186
874

684

Y-1916

97,564

**FIRE SPRINKLER NETWORK
ANALYSIS PROGRAM**

J. T. Blackmon
E. T. Stickle

**UNION
CARBIDE**

OAK RIDGE Y-12 PLANT
OAK RIDGE, TENNESSEE

*prepared for the U.S. ATOMIC ENERGY COMMISSION
under U.S. GOVERNMENT Contract W-7405 eng 26*

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road, Springfield, Virginia 22151
Price: Printed Copy \$5.45; Microfiche \$0.95

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

FIRE SPRINKLER NETWORK ANALYSIS PROGRAM

J. T. Blackmon

E. T. Stickle

Oak Ridge Y-12 Plant

P.O. Box Y, Oak Ridge, Tennessee 37830

NOTICE

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

Date Issued - April 2, 1974

Prepared for the U.S. Atomic Energy Commission
Under U.S. Government Contract W-7405-eng-26

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

ABSTRACT

Hydraulically designed fire protection systems allow the designer to supersede the existing rules for pipe sizing. The advent of large industrial buildings, exhibition halls, high-rise buildings, and warehouses with their new techniques of high-piles storage have necessitated a change in philosophy from set rules to hydraulically designed systems. Computers assist in the design of systems and permit a more efficient use of a designer's time by reducing the tedious time-consuming calculations that are the natural result of using the hydraulically designed system. This report describes how the n -dimensional Newton-Raphson solution technique is utilized to reduce computer time and the storage required for the solution of a problem.

CONTENTS

SUMMARY	4
INTRODUCTION	5
SPRINKLER NETWORK ANALYSIS	6
Y-12 Plant Methods	6
Hardy-Cross Method	6
Matrix Method	6
Usage	7
General Job Information	7
Operation and Setup Instructions	7
Error Checks and Diagnostic Printout	13
Physical System and Problem Definition	13
Network Preparation	13
Programs and Equations	14
Output List for Problem Solution	14
Fortran List	16
Limitations	62
Sample Problem	62
REFERENCES	73
APPENDIX	74
Basic Program Description	74

SUMMARY

The computer program presented herein has been used successfully for approximately three years at the Oak Ridge Y-12 Plant^(a) and was an adaption of a program developed at the University of Michigan by M. A. Stoner.^(b) This program is an improvement over previous computer programs utilized by the Y-12 Plant prior to implementation of this system.

This program will provide a direct solution for combinations of unknowns which include pressures (heads), water consumption (sprinkler discharge), pipeline resistances, and pipeline sizes. The Hazen-Williams equation, commonly used for fire main and sprinkler calculations, is the law relating flow and pipeline energy. Sprinkler heads are treated as orifices. Sprinkler heads, pipelines, and junction locations may be added or subtracted as required to suit the problem or to change the conditions.

-
- (a) Operated for the US Atomic Energy Commission by the Union Carbide Corporation's Nuclear Division.
 - (b) Stoner, M. A.; "A General Gaussion Elimination Solution of a Set of Simultaneous Linear Equations Using Spence Matrix Programming," *Flow of Natural Gas in Transmission & Distribution Systems*, C-106; University of Michigan Engineering Summer Conference, Ann Arbor, Michigan; June 1970.

INTRODUCTION

In 1895, a meeting took place in the Boston office of Everett U. Crosby.⁽¹⁾ The group that met that day did not know that their small meeting would eventually evolve into what is now known as the National Fire Protection Association (NFPA). Their concern on that day was simply what to do about the nine different standards being used to size sprinkler piping and space sprinkler heads. Their conclusions were that sprinkler protection was worthwhile and that standards must be developed and enforced.

The first edition of NFPA Standard 13, entitled "Standard for the Installation of Sprinkler Systems", appeared in 1896.⁽²⁾ In the 75 years since the appearance of the first edition, some 41 subsequent editions have appeared. In the standard, the method of pipe sizing is by use of pipe schedules that stipulate how many heads may be installed on a given pipe size. These schedules, which are based upon loss experience, judgment, and hydraulic considerations, perform well for most light and ordinary hazard occupancies.⁽³⁾ However, the advent of large industrial buildings and exhibition halls, with their myriad of combustible types and material-handling arrangements, have necessitated a change of philosophy in the approach to the design of sprinkler systems.

During the 1960s, the sprinkler code writing groups became cognizant of this change in philosophy by recognizing the hydraulically designed sprinkler system. Prior to this time, in the late 1950s, the Atomic Energy Commission recognized this need and implemented a policy of hydraulically designed systems for its Gaseous Diffusion Plants in Oak Ridge, Paducah, and Portsmouth. In the Y-12 Plant, hydraulically designed systems are the standard, rather than the exception, in large, multistory, multiple-occupancy buildings, especially in areas containing high-pressure hydraulic oil systems and high-rack storage areas.

A hydraulically designed system permits the selection of pipe sizes on a pressure-loss basis to provide a prescribed density, distributed with a reasonable degree of uniformity over a specified area in accordance with the characteristics of the water supply available.⁽⁴⁾ Prior to the advent of the large-scale use of high-speed digital computers, hydraulic calculations presented a problem to the designer as he was required to expend many manhours in tedious, time-consuming calculations. These calculations were particularly difficult unless the system under consideration had a certain degree of symmetry. This factor made a study of alternate designs an impractical chore.

SPRINKLER NETWORK ANALYSIS

Y-12 PLANT METHODS

Hardy-Cross Method

In 1963, the Engineering Division at the Oak Ridge Gaseous Diffusion Plant adapted a program written by the Los Angeles County Engineer for use in studies of the Plant underground fire mains. In 1966, the authors of this report adapted that program for use in checking the hydraulics of fire sprinkler systems. This program used the Hardy-Cross Iterative Method for achieving the solution by balancing pressure drop around loops.

The Hardy-Cross method is the one most commonly used for hand calculations, and many computer programs exist to perform the work.⁽⁵⁾ The program used by the Y-12 Plant had several undesirable features: (1) it required initial values of the flow in each line before the main calculations could begin; (2) when a large network was encountered, the procedure was excessively time consuming; (3) the program required a rather precise method of numbering lines, loops and junctions; and (4) when it was necessary to add sprinkler heads, additional lines, or loops, the system required a complete renumbering of lines and junctions. In addition, elements such as pumps or control valves could not be added directly to the program because of the solution method (Hardy-Cross).

Matrix Method

With the aforementioned limitations in mind, an adaptation was made to an existing program that utilizes the n-dimensional Newton-Raphson solution technique (Appendix). The simultaneous equations derived from this technique during each iteration are solved by Gaussian elimination. Sparse matrix programming is utilized to reduce the computer time and storage required for this solution.

This program will solve directly for combinations of unknowns which include pressures (heads), water consumption (sprinkler discharge), pipeline resistances, and pipeline sizes. The particular version of the program contained in this report is not capable of incorporating pumps or pressure control valves, but these elements could be directly incorporated by a program change. The Hazen-Williams equation, commonly used for fire main and sprinkler calculations, is the law relating flow and pipeline energy loss. Sprinkler heads are treated as orifices.

The advantages of the use of this matrix program are:

1. The program has flexibility for handling changes in input data.
2. The program, as implemented, allows selection of unknown pressures, sprinkler outflows, or pipe resistances.
3. Sprinkler heads (nodes), pipelines (node-connecting elements), and junction locations (nodes) may be added or subtracted as required to suit the problem or to change the conditions. This choice makes it particularly advantageous when analyzing an existing sprinkler system that may need altering because of changes in occupancy or the process.

USAGE

This section of the report covers the application of the Matrix program and serves as an instruction manual for its use. It includes detailed instructions for preparing the input data and an illustrative example.

General Job Information

The program calculates the flow in each line in the network, the pressure at every junction, the flow out of each sprinkler head, and the percent deviation from normal flow for every sprinkler head.

