutines. The option PLOTH determines the height (i.e. the size in the y direction) of the plot in inches. It is the responsibility of the user to ensure that this plot height is less than or equal to the width of the plot paper being used by one
inch. The extra inch is used at the bottom of the plot for a title if this title is requested. The option NLETTE determines the height of the lettering that will be used on the plot in inches. Some of the lettering will be smaller than this magnitude in order to help distinguish what the numbers represent.

Post Processor OFPLOT

The above section applies strictly only for USU-NETWK on a larger computer such as a VAX that has the entire program that includes the graphics routines as part of the program. However, essentially the same capability is available through the post processor program OFPLOT on a PC. Also an alternative available for calcomp plotting on larger computers is though the post processor program OFPLOT. This plotting program using the general graphics data file written by USU-NETWK when NETPLT = 3 or = 4. The same plotting capabilities are done by this program as described above for the plot routines that are incorporated within USU-NETWK. The difference is that this post processor contains a driver program that reads in the information from the graphics data file. OFPLOT can be used with the complete program of USU-NETWK because the PC version does not contain these necessary plotting routines.

In addition to the version of OFPLOT that is designed to drive a Calcomp plotter or other plotting device, a special version of OFPLOT is available to run under DOS on PCs. This special version of OFPLOT does not allow continuous changes of letter sizes as can be done when using the usual calcomp plot program OFPLOT, but selected sizes are available.

OFPLOT and Internal Plot Routines

The use of the post processor OFPLOT is described in this section. Before executing OFPLOT it is necessary that USU-NETWK be used to solve a network problem, and that in the data for this problem the option NETPLT = 3 exist, which requests that a general purpose graphics file be written. (The information written to this file is described subsequently in this appendix.)

Before the plot is make the following menu will be displayed by OFPLOT on the terminal:

THE PARAMETERS FOR THE PLOT ARE SET AS:

1. HEIGHT = 12.00
   WITH ACCOM. LENGTH = 34.45
2. LETTER SIZE = 0.13
3. SPACING BETWEEN LETTERS = 0.10
4. BASIC UNITS FOR DEMAND = T
5. WRITE ELEV. OF NODES = T
6. WRITE PRESSURES AT NODES = T
7. WRITE FLOWRATES IN PIPES = T
8. WRITE PIPE LENGTH & DIAM = T
9. PROVIDE TITLE TO PLOT = T
10. SHOW DEMANDS = T

IF O.K. GIVE 0, OTHERWISE NO OF ITEM THAT SHOULD BE CHANGED

The descriptions in this menu are followed by the default values, which may be different in your program from those given above. The capital T, after the equal sign, indicates "true" for this item or if an F (for false) follows the equal sign it denotes this item will not be placed on the plot. The second length given in menu item # 1 is the length of the graph corresponding to the given height. Both of these values are in inches. This value will change as the first value in item # 1 is the length of the graph corresponding to the given height. Both of these values are in inches. This value will change as the first value in item # 1 is changes to preserve the same scale in the x and y directions. To change any value for one of these menu items type in the number (integer) of the item followed by the new value when this new value is ask for. Selection of a true or false item toggle it to the opposite.

The above menu will be displayed repeatedly until a 0 (the number zero, not letter o) is entered at the keyboard. Therefore, it is possible to change any item again if a mistake is made the first time it is changed and as many items as desired can be changed.

After USU-NETWK has terminated execution, then system instructions must be provided to have the actual plot made. That is USU-NETWK has now written a "calcomp plot" file that needs to be properly directed to the plotter. At the USU installation using the VAX 8650, the user must provide the system command:
in which NETWKPL.PLT is the name of the disk file on which the instructions to make the calcomp plot have been written.

Figure C-2 shows a small network that has been produced by setting NETPLT=-1. The Calcomp plotter operator was requested to "use liquid ink and # 3 pen." Also, the height of the graph was specified as 9 inches, with a letter height of 0.1 inches. The size of the actual plot has been reduced in size for its inclusion in this manual. The options were turned on for "WRITE ELEV. OF NODES", and "WRITE PIPE LENGTH & DIAS". The input data for this solution are given in Table C.2, instead of the plotted network.

Identically the same plot could be produced by OFPLOT. The difference would be that the option NETPLT=3 in the input to USU, and then OFPLOT would read the information in the file that would be written and make the plot shown in Figure C-2.

Table C.2. Data input to USU-NETWK that is plotted in Figure C-2.

```
EXAMPLE NETWORK TO ILLUSTRATE PLOTTING ON PC'S
/*
SSSPECIF PEAKF=1.3,NFLOW=0,WPGM=0,NODESF=1,
NETPLT=1,LENGON=0 SEND
PIPES
1 2 21 12 .005
2 21 1 12/
3 1 2 10/
4 1 3 /
5 3 4 /
6 4 23/
7 2 5 8 /
8 6 5 /
9 2 7 /
10 8 3 10 /
11 10 8 /
12 8 9 6 /
13 11 6 8 /
14 7 6 /
15 10 7 10 /
16 13 10 /
17 13 14 /
18 14 9 6 /
19 17 11 10 /
20 15 11 /
21 12 7 /
22 15 12 /
23 18 15 12 /
24 18 17 10 /
25 16 12 12 /
26 19 18 /
27 16 13 /
28 19 16 /
29 20 14 /
30 19 20 /
31 24 19 20 /
RESER
22 500
23 400
PUMPS
24 10 150 20 120 30 40 450
RUN
```

In making plots of large networks, it is necessary that large size paper is used and that the plot height be large enough and letter size be small enough so that the lettering that gives pipe numbers, node numbers, pipe diameters, pipe length, flowrates, etc., do not overwrite each other. Should the plot become too cluttered despite using large plot heights and/or small letter heights, it is possible to not display all of the information on the same plot by turning off some of the items that are written by using the above menu. If necessary, two or more separate plots might be created, the first consisting of the network's physical features, and the others consisting of the solution parameters such as flow rates, and nodal pressures. An advantage of using the off line plot program OFPLOT is that several separate plots of the same solution can be made. Changing the pitch of the lettering, e.g. the spacing between the letter in relationship to the letter height, allows additional flexibility to make the lettering readable. The standard spacing between characters is 0.8 times the height.

Below is input data for a small network.

In the SSSPECIF list of options the inclusion of NETPLT=3 tells USU-NETWK that the general purpose graphics file is to be created.
Figure C-2. Plot of network whose input data file is in Table C-2.
The general purpose graphics file that USU-NETWK has written for this small network consists of that given in Table C-4.
This data file consists of the following information and is written with the formats given below:

1. **Information in graphics file when NETPLT=3**
   1. The x and y coordinates of the nodes of the network given as pairs. This list of pairs will consist of the number of junctions in the network, and if NODESP=0 these will be actual junctions; however if NODESP=1 then the source pumps and reservoirs, respectively will have their coordinates included in the list in the order in which they are given in the input data. These coordinates will be written to the file in the case of the steady state version of the program only if coordinates are provided with the input data, i.e. the option LENGN=0 is given in the $SPECIF list such that x and y coordinates are provided for the nodes rather than the lengths of pipes. This list of pairs of coordinates is terminated with two 999 999 or with a / depending upon which version of USU-NETWK you are using. The / is intended for use with FORTRAN programs so that the list directed READ terminates without knowing in advance the number of pairs in the list. If you are using a PASCAL based post processor the 999's are used to terminate the read, and if a / exists in the file it must be changed to 999's. Likewise, when using a post processor written in FORTRAN and two 999's occur in the file they must be changed to a / (or a / placed in front of the 999's) with an editor before using a post processor whose source was written in FORTRAN. If coordinates are not given in the input data, and therefore are not written to this file, then the coordinates must be added to this file with an editor before use of a graphics post processor program. FORMAT(10F8.1) as default, but if coordinates are small, or large the .1 may be .2 or .0, respectively.

2. The next input will be written only if the option NODESP=1 is used. It is also terminated with a / or 999's and contains the internal node numbers (i.e. the order of the occurrence of nodes in the input data) for the sources of supply (i.e. the RESERVoirs and source PUMPS). The internal node numbers for the reservoirs are listed first followed by the source pumps. (The external node numbers corresponding to these internal numbers are given later in the information about reservoir, and pumps.)

3. This line contains a list of integer values that give the number of components in the network. These variable are:
   a-NP the number of pipes in the network,
   b-NJ the number of real junctions in the network,
   c-NRES the number of reservoirs plus nozzles in the network,
   d-NPUMP the number of source pumps in the network,
   e-NPUMP the total number of pumps (source pumps plus booster pumps),
   f-NBPUMP the number of booster pumps in the network,
   g-NVALVE the number of pressure reduction valves in the network,
   h-NBPV the number of back pressure valves in the network,
   i-NRESI the number of reservoirs in the network,
   j-NOZZLE the number of nozzles in the network,
   k-NML the number of minor loss devices in the network,
   l-NODESP the value of this option, a 0 or a 1,
   m-NCKVAL the number of check valves in the network,
   n-ISIML the option for a simulation solution which will be 0 unless the file has been written from a time increment of a time dependent solution in which case this will be a 1.
   o-NUNIT the option that denotes the units used in defining the network. FORMAT(15I5)

4. The external pipe numbers listed in the order in which they are given in the input data file, NOP(I). FORMAT(2415)

5. The external node numbers listed in the order in which they are given in the input data file, NODS(I). FORMAT(2415)

6. The values of the pointer array that
7. The internal pipe numbers that join at the separate nodes of the network, JN(J). The above pointer array NN(I) separates this one dimensional array into individual nodes listed in the order in which the node data occur in the input data file, i.e. according to internal node numbers. If the flow was specified into a node, then this internal pipe number will be negative. FORMAT(24IS)

8. The upstream and downstream node numbers as pairs of integers for the pipes of the network listed in the order in which the pipes are given in the input data, i.e. the internal pipe numbers, L1(I) and L2(I). If a source exist at one end of the pipe this node will be given as 0 even if NODESP=1 in the $SPECIF list of options. From the information given in #2 above the source nodes may replace these 0's if desired, provided the option NODESP=1 is used. However, for plotting the 0's provides a convenient means for identifying this has a source node at the end of a pipe. FORMAT(24IS)

9. The following data for each pump in the network:
   a- IPUMP(I), The internal pipe number that contains the pump,
   b- IPIUMP(I), The external node number given the pump if NODESP=1, otherwise a 0,
   c- ELEP(I), The elevation of the source pumps water surface elevation, or a zero for booster pumps,
   d- EIPUMP(I), The ground elevation for the pump given as part of the NODES data for the node assigned to the pump when the option NODESP=1 is used, otherwise a zero. FORMAT(4(2IS,2F10.2))

10. A list of heads produced by the pumps with source pumps listed first and booster pumps thereafter, and within each of these listing the order is according to the order in which the pumps are given in the input data. FORMAT(12F10.2)

11. For each pressure reduction valve, PRV, the following four items:
   a- IVPIP, The internal number of the pipe that contains this PRV,
   b- HGLVAL, The HGL elevation setting for this PRV,
   c- DVALVE, The distance of this PRV from its upstream node, and
   d- DHVAL, The drop in pressure head across the PRV. FORMAT(15,3F10.4)

12. For each reservoir in the network the following four items:
   a- IRES, The internal number of the pipe that connect the reservoir to the network,
   b- IIRESI, The external node number that the reservoir has been assigned if the option NODESP=1 is given, otherwise a zero.
   c- ELE, The water surface elevation of the reservoir, and
   d- EIRES, The ground elevation given for the node assigned to the reservoir under the NODES command when the option NODESP=1 is used, otherwise a zero. FORMAT(4(2IS,2F10.2))

13. A listing of the elevations of the nodes given under the NODES command for each node, or junction of the network in the order in which the nodes are given in the input data, ELEV(I). When the command PIPE- is used the order is established in the order in which a node number first appears in the input data. Elevations for nodes assigned to sources (source pumps and reservoirs) when using the option NODESP=1 are omitted from this listing. FORMAT(12F10.2)

14. A listing of the pressure heads from the solution at the nodes in the order in which the nodes are given in the input data. When using ES units, these values will be in feet, and when using SI units these values will be in meters, AR(I). FORMAT(12F10.2)
15. A listing of the demands at the nodes of the network in the order in which the nodes are entered in the input data, \( Q(I) \). These values will be given in basic units, e.g. cfs when using ES units and \( \text{m}^3/\text{s} \) when using SI units, unless weight or mass flow rates are used. FORMAT(12F10.3)

16. A listing of the pipe diameters of the network in the order in which the pipes are entered in the input data, \( D(I) \). These values will be given in basic units, e.g. ft when using ES units, and meter when using SI units. FORMAT(12F10.3)

17. A listing of the pipe lengths of the network in the order in which the pipes are entered in the input data, \( L(I) \). FORMAT(12F10.2)

18. A listing of the pipe roughness coefficients of the network in the order in which the pipes are entered in the input data, \( E(I) \). When using the Darcy-Weisbach equation these values will be the relative roughness of the pipe \( e/D \), and when using the Hazen-Williams equation these values with the Hazen-Williams coefficient. FORMAT(12F10.2)

19. A listing of the flow rates in the pipes as determined by the solution given in the order in which the pipes are listed in the input data, \( Q(I) \). These values will be in basic units, e.g. cfs when using ES units, and \( \text{m} / \text{s} \) when using SI units unless weight are mass flow rates are used in the input data by setting the option NFLOW>4. FORMAT(12F10.3)

20. A listing of the internal pipe numbers that contain minor loss devices in the order in which these are entered in the original input data, \( IML(I) \). FORMAT(24I5)

21. A listing of the internal pipe number that contain check valves in the order in which these are entered in the original input data, \( ICKVAL(I) \). FORMAT(24I5)

22. The conversion factor for changing the demands into basic units from the given units. For example if gpm are given then this is \( \text{CONVF} = 448.83 \).