The data required includes the diameter, length, elevation, and Hazen-Williams coefficient for each pipe in the network, the K factor and elevation for each sprinkler-head orifice, and an estimated flow into the network. The program is applicable to many different network configurations and, while a prescribed convention of numbering lines and junctions should be followed, it is not mandatory.

Operation and Setup Instructions

Type of Job: IBJOB Monitor, Fortran IV.

Security Classification: Unclassified.

Required Service: Normal handling. Execution time varies greatly due to the great variation in the complexity of networks. (Some long networks may require up to 10 minutes on an IBM 360/91.)

Input Data Information:

1. All input to the program is on cards keypunched from data sheets and arranged as shown in Figure 1. The number of cards depends on the type of system being handled.

2. Format description of input data (for one network):

(a) DIN card: One card required. This card is a NAMELIST DATA card (see any IBM Fortran IV manual) and has the following format:

Column 1 Blank.

Columns 2 - 5 &DIN.

Column 6 Blank.

Columns 7 - 15 TOL = X.XX, where a value, such as 0.05, replaces the Xs. This is the tolerance on flow imbalance at each node that must be satisfied before the network is considered balanced. It has been determined that 0.05 is a good value. Decreasing the tolerance

Figure 1. DATA CARDS REQUIRED FOR EACH SPRINKLER NETWORK.

to a value lower than 0.05 increases the computer time without significantly affecting the results.

- Columns 16 - 41 MA = 25,1,20,20,2,5,2,1,0,0 . These variables are described in the Appendix under Subroutine BALALG on Page C-82 and can be changed to suit the present needs. ISN 0006 of the main program assigns values to this variable; and, if these values are satisfactory, the given values should not be included on this card.
- Columns 42 - 45 &END. If the MA values are left off, this would be in Columns 16 - 19.

(b) Basic Network Data: one card required.

- Columns 1 - 4 Number of data cards, not including DIN, BASIC NETWORK, DENSITY, and RERUN cards. Integer right adjusted. This value should equal the number of lines and heads in the network.
- Column 5 Blank.
- Columns 6 - 9 Key node identification, alphameric. (Key sprinkler head or location of flow into system.)
- Column 10 Blank.
- Columns 11 - 14 Pressure in psig at key node. Decimal assumed between Columns 13 and 14.
- Column 15 Blank.
- Columns 16 - 22 Key node elevation in feet. Decimal assumed between Columns 20 and 21.
- Column 23 Blank.
- Columns 24 - 29 Factor. (This variable was designed to be used for another job, not sprinkler systems. For sprinkler work, its value is 1.0.) Decimal assumed between Columns 24 and 25.
- Columns 30 - 32 Blank.
- Columns 33 - 80 Alphameric identification of network.

(c) Line Data: one card for each line in network. (Instructions differ slightly if the key node is a sprinkler, see Subsection (d), Page 11, Column 26.)

Columns 1 - 4	Name of the predecessor node. The line direction is away from this node. Alphameric field.
Column 5	Blank.
Columns 6 - 9	Name of the successor node. The line direction is toward this node. Alphameric field.
Column 10	Blank.
Columns 11 - 12	Type of node-connecting element. Alphameric field (Pipe P1; Check Valve CV; Sprinkler Head S1).
Columns 13 - 15	Blank.
Column 16	An asterisk is located in the pressure-status column and should appear on all line cards except the key node. The asterisk is associated with the successor node. Note: Asterisks indicate unknowns to be calculated. The total number of asterisks on all cards (line data and sprinkler-head data) must equal the number of nodes.
Column 17	Blank.
Columns 18 - 24	Estimated flow in gpm entering or leaving the key successor node. Flow into the network is positive. Decimal assumed between Columns 23 and 24. A flow estimate is required only for the line leading to the key node. On other line cards this space is blank.
Column 25	Blank.
Column 26	An asterisk is located in the flow status column only when a nonzero flow estimate appears in Columns 18 - 24 (ie, when successor is the key node). Although flow will be calculated for every line, only one asterisk is required.
Column 27	Blank.
Columns 28 - 32	Diameter in inches of pipe between the predecessor node and successor node. Decimal assumed between Columns 30 and 31.
Columns 33 - 38	Line length or equivalent length of pipe in feet between predecessor node and successor node. Decimal assumed between Columns 37 and 38.
Columns 39 - 42	Hazen-Williams pipe coefficient. Decimal assumed at right of Column 42.

Columns 43 - 49	Elevation in feet of successor node. Decimal assumed between Columns 47 and 48.
Column 50	Blank.
Columns 51 - 56	Pressure in psig at the successor node. Only one pressure is to be given for one network at one time and should be given on the card whose successor is the key node. This is not an estimate but is the actual pressure available to the system. Decimal assumed between Columns 54 and 55.
Columns 57 - 76	Blank.
Columns 77 - 80	Alphameric identification of line (Example, L1).

(d) Sprinkler Head Data: One card per sprinkler head.

Columns 1 - 4	Alphameric identification of sprinkler connection. (Example: S1 would be the fitting into which a sprinkler head is placed.)
Columns 5 - 9	Alphameric identification of sprinkler head. (Example: H1 would be the actual sprinkler head.)
Column 10	Blank.
Columns 11 - 12	S1.
Columns 13 - 17	Blank.
Columns 18 - 24	Approximation of flow in gpm out of the sprinkler head. (The closer the guess to the calculated answer, the faster the program will run.) Decimal assumed between Columns 23 and 24.
Column 25	Blank.
Column 26	Asterisk. It is possible to use a sprinkler as the key node, in which case the flow given in Columns 18 - 24 would be the demanded flow from the key node and Column 26 of the key node card would not contain an asterisk. If the key node is a sprinkler, every line data card must have an asterisk in Column 16 and the line card whose successor is the key node will still contain an asterisk in Column 26 and a pressure in Columns 51 - 56. The number of asterisks will still equal the number of nodes. If this method is used, all line pressures will be dependent variables.
Column 27	Blank.

Columns 28 - 32 K factor of the sprinkler head. Decimal assumed between Columns 30 and 31. (K factors may be found in sprinkler manufacturers catalogues, Underwriters, or Factory Mutual approval lists.)

Columns 33 - 76 Blank.

Columns 77 - 80 Alphameric identification of sprinkler line. (Example: Add the letter K before the sprinkler connection number given in Columns 1 - 4; ie, K1, K10.)

- (e) Density and Rerun Subroutines: If the designer does not wish to receive the information concerning density and/or have the program automatically recalculated, insert blank cards in the data deck at the end of the program to represent these functions. If these functions are wanted, the following cards must be substituted for the blank cards:

(1) Density

Columns 1 - 4 Line name whose successor node is the node where the water enters the system.

Column 5 Blank.

Columns 6 - 10 Total square feet covered by the system.

Column 11 Blank.

Columns 12 - 16 Coverage in square feet per sprinkler head.

Columns 17 - 80 Blank.

- (2) Rerun. With this subroutine, the key pressure and key node may be reset. For example, if on the first run the key node was associated with the farthest sprinkler head, the node where water enters the system can be made the key node and a new key pressure assigned to this node.

Columns 1 - 4 New key node name.

Column 5 Blank.

Columns 6 - 9 Pressure at new key node. Decimal assumed between Columns 8 and 9.

Column 10 Blank.

Columns 11 - 14 Line name whose successor node is the new key node.

ERROR CHECKS AND DIAGNOSTIC PRINTOUT

The input data are printed out to facilitate checking of decimal points and other information. All other diagnostic printouts are explained in the Appendix.

PHYSICAL SYSTEM AND PROBLEM DEFINITION

Network Preparation

A sketch of the network should first be made showing all items such as lines, pipe junctions, sprinkler heads, and elevation changes. Each sprinkler head should be represented as a node as should all actual pipe connections. Any numbering or naming system may be used since subroutines in the program do the actual indexing. The method used at Y-12 is as follows:

1. Starting at the junction (node) where water enters the system, all pipe junctions (nodes), except sprinkler heads, are numbered 1, 2, 3,... in increasing order (see Figure 4).
2. All sprinkler-head nodes are named in a convenient manner, starting with S1, S2, S3, until all such junctions are numbered.
3. Line directions are assigned to each line in the system. These assigned directions have nothing to do with the actual direction of flow, but they are used to determine the actual direction of flow in each line of the balanced network (see sample problem results on Page 62). The friction loss in each line (fourth column on Page 67) will be positive or negative. If it is positive, the actual flow is in the same direction as the assigned line direction; if it is negative, the actual flow is in the opposite direction from the assigned line direction. Note: The line direction of sprinkler heads must always be toward the head (Example: S1 to H1).
4. After all line directions have been assigned, every node should be checked to see that at least one line is entering that node. If this sequence doesn't occur, the line direction of one of the lines connected to the junction should be changed to make the condition true. The reason for this requirement is that the elevation and flow on a line data card and the flow on a sprinkler data card are associated with the successor node. If no lines enter a node, it will never be a successor node and the elevation or flow of that node will be missing. If it is not possible to change the line direction of one of the lines connected to the node (as could be the case if only one line were connected to the node), then a line of 0.0 diameter and any length can be added to make the node a successor node. (See Line LO in Figure 4.) This step should be avoided whenever possible since the computer time increases whenever a line is added.
5. The sprinkler head which is thought to have the least pressure (usually the one farthest and highest from the supply point) may be designated as the key node, and the pressure assigned to this node should be such that the calculated flow from that sprinkler head is the estimated flow needed to achieve the specified density as required for the problem. The program will then calculate the flow from the remaining heads and the flow and pressure input to the system which is required to yield the desired flow at the key node. Initially, all sprinkler-head nodes which are shown as flowing must be assigned this

estimated flow (or a better estimate if one is available). Alternately, the junction where water enters the system may be designated as the key node. The pressure assigned to this node should be the actual pressure available from the existing water source. In this case, the program will calculate the performance of the system at the given source pressure. Note: Computer time is reduced if the key node is not a sprinkler.

Programs and Equations

Program - The working philosophy and mechanics of the main program and subroutines are described in the Appendix. The Fortran lists are contained in the section that follows.

Equations -

1. To compute relative resistance of each pipe:

$$F = \frac{10.43 S}{C^{1.85} D^{4.87}}$$

where:

- F represents the relative resistance,
- S the length in feet,
- C the Hazen-Williams coefficient, and
- D the diameter in inches.

2. To compute sprinkler outflow:

$$Q = K \sqrt{P}$$

where:

- Q represents the flow in gallon per minute,
- K the sprinkler orifice coefficient, and
- P the pressure in pounds per square inch.

Output List for Problem Solution

The following information will be printed for the problem answer (reading from right to left):

1. Line number or sprinkler line number.
2. Type element of the pipeline (P1 or S1).

3. Flow in gpm through the pipeline.
4. Loss of psi because of friction in the pipeline.
5. Loss of psi due to elevation change in the pipeline.
6. Number of the predecessor node.
7. Number of the successor node.
8. Pressure in psig maintained at the successor node. (Pressure is not listed in the sprinkler section of the printout.)
9. Flow out of the system from the sprinkler node.
10. Residual imbalance at each node. If the system is perfectly balanced, the algebraic sum of the flows at a node would be zero. The program considers the system balanced when all residuals are less than a specified tolerance value. See DIN Card, Page 7.
11. Immediately following the above data, the program prints out the line number of any line whose friction loss is greater than 3.0 psi and lists the recommended diameter (in terms of the inside diameter of schedule 40 pipe) and the approximate resulting friction loss. This friction loss is approximate because the program does not rebalance the network using the recommended diameters. The printout of the results from the DENSITY subroutine will follow this if a DENSITY card has been included. The printout of the results from the RERUN subroutine will follow this if a RERUN card has been included.

FORTTRAN LIST

LEVEL 20.1 (AUG 71)

DS/360 FORTTRAN H

COMPILER OPTIONS - NAME= MAIN,OPT=00,LINECNT=60,SIZE=0000K,
SOURCE,ERCDIC,NOLIST,NODECK,LOAD,MAP,NOCFIT,TD,NOXREF

```

C
C      ** REVISED VERSION OF ISAP FOR WATER SYSTEM ANALYSIS **
C      THIS IS THE MAIN PROGRAM FOR THE REVISED VERSION OF THE
C      INTEGRATED SYSTEM ANALYSIS PROGRAM (ISAP) FOR WATER DISTRIBUTION
C      NETWORKS. THE ORIGINAL PROGRAM WAS PREPARED FOR NATURAL GAS
C      DISTRIBUTION NETWORKS BYO M.A. STICKER, PH.D.
C      GAS SUPPLY PLANNING DEPARTMENT
C      CONSUMERS POWER COMPANY
C      JACKSON, MICHIGAN
C      THE REVISED PROGRAM IS USED TO BALANCE A SYSTEM OF NETWORK
C      EQUATIONS WHICH CAN INCLUDE PIPELINES, BOOSTER PUMPS, CHECK VAL-
C      VES AND HEAD TANKS. THE HAZEN AND WILLIAMS EMPIRICAL FORMULA
C      FOR FLOW OF WATER IS THE PRESSURE-FLOW RELATIONSHIP USED.
C      THE NEWTON-RAPHSON SOLUTION IS IMPLEMENTED BY GAUSSIAN ELIMIN-
C      ATION USING SPARSE MATRIX PROGRAMMING.
C      THIS REVISED PROGRAM WAS PREPARED BYO E.T. STICKLE
C      ENGR. MECHANICS DEPART.
C      Y-12 PLANT
C      OAK RIDGE, TENNESSEE
C
ISN 0002      IMPLICIT INTEGER*2(I-N)
ISN 0003      INTEGER*4 LT1,LT3,NFN,NTN,NODE,IDENT,KFYJUN
ISN 0004      DIMENSION NCETYP(4),MA(10),IS(2),AD(12),
C      QG(600),MMQ(600), LNFM(10),LNTN(10),LNCLL(10),LIUN(1
20),LMMP(10),OQQ(10),LMMQ(10),OX1(10),OX2(10),OX3(10),OX4(10),OCONP
3(10),LIDENT(10),KHVZ(10),JJJ(10)
ISN 0005      DATA NCETYP/'P','S','CV','IQUES','/',BLANK/'/'
ISN 0006      DATA MA/50,3,50,10,2,4,3,0,1,50/
ISN 0007      COMMON LT1,LT3, NFN(600),NTN(600),NODE(600),A(2000), CC(600)
1,DELV(600),F(600),FF(600),P(600),Q(600),CN(600),TOL,X1(600),X2(600
2),X3(600),X4(600),ICENT(600),CONP(600),MMP(600),IA(600),INDF(600),
3IPN(7000),IPT(600),IPTIPN(300),IRIM(600),IQIM(600),IUN(600),IX(180
40),IXR(5000),JA(3000),NCEL(600),NFUN(600),NPUN(600),NN,NNCE,NNI,NU
5F,NUP,NUNCE,INDP(600),INDR(600),IPTM(600),NCLL(600)
ISN 0008      NAMELIST/DIN/TOL,MA
ISN 0009      LT3=5
ISN 0010      LT3=6
ISN 0011      1 READ (LT1,DIN,END=999)
ISN 0012      WRITE(LT3,2)
ISN 0013      2 FORMAT('H',' SYSTEM VARIABLES AND EXECUTION CONTROLS')
ISN 0014      WRITE(LT3,DIN)
ISN 0015      KV=0
ISN 0016      775 DO 3 I=1,600
ISN 0017      DELV(I)=BLANK
ISN 0018      3 IRIM(I)=0
ISN 0019      DO 8 I=1,500
ISN 0020      IA(I)=0
ISN 0021      IPIM(I)=0
ISN 0022      8 IQIM(I)=0
ISN 0023      INDR(I)=0
ISN 0024      IF(KV,GT,0)GO TO 700
C      READ SYSTEM IDENTIFICATION CARD