23. If a time-dependent solution is being obtained the time in seconds for which this solution applies.

The following data in this file is generally not needed

24. Three integer values that consist of: (a) the number of loops in the network, (b) the number of these which are pseudo loops, and (c) the number of these which are real loops.

25. A pointer to the next array that contains the loop information. This pointer array \( MN(I) \) separates the loop data in the following LP(J) one dimensional array. The pointer of \( MN(I+1) \) points to the end of data for loop 1.

26. The internal pipe numbers that defines the loops of the network. These values are in a one dimensional array LP(J), and the previous pointer array MK separate the individual loops.

27. A pointer array MK(I) that separate the next array.

28. A one dimensional array IK that defines the corrective flow rates circulating through each pipe of the network. The previous array MK separates this list into individual pipes.

Graphics file obtained with option \( \text{NETPLT}=1 \)

An alternative graphics file can be obtained by setting \( \text{NETPLT}=1 \) rather than \( =3 \), using the PC version of USU-NETWK. This is the graphics file that the screen post processor for PCs, PLTNET, uses. The name of this file will be PLTNET.DAT. A listing of this graphics file is given in Table C-5 for the same input data file as that used for the above general purpose graphics file, the only difference is the the option \( \text{NETPLT} \) was changed to equal 1. Depending upon the special graphics you are developing, you may find this file more convenient to use then the one created with \( \text{NETPLT}=3 \).
Table C-5. Listing of file PLTNET.DAT obtained with NETPLT=1 and using input file, Table C-2.

<table>
<thead>
<tr>
<th>NODESP</th>
<th>SOURCES</th>
<th>RESERVOIRS</th>
<th>BOOSTER</th>
<th>MINOR</th>
<th>NODESP</th>
<th>SOURCES</th>
<th>RESERVOIRS</th>
<th>BOOSTER</th>
<th>MINOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>34</td>
<td>24</td>
<td>16</td>
<td>12</td>
<td>8</td>
<td>19</td>
<td>16</td>
<td>12</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>2300</td>
<td>2500</td>
<td>3000</td>
<td>3000</td>
<td>3000</td>
<td>5500</td>
<td>5500</td>
<td>5500</td>
<td>5500</td>
<td>5500</td>
</tr>
<tr>
<td>5000</td>
<td>10300</td>
<td>1500</td>
<td>8000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The information contained in the graphics file that post processor PLTNET utilizes consists of the following:

1. A list of integers defining the components of the network, using the FORMAT(16I5). These are:
   (a) The number of pipes,
   (b) The number of real nodes, (If sources are numbered using the option NODESP=1, then this number written to this file will be smaller by the number of sources of supply than the number of nodes reported by USU-NETWK. To get the latter values the number of source pumps and reservoirs (given later in this list) must be added to this value.)
   (c) The number of source pumps,
   (d) The number of booster pumps,
   (e) The number of reservoirs, not including reservoirs for source pumps,
   (f) The number of minor loss devices,
   (g) The value of the option NODESP.
   (This will be a 1 if NODESP=1 in the $SPECIF list of options, and will be 0 otherwise.)
   (h) The option for the type of equation used for defining the head loss flow rate relationship; i.e. a 0 if the Darcy-Weisbach equation was used, a 1 if the Hazen-Williams equation was used, a 2 if the Hazen-Williams equation was used with a 1 in the $SPECIF.}

equation was used, and a value larger than 1 if the
Mannings equation was used.

2. A list of the x-coordinates of the nodes
of the network. This list of x values will be in the
same order as the nodes are entered in the
original input data, i.e. the internal order. FORMAT(16I5).

3. A list of the y-coordinates of the nodes
of the network corresponding to the x coordinates
given in the previous list. The number of values
in this list equals the number of nodes, i.e. the
value in Ib above if NODESP=0 and the sum of
the number of nodes plus sources of supply if
NODESP=1, or the sum of 1b + 1c + 1e.
PLTNET assumes that the x- and y- coordinates
for the sources follow the real nodes if
NODESP=1 is used. This requires that in
providing the input to USU-NETWK that you
include source nodes following the real nodes
under the
NODES
command, or if the PIPE-
command is used that pipes to sources follow
other pipes of the network. Furthermore, the
coordinates for source pumps precede coordinates
of reservoirs, and the original input must reflect
this order also. FORMAT(I0F8.1)

3a. If the option NODESP=1 is given
then a list of internal node number for sources of
supply are given, with reservoirs listed first and
source pumps thereafter. This list is needed to
associate x and y coordinates with sources of
supply. It is good practice to list nodes for
sources (if they are numbered using the option
NODESP=1) after the real nodes under the
NODES command so that these coordinates will
come at the end of the list under # 3.

4. The list of external pipe numbers listed
in the order in which the pipes were included in
the original input data, i.e. the internal order of
pipes. FORMAT(I0F8.1)

5. The list of internal upstream node
numbers corresponding to the pipes listed in # 4.
To get the external node number used the
information listed under #7 below. This list will
contain a zero if a source of supply is attached to
the pipe regardless of whether NODESP=1, or 0.
FORMAT(16I5).

6. The list of internal downstream node
numbers corresponding to the pipes listed in # 4.
FORMAT(16I5).

7. This list consist of node data and will
contain the following values in each line of the
list: (FORMAT(15,14,F8.3,12I5).
(a) The external node number. The
internal node number is established by the order
of this list.
(b) The number of pipes that join at this
node.
(c) The demand in basic units at this
node. By basic units is meant that if ES units are
used these values will be in cfs, and if SI units are
used these values will be in cubic meters per
second.
(d) A list of internal pipe numbers that
join at this node. If the assumed direction of
flow, as given in the input data, is into this node
this number will be negative.

8. A list of the flow rates in each pipe of
the network as determined by the solution for
which this file applies. FORMAT(I0F8.3).

9. A list of pipe diameters in the units
used in the input data. FORMAT(I0F8.3).

10. A list of pipe lengths in basic units.
These are the hydraulic lengths of pipe, and if
minor losses are included then the list will consist
of the actual pipe length plus the equivalent
length of pipe for the minor loss. FORMAT(I0F8.1).

11. A list of pipe roughness coefficients.
The format will be FORMAT(I0F8.6) if the
Darcy-Weisbach equation is used and
FORMAT(I0F8.1) if the Hazen-Williams equation
is used.

12. A list of elevations of the nodes of
the network according to the order that real nodes
are given in the original input data. FORMAT(I0F8.2).
13. A list of the pressure heads obtained as the solution at the nodes of the network. FORMAT(10F8.2).

14. A list of reservoir information consisting of the following for each reservoir in the order they were given under the RESER command. FORMAT(4(2I5,FB.2)).
(a) The internal pipe number that connects the reservoir to the network.
(b) The node number assigned to this reservoir with NODESP=1; otherwise a zero.
(c) The water surface elevation of the reservoir.

15. A list of information for the pumps in the network. Source pumps occur first in this list according to the order they are entered in the input under the PUMPS command, and booster pumps follow source pumps again in the order of their entry under the command BPUMP or BOOSTER. FORMAT(215,5F10.3)
(a) The internal pipe number that contains the pump.
(b) The node number assigned to this source pump if NODESP=1; otherwise a zero.
(c) This value and the next two values are the coefficients of a 2nd degree polynomial that define the pump curve, i.e. the head the pump produces as a function of the flow rate in basic units, i.e. \( H = aQ^2 + bQ + c \). This value is the coefficient \( a \) that multiplies the flow rate squared.
(d) The coefficient \( b \) in the above equation.
(e) The coefficient \( c \) in the above equation. If more than 3 pairs of values are used in defining a pump's characteristic curve, then the coefficients \( a, b \) and \( c \) will be those needed to give a 2nd degree polynomial fit through the 3 points that bracket the flow rate passing through the pump as determined by the solution.
(f) The water surface elevation for source pumps, and a zero for booster pumps.
(g) The normal capacity of the pump. When 3 or more pairs of \( Q \) versus \( H \) values are used in the input to define the pump curve this will be that given normal capacity converted to basic flow rate units of cfs if ES units are used and \( m^3/s \) if SI units are used.

16. A list of internal pipe numbers, and loss coefficients for minor loss devices in the network. FORMAT(6(I5,F8.2)).

Graphics Program PROFILM

As described above the auxiliary, or post processor program PROFILM is designed to help with understanding the output from the solutions to USU-NETWK by: 1. providing pressure profiles through selected sequences of pipes, 2. displaying pressure bars at the nodes of the network, or 3. displaying a contour type map of pressures. PROFILM utilizes the general purpose graphics file that USU-NETWK is requested to write by inclusion of the option NETPLT=3 in the $SPECIF list, and which is described above under the use of the off line Calcomp Plot program OFPLOT. This is the same graphics file that program PLTNET3 utilizes. Both the steady-state version of USU-NETWK and the version that also allows time-dependent solutions can write this special file, and if not given a different name its name is PLOT10.DAT.

There are 3 versions, or variations of program PROFILM, depending up the computer using it. One version is designed to drive a graphics device such as a Calcomp plotter. This version assumes that the computer that it is being used on has standard library calls such as PLOT, NUMBER, SYMBOL, etc. that provide the proper instructions to move the plouing pen etc. on a graphics device such as a Calcomp plotter. This program is in FORTRAN.

The other two versions are designed more for use on PC's, the first of these is essentially the same as the first version mentioned above, with the exception that the calls to PLOT, NUMBER, etc. have been changed to produce the plotting, etc on the monitor of the PC. This version should execute properly on IBM compatible PC's, under DOS that have IBM type graphics cards,
and compatible monitors. It is also in FORTRAN. The third version is written in PASCAL and attempts to determine what the PC's configuration consists of, and then utilizes drivers appropriate to the graphics adaptor that the PC has. It contains an options' menu similar to that in the post processor PLTNET.EXE and one version of PLTTIM (described below), that allow the size of the lettering, size of the plot, and colors to be used for background and foreground, etc. to be changed.

After giving the command to execute PROFILM, you will first be asked: GIVE FILENAME WITH DATA (cr if PLOT10.DAT). In response to this you need only press the Enter key, if the default name contains the data for general purpose graphics, or type in the name of the file that does contain this data. (The third version of PROFILM allows you to change options prior to this.) Next you will see the following displayed:

SELECT THE NO. FOR THE TYPE OF PLOT YOU DESIRE
1 - Head Profiles over selected sequences of pipes,
2 - Pressure bars at nodes of network, or
3 - Contour map of heads or pressures.
You should type in 1, 2 or 3 to indicate what you want done.

Below a description of what occurs if you select 1, 2 or 3 is given.

1. Profiles: Profiles show the elevation of the pipe, and the HGL-elevation as obtain for the selected solution above these pipes. When you select to plot profiles you will be asked to indicate how many profiles you want to have displayed on this graph. Up to 8 such profiles can be placed on any graph, and it is possible to create any number of such graphs, and therefore as many such profiles can be created as you desire. Next, you will be asked to supply a sequence of connected pipes through which each of these profiles is to be plotted. Therefore, be sure that you are familiar with the connectivity of the pipes so that you can give these sequences of pipes. If consecutive pipe numbers that you give are not connected you will be informed of this and must supply a correct pipe number or abort execution.

If several profiles are placed on a graph, then PROFILM will decide on the vertical axis scales to use, such that the tick marks for elevations are some appropriate integer, and the profiles won't be of the same vertical height generally. The summation of the lengths of pipes in the longest profile on the graph will determine the horizontal scale used, and since this same horizontal scale will be used for all profiles on this graph, shorter profiles will extend only partially through the graph. Therefore, it is generally best to have profiles of approximately the same length plotted on a given graph. For the network given in Figure C-1 (and whose input data is given in Table C-1) Figure C-3 shows a graph with 4 pressure profiles plotted. Note on this graph that the top profile, and the second profile from the bottom are through the same sequence of pipes but in opposite directions. Thus the pressure drop through the pressure reduction value in pipe 25 is a rise in the HGL-elevation from left to right in the top profile but a drop in the second from the bottom profile. Figure C-4 shows two profiles from the network in Figure C-2 (whose input data is in Table C-2).

2. Pressure Bars: If you select 2 above then a display of the network's layout or rectangular portion thereof will be displayed with bars showing the pressure at each node within this display. When you select this option you will first be asked: SHOULD ENTIRE NETWORK BE PLOTTED? (Y or N). If you type N for no, the x and y coordinates at the lower left hand and upper right hand extremes of the network will be displayed, i.e. the minimum and maximum x and y coordinates will be given. You will be asked to supply 4 values, namely the x and y coordinates of the lower left corner, and upper right corner of the rectangle of the network that should be displayed. Next you will be asked to respond to the following menu that will be repeated with the new values you give until you indicate that you do not want to see it again.

GIVE THE FOLLOWING:
   1 - Type of bar =
       (1) pressure (psi or Kpa)
Figure C-3. Four profiles plotted using a Calcomp plotter from the network shown in Figure C-1 (with input data in Table C-1). This plot was created with program PROFILM.
Figure C-4. Two profiles plotted through consecutively connected pipe from Figure C-2.
A default value will follow each equal sign. Only with the FORTRAN programs can you only type in a / to accept all of these values supplied under items 1, 2 etc. In the case of the PASCAL program you must type in all requested values even if they are the same as the default values given. The scale factor (item # 2), to use for height of bars, is the multiplier of the quantity being plotted from item # 1 to establish the height of the bars. If this factor is very small the bars will be short, but if this factor is too large then the bars will extended above the top of the graph and will not be fully displayed or the size of the network will be small. For example, if a graph is to be plotted on a Calcomp plotter, and if pressure is selected for the bar, and the pressure at a node is 40 psi, then a scale factor of .02 will result in this bar being 40 x .02 = 0.80 inches high; a scale factor of 0.01 will cause this bar to be 0.40 inches high, etc. Similar results occur on the screen of the PC, but the results will not be in inches.

If you have a graph with bars that are either too large or too small, then either increase or decrease the scaling factor and create another display, or plot. After a little experience with the graphics adaptor that you are using you will gain a feeling for what appropriate scale factors are.

Item # 3, the width of the bars, is their horizontal dimension in inches, if sent to a Calcomp plotter, and again a measure to their width if displayed on a PC’s monitor. All bars will be this same width. The grid spacing in bars, item # 4, represents the distance in inches between consecutive 30 degree diagonal lines that will be used in filling the bars, when the plot is produced by a Calcomp plotter. For the PASCAL program the bars are filled in and this item does not appear. For very small values of this distance, in the order of the pen widths, the bars become solid, but also more plotting effort and time is needed in making the bars. Item # 5 allows for plotting of problems that use SI units rather than ES units.

Figure C-5 shows the pressure bars from the solution to the network in Figure C-1.

3. Contour maps: A real contour map consist a lines over an area that represent a constant elevation. Individuals experienced in reading contour maps can get a quit impression of area changes in elevation and wish to utilize this same type of plot to get impressions of how pressure, and/or HGL elevations change over the area served by a network. The contouring capability of program PROFILM, that allows contour type maps of pressure or HGL elevations, is intended to cater to this need. It is important to recognize, however, that the manner in which the HGL elevation, or the pressure, changes over the area serviced by a pipe network is not the same as the continuous changes that may take place in elevation of the ground surface. Rather the piping system supplies pressures only along the actual pipe lines and not between the pipes. Therefore, a contour type map of the HGL elevation (or pressure) has meaning only along pipes. The pressure will vary linearly along any pipe line, except where devices such as booster pumps, or valves, etc. exist where the HGL elevation will be a step function. To make the user aware of this difference the contour of HGL elevation, or pressure produced by program PROFILM are shown as straight lines in the spaces between the pipes. These straight lines will not be smooth as they cross pipe lines. In reading these contour type maps you must recognize that the values are only meaningful along the pipe lines themselves, and only if additional pipes lines are included as part of the network do the values have meaning in the spaces between the lines. However, the addition of more pipes will alter the solution even if the demands are keep the same, and the sources of supply do not change.

When you select to have PROFILM create a contour type map, you will first be ask to give what is to be contoured. Next you will be given the range of the item that you selected to have contoured given, and you will be ask to select the smallest value for which contours are to
PRESSURE BARS AT NODES OF SAMPLE NETWORK

Figure C-5. Pressure bars displayed at nodes of the Network of Figure C-1 by program PROFILM.
Figure C-6.
Contour type map displaying the Hydraulic Grade Line elevations from the solution obtained for the network displayed in Figure C-2 (with input data in Table C-2.)
be drawn, the number of contours to be drawn, and the interval between these contours. For the PC version of PROFILM written in PASCAL you will also be requested to indicate the frequency of heavier contour lines. In addition to the display of contour lines the pipe lines of the network will be displayed, but will not be numbered unless you select the option to have these numbers given. An alternative is to have the node numbers of the network displayed on the contour type map.

Figure C-6 shows the contours of the HGL elevations plotted.

In using the PC version of PROFILM you can change color, line widths, etc., as options in the same manner as described below under the graphics program PLTTIM.

**Use of PLTTIM to Make Graphs of Solution Parameter v.s. Time**

The graphics program PLTTIM shows variations of nodal pressures, pipe flow rates, water surface elevations in tanks, etc. as functions of time, and as such is only useful in connection with understanding solutions obtained from the version of USU-NETWK that can also solve time dependent problems. There are three versions of PLTTIM available, similar to those of PROFILM. Two versions are in FORTRAN; one of which is designed to write a "Calcomp" type plot file, or whether it is the PC version written in PASCAL that is displaying the result on the monitor. The PC version first asks if you wish to change any options. If you respond with Y (for yes) then the menu shown below will be displayed on the monitor.

```
Give No. of option you wish to change.
0 - Options all O.K.
1 - Background Color
2 - Text & Line Color
3 - Character Size
4 - Text Style
5 - Line Thickness
6 - Identify Curves
7 - Size of Plot
Give 0,1,2,3,4,5,6 or 7
```

Options 1 and 2 from this menu allow up to 16 different colors to be selected (assuming that your graphics card will allow this). When you type either the number 1 or 2 the following box will appear on the monitor:

```
Select from available colors
0 = Black
1 = Blue
2 = Green
3 = Cyan
4 = Red
```
After making the change to the color you desire for either the background (#1) or the foreground (#2), you will notice that the cursor returns to the end of the options menu waiting for you to type in another number for an additional option, or 0 to indicate that you do not wish to change any other options. If you select #3 the following will appear at a different location on the screen, and in a different color if you have a color monitor:

Give the following 4 integers
1. Width Numerator, 2. Width Denominator
3. Height Numerator, 4. Height Denominator
Default Values are: 1 1 1 1

The meaning of these integer values are that their ratios determine the width and heights of the characters that will be displayed. For example giving 5 4 5 4 would increase the size of the letters modestly, but keep their spacing in the same proportion.

Selection of # 4 is an alternative to changing the size of the letters as well as the style of these letters. Upon selection of # 4 you will see the following appear on the screen:

Give No. for Text Style= 2
0 - normal
1 - triple size
2 - small
3 - gothic

Give CharSize factor(1-10)
CharSize= 4

The "small" are vector drawn characters and are the default (i.e. Text Style= 2), these can be made smaller, or larger by changing the CharSize factor less than or greater than 4 respectively. (Also menu item # 3 allows control over character size.)

Selection of option # 7 from the above menu will display the following:

Full screen is used
Give No. of x pos. & No. of y pos.

The "No. of pos." are the number of pixels in the graphics mode of the monitor. A CGA graphics adaptor has 320 X 200 in the x and y directions, respectively. A EGA board allows 640 X 350 and a VGA board allows 640 X 480 pixels. After you have indicated the size of the screen to be used, then instead of full screen the size you used the last time will be displayed for both the "No. of x positions" and the "No. of y positions." These sizes will depend upon the monitor you are using. If you give values larger than those allowed by your graphics card, then the full screen size will be utilized.

When you have selected 0 to finish changing options, then you will be ask for a file that should be used for input to PLTTIM. This must be a file that contains appropriate data such as...
Figure C-7. Plot obtained from an IBM-XT with CGA graphics by printing the monitor's display produced by PLTTIM on an EPSON dot matrix printer.
Figure C-8. Time dependent plot from PLTTIM from a Calcomp Plotter.
as written by USU-NETWK with the option PRINTT greater than one, or written by program PRINT. Depending upon what this file contains you will see something similar to the following:

THIS FILE CONTAINS PRESSURES AT NODE NO 7 FOR TIMES
0.00 2.00 4.00 6.00 8.00 10.00 12.00
14.00 16.00 18.00 20.00 22.00 24.00
Give NODE NO that you want plotted against time up to 7 No.s allowed. Term. with 999

For example if you wanted all 7 columns of data plotted on the same graph you could type in the following in response to this:
1 2 3 4 5 6 7 999
The order of giving the numbers determines the order in which the curves will be plotted on the graph, and their symbols will be different.

Below in Figures C-7 and C-8 you see the graphs as they have been obtained by different methods. The one obtained from an IBM XT PC with only a CGA graphics card was send to an Epson printer by pressing the <Shift>PrtScr key. (The command GRAPHICS was given to DOS prior to executing PLTTIM.) The other graph was obtained from a Calcomp plotter.

Another possibility is to have the graphs printed with an HP-Laser Printer. For example the screen image might be captured by WordPerfect using the GRAB utility supplied with WP version 5, and then incorporated as part of the WordPerfect document using the Graphics key.

Use of PLTNET3 to make Three-Dimensional graphics displays of Network and their Solutions

This section describes the types of graphics available through the use of the auxiliary program PLTNET3. Program PLTNET3 utilizes the information in the general purpose graphics file, that USU-NETWK can be instructed to write. Unless given another name when prompted by USU-NETWK this file will be given the default name PLOT10.DAT. PLTNET3 interfaces with the DISSPLA software package to create any of the following displays:

1) A contour type map of: (a) the elevation of the HGL, (b) the pressure head, (c) the pressure (psi when ES units are used and KPa when SI units are used) or (d) the ground elevation,

2) A three-dimensional perspective view of the network and its solution from any desired viewpoint, and

3) A three-dimensional perspective view of the networks HGL elevation, or pressure head, etc. superimposed on "bed posts" over a contour map of the same quantity. In order to use program PLTNET3, the computer on which PLTNET3 runs must contain the DISPPLA software package.

Upon execution of program PLTNET3 you will be ask:

Type in No. for GRAPHICS device:
1 - CALCOMP 2 - CITOH2 3 - CITOH4
4 - GIGI 5 - HP7475 6 - IDystems
7 - KERMIT-TEK 8 - PRINTRONIX 9 - RAMTEK
10 - RETROGRAPH 11 - SELANAR 12 - TEK4010
13 - TEK4014 14 - TEK4510 15 - TEK4662
16 - TEK41 17 - VT240 18 - 4662
19 - LASERJET

(This list of devices may change with time & installation.)

Since DISSPLA supports any graphics device including printers such as Laserjet and Printronix printer, the results can be send to any device that accepts graphics.