```

ISN 0026	READ(LT1,65) NNCE,KEYJJN,PRESKY,ELEVKY,FACTOR,JUPD,(AD(I),I=1,12)	47
ISN 0027	65 FORMAT(14,1X,A4,1X,F4.2,1X,F7.2,1X,F6.5,1X,1,1X,12A4)	48
	C READ SYSTEM DESCRIPTION CARDS. NUMBER TO BE READ IS DETERMINED	49
	C FROM SYSTEM ID CARD	50
ISN 0028	READ(LT1,100)(NFN(I),NTN(I),NCLL(I),TUN(I),MMP(I),QQ(I),MMQ(I),X1	51
	1(I),X2(I),X3(I),X4(I),CONP(I),IDENT(I),I=1,NNCE)	52
ISN 0029	100 FORMAT(A4,1X,A4,1X,A2,1X,A1,1X,A1,1X,F7.1,1X,A1,1X,F5.2,F6.1,F4.0	53
	1,F7.2,1X,F6.2,20X,A4)	54
	C IF MA(8)=1, SYSTEM DESCRIPTION DATA WILL BE WRITTEN	55
ISN 0030	IF(MA(8).NE.1)GO TO 400	56
ISN 0032	IF(FACTOR .EQ. 1.0) GO TO 700	A56
ISN 0034	DO 200 I=1,NNCE	B56
ISN 0035	QQ(I)=QQ(I)*FACTOR	C56
ISN 0036	200 CONTINUE	D56
ISN 0037	700 WRITE(LT3,101)(AD(I),I=1,12)	57
ISN 0038	101 FORMAT(1H,10X,'SYSTEM DESCRIPTION DATA FOR '12A4//' NCE FROM	58
	1 TO NCE NCE PRESS. FLOW FLOW DIAM. LENGTH	59
	2 C ELEV. CONSTANT// NUM NODE NODE TYPE STATUS	60
	3 STATUS (GPM) STATUS (IN.) (FT) FACTOR (FT) PRE	61
	4SSURE'//)	62
ISN 0039	WRITE(LT3,102)(IDENT(I),NFN(I),NTN(I),NCLL(I),TUN(I),MMP(I),QQ(I),	63
	1MMQ(I),X1(I),X2(I),X3(I),X4(I),CONP(I),I=1,NNCE)	64
ISN 0040	102 FORMAT(1H,A4,3X,A4,3X,A4,5X,A2,5X,A1,7X,A1,2X,F9.2,4X,A1,5X,F5.2	65
	1, F9.2,3X,F5.0,3X,F8.2,3X,F7.2)	
ISN 0041	400 DO 410 I=1,NNCE	67
ISN 0042	DO 105 J=1,3	68
ISN 0043	IF(NCLL(I)-NCETYP(J))105,104,105	69
ISN 0044	104 NCLL(I)=J	70
ISN 0045	GO TO 410	71
ISN 0046	105 CONTINUE	72
ISN 0047	GO TO 173	72
ISN 0048	410 CONTINUE	74
ISN 0049	GO TO 405	A74
ISN 0050	113 WRITE(LT3,111)	75
ISN 0051	111 FORMAT(10 'INCORRECT NODE CONNECTING ELEMENT TYPE SPECIFICATION')	76
ISN 0052	GO TO 848	
ISN 0053	405 CALL INDX1	78
ISN 0054	IF(KV.GT. 0) GO TO 125	A78
	C	79
	C	80
	C PRINT NODE CORRESPONDENCE TABLE AND LIST OF ADJACENT NODES.	A80
ISN 0056	WRITE(LT3,121)	B80
ISN 0057	121 FORMAT(1 'NODE CORRESPONDENCE TABLE AND LIST OF ADJACENT NODES'	C80
	1//2X,'INTERNAL '3X,'NODE',5X,'ADJACENT NODES(BY NAME)'	D80
	2' NODE NUMBER',2X,'NAME'//)	E80
ISN 0058	DO 122 I=1,NN	F80
ISN 0059	II=IPT(I)+3	G80
ISN 0060	III=II+IXR(IPT(I)+1)*2-3	H80
ISN 0061	122 WRITE(LT3,123) I,NODE(I),(NODE(IXR(J)),J=II,III,2)	I80
ISN 0062	123 FORMAT(3X,14,5X,A4,5X,10(A4,1X))	J80
	C	81
	C	82
	C	83
	C SET UP A VECTOR OF THE NODE CONNECTING ELEMENTS WITH UNKNOWN	84
	C PARAMETERS.	85

ISN 0063	125 NUNCE=C	86
ISN 0064	DO 130 I=1,NNCE	87
ISN 0065	IF(IUN(I).NE.IQUES) GO TO 130	88
ISN 0067	NUNCE=NUNCE+1	89
ISN 0068	INDR(NUNCE)=I	90
ISN 0069	130 CONTINUE	91
ISN 0070	NUP=0	92
ISN 0071	NUF=0	93
ISN 0072	DO 420 I=1,NNCE	94
ISN 0073	DO 46 N=1,MN	95
ISN 0074	IF(NTN(I).EQ.NODE(N))GO TO 45	96
ISN 0076	46 CONTINUE	97
ISN 0077	45 IF(IA(N).EQ.1)GO TO 420	98
ISN 0079	IA(N)=1	99
ISN 0080	NFUN(N)=MMQ(I)	100
ISN 0081	NPUN(N)=MMP(I)	101
ISN 0082	QN(N)=CQ(I)	102
	C	103
	C THE MAXIMUM STATIC PRESSURE (DUE TO ELEVATION) IS USED AS THE	104
	C INITIAL ESTIMATE OF THE UNKNOWN PRESSURE AT EACH JUNCTION.	105
ISN 0083	IF(KV .GT. 0) GO TO 423	A105
ISN 0085	IF(NPUN(N).NE.IQUES)GO TO 423	106
ISN 0087	P(N)=.4331*(ELEVKY-X4(I))+PRESKY	107
ISN 0088	GO TO 425	A107
ISN 0089	423 P(N)=CCNP(I)	AA107
ISN 0090	425 IF(P(N).LT.0.1)GO TO 420	108
	C	109
	C DETERMINE THE 'UNKNOWN' STATUS OF NODE'S PRESSURE AND FLOW.	110
	C MAKE THE APPROPRIATE ENTRY(S) IN UNKNOWN NODE LISTS.	111
ISN 0092	426 IF(MMP(I).NE.IQUES)GO TO 48	112
ISN 0094	NUP=NUP+1	113
ISN 0095	INDP(NUP)=N	114
ISN 0096	48 IF(MMQ(I).NE.IQUES)GO TO 420	115
ISN 0098	NUF=NUF+1	116
ISN 0099	INDF(NUF)=N	117
ISN 0100	420 CONTINUE	118
ISN 0101	GO TO 50	119
ISN 0102	430 WRITE(LT3,51) NTN(I),P(N)	120
ISN 0103	51 FORMAT(1X,'NODE ',A4,'HAS INITIAL PRESSURE ESTIMATE OF ',F7.2,' PS	121
	1IG')	122
ISN 0104	GO TO 848	
ISN 0105	50 DO 450 I=1,NNCE	124
ISN 0106	KU=IX(3*I-1)	125
ISN 0107	KD=IX(3*I)	126
ISN 0108	DO 445 J=1,NNCE	127
ISN 0109	KDOFKU=IX(3*J)	128
ISN 0110	IF(KDOFKU.EQ.KU)GO TO 445	129
ISN 0112	445 CONTINUE	130
ISN 0113	WRITE(LT3,460) NODE(KU)	131
ISN 0114	460 FORMAT(1X,'NODE ',A4,' IS NOT A TO NODE FOR ANY NCE')	132
ISN 0115	446 DO 447 K=1,NNCE	134
ISN 0116	KD1=IX(3*K)	135
ISN 0117	IF(KD1.EQ.KD)GO TO 449	136
ISN 0119	447 CONTINUE	137
ISN 0120	449 IF(NCELI(I).EQ. 2) GO TO 450	A137