Next you are ask to:

GIVE FILENAME WITH DATA. ( cr if PLOT10.DAT)

The response to this prompt can be a press to the "Enter" key or typing PLOT10.DAT if you have not assigned another name to the general purpose graphics file that USU-NETWK will write with the option NETPLT=3. Otherwise type in the file name that contains the general purpose graphics information written by USU-NETWK. The information contained in this general purpose graphics file is described earlier in this Appendix. Next you will be ask:

HOW LARGE SHOULD PLOT BE X-DIST, & Y-DIST
The x and y distances request here are in inches if a calcomp plot is requested and might vary slightly from this for other graphics devices. The actual size of the plot will be larger than these dimensions by the title and labeling on both the x and y axes. The next requested information is:

GIVE NO. OF WHAT YOU WANT PLOTTED:
1 - HGL Elev.
2 - Pressure Head
3 - Pressure in psi (or -3 if kPa)
4 - The ground elevation
GIVE 1,2,3 or 4 (or 0 to STOP)

The next menu allow for you to select from the three different types of displays that can be created. This menu consists of:

Should this be:
1. a 3-Dim. display,
2. a Contour plot? or
3. a 3-Dim. display with a contour map underneath

In response this this question you type in the number 1, 2 or 3.

To assist you in deciding an appropriate viewpoint for the display PLTNET3 next give a message such as:

THE SIZE OF 3-D WORKBOX IS 4600 4000 4300
A DEFAULT VIEWPOINT IS AT -9200 -4800 13800
DO YOU WANT TO GIVE A DIFFERENT VIEWPOINT?

If you want to accept this viewpoint which is from behind in both the x and y directions, and well above the network at a distance of 13800 ft in the z-direction in the above example you would respond with a N for no. A Y response allows you to determine the viewpoint by having you provide the x y and z coordinates from which the viewer looks at the network and its solution which will be displayed. If the network is large in area size, then these coordinates need to be relative large also. If the network extended over a relative small are, then these coordinates need to be smaller also.

Next you will be ask:

THE HGL SURFACE HAS BEEN PLACE OVER

At this time the information has already been written to the plotter file that is needed to generate the requested perspective view. If the response to the above question is N, for no, then the closure to the plot will be written, and execution terminated. A response of Y, for yes, will result in similar information being generated to display the ground surface and in this perspective view the pipes will be shown as lines. This ground perspective will be drawn with dashed lines so it can be distinguished from the previous perspective display.

Figures C-9, C-10 and C-11 illustrate the three different types of plot that can be produced by PLTNET3. The information to plot or print will be written to a disk file such as CALCMP.PLT, FOR032.DAT, LASER.PLT, etc. (which may be a different name depending upon the installation). If the selection above is the PRINTRONIX printer, then the file is FOR032.DAT, and the file will be CALCMP.PLT if the selection above is CALCOMP, etc. This output file must now be sent to the appropriate device for plotting of displaying. This output file might now be deleted, if desired. It is possible to reuse the general purpose graphics file written by USU-NETWK any number of times using different viewpoints and/or sizes for the plot or different plot devices, until the desired results are obtained. For example if you are connected to a host computer, that has DISSPLA software available, by means of a PC running KERMIT you can select KERMIT (No. 7) as the graphics device. This will display the results on the PC's monitor. After adjusting the viewpoint, etc. until the desired plot appears, you could then repeat the plot, but for this final time select a hard copy graphics device, such as the Calcomp Plotter, or a Laserjet printer. After a little experience, a user generally has a good idea about the viewpoint position he wishes to use.

C-28
Figure C-9. Perspective view of the HGL-elevation from the solution to a network. This is the first type of graphics that PLTNET3 can produce. The results were obtained from a Laserjet printer.
Figure C-10. Contour map of the HGL-elevation. This is the second type of graphics that PLTNET3 can produce.
Figure C-11. Perspective view of HGL-elevation superimposed above the contours of this surface in a horizontal plane that also shows the pipes of the network. This plot was obtained from PLTNET3 using the third type of graphics, e.g. "a 3-Dim. display with a contour map underneath".
APPENDIX D - IMPROVING DESIGN

(SETTING OPTION DESIGN=2)

Assistance in improving upon the design of an existing looped water system, or designing such a new system can be obtained from NETWK by setting the option DESIGN=2 in the $SPECIF list. With DESIGN=2 a special portion of NETWK is called upon that assists in doing the following (a) sizing pipes, (b) locating possible pipes that should contain pumps, (c) determining water surface elevations of reservoirs (or tanks), and (d) determining heads and flowrates that source pumps should supply. These are determined such that the network results in least cost, or near least cost. This portion of NETWK is interactive, and the user is lead through the design by a series of questions that are asked of the user. This appendix discusses how a user can utilize this special portion of NETWK. Also a brief discussion of the methods that NETWK uses is given to assist in understanding what is needed on the part of the user.

With DESIGN=2 the data may be prepared in the usual manner for a regular analysis in which a solution of the flows and pressures throughout a network, i.e., all pipe diameters may be specified, even though NETWK will determine the best sizes of pipe to use in most of the lines of the network. The user will be asked if given diameters should be ignored or not. If the response is Y for yes then NETWK will decide which pipes should be sized with the minimum allowable diameter (that the user can give), and which pipes should be sized to meet the demands, etc. that are given such that the slope of the HGL is equal or nearly equal to that which will result in the least overall costs. Alternatively, one can prepare the input data similarly to that used with DESIGN=1, i.e., J pipe diameters are given as zero to denote pipes whose diameters are to be solved for, and the remaining pipes are given the given diameter. When the latter option is selected, then the pipes with given diameters will retain these diameters, and in the methods used they will be the "loop forming pipes." What the "loop forming pipes" means is explained below.

When NETWK first begins execution of this portion of the code you will be asked if you wish to accept the default values that define the least cost slope of the HGL versus flowrate relationship i.e., the optimum S-Q relationship, as given in Table D-1. The relationship in Table D-1 is based on the default costs of pipes, electrical energy costs of $0.10/kilowatt-hour, a 50 year life expectancy, and an interest rate of 10 percent. If your response is N (for no), then you will be asked for some basic cost information such as interest rate, capital cost of different pipe sizes per unit length, life of system, and energy costs. NETWK utilizes these costs data to obtain the optimum S-Q relationship by the methods described in a subsequent section of this appendix.

With the optimal S-Q relationship given, NETWK next separates the given network into: (1) a primary branched system, and (2) additional loop forming pipes. If J (number of nodes in the network) pipes are given zero diameters as when DESIGN=1, then the pipes whose diameters are given greater than zero are the loop forming pipes. If all pipes are given a diameter then NETWK makes the decision about which pipes should constitute the loop forming pipes. With these pipes removed a continuous path of pipes must exist to all nodes of the network, i.e., at least one pipe with a given zero diameter must occur at each node. It is these pipes whose diameters are given as zero, and which must form a branched system, that will be sized to satisfy the demands at "least cost". These pipes are sized so that the slope of the HGL is as close to the optimum slope using standard pipe sizes as possible. The additional loop forming pipes are given diameters equal to the minimum allowed if all pipes are given diameters in the original data, or as specified by the user. The method used by NETWK in accomplishing this can be described in the following six steps:

1. The two most dominant sources of supply are identified. If the user does not wish to specify which these are, then those will be selected with the two largest total heads, as given by the input data. The water surface elevation of reservoirs is used for this purpose, and the head produced by pumps, assuming normal capacity is added to their supply water surfaces. Next these two most dominant sources of supply are connected by the shortest path of pipes between them. For subsequent descriptions of the methodology this path will be referred to as the "dominant path". All other paths of the branched system that will be defined, will ultimately terminate at one of the nodes of this dominant path. If only one source of supply exists this dominant path is not defined.

2. The other sources of supply are connected by the shortest path of pipes to one of the nodes of the dominant path. These additional
paths will be referred to as primary paths. These primary paths will each be formed in the ascending order of the total head available from the source at its terminus as defined in the input data if the user does not wish to indicate which source should be used for each succeeding primary path. Any such primary path is terminated (i.e., merges with) when it intersects a previous primary path. Should only two sources of supply exist, then they constitute the dominant sources and this step is omitted.

3. The remaining nodes of the network, that are not included in the dominant path, or any primary path are connected by the shortest path to one of the nodes of the dominant path. Whenever any such path intersects a node in a previously defined path it is terminated. These paths will be referred to as secondary paths. The order in which these secondary paths are formed is first from nodes of degree 1, i.e., dead end pipes, next from nodes of degree 2, i.e., that have only two connecting pipes, etc. The order in which nodes are selected within a given degree is according to their descending elevations.

In completing the above three steps a branched system of pipes is formed that includes all nodes of the network and presumably contains the pipes that will convey the vast majority of the flow from the sources of supply to the various demand points. Those pipes not included in this branched network are identified, and the user has the option of including those that can be appropriately incorporated in the branched system in place of those that the computer selects.

4. An appropriate head is established at each node of the network so that the slope of the hydraulic grade line is as close as possible to the optimum S-Q relationship using standard pipe sizes. The procedure for doing this consists of working through paths in the reverse order of their formation. At the beginning node of each path, the total head is equated to the minimum allowable pressure head (which the user can specify if he does not wish to accept the default value) plus the elevation of the node. Proceeding from this node to succeeding nodes of the path, the total flowrate that each pipe must carry, when the carrying capacity of the additional loop forming pipes is ignored, is computed. With this flowrate known, the optimum S-Q relationship provides the slope of the hydraulic grade line that should exist for this pipe, and therefore its diameter is computed to satisfy the head loss equation. In this process of proceeding along paths a check is made to determine whether the computed head has been previously assigned to any node. If so the previously computed head is composed with the currently computed value, and the larger of the two selected. Upon establishing these heads, the pressure head (e.g., the difference between the total head and the elevation) of the node is examined. Should this pressure head exceed a maximum specified amount, that merits inclusion of a booster pump (which the user can give), then a pipe is suggested that might contain the pump and an appropriate head for this pump is given. The user can accept these suggestions or provide an alternative. Also a list of consecutive nodes, with their connecting pipes, that exceed the maximum head is given to assist the user in deciding were to place a booster pump if the suggested pipe is not acceptable. The total head at the nodes upstream from the pipes in which a booster pump is placed are reduced by the head of the pump. This process of identifying pipes that should contain booster pumps continues whenever the pressure head exceeds the specified maximum pressure head.

When this procedure for establishing HGL elevations advances back to the primary paths it is necessary to have a flowrate that each reservoir must supply that is at the terminus of each primary path. The user is requested to indicate what fraction of the total demand (either positive or negative) he believes this reservoir should supply. The HGL elevations along primary paths are established to just meet the minimum head at some node along the path. The water surface elevation of the reservoir at the terminus of each primary path is thus established.

5. The total heads at nodes of the dominant path that have not been previously assigned, are determined last by a process designed to not only allow for optimum, or near optimum sizing of pipe, but also to assist in determining minimum allowable heads that the two dominant sources of supply should have. To assist in describing how these heads are determined it will be assumed that \( N_d \) nodes exist along the dominant path, excluding the two dominant sources of supply. Figure D-1 shows a sketch in which \( N_d = 3 \). \( N_d \) different cases are examined that assume the flow is directed toward the \( N_d \) nodes from both sides. For case 1 the hydraulic grade line must slope from both directions toward node #1 that is closest to one of the dominant sources; for case #2 the HGL slopes from both directions to the next node, etc. until for the \( N_d \)-th case the HGL slopes toward the node closest to the other dominant source. The elevations of the HGL for each case starts at the node at either the minimum head, \( H_{min} \), above the elevation of the node, or at the head \( H_p \) that may have been established during step 4. The slope of HGL is based on the optimum S-Q relationship corresponding to the flowrate carried by the pipe. These flowrates can be determined since the flowrates, or demands, leaving from each of the nodes of the dominant path are now known in completing step 4. When at the starting node for each case then the demand
at this node can be distributed according to the ratio of the number of nodes to the source of supply and \( H_d + 1 \). Should the HGL fall below the minimum head, \( H_{\text{min}} \), or the total head \( H_t \), then the entire HGL is raised. Raising of the HGL is illustrated in case 2 and 3 of Figure D-1. For each case, i, the required total heads from the two dominant sources of supply \( H_{1i} \) and \( H_{2i} \) are computed. The case that produces the minimum sum of these source heads is selected. The heads given for these two dominant sources of supply is altered, or if the source is a pump, then the head supplied by the pump is altered.