ISN 0122	IF((X4(J).GT.0).AND.(X4(K).GT.0)) GO TO 451	128
ISN 0124	WRITE(LT3,465) NODE(KU),X4(J),NODE(KD),X4(K)	129
ISN 0125	465 FORMAT(1X,'THE ELEVATION OF NODE ',A4,' IS ',F8.2/' THE ELEVATION	140
	OF NODE ',A4,' IS ',F8.2)	141
ISN 0126	GO TO 848	
ISN 0127	451 DELV(I)=-.4331*(X4(K)-X4(J))	144
ISN 0128	450 CONTINUE	145
	C	149
	C WRITE OUT PROBLEM SUMMARY	150
ISN 0129	WRITE(LT3,172) NN,NNCE,NUNCE,NUP,NUF	151
ISN 0130	172 FORMAT(' - *** PROBLEM SUMMARY ***/'0',4X,14,' NODES IN THE SYSTE	152
	'M'/5X,14,' NODE CONNECTING ELEMENTS'/5X,14,' UNKNOWN NCE PARAMETER	153
	2S'/5X,14,' UNKNOWN PRESSURES'/5X,14,' UNKNOWN FLOWS')	154
ISN 0131	IF(NUP+NUF+NUNCE-NN) 179,178,179	155
ISN 0132	179 WRITE(LT3,176)	156
ISN 0133	176 FORMAT('0 NUMBER OF UNKNOWNNS DOES NOT EQUAL NUMBER OF NODES')	157
ISN 0134	GO TO 848	
	C	159
	C NODE INFORMATION HAS BEEN SUCCESSFULLY READ IN . SET UP INDEX	160
	C VECTORS FOR EFFICIENT PROBLEM SOLUTION.	161
ISN 0135	178 CALL INDX2	162
	C	163
	C PRINT OUT LIST OF INDEX VECTORS.	164
ISN 0136	IN1=IPT(NN)+IXP(IPT(NN)+1)*2+1	165
ISN 0137	IN2=IA(NN1)-1	166
ISN 0138	IN3=IPTIPN(NN1)-1	167
ISN 0139	WRITE(LT3,140) IN1,IN2,IN3	168
ISN 0140	140 FORMAT('1 LIST OF INDEX VECTORS. INTERNAL NUMBERING USED.'/'0	169
	1MAXIMUM INDICIES OF SYSTEM DEPENDENT INDEX VECTORS.'/'0 IXP',15/	170
	2' JA ',15/' IPN',15)	171
ISN 0141	WRITE(LT3,141) (INDR(I),I=1,NUNCE)	172
ISN 0142	141 FORMAT('0 INDR'/(10X,10I5))	173
ISN 0143	WRITE(LT3,142) (INDP(I),I=1,NUP)	174
ISN 0144	142 FORMAT('0 INDP'/(10X,10I5))	175
ISN 0145	WRITE(LT3,143) (INDF(I),I=1,NUF)	176
ISN 0146	143 FORMAT('0 INDF'/(10X,10I5))	177
ISN 0147	WRITE(LT3,144) (IRIM(I),I=1,NNCE)	178
ISN 0148	144 FORMAT('0 IRIM'/(10X,10I5))	179
ISN 0149	WRITE(LT3,145) (IPIM(I),I=1,NN)	180
ISN 0150	145 FORMAT('0 IPIM'/(10X,10I5))	181
ISN 0151	WRITE(LT3,146) (IQIM(I),I=1,NN)	182
ISN 0152	146 FORMAT('0 IQIM'/(10X,10I5))	183
ISN 0153	WRITE(LT3,147) (IPTIPN(I),I=1,NN)	184
ISN 0154	147 FORMAT('0 IPTIPN'/(10X,10I5))	185
ISN 0155	WRITE(LT3,148) (IPN(I),I=1,IN3)	186
ISN 0156	148 FORMAT('0 IPN'/(10X,10I5))	187
ISN 0157	I=0	188
	C	189
	C COMPUTE NCE CONSTANTS(PIPES ONLY) AND ASSIGN NCE VARIABLES TO	190
	C THE VECTOR FF.	191
ISN 0158	DO 10 J=1,NNCE	192
ISN 0159	LNTYP=NCCL(J)	193
ISN 0160	GO TO (11,12,11),LNTYP	194
ISN 0161	11 CO(J)=.4430*X3(J)/X2(J)**.54	195
ISN 0162	12 FF(J)=ABS(X1(J))	196

ISN 0163	10 CONTINUE	197
	C	198
	C CALL BALALG TO BALANCE THE SYSTEM	199
ISN 0164	CALL BALALG(MA,NCETYP,8848)	200
ISN 0165	WRITE(LT3,64) AD	201
ISN 0166	64 FORMAT(1H1,13X,12A4// ' NCE NCE FLOW FRICTION ELEVATION	202
	1 FROM TO PRESSURE NODE FLOW RESIDUAL// ' NUM TYPE (G	203
	2 PM) LOSS(PSI) LOSS(PSI) NODE NODE (PSIG) (GPM)	204
	3 (GPM)///)	A204
ISN 0167	DO 61 I=1,NNCE	205
ISN 0168	KU=IX(3=I-1)	206
ISN 0169	KD=IX(3=I)	207
ISN 0170	DP=BLANK	A207
ISN 0171	IF(NCEL(I).EQ. 2) GO TO 60	B207
ISN 0173	DP=P(KU)-P(KD)-DELV(I)	208
ISN 0174	59 WRITE(LT3,63) IDENT(I),NCETYP(NCEL(I)),Q(I),DP,DELV(I),NFN(I),NTN(I	209
	1),P(KD),QN(KD),F(KD)	210
ISN 0175	63 FORMAT(1H ,A4,4X,A2,2X,F8.2,F8.2,4X,F7.2,5X,A4,3X,A4,3X,F7.2,4X,F8	211
	1.2,4X,F9.3)	212
ISN 0176	GO TO 61	213
ISN 0177	60 WRITE(LT3,66) IDENT(I),NCETYP(NCEL(I)),Q(I),NFN(I),NTN(I),P(KD),QN	220
	1(KD)	221
ISN 0178	66 FORMAT(1H ,A4,4X,A2,2X,F8.2,24X,A4,3X,A4,3X,F7.2,4X,F8.2)	222
ISN 0179	61 CONTINUE	220
ISN 0180	790 CALL DENSITY(KV,DSITYA,DSITYB,QNA,QNB,ALOWQA,ALOWQB,LOWIA,LOWIB,DNL	240
	LOWA,DNLOWB)	
ISN 0181	IF(KV .GE. 2) GO TO 825	249
ISN 0183	800 CALL REPJN(INEW,IOLD,6775,6830)	250
ISN 0184	825 IF(QNA .EQ. 0.0) GO TO 830	
ISN 0186	WRITE(LT3,826) NTN(IOLD),NTN(INEW),NTN(LOWIA),NTN(LOWIB),ALOWQA,ALO	
	1WQB,DNLOWA,DNLOWB,DSITYA,DSITYB,QNA,QNB	
ISN 0187	826 FORMAT(///, ' ,2EX,'CONSTANT PRESSURE AT ' ,A4,4X,'CONSTANT PRESSU	
	1RE AT ' ,A4,///, ' ,17X,'HEAD WITH LEAST OUT FLOW IS ' ,A4,33X,'HEAD	
	2WITH LEAST OUT FLOW IS ' ,A4,///, ' ,20X,'LEAST OUT FLOW = ' ,F5.1, ' ,	
	3GPM',40X,'LEAST OUT FLOW = ' ,F5.1, ' GPM',///, ' ,15X,'LEAST FLOW DE	
	4NSITY = ' ,F5.3, ' GPM/SQ FT',30X,'LEAST FLOW DENSITY = ' ,F5.3, ' GPM	
	5/SQ FT',///, ' ,17X,'SYSTEM DENSITY = ' ,F5.3, ' GPM/SQ FT',38X,'SYST	
	6EM DENSITY = ' ,F5.3, ' GPM/SQ FT',///, ' ,19X,'IN FLOW AT MAIN = ' ,F	
	77.1, ' GPM',41X,'IN FLOW AT MAIN = ' ,F7.1, ' GPM',////)	
	830 CALL FRICTN	
	GO TO 1	
ISN 0188		
ISN 0189		
ISN 0190	848 READ(LT1,849)MAIN,TOLWPR	295
ISN 0191	849 FORMAT(A4)	296
ISN 0192	850 GO TO 1	295
ISN 0193	999 WRITE(LT3,135)	297
ISN 0194	135 FORMAT(' - RUN COMPLETED')	298
ISN 0195	STOP	299
ISN 0196	END	400

NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.
A	C	R*4	N.R.	F	F	C	R*4 0050C8	I	SF	I*2	000978	J	SFA	I*2	00097A
K	SF	I*2	00097C	N	SF	I*2	00097E	P	SF	C	R*4 007088	Q	F	C	R*4 0079E8
AD	SF	R*4	000EAC	CO	S	C	R*4 004808	DP	SF	R*4	0009A4	FF	S	C	R*4 006728
IA	SF	C	I*2 00C99C	II	SF	I*2	000980	IS		I*2	N.R.	IX	F	C	I*2 011EC4
JA	C	I*2	N.R.	KD	SF	I*2	000982	KU	SF	I*2	000984	KV	SFA	I*2	000986
MA	SFA	I*2	0009E0	NN	F	C	I*2 017964	QN	SF	C	R*4 008748	QO	SF	R*4	000EDC
X1	SFA	C	R*4 008CAC	X2	SF	C	R*4 00960C	X3	SF	C	R*4 009F6C	X4	SF	C	R*4 00A8CC
DIN	F		000002	III	SF	I*2	000988	IN1	SF	I*2	00098A	IN2	SF	I*2	00098C
IN3	SF	I*2	00098E	IPN	F	C	I*2 00D2FC	IPT	F	C	I*2 0109AC	IUN	SF	C	I*2 011A14
IXR	F	C	I*2 012C04	JJJ		I*2	N.R.	KD1	S	I*2	000990	LT1	SF	C	I*4 000000
LT3	SF	C	I*4 000004	MMP	SF	C	I*2 00C4EC	MMQ	SF	I*2	0009F4	NFN	SF	C	I*4 000008
NN1	F	C	I*2 017968	NTN	SF	C	I*4 000968	NUF	SF	C	I*2 01796A	NUP	SF	C	I*2 01796C
QOQ		R*4	N.R.	OX1		R*4	N.R.	OX2		R*4	N.R.	OX3		R*4	N.R.
OX4		R*4	N.R.	QNA	SFA	R*4	000948	QNB	SFA	R*4	0009AC	TOL	C	R*4	008CA8
CONP	SF	C	R*4 008B8C	DELV	SF	C	R*4 005468	INDF	SF	C	I*2 00CE4C	INDP	SF	C	I*2 017970
INDR	SF	C	I*2 017F20	INEW	SFA	I*2	000992	IOLD	SFA	I*2	000994	IPIM	SF	C	I*2 0182D0
IQIM	SF	C	I*2 011564	IRIM	SF	C	I*2 0110B4	JUPD	S	I*2	000996	KHVA		I*2	N.R.
LIUN		I*2	N.R.	LMMP		I*2	N.R.	LMMQ		I*2	N.R.	LNFN		I*2	N.R.
LNTN		I*2	N.R.	MAIN	S	I*2	000998	NCEL	SF	C	I*2 016854	NCLI	SF	C	I*2 018780
NFUN	S	C	I*2 017004	NNCE	SF	C	I*2 017966	NODE	F	C	I*4 0012C8	NPUN	S	C	I*2 0174B4
BLANK	F	R*4	00098C	IDENT	SF	C	I*4 00B22C	INDX1	SF	XF	I*2 000000	INDX2	SF	XF	I*2 000000
IQUES		I*2	00099A	LNCLL		I*2	N.R.	LNTP	SF	I*2	00099C	LDWIA	SFA	I*2	00099E
LDWIB	SFA	I*2	0009AC	NUNCE	SF	C	I*2 01796E	OCOMP		R*4	N.R.	REPUN	SF	XF	R*4 000000
RXPFR	XF	R*4	00000C	ALOWQA	SFA	R*4	0009B4	ALOWQB	SFA	R*4	0009B8	BALALG	SF	XF	R*4 000000
NLCWA	SFA	R*4	0009BC	DNLOWB	SFA	R*4	0009C0	DNSITY	SF	XF	R*4 000000	DSITYA	SFA	R*4	0009C4
SITYB	SFA	R*4	0009C8	ELEVKY	SF	R*4	0009CC	FACTOR	SF	R*4	0009D0	FRDNL	F	XF	R*4 000000
RICTN	SF	XF	R*4 00000C	FWRNL	F	XF	R*4 000000	IBCOM	F	XF	I*4 000000	IPTIPN	F	C	I*2 010E5C
DOFKU	S	I*2	0009A2	KEYJUN	S	I*4	0009D4	LIDENT		I*2	N.R.	NCETYP	SFA	I*2	000EA4
RESKY	SF	R*4	0009C8	TOLNPR	S	R*4	0009DC								

***** COMMON INFORMATION *****

NAME OF COMMON BLOCK * * SIZE OF BLOCK 018C30 HEXADECIMAL BYTES

VAR. NAME	TYPE	REL. ADDR.	VAR. NAME	TYPE	REL. ADDR.	VAR. NAME	TYPE	REL. ADDR.	VAR. NAME	TYPE	REL. ADDR.
LT1	I*4	000000	LT3	I*4	000004	NFN	I*4	000008	NTN	I*4	000968
NODE	I*4	0012C8	A	R*4	N.R.	CO	R*4	004808	DELV	R*4	005468
F	R*4	0050C8	FF	R*4	006728	P	R*4	007088	Q	R*4	0079E8
QN	R*4	008348	TOL	R*4	008CA8	X1	R*4	008CAC	X2	R*4	00960C
X3	R*4	009F6C	X4	R*4	00A8CC	IDENT	I*4	00B22C	CONP	R*4	008B8C
MMP	I*2	00C4EC	IA	I*2	00C99C	INDF	I*2	00CE4C	IPN	I*2	00D2FC
IPT	I*2	0109AC	IPTIPN	I*2	010E5C	IRIM	I*2	0110B4	IQIM	I*2	011564
IUN	I*2	011A14	IX	I*2	011EC4	IXR	I*2	012C04	JA	I*2	N.R.
NCEL	I*2	016854	NFUN	I*2	017004	NPUN	I*2	0174B4	NN	I*2	017964
NNCE	I*2	017966	NNJ	I*2	017968	NUF	I*2	01796A	NUP	I*2	01796C
NUNCE	I*2	01796E	INDP	I*2	017970	INDR	I*2	017F20	IPIM	I*2	0182D0
NCLL	I*2	018780									

LABEL	ADDR	LABEL	ADDR	LABEL	ADDR	LABEL	ADDR	PAGE 007
1	001A80	775	C01AF0	3	00180C	8	001F66	
200	001D92	7C0	001DAC	400	001F1A	104	001F54	
105	001F6E	410	C01F8A	112	001FAA	405	001FC6	
122	00203E	125	C020CA	130	00211A	46	002182	
45	00219C	423	C0229A	425	0022C2	426	0022DA	
48	002312	420	C0234A	430	00236A	50	0023AE	
445	002412	446	C02454	447	002480	449	00249A	
451	00256E	450	0025AE	179	00261E	178	00263A	
11	002986	E2	0029E0	10	002A0A	59	002AFE NR	
60	002B0C	E1	002C82	790	002C9C	800	002CBB	
825	002CD8	830	002D7C	848	002D8E	850	002DB4 NR	
999	002DBA							