6. With the total heads and flowrates known for each pipe in the branched system obtained through steps 1-5 above, the diameters of each pipe is next computed. These diameters may be determined from any of the frictional head loss equations, i.e., the Darcy-Weisbach equation, the Hazen-Williams equation or the Manning equation. The computer diameters are changed to: (a) the closest standard diameter, (b) the next larger standard diameter, (c) the next smaller diameter as selected by the user.

After the standard diameters are determined, these are utilized in a regular solution by NETWK.

### Least cost slope of the HGL versus flowrate relationship (optimum S-Q relationship)

A vital component of the above procedure is to have the optimum slope, \( S \), that the hydraulic grade line should have as a function of the flowrate, \( Q \), i.e., the optimum S-Q relationship. This relationship cannot be defined algebraically when utilizing the Darcy-Weisbach equation, but is obtained numerically, and used in a table look-up interpolation. In the algorithm used by NETWK at present only the following three major costs are used and called for: (1) pipes and their installation, i.e., pipe capital costs, (2) energy for pumps, and (3) reservoir capacity and head. Additional costs might be included by adding additional code. The capital costs of pipe has the same default values as used with the ICOST=1 option, but may be altered if desired.

<table>
<thead>
<tr>
<th>Flowrate, Q (cfs)</th>
<th>Optimum Slope of Hydr. Grade line</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
<td>.01</td>
</tr>
<tr>
<td>40</td>
<td>.009</td>
</tr>
<tr>
<td>30</td>
<td>.008</td>
</tr>
<tr>
<td>20</td>
<td>.007</td>
</tr>
<tr>
<td>15</td>
<td>.006</td>
</tr>
<tr>
<td>12.5</td>
<td>.005</td>
</tr>
<tr>
<td>10</td>
<td>.004</td>
</tr>
<tr>
<td>8</td>
<td>.0035</td>
</tr>
<tr>
<td>7</td>
<td>.0030</td>
</tr>
<tr>
<td>6</td>
<td>.0028</td>
</tr>
<tr>
<td>5</td>
<td>.0025</td>
</tr>
<tr>
<td>4</td>
<td>.0023</td>
</tr>
<tr>
<td>3</td>
<td>.0021</td>
</tr>
<tr>
<td>2</td>
<td>.0020</td>
</tr>
<tr>
<td>1</td>
<td>.0019</td>
</tr>
<tr>
<td>.6</td>
<td>.0018</td>
</tr>
<tr>
<td>.3</td>
<td>.0015</td>
</tr>
<tr>
<td>.1</td>
<td>.0010</td>
</tr>
<tr>
<td>.05</td>
<td>.0005</td>
</tr>
<tr>
<td>.01</td>
<td>.0003</td>
</tr>
</tbody>
</table>
Figure D-1. Illustration of procedure used in determining the optimum hydraulic grade line to use along the dominant path between two major sources of supply.
APPENDIX E - AUXILIARY PROGRAM PRINT

(RETRIEVAL OF INDIVIDUAL INFORMATION FROM SIMULATION SOLUTION)

Introduction

This appendix of the manual describes how any item of information can be retrieved from the direct access file that will be created by a simulation of a network with the program USU-NETWK provided NTRAND in the SSPECIF list is greater than 0, by using the auxiliary, or post processor, program PRINT. Since the assignment of direct access files may vary slightly from system to system, there could be slight variations in the characteristics of the files that USU-NETWK writes for PRINT to use. However, the use of the program PRINT will be the same as far as you are concerned, as a user. The difference will be whether the information in the files that PRINT uses is formatted, or unformatted. The full program for larger computers such as VAXes for example has an option that allow the file to be written as a formatted file or an unformatted file, whereas the PC version only permits USU-NETWK to write formatted files. To instruct the VAX version of USU-NETWK, for example that it should write formatted, i.e. ASCII, files for PRINT to utilize the option NTRAND is given the value 3 in place of 1, or NTRAND=4 in place of 2 to denote that both a formatted direct access file, as well as the regular solution tables are to be written. The VAX version of PRINT also has an option that allows it to read either a formatted, or an unformatted file, whereas the PC version of PRINT only reads formatted files.

The illustrations used in this appendix will be taken from using PRINT on a PC under DOS. Identically the same can occur when PRINT is used on a VAX computer for example, with the exception that if you are using the VAX from a remote terminal the results will appear there instead of the PC's monitor. However, PRINT might also be used in batch mode on the VAX whereas this is not possible with a PC under DOS.

Program PRINT utilizes two files FILE30 and FILE31 (these may contain the extensions .DAT) The file FILE31 contains information about the network, and file FILE30 contains information about the solution of the network at the time increments for which the simulation solution was obtained by USU-NETWK. File FILE30 is a direct access file. These two files will be written by USU-NETWK if the option NTRAND = 1 or = 2 (or = 3 or = 4) as described under this option in the body of this manual. The files FILE30 and FILE31, that contain the information from USU-NETWK about the network that you wish to selectively examine, must be present in the same directory from which you are executing PRINT. PRINT is designed to execute in either batch or interactive mode, and therefore, unless you are using a version that is designed solely for interactive mode the first prompt you will receive if you are using PRINT interactively is to indicate whether interactive of batch use is occurring. Next you will be ask to indicate whether you wish to change any of PRINTS options.

Input Data to Auxiliary Program PRINT

The first line of input to program PRINT requires the word HOURS, DAYS, or MINUTES to communicate the units of the times that will be used on subsequent commands.

After this first input other input consist of a command followed by the data called for by this command. The commands may be in any order. Each command word is followed by a line containing numbers (integers) separated by a blank (or blanks), a comma "", or a dash "-" (minus sign) between two numbers to denote a range of values. The first of these integers (or range of integers) gives the time (or times) and the remaining integers are pipe numbers, node numbers, etc. For some commands only times are called for; in which event the list of integers denotes times. Each such list of numbers is terminated by a "/." If you leave off the / it will work properly only if the line is at least as long as previous lines; otherwise the number in the list after the last value given will generally be appended to your line. It is also possible to place these numbers on the same lines as the command name if desired, as described below. For example after the command PIPES you may give 6-12 5 7 50-70/ to indicate that for times 6 through 12, on the increment of the solution, the flow rates and head losses should be given for pipes 5, 7, 50 through 70.

There are seventeen valid command names. These are: PIPES, NODES, RESERVOIRS (which may be truncated to RESERV), PUMPS, VALVES, PIPE TABLES (which may be truncated to PIPE T), NODE TABLES (which may be
truncated to NODE T), PIPE FLOWRATE (which may be truncated to PIPE F), PIPE HEAD LOSS (which may be truncated to PIPE H), NODE PRESSURE (which may be truncated to NODE P), NODE HGL-ELEV (which may be truncated to NODE H), NODE DEMAND (which may be truncated to NODE D), OPTIONS, SAME CARD, NEXT CARD, END (or STOP). Each of these names are listed below, followed by a description of the record which follows it. Each such description is followed by an example.

**PIPS**

The time, or range of times, followed by the pipe numbers whose flow rate and head loss are to be displayed for this time, or range of times.

**PIPS**
12 8 12-25 4/  
(At time 12 the flow rate and head losses are to be given for pipes 8, 12 through 25 and 4.)

**NODES**

The time, or range of times, followed by the node numbers whose demands and pressures are to be displayed for this time, or range of times.

**NODES**
4-8 12 14/  
(At times 4 through 8 on the increment of the solution give the demands and pressures at nodes 12 and 14 are to be displayed.)

**RESERV**

The time, or range of times, followed by the reservoir's designations (pipe numbers if NODESP = 0, or node numbers if NODESP = 1 when the data file was created by USU-NETWK), whose water surface elevation and flow rate are to be displayed.

**RESERV**
12 5/  
(At time 12 the water surface elevation in feet or meter, depending upon whether ES or SI units are used, respectively at reservoir designated by 5.)

**PUMPS**

The time, or range of time, followed by the pump designations (pipe numbers if NODESP = 0, or node numbers if NODESP=1)

**PUMPS**
12 1 8 6/  
(At time 12 give the flow rates and HGL elevations for pumps designated by 1 8 and 6.)

**VALVES**

The time, or range of times, followed by the PRV numbers whose upstream and downstream HGL's are to be displayed.

**VALVES**
12 1/  
(At time 12 give the upstream HGL elevation and the HGL elevation of the setting for pressure reduction valve, PRV, number 1.)

**PIPE F, PIPE H, NODE P, NODE H, and NODE D**

These five commands retrieve basically the same data as the PIPES and the NODES commands do. They place the data in a table in which time is in the first column, and the requested items of data associated with the pipe or node numbers, that are given in the list, are given in subsequent columns. The width of these tables allows only up to 11 different pipe or node numbers in any given table, but as many tables as desired can be created. The tables created by these commands are identical to the special tables that can be written by USU-NETWK when PRINTT is given a value other than zero in the $TDATA list of options. Therefore you can use PRINT to create special tables that can be used directly with the post processor PLTTIM described in Appendix C for special graphics, for example. In fact there is a special option in PRINT that allows you to have each such table you create placed in a separate file so that it is not necessary to extract a given table for use with PLTTIM if you ask PRINT to give a number of such tables.

**PIPE Flowrate**

The range of times followed by the pipe numbers whose flow rates are to be place in a special table with time as the first column. If only a single time is given then this table will contain but one line of data, and therefore this command is generally used with a range of times.

**PIPE F**
0-24 5-15/  
(The special table for pipe flow rates will be for times 0 through 24 with the increment used in the solution with pipe numbers 5 through 15 as the separate column headings in this table.)
PIPE Headloss

The range of times followed by the pipe numbers whose head losses are to be place in a special table with time as the first column. If only a single time is given then this table will contain but one line of data, and therefore this command is generally used with a range of times.

PIPE F
0-24 30,35,40-47/
(The special table for pipe head losses will be for times 0 through 24 with the increments used in the solution with pipes 30, 35, 40, 41, 42, 43, 44, 45, 46 and 47 as the subsequent headings for individual columns after time as the first column in this table.)

NODE Pressure

The range of times followed by the node numbers whose pressures are to be place in a special table with time as the first column. If only a single time is given then this table will contain but one line of data, and therefore this command is generally used with a range of times.

NODE P
1-13 1-7/
(The special table for nodal pressures will be for times 1 through 13 with the increments used in the solution with nodes 1 through 7 in separate columns in this table. If the increment of the solution from USU-NETWK were 2 hours for example, then the values in this table would be for times midway between those given in the original solution.)

NODEH HGL-elevation

The range of times followed by the node numbers whose elevations of the HGL are to be place in a special table with time as the first column. If only a single time is given then this table will contain but one line of data, and therefore this command is generally used with a range of times.

NODE H
0-24 5-15/
(The special table for HGL-elevations will be for times 0 through 24.)

NODE Demand

The range of times followed by the node numbers whose demands are to be place in a special table with time as the first column.

NODE D
12-24 35-46/
(The special table for nodal demands will be for times 12 through 24 with the increments used in the solution with nodes 35 through 46 in separate columns in this table. The demands could be flow rates from sources of supply if these are nodes within the list.)

PIPE T

The times (which may be given as a range) for which complete tables of pipe information are to be printed.

PIPE T
12,24/
(Give tables of pipe data for times 12 and 24.)
12-24
(Give tables of pipe data for times 12 through 24 using the increment of the solution.)

NODE T

The times (which may be given as a range) for which complete tables of node information are to be printed.

NODE T
1 5 6-20/
(Give complete tables of node data for time 1, 5 and 6 through 20. For example if the solution was obtained on a 2 hour increment, and the time units was selected as hours, then the first two such tables would be interpolated midway between the available solution from time increments, and the tables for hours 6 through 20 would be on a two hour increment and agree with the solutions for these time increments. If, however, the line had read 1 5-20/ then the tables would be given at hours 1, 5, 7, 9, 11, 13, 15, 17 and 19 hours respectively.)

BOTH T

The times (which may be a range if two integers are separated by a -) for which complete tables (both pipe data and node data tables) of information are to be printed.

BOTH T
18/
(Give both the Pipe data table and the Node data
table for time 18.)

SAME C

No data follows this command. Rather the next command follows. The command SAME CARD indicates that the input data that usually follows the command will be given on the same line as the command until the NEXT CARD command is issued. (This does not prohibit entry of data after the command as well, since data is accepted under a given command until a new command is given).

NEXT C

No data follows this command. Rather the next command follows. The command NEXT CARD toggles the mode back so that data will be ignored if entered on the same line as the command.

OPTIONS

The command OPTIONS can be given anytime when a command is allowed and its gets you to the options menu again. Access to the options menu can be requested when you first execute PRINT by responding Y (for yes) when you are asked if you wish to change any options, and generally you will change options, if this needs to be done, from their default values at this time, but if for some reason you wish to change the options later, then this can be done by giving the command OPTIONS.

The options consist of: (a) A switch that allows you to indicate whether the file FILE30 and FILE31 are binary, or ASCII, (b) the ability to set FORMATS for different outputs that you may be using later, and (c) to designate whether the special time dependent tables created by the command PIPE F, PIPE H, NODE P, NODE H, and NODE D are to be place in separate files. If you select these separate files, then they will be under the names SPECIAxx.DAT in which xx starts with the number 01 and is increment for each new such file.

END or STOP

These commands terminate execution.

From the above examples you probably noted that separate numbers can be separated by either one or more blanks or a comma (which may be followed by blanks). To denote a range of numbers a "-" is used. For example, 12-25 causes the same action as listing the numbers 12, 13, ..., 25.

Examples of Use of Retrieval Program PRINT

Examples of the commands given to the retrieval program PRINT and the results it prints are given below. These examples use the interactive mode on a PC. Identical results could be obtained under time-shared use of a computer systems or by submitting a batch job to such a system.

The problem that will be used in illustrating use of Program PRINT is Example No. 1 for which a simulation has been obtain by USU-NETWK over a 24 hour period of time in 2 hour increments. The input data for this problem is given below.

EXAMPLE NUMBER 1 OF SIMPLE NETWORK

```
/* $SPECIF NPRINT=-1 NODESP=0,ISIML=1,NTRAND=2
  NSORTP=0,NODESO=0 SEND 
  PIPE- 
  3 12. 300. 7-9. 300. 2. 300. 
  5 10. 1300. 2 11. 280. 
  5 8. 1600. 3 4 6. 200. 
  4 10. 2000. 1 4 
  2 8. 2200. 2 3 1.5 280. 
  6 8. 2200. 2 3 
  8 8. 3000. 3 4 6. 200. 
  10 10. 2800. 4 6 3. 200. 
  9 6. 1600. 3 6 
  7 10. 1000. 5 1. 270. 
  11 12. 500. 1 
  12 17. 1. 7 
  PRINT 
  7 3 80 4 77 5 72 350 
  RESER 
  12 400 
  11 350 
  MINOR 
  4 10 
  VALVE 
  8 1500 320 
  RUN 
  $DATA NPUNOD=2,NPNRES=1,INCHR=2,ISUNIT=0, 
  LINEAR=1 END 
  DEMAND FUNCTION 
  1 0.1 1 1.2 0.16 .5 20 .6 24 1/ 
  2 5 6/ 
  4 3 1.3 6 1.5 10 1 14 .6 18 .5 24 1/ 
  1 4/ 
  PUMP RULES 
  7 2 11 3 348 4 352 3 355 2 358 1/ 
  STORAGE FUNCTION 
  1 35 0 350 360 359 320 360 59905/ 
  1/1 
  2 392 0 397 314160 402 628520/ 
  END 
```

Note in the $SPECIF options list that the options NSORTP=0 and NODESO=0 tell USU-NETWK that the pipes and nodes should be listed in the order of their input under these commands. The problem specifies that during this simulation a constant rate of inflow is occurring to reservoir number 1. This specification is accomplished by putting node 7 one foot downstream from reservoir # 1.

Below the dialogue between the PRINT program and the user is shown. To help you distinguish between what PRINT gives, and what the user requests, the user's input is in bold lettering. (The actual process by which the output from PRINT has been obtained below was to have PRINT also write the results to a file (an option
allowed), and then this file has been retrieved as part of this document with the user requests inserted at the correct positions.

PRINT
IF I AM USED FROM TIMESHARE GIVE 1, FROM BATCH 0
1

TYPE IN THE UNITS OF TIME: DAYS, HOURS, MINUTES OR SECONDS
DAYS
DO YOU WANT RESULTS ON FILE ALSO? (Y or N)
Y

NOW GIVE A COMMAND SUCH AS:
Pipes, Nodes, Reservoirs, Pumps, Valves, Pipe T, Node T,
both T followed by time & range, etc.

PIPE T

NOW GIVE NODE DATA.

NODE DATA. (TIME= 7.0)

NODE NO. DEMAND (CFS) (GPM) ELEV PRESSURE HEAD HLOSS ELEV
7 -8.000 -3590.64 300. 99.74 43.22 399.74
1 1.000 444.83 285. 65.53 28.40 34.53
3 3.663 1063.71 200. 58.43 25.12 33.44
4 8.220 3702.85 200. 88.47 54.34 288.47
9 4.000 1413.81 250. 133.89 37.67 403.09
6 1.150 471.87 250. 85.28 37.67 288.47

PIPES
6 1-6/

TIME= 6.0

FLOW RATE AND HEAD LOSS IN PIPE
1 .902984E+01 .197942E+02
2 .232591E+01 .491849E+02
3 .281903E+01 .522276E+02
4 .470546E+01 .678469E+02
5 .450393E+01 .335657E+02
6 .274312E+01 .680523E+02

Notice that the pipes and nodes are listed in the data tables in the same order as the input data.
rather than in their sorted order that would occur if NODESO and NSORTP had been left with their default values. Furthermore since the times specified were between time increments for the solution these data tables are actually obtained by interpolating the given solutions.

The following changes will be made to the above input to USU-NETWK: (1) the option NODESP=0 will be removed from the $SPECIF list. Since if ISIML=1 is given, the default assumes sources of supply are numbered as nodes, the removal of ISIML=1 means that the nodes are numbered. The reservoirs are given the nodes 8 and 9 and the source pump is given node 10. (2) the appropriate source nodes are added to the data under the PIPE- command, (3) the options NODESO=0 and NSORTP=0 have been removed from the $SPECIF list, and (4) the designation of the pumps and reservoirs is changed to node numbers. With these changes the input data now is as given below.

```
EXAMPLE NUMBER 1 OF SIMPLE NETWORK
$SPECIF NPRINT=-1,ISIML=1,NTRAND=2 SEND
PIPE- 1 12 500 7 -8 300 2 2 300.
  5 10 1300 2 1 1 280.
  8 1800 3 4 6 200.
  3 10 2000 1 4
  6 2200 2 3 1 5 200.
  6 2200 3 3
  8 3500 3 4 6 200.
  10 10 2000 4 6 3 200.
  9 6 2400 5 6
  7 12 1000 10 5 1 270.
  11 12 500 9 1
  12 12 1 7 8
PUMPS
  10 3 80 4 77 5 72 350
RESER
  8 400
  9 350
MINCR
  4 10
VALUE
  6 1500 320
RUN
$DATA NPUNOD=2, NPRES=1, INCHER=2, ISUNIT=0, LINEAR=1 SEND
$SPECIFIC FUNCTION
  1 4 1 2 8 1 12 .8 16 .5 20 .6 24 1/
  2 5 6/
  3 1.3 6 1.5 10 1 14 .6 18 .5 24 1/
  4 1/
PUMP RULES
  7 2 11 3 348 4 352 3 355 2 358 1/
STORAGE FUNCTION
  1 345 0 350 19635 355 39270 360 36905/
  11/
  2 392 0 397 314160 402 626320/
END SIMULATION
```

Now calling on program PRINT again to retrieve information that has been place on the files FILE30 and FILE31 by USU-NETWK.
### PIPE F

#### 1-23 1-10/

<table>
<thead>
<tr>
<th>TIME</th>
<th>PIPE NO.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### PIPE NODES

<table>
<thead>
<tr>
<th>PIPE NODES</th>
<th>(FEET)</th>
<th>(PSI)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### NODE P

<table>
<thead>
<tr>
<th>NODE P</th>
<th>(PSI)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### BOTH T

<table>
<thead>
<tr>
<th>BOTH T</th>
<th>7-9/</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### UNITS OF SOLUTION ARE

<table>
<thead>
<tr>
<th>UNITS OF SOLUTION ARE</th>
<th>FEET</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### NODES AS

<table>
<thead>
<tr>
<th>NODES AS</th>
<th>(PSI)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### DESIGNATED PIPES AS

<table>
<thead>
<tr>
<th>DESIGNATED PIPES AS</th>
<th>(PSI)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### DATA (TIME=

<table>
<thead>
<tr>
<th>DATA (TIME=</th>
<th>7.0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### PIPE NODES

<table>
<thead>
<tr>
<th>PIPE NODES</th>
<th>(FEET)</th>
<th>(PSI)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### PIPE DATA (TIME= 7.0)

<table>
<thead>
<tr>
<th>PIPE DATA</th>
<th>(TIME=</th>
<th>7.0</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### PIPE DATA (TIME= 9.0)

<table>
<thead>
<tr>
<th>PIPE DATA</th>
<th>(TIME=</th>
<th>9.0</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
9 5 6 1400. 5 .006850 2.031 10.25 108.07 76.40
10 4 6 2000. 8 .006850 .919 1.50 1.92 .96
11 9 7 500. 1 .006850 .428 .54 .09 .15
12 8 7 500. 1 0.006850 .447 .59 .00 .16

NODE DATA. (TIME= 0.0)

<table>
<thead>
<tr>
<th>NODE NO.</th>
<th>DEMAND</th>
<th>HGL</th>
<th>ELEV</th>
<th>HEAD</th>
<th>PRESSURE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-8.000</td>
<td>3590.64</td>
<td>280.</td>
<td>119.65</td>
<td>51.85</td>
</tr>
<tr>
<td>2</td>
<td>6.750</td>
<td>3029.60</td>
<td>300.</td>
<td>6.16</td>
<td>2.87</td>
</tr>
<tr>
<td>3</td>
<td>1.000</td>
<td>448.63</td>
<td>230.</td>
<td>64.55</td>
<td>27.97</td>
</tr>
<tr>
<td>4</td>
<td>1.688</td>
<td>757.40</td>
<td>200.</td>
<td>152.44</td>
<td>66.06</td>
</tr>
<tr>
<td>5</td>
<td>1.900</td>
<td>652.76</td>
<td>270.</td>
<td>112.21</td>
<td>48.62</td>
</tr>
<tr>
<td>6</td>
<td>0.000</td>
<td>426.39</td>
<td>200.</td>
<td>211.22</td>
<td>91.92</td>
</tr>
<tr>
<td>7</td>
<td>0.000</td>
<td>300.</td>
<td></td>
<td>-269.21</td>
<td>-129.66</td>
</tr>
<tr>
<td>8</td>
<td>-0.128</td>
<td>-56.56</td>
<td>0.</td>
<td>200.20</td>
<td>30.75</td>
</tr>
<tr>
<td>9</td>
<td>-0.349</td>
<td>-3114.98</td>
<td>0.</td>
<td>350.00</td>
<td>151.67</td>
</tr>
<tr>
<td>10</td>
<td>-6.940</td>
<td>0.</td>
<td></td>
<td>350.00</td>
<td>74.78</td>
</tr>
</tbody>
</table>

VALVES
12 1/

TIME= 12.0
HGL S ELEV. AT GIVEN PRV
8 .421638E+03 .320000E+03

You should note that the last data tables produces by PRINT have Pipes and Nodes listed in ascending order. This occurs because USU-NETWK was told to list the pipe and nodes in this order since options NSORTP=1 and NODES0=1 (the default values). USU-NETWK places pipes and nodes in the files FILE30 AND FILE31 in the same order as they are listed in the output tables for "pipe data" and "node data".