OPTIONS IN EFFECT NAME= MAIN,OPT=00,LINECNT=60,SIZE=0000K,

OPTIONS IN EFFECT SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NODEIT,ID,NOXREF

STATISTICS SOURCE STATEMENTS = 195 ,PROGRAM SIZE = 11778

STATISTICS NO DIAGNOSTICS GENERATED

***** END OF COMPILATION *****

87K BYTES OF CORE NOT USED

EVEL 20.1 (AUG 71)

OS/360 FORTRAN H

DATE 72.227/08.19.02

```

COMPILER OPTIONS - NAME= MAIN,OPT=00,LINECNT=60,SIZE=0000K,
SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NODEIT,ID,NOXREF
ISN 0002 SUBROUTINE BALALG(MA-NCETYP,*)
C
C ** REVISED VERSION OF ISAP FOR WATER SYSTEM ANALYSIS **
C THIS ROUTINE DIRECTS THE BALANCING PROCEDURES. THE SEQUENCE
C AND FREQUENCY OF CALLS ON THE SPARSE MATRIX ROUTINES ARE GIVEN
C BY THE CONTENTS OF MA.
C
ISN 0003 IMPLICIT INTEGER*2(1-N)
ISN 0004 INTEGER*4 LT1,LT3,NFN,NTN,NODE,IDENT,KEYJUN,I,NN2,NN3,IHOW
ISN 0005 DIMENSION RHS(500),IU(500),JU(30000),IHOW(40000),DX(500),DX1(500),
1 ACEL(500),U(30000),MA(1),NCETYP(1)
ISN 0006 COMMON LT1,LT3,NFN(600),NTN(600),NODE(500),AI(3000),CD(600)
1,DELV(600),F(500),FF(600),P(600),Q(600),QN(600),TDL,X1(600),X2(600)
2,X3(600),X4(600),IDENT(600),CONP(600),MMP(600),IA(600),INDF(600),
3IPN(7000),IPT(600),IPTIPN(300),IRIM(600),IQIM(600),IUN(600),IX(180
40),IXR(5000),JA(3000),NCEL(600),NFUN(600),NPUN(600),NN,NNCE,NN2,NU
5F,NUP,NUNCE,INDP(600),INDR(600),IPIM(600),NCLL(600)
ISN 0007 IC=0
ISN 0008 KO=0

```

1
2
3
4
5
6
7
8

17
18

ISN 0009	LO=0	19
ISN 0010	MASIX=MA(6)	
ISN 0011	226 IC=IC+1	20
ISN 0012	IF(IC.GT. MA(1)) GO TO 240	21
ISN 0014	IF(IC-MA(2)) 10,10,20	22
ISN 0015	10 KO=0	23
	C	24
	C DETERMINE THE NODE IMBALANCE OR RESIDUALS, F	25
ISN 0016	CALL SETQAF(ITOL,IC,MASIX)	26
ISN 0017	IF(ITOL) 11,240,11	27
	C	28
	C CALCULATE THE COEFFICIENT MATRIX OR JACOBIAN	29
ISN 0018	11 CALL JACOB	30
ISN 0019	IF(IC.GT.MA(9)) GO TO 282	31
ISN 0021	WRITE(LT3,280)	32
ISN 0022	280 FORMAT('1 SUMMARY OF SIMULTANEOUS LINEAR EQUATIONS'// ' EQUATI	33
	ION (NODE) NUMBER, RIGHT HAND SIDE (F), AND NON-ZERO TERMS OF THE	34
	?JACOBIAN'//)	35
ISN 0022	282 DO 230 I=1,NN	36
ISN 0024	RHS(I)=-F(I)	37
ISN 0025	IF(IC.GT.MA(9)) GO TO 250	38
ISN 0027	K1=IA(I)	39
ISN 0028	K2=IA(I+1)-1	40
ISN 0029	WRITE(LT3,237)I,RHS(I)	41
ISN 0030	WRITE(LT3,239)(A(J),J=K1,K2)	42
ISN 0031	WRITE(LT3,236)(JA(J),J=K1,K2)	43
ISN 0032	237 FORMAT(2X,'NODE',I4,' F=',F8.2)	44
ISN 0033	239 FORMAT(6X,8F15.4)	45
ISN 0034	236 FORMAT(2X,8I15)	46
ISN 0035	230 CONTINUE	47
	C	48
	C TRIANGULARIZE THE JACOBIAN AND STORE PIVOTING INFORMATION IN IHOW	49
ISN 0036	CALL ELMPIV(NN,JA,JA,A,IU,JU,U,IHOW,MARK)	50
ISN 0037	IF(MARK) 232,120,232	51
ISN 0038	120 WRITE(LT3,2364) (IHOW(I),I=1,NN)	52
ISN 0039	2364 FORMAT('1 PIVOT ROW LIST. IHOW(1) TO IHOW(NN)'//('015))	53
ISN 0040	WRITE(LT3,78) IU(NN1),IHOW(2*NN)	54
ISN 0041	78 FORMAT('0 MAXIMUM INDEX OF JU AND U IS ',I6,' MAXIMUM INDEX OF I	55
	IHOW IS ',I6)	56
ISN 0042	IF(IC.GT.MA(9)) GO TO 243	57
ISN 0044	NN2=2*NN	58
ISN 0045	WRITE(LT3,2362)(IHOW(I),I=NN1,NN2)	59
ISN 0046	2362 FORMAT('0 POINTERS TO COLUMN ELIMINATION GROUPS. IHOW(N+1) TO	60
	1 IHOW(2N)'//('015))	61
ISN 0047	NN2=NN2+1	62
ISN 0048	NN3=IHOW(2*NN)	63
ISN 0049	NN3=IHOW(NN3+2)*2+2+NN2	64
ISN 0050	WRITE(LT3,2363)(IHOW(I),I=NN2,NN3)	65
ISN 0051	2363 FORMAT('0 IHOW(2N+1) TO THE END OF IHOW'//('015))	66
ISN 0052	505 IF(IC.GT. MA(9)) GO TO 243	67
ISN 0054	WRITE(LT3,281)	68
ISN 0055	281 FORMAT('1 NON-ZERO TERMS OF THE TRIANGULARIZED MATRIX U')	69
ISN 0056	DO 242 I=1,NN	70
ISN 0057	K1=IU(I)	71
ISN 0058	K2=IU(I+1)-1	72