Since PRINT can reproduce the original tables from the solution, and the files FILE30 and FILE31 are smaller than the complete output from a computer simulation, Program PRINT might be used to print the output using a PC when USU-NETWK was executed on a larger computer. The option NTRAND=3 should be used in this event to have the files FILE30 and FILE31 written as ASCII files rather than binary files. Then files FILE30 and FILE31 (with the extension .DAT) are transferred to the PC, program PRINT is executed with the command BOTH T followed by 0-24/, and the results sent to the printer driven by the PC.
APPENDIX F - PROGRAM PMPCUR
for display of pump curves

Introduction

An important component in providing data for a network is the description of the performance of pumps, or giving data after the PUMPS and BOOSTER commands. USU-NETWK allows this description to be given in 3 different ways: (1) By giving 3 or more pairs of flowrate and head that a pump produces to define its operating characteristics, (2) By giving the power the pump supplies to the fluid and its normal capacity (flowrate at peak efficiency), or (3) giving the coefficient for a 2nd degree polynomial that describes the pump curve, e.g. giving a,b and c in the equation,

\[ h_p = aQ^2 + bQ + c \]  \((1)\)

The utility program PMPCUR, whose use is described in this appendix, is designed to help you visualize how any of these descriptions of a pump's operating characteristics are interpreted by USU-NETWK by displaying the pump curve thus defined. This utility also allows you to modify the description of the pump curve until you are satisfied that it adequately describes the performance of the pump, and then provides you with the data you can give in the input data file for USU-NETWK to use in defining the operation of pumps.

Program PMPCUR contains menus that inform you what can be done, and lets you select the items to accomplish what you want. The description provided herein is therefore brief, and only provides a general overview of what PMPCUR does. To get acquainted with the operation of PMPCUR you need to use it, study how the above methods that provide the mathematical description of a pump's operating characteristics, and determine how you can effectively change these descriptions to duplicate the actual operation of pumps in your piping systems.

How to use PMPCUR

When PMPCUR is executed the first menu that appears reads:

Select input to use
1. 3 pairs of Flowrate versus Head data
2. Pump Fluid Power and Normal Capacity
3. More than 3 pairs of data
4. Change options
5. Stop

The option desired from the above menu can be selected by any of the following methods: (1) typing in the number 1 through 5 for the option, (2) Moving the highlighting bar by pressing the up or down arrow keys, or (3) if your PC has a mouse you can move the mouse to the appropriate line and press the left mouse button. When the item has been properly selected press the right mouse button. The effect of these options is as follows:

1. By selecting 3 pairs of flowrate versus head data, the pump curve will be defined according the to default method USU-NETWK utilizes, e.g. a 2nd degree polynomial will be passed through the three points you will be required to give, and this polynomial equation will be used to define how much head the pump supplies to the fluid as a function of the flowrate passing through the pump.

2. If you select item 2 from the above menu you will be asked to supply the Power (in either horsepower when using ES units, or kilowatts when using SI units), which is an alternative method that USU-NETWK allows you to use in defining the operating characteristics of a pump. PMPCUR will display the pump curve that USU-NETWK using.

3. Selection of item 3 from the above menu allows you to define the operation of a pump with more than 3 pairs of flowrate versus head data, as occurs in USU-NETWK if you set the option PCHAR3=0 in the $SPECIF list. PMPCUR will provide you with the coefficient for the 2nd degree polynomials that passes through...
4. Upon section of item 4 you are permitted to change the default options associated with pump operations. These consist of (a) whether ES or SI units are being used, (i.e. the control you have with the option NUNIT in USU-NETWK), (b) the units that the flowrate values will be given in (i.e. the control you have with the option NPGPM in USU-NETWK), and (c) the value of the specific weight of the fluid being pumped (i.e. the control you have with the option GAMMA, or the command WEIGHT in USU-NETWK).

5. Selection of 5 will terminate the execution of PMPCUR.

Depending upon the above menu item you select you will be prompted by PMPCUR for the necessary input data. Upon supplying these requested values a plot will appear on the monitor that shows the pump curve being defined, and that will be used by USU-NETWK in solving the network. This display consists of the flowrate across the abissc of the graph, and the head supplied by the pump to the fluid as the ordinate of the graph. In the bottom left had corner you will see the prompt Press Ret.

When you are through examining the graph press the "Enter" or "Return" key depending upon how it is labelled on your keyboard, and you will see the following menu displayed:

1. Terminate
2. New graph using same pump with dif. range of flowrates
3. New graph for different pump
4. Change shape of curve
5. Print polynomial coefficients

Selection of one of the above items has the effect described. When you select 4 to "Change shape of curve" you will be prompted with the following:
You can:
1. change polynomial coef
2. change value of points
3. change range of Qs
4. terminate changes

This menu will repeat until you select 3.

Selection of 1 permits you to give the values of a, b and c in the above 2nd degree polynomial equation to define a new pump curve. Upon giving these 3 values you will see the new pump curve displayed on the monitor. Selection of 2 will result in you being shown what the current points are (i.e. pairs of flowrates versus heads) and allows you to change any or all of these pairs of values. Selection of 3 lets you show the pump curve over larger or smaller ranges of flowrates. When displaying the pump curve from the changes permitted by items 1 or 2, the range of flowrates will be from the flowrate given with the first point to the flowrate given with the last point. If a different range of flowrates is desired for the abissc of the graph then you must select item # 3 subsequently. Upon indicating that you do not wish to change any additional points, the newly defined pump curve will be displayed.

When you are satisfied that the description of the pump that you have given agrees with the actual operation of the pump you should terminate PMPCUR. Upon indicating you wish to terminate, PMPCUR will display the final data that might be used under the PUMPS or BOOSTER command in USU-NETWK to define this pump. You will also be ask if you wish to have this information written to a file. If you indicate that you do, then you will be ask for a file name and the information as it appears on the monitor will be written to the indicated file for you to refer to subsequently.

Illustration of Use

As an example assume that the pump curve is defined by the following 3 pairs of flowrate versus head produced.

<table>
<thead>
<tr>
<th>Flowrate (cfs)</th>
<th>Head Produced (ft)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>50</td>
</tr>
<tr>
<td>2</td>
<td>48</td>
</tr>
<tr>
<td>3</td>
<td>42</td>
</tr>
</tbody>
</table>

The resulting curve produced by PMPCUR is shown in Figure F-1. This graph was obtained by utilizing the "Print Screen" and an Epson printer.
that was attached to a IBM XT that was used to run the program PMPCUR. In making this graph the range of flowrates was changed from the default of 1 to 3 (the range given in the data above) to 0 to 4, when PMPCUR ask:

Default range for flowrates 1 - 3
Should these be changed? (Y or N)

Note that the 2nd degree polynomial that passes through the three points given to define the pump curve results in the shut-off head (the head produced by the pump at zero flowrate) being less than the maximum amount produced by the pump. This might be the actual situation if the pump has strictly radial flow through its impeller, but most centrifugal pumps are "mixed flow" devices and have larger shut-off heads than the head associated with their normal capacities.

It is possible to specify operating characteristics for a pump so that there is no solution possible for the network problem. Such a "no solution" situation will result if the maximum head that a pump can produce is less than that required in order meet the head that other component of the network dictate exist at its downstream node. Mathematically the equation associated with the pump's operation for such situations has a saddle point, causing the solution to become imaginary.

To illustrate such a situation take the small network illustrate in example problem # 4 in which 6 pumps supply water to a reservoir through a network of branched pipes. In this example assume that the pump at node # 2 has the water surface elevation of its source 175 feet lower than that given in the input data for this problem, i.e. the source elevation is 5170 feet rather than 5345 feet. The input data for this problem would then be:

Pump 2 has w.s. elevation reduced 175' to 5170'. No solution possible because max. head the this pump can produce is too small for HGL elevation at node 3.

$SPECIF NPRINT=-2,NFLOW=1,NPGPM=1, NODESP=1,COEFRO=120$<br>
PIPE:<br>1 8. 1670. 1 5350. 3 5380.<br>2 8. 1200. 2 5340. 3

When attempting a solution USU-NETWK fails to converge. A portion of this solution attempt is shown below in which the maximum 15 Newton iterations is exceeded, and the SUM OF DIFFERENCES between the absolute changes in flowrates between consecutive iterations has not reduced to a value less than the allowed error. In fact this sum of differences does not become smaller with succeeding Newton iterations.

```
ITERATION= 1 SUM OF DIFFERENCES=.431E+01
ITERATION= 2 SUM OF DIFFERENCES=.252E+01
ITERATION= 3 SUM OF DIFFERENCES=.140E+02
ITERATION= 4 SUM OF DIFFERENCES=.708E+01
ITERATION= 5 SUM OF DIFFERENCES=.366E+01
ITERATION= 6 SUM OF DIFFERENCES=.203E+01
ITERATION= 7 SUM OF DIFFERENCES=.173E+01
ITERATION= 8 SUM OF DIFFERENCES=.238E+01
ITERATION= 9 SUM OF DIFFERENCES=.161E+01
ITERATION= 10 SUM OF DIFFERENCES=.675E+01
ITERATION= 11 SUM OF DIFFERENCES=.495E+01
ITERATION= 12 SUM OF DIFFERENCES=.261E+01
ITERATION= 13 SUM OF DIFFERENCES=.166E+01
ITERATION= 14 SUM OF DIFFERENCES=.356E+01
ITERATION= 15 SUM OF DIFFERENCES=.205E+01

DID NOT CONVERGE IN 15 ITERATIONS - SUM OF DIFFERENCES = .204E+01
```

LOSSES DUE TO FL. FRIC. IN ALL PIPES
POWER LOSS=59.027 H.P.=44.034 KWATTS.
ENERGY LOSS=1056.822 KWHRS/DAY

THE FLOW RATE 71.564 IS QUITE DIFFERENT THAN THE NORMAL CAPACITY 600.000
AS GIVEN BY THE PUMP CHARACTERISTICS FOR PUMP 2 IN PIPE 2.

IS THE PROBLEM CORRECTLY SPECIFIED?

PUMPS:
NODE PIPE HEAD FLOW HORSEPOWER KILOWATTS KW-HRS/DAY
1 1 215 732 39.80 28.69 712.51
2 2 355 71 6.43 4.78 113.06
4 4 180 771 37.07 27.66 563.77
5 6 102 762 36.14 26.96 648.87
8 8 172 757 34.94 25.99 623.77
10 10 142 840 30.21 22.54 540.95

F-3
The pump curve for the # 2 pump is shown in Figure F-2. The maximum head this pump can produce as dictated by the 2nd degree polynomial through the three points defined in the input data is 369 ft. When this head is added to the specified water surface elevation of 5,170 ft, the value of 5539 ft is obtained, but this value is less than the 5549.13 feet shown in the NODE DATA: solution table above. Thus no solution is possible, and the Newton iterative solution technique simple can not converge to a non existent solution.

Should you encounter a similar situation in which a solution is not produced, i.e. the SUM OF DIFFERENCES from the Newton iterative method does not consistently becomes smaller with each subsequent iteration, then a likely cause of the problem may be a pump specification that results in a system of equations for which no solution is possible. Program PUMCUR can assist you in determining whether this is the problem.
Figure F-1. Pump curve displayed by PMPCUR.

Figure F-2. Pump operating characteristics for pump # 2 in illustrate example problem # 4 as defined by a 2nd degree polynomial passing through the 3 pairs of input data given.
## INDEX

<table>
<thead>
<tr>
<th>%</th>
<th>47</th>
<th>economic analysis</th>
<th>78</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ASPECI</td>
<td>A-1</td>
<td>EFFI CIency</td>
<td>74,156</td>
</tr>
<tr>
<td>SEND</td>
<td>111</td>
<td>ELECTricity</td>
<td>74,77,78,133</td>
</tr>
<tr>
<td>$DATA</td>
<td>82,128,140</td>
<td>ELEV</td>
<td>18</td>
</tr>
<tr>
<td>ADDITional</td>
<td>72,76,78</td>
<td>ELNODE</td>
<td>67</td>
</tr>
<tr>
<td>ALLCOefficients</td>
<td>67,121,150</td>
<td>ELPUMPs</td>
<td>67</td>
</tr>
<tr>
<td>Alpha-numeric</td>
<td>39,41</td>
<td>ELREServois</td>
<td>67,122</td>
</tr>
<tr>
<td>ALTV</td>
<td>82,154</td>
<td>END</td>
<td>xxi,49,76</td>
</tr>
<tr>
<td>APPURTENCES</td>
<td>75,78</td>
<td>END TABLES</td>
<td>85,130</td>
</tr>
<tr>
<td>BOOSTer</td>
<td>xxi,48,123</td>
<td>engineering cost</td>
<td>5,71,D-1</td>
</tr>
<tr>
<td>BOTH TABLES</td>
<td>E-3</td>
<td>EQUDIA</td>
<td>16,29</td>
</tr>
<tr>
<td>BOTTOM</td>
<td>85,159</td>
<td>ER8</td>
<td>49</td>
</tr>
<tr>
<td>BPUMP</td>
<td>48,123</td>
<td>ERROR</td>
<td>xxi,16,49,159</td>
</tr>
<tr>
<td>BPVAL</td>
<td>xxi,49,149,150</td>
<td>English units</td>
<td>3</td>
</tr>
<tr>
<td>branched system</td>
<td>21,76,134</td>
<td>executing NETWK</td>
<td>iv</td>
</tr>
<tr>
<td>CAPlital</td>
<td>75,78,156</td>
<td>flow coefficient</td>
<td>88</td>
</tr>
<tr>
<td>capital costs</td>
<td>74,78</td>
<td>FLOW RULES</td>
<td>88,155</td>
</tr>
<tr>
<td>CHANGE</td>
<td>67,69,122</td>
<td>FLOWFC</td>
<td>16,44</td>
</tr>
<tr>
<td>change data</td>
<td>67,122</td>
<td>FLOWS (cost)</td>
<td>78</td>
</tr>
<tr>
<td>CHECK</td>
<td>xxi,49</td>
<td>fluid mechanics</td>
<td>xvii</td>
</tr>
<tr>
<td>check valve</td>
<td>64,102</td>
<td>FORMAT</td>
<td>xxi,23,52</td>
</tr>
<tr>
<td>COEFFicients</td>
<td>12,118,67</td>
<td>format</td>
<td>42,47</td>
</tr>
<tr>
<td>COEFRO</td>
<td>12,115,135</td>
<td>general specifications</td>
<td>10,11</td>
</tr>
<tr>
<td>commands</td>
<td>47,48</td>
<td>graphics</td>
<td>6,30,C-1</td>
</tr>
<tr>
<td>coordinates</td>
<td>24</td>
<td>HGLSET</td>
<td>89,169</td>
</tr>
<tr>
<td>CORNOD</td>
<td>11,120</td>
<td>HLETTE</td>
<td>16</td>
</tr>
<tr>
<td>CORPIN</td>
<td>11,120</td>
<td>HOURS</td>
<td>E-1</td>
</tr>
<tr>
<td>CORPIP</td>
<td>11,118</td>
<td>HTIME</td>
<td>83,130,140,154</td>
</tr>
<tr>
<td>CORPUM</td>
<td>14</td>
<td>ICOST</td>
<td>17,20,121,133,156</td>
</tr>
<tr>
<td>cost data</td>
<td>10,71</td>
<td>IHGL</td>
<td>17,20,21,115,125,156,157,166</td>
</tr>
<tr>
<td>crf</td>
<td>72</td>
<td>INCHR</td>
<td>83,141,154</td>
</tr>
<tr>
<td>DATVAL</td>
<td>87</td>
<td>INCHR</td>
<td>83</td>
</tr>
<tr>
<td>DAYS</td>
<td>74,E-1</td>
<td>INCMIN</td>
<td>83,159</td>
</tr>
<tr>
<td>DEMAND</td>
<td>67</td>
<td>INCMIP</td>
<td>83</td>
</tr>
<tr>
<td>DEMAND FUNCT</td>
<td>87,141,145,147,155</td>
<td>INITIAL</td>
<td>22,154</td>
</tr>
<tr>
<td>DEMC</td>
<td>75</td>
<td>INPUT</td>
<td>22</td>
</tr>
<tr>
<td>detailed data</td>
<td>10,47</td>
<td>INPUTA</td>
<td>23</td>
</tr>
<tr>
<td>DESIGN(option)</td>
<td>14,20,21,115,126,158,175,E-1</td>
<td>installation</td>
<td>iii</td>
</tr>
<tr>
<td>DESIGN(command)</td>
<td>xxi,49,133</td>
<td>INST_NET</td>
<td>iii</td>
</tr>
<tr>
<td>DFRAC(command)</td>
<td>xxi,49</td>
<td>INTEREST</td>
<td>72,121,133,157,158</td>
</tr>
<tr>
<td>DFRAC(change)</td>
<td>67,121</td>
<td>international units</td>
<td>3</td>
</tr>
<tr>
<td>DHEAD(command)</td>
<td>xxi,49,129,138,173,174</td>
<td>IOEX</td>
<td>125</td>
</tr>
<tr>
<td>DHEAD(changes)</td>
<td>67,121b</td>
<td>IOUT</td>
<td>23</td>
</tr>
<tr>
<td>DHEAD(simul.)</td>
<td>90</td>
<td>IOUTI</td>
<td>23</td>
</tr>
<tr>
<td>DHEAD(special I)</td>
<td>19,20</td>
<td>IUTO</td>
<td>23</td>
</tr>
<tr>
<td>DIAMEters</td>
<td>83</td>
<td>IRESID</td>
<td>24</td>
</tr>
<tr>
<td>direct access file</td>
<td>145,E-1</td>
<td>ISIML</td>
<td>24,128,140,144,146</td>
</tr>
<tr>
<td>ISUNIT</td>
<td>83,128,144</td>
<td>NOMSOL</td>
<td>20,32,115,126,134,157,175</td>
</tr>
<tr>
<td>------------</td>
<td>------------</td>
<td>------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>ITERAtion</td>
<td>xxi,24,52</td>
<td>NOPARP</td>
<td>33</td>
</tr>
<tr>
<td>kinematic viscosity</td>
<td>11,45,65,131</td>
<td>NOPUMF</td>
<td>84</td>
</tr>
<tr>
<td>LCALGO</td>
<td>92,159</td>
<td>NOSTOP</td>
<td>33</td>
</tr>
<tr>
<td>LENGNON</td>
<td>xvi,24,117</td>
<td>NOZDIA</td>
<td>A-2</td>
</tr>
<tr>
<td>LENGTHs</td>
<td>67</td>
<td>NOZHG</td>
<td>A-2</td>
</tr>
<tr>
<td>LENORG</td>
<td>25</td>
<td>NOZZLE</td>
<td>xxi,5,52,149</td>
</tr>
<tr>
<td>LIFE</td>
<td>72,134,156</td>
<td>NPARAL</td>
<td>A-2</td>
</tr>
<tr>
<td>LINEAR</td>
<td>83,130,154</td>
<td>NPERCT</td>
<td>33</td>
</tr>
<tr>
<td>LNODE</td>
<td>xxi,52,194</td>
<td>NPGPM</td>
<td>11,20,34,111,141</td>
</tr>
<tr>
<td>LOOPRD</td>
<td>25,114</td>
<td>NPIPE</td>
<td>A-1</td>
</tr>
<tr>
<td>loops</td>
<td>27</td>
<td>NPLENG</td>
<td>20,34,112,121,150</td>
</tr>
<tr>
<td>LOOPSE</td>
<td>26</td>
<td>NPNRES</td>
<td>34,141</td>
</tr>
<tr>
<td>LOOPSR</td>
<td>A-1</td>
<td>NPOWE</td>
<td>A-2</td>
</tr>
<tr>
<td>LOOPSS</td>
<td>A-1</td>
<td>NPRINT</td>
<td>20,34,78,111</td>
</tr>
<tr>
<td>LPIPE</td>
<td>xxi,52,94</td>
<td>NPRI</td>
<td>75</td>
</tr>
<tr>
<td>loss coefficients</td>
<td>12,53,67</td>
<td>NPPRPV</td>
<td>34</td>
</tr>
<tr>
<td>MANNI</td>
<td>28</td>
<td>NPRPUM</td>
<td>35,154</td>
</tr>
<tr>
<td>mass flowrate</td>
<td>31,34</td>
<td>NPRRES</td>
<td>35,84,130,154</td>
</tr>
<tr>
<td>maximum velocity</td>
<td>45</td>
<td>NPRTAB</td>
<td>84</td>
</tr>
<tr>
<td>minimum velocity</td>
<td>45</td>
<td>NPRTRM</td>
<td>35</td>
</tr>
<tr>
<td>MINOR(command)</td>
<td>xxi,53,111,159</td>
<td>NPSERI</td>
<td>35,127,149</td>
</tr>
<tr>
<td>MINOR(change)</td>
<td>67</td>
<td>NPESRR</td>
<td>A-2</td>
</tr>
<tr>
<td>MTIME</td>
<td>83</td>
<td>NPUMP</td>
<td>A-1</td>
</tr>
<tr>
<td>NAMELIST</td>
<td>10</td>
<td>NPUNOD</td>
<td>84,130,141,154</td>
</tr>
<tr>
<td>MAX</td>
<td>24</td>
<td>NRES</td>
<td>A-2</td>
</tr>
<tr>
<td>NBPUMP</td>
<td>A-1</td>
<td>NSORTP</td>
<td>36</td>
</tr>
<tr>
<td>NBPV</td>
<td>A-2</td>
<td>NSYMMT</td>
<td>36,154</td>
</tr>
<tr>
<td>NCKVAL</td>
<td>A-2</td>
<td>NTIMED</td>
<td>84,159</td>
</tr>
<tr>
<td>NCONTI</td>
<td>28</td>
<td>NTRAND</td>
<td>38,145,147</td>
</tr>
<tr>
<td>NELEV</td>
<td>28,114,130,</td>
<td>NUMPIP</td>
<td>38,114,123,136</td>
</tr>
<tr>
<td>NEQUAT</td>
<td>29</td>
<td>NUNIT</td>
<td>20,38,136,141</td>
</tr>
<tr>
<td>NETCHK</td>
<td>29,114,130,134,154</td>
<td>NVALVE</td>
<td>A-2</td>
</tr>
<tr>
<td>NETPLT</td>
<td>6,30,166,167</td>
<td>OFPLOT</td>
<td>C-7</td>
</tr>
<tr>
<td>NEWSTR</td>
<td>31</td>
<td>OPERATing</td>
<td>134</td>
</tr>
<tr>
<td>NEXT CARD</td>
<td>E-4</td>
<td>options</td>
<td>10</td>
</tr>
<tr>
<td>NFLOW</td>
<td>11,21,31,103,140</td>
<td>OPTIONS</td>
<td>E-4</td>
</tr>
<tr>
<td>NHAZEN</td>
<td>32</td>
<td>order of input</td>
<td>A-1</td>
</tr>
<tr>
<td>NLNODE</td>
<td>xxi,52,94</td>
<td>OUTPUTI</td>
<td>38,111</td>
</tr>
<tr>
<td>NLPIPE</td>
<td>xxi,52,94</td>
<td>orifice</td>
<td>52</td>
</tr>
<tr>
<td>NOCHK</td>
<td>32</td>
<td>output</td>
<td>2,4</td>
</tr>
<tr>
<td>NODE DEMAND</td>
<td>E-3</td>
<td>PARAL(command)</td>
<td>xxi,55,122,161</td>
</tr>
<tr>
<td>NODE HGL</td>
<td>E-3</td>
<td>PARALL(change)</td>
<td>67,122</td>
</tr>
<tr>
<td>NODE PRESSURE</td>
<td>E-3</td>
<td>PAYMENT</td>
<td>74</td>
</tr>
<tr>
<td>NODE TABLE</td>
<td>85,130,161</td>
<td>PCALGO</td>
<td>94,159</td>
</tr>
<tr>
<td>NODES</td>
<td>xxi,53,76</td>
<td>PCHAR3</td>
<td>43,130,135</td>
</tr>
<tr>
<td>NODESO</td>
<td>32</td>
<td>peak factor</td>
<td>44</td>
</tr>
<tr>
<td>NODESP</td>
<td>xvi,11,32,112,117</td>
<td>PEAKF(option)</td>
<td>44,136</td>
</tr>
<tr>
<td>NOMDIA</td>
<td>30,32</td>
<td>PEAKF(command)</td>
<td>xxi,55</td>
</tr>
</tbody>
</table>
This index does not cite all pages where commands or options are used. Generally only the page number where the command or option is described is given, and one or two of example problems where the command, or option, is used are cited. For example if the commands PIPES and NODES were cited for each example problem in which they are used, then practically all pages giving input listing in the example problems would be cited. Command names, Option names, and Program names are capitalized.