ISN 0059	WRITE(LT3,506) I	73
ISN 0060	WRITE(LT3,239) (U(J),J=K1,K2)	74
ISN 0061	242 WRITE(LT3,236) (JU(J),J=K1,K2)	75
ISN 0062	506 FORMAT(2X,'NODE',I5)	76
ISN 0063	GO TO 243	77
ISN 0064	232 GO TO(5,6,7,8), MARK	78
ISN 0065	5 WRITE(LT3,1)	79
ISN 0066	1 FORMAT('SINGULAR MATRIX - ROW OF ZEROS.')	80
ISN 0067	GO TO 9	81
ISN 0068	6 WRITE(LT3,2)	82
ISN 0069	2 FORMAT('SINGULAR MATRIX - NO PIVOT ELEMENT IN COLUMN.')	83
ISN 0070	GO TO 9	84
ISN 0071	7 WRITE(LT3,3)	85
ISN 0072	3 FORMAT('SINGULAR MATRIX - ZERO PIVCT ELEMENT IN ELMPIV.')	86
ISN 0073	GO TO 9	87
ISN 0074	8 WRITE(LT3,4)	88
ISN 0075	4 FORMAT('SINGULAR MATRIX - ZERO PIVCT ELEMENT IN ELIMA.')	89
ISN 0076	9 WRITE(LT3,233) IC	90
ISN 0077	233 FORMAT('0 ITERATION NUMBER',I4)	91
ISN 0078	RETURN 1	92
ISN 0079	20 KO=KO+1	93
ISN 0080	IF(KO-MA(3)) 30,20,30	94
ISN 0081	30 IF(IC-MA(4)) 40,40,50	95
ISN 0082	40 LO=0	96
ISN 0083	CALL SETQAF(ITOL,IC,MA5IX)	97
ISN 0084	IF(ITOL) 41,240,41	98
ISN 0085	41 CALL JACOB	99
ISN 0086	DO 42 I=1,MN	100
ISN 0087	42 RHS(I)=-F(I)	101
ISN 0088	IF(IC-MA(9)) GO TO 43	102
ISN 0090	WRITE(LT3,380)	103
ISN 0091	DO 45 I=1,MN	104
ISN 0092	K1=IA(I)	105
ISN 0093	K2=IA(I+1)-1	106
ISN 0094	WRITE(LT3,237) I,RHS(I)	107
ISN 0095	WRITE(LT3,239) (A(J),J=K1,K2)	108
ISN 0096	45 WRITE(LT3,236) (JA(J),J=K1,K2)	109
	C	110
	C TRIANGULARIZE THE JACOBIAN USING IHOW INFORMATION FOR PIWOTING	111
ISN 0097	43 CALL ELIMB(INN,IA-JA,A,IU,JU,U,IHOW,MARK)	112
ISN 0098	IF(MARK) 232,505,232	113
ISN 0099	50 LO=LO+1	114
ISN 0100	IF(LO-MA(5)) 60,40,60	115
ISN 0101	60 CALL SETQAF(ITOL,IC,MA5IX)	116
ISN 0102	IF(ITOL) 61,240,61	117
ISN 0103	61 DO 62 I=1,MN	118
ISN 0104	62 RHS(I)=-F(I)	119
ISN 0105	243 CALL ELIMB(INN,IU,JU,IHOW,U,RHS)	120
ISN 0106	CALL BACSUR(INN,IU,JU,IHOW,U,RHS,DX)	121
	C	122
	C DETERMINE ACCELERATION VALUES FOR EACH UNKNOWN	123
	C	124
ISN 0107	250 ITAB=0	125
ISN 0108	IF(IC-MA(6)) 350,350,351	126
ISN 0109	350 ACCF=IC/(MA(6)+1.)	127

ISN 0110	GO TO 262	128
ISN 0111	351 ACCE=1.	129
ISN 0112	IF((IC-MA(6))/MA(7)*MA(7).NE.(IC-MA(6))) GOTO 257	130
ISN 0114	ITAB=1	131
ISN 0115	DO 251 I=1,NN	132
ISN 0116	IF(DX1(I)*DX(I)) 252,269,253	133
ISN 0117	253 RA=DX(I)/DX1(I)	134
ISN 0118	IF(RA-1.) 268,269,269	135
ISN 0119	269 ACCL(I)=3.	136
ISN 0120	GO TO 251	137
ISN 0121	268 ACCL(I)=1.+2.*RA	138
ISN 0122	GO TO 251	139
ISN 0123	252 RA=ABS(DX(I)/DX1(I))	140
ISN 0124	IF(RA-1.) 254,254,256	141
ISN 0125	254 ACCL(I)=1.-.5*RA	142
ISN 0126	GO TO 251	143
ISN 0127	256 IF (RA.GT.10.) GO TO 2560	144
ISN 0129	ACCL(I)=.5**RA	145
ISN 0130	GO TO 251	146
ISN 0131	2560 ACCL(I)=0.	147
ISN 0132	251 CONTINUE	148
ISN 0133	257 DO 258 I=1,NN	149
ISN 0134	258 DX1(I)=DX(I)	150
ISN 0135	IF(ITAB) 261,262,261	151
ISN 0136	262 DO 263 I=1,NN	152
ISN 0137	263 ACCL(I)=ACCE	153
ISN 0138	261 CONTINUE	154
C		155
C	APPLY CORRECTION VALUES TO UPDATE THE UNKNOWN AND CHECK FOR	156
C	LOWER LIMITS.	157
ISN 0139	231 IF(NUNCE.EQ.0) GO TO 301	158
ISN 0141	DO 234 I=1,NUNCE	159
ISN 0142	II=INDR(I)	160
ISN 0143	FF(II)=FF(II)+DX(I)*ACCL(I)	161
ISN 0144	LNTYP=NCEL(II)	162
ISN 0145	GO TO (81,84,87,90),LNTYP	163
ISN 0146	81 IF(FF(II).LT.0.) FF(II)=1.0	164
ISN 0148	GO TO 234	165
ISN 0149	84 IF(FF(II).LT.0.) FF(II)=10.	166
ISN 0151	GO TO 234	167
ISN 0152	87 IF(FF(II).LT.0.) FF(II)=.00005	168
ISN 0154	GO TO 234	169
ISN 0155	90 IF(FF(II).LT.0.) FF(II)=1.	170
ISN 0157	234 CONTINUE	171
ISN 0158	301 IF(NUP.EQ.0) GO TO 311	172
ISN 0160	DO 238 I=1,NUP	173
ISN 0161	II=INDP(I)	174
ISN 0162	P(II)=P(II)+DX(IQIM(II))*ACCL(IQIM(II))	175
ISN 0163	IF(P(II).LT.0.) P(II)=14.7	176
ISN 0165	238 CONTINUE	178
ISN 0166	311 IF(NUF.EQ.0) GO TO 314	179
ISN 0168	DO 312 I=1,NUF	180
ISN 0169	II=INDF(I)	181
ISN 0170	312 QN(II)=QN(II)+DX(IQIM(II))*ACCL(IQIM(II))	182
- ISN 0171	314 CONTINUE	183

```

ISN 0172      IF(IC-MA(10)) 143,143,226
C
C      PRINT OUT FOR EACH ITERATION UP TO AND INCLUDING THE VALUE OF
C      MA(10).
ISN 0173      143 WRITE(LT3,147) IC
ISN 0174      147 FORMAT('1 SYSTEM VALUES AFTER ITERATION NUMBER',I4,'//20X','NODE I
INFORMATION'//' NODE PRES FULL ACEL NODE FULL ACEL
INODE RESIDUAL(F)'// NAME PSIG CORR VALUE FLOW CORR V
'ALUE START OF ITERATION'//)
ISN 0175      DO 145 I=1,NN
ISN 0176      PNK=P(I)
ISN 0177      WRITE(LT3,146) NODE(I),PNK,NPUN(I),QN(I),NFUN(I),F(I)
ISN 0178      146 FORMAT(1X,A4,F9.2,A1,14X,F8.2,A1,14X,F9.3)
ISN 0179      IF(IPIM(I)) 148,149,148
ISN 0180      148 WRITE(LT3,165) DX(IPIM(I)),ACEL(IPIM(I))
ISN 0181      165 FORMAT(1H+,14X,F9.3,F5.2)
ISN 0182      149 IF(IQIM(I)) 166,145,166
ISN 0183      166 WRITE(LT3,167) DX(IQIM(I)),ACEL(IQIM(I))
ISN 0184      167 FORMAT(1H+,37X,F9.3,F5.2)
ISN 0185      145 CONTINUE
ISN 0186      IF(NUNCE) 150,160,150
ISN 0187      150 WRITE(LT3,151)
ISN 0188      151 FORMAT('1- NODE CONNECTING ELEMENTS WITH UNKNOWN PARAMETERS'//
1X,'FROM TO TYPE FLOW NCE FULL ACCELERATION'// N
NODE NODE',17X,'PARAM CORR VALUE'//)
ISN 0189      DO 155 I=1,NUNCE
ISN 0190      II=INDE(I)
ISN 0191      155 WRITE(LT3,156) NFN(II),NTN(II),NCETYP,NCEL(II),IUN(II),Q(II),FF(
II),DX(II),ACEL(II)
ISN 0192      156 FORMAT(2(1X,A4),4X,A2,1X,A1,F7.2,F10.4,2F9.5)
ISN 0193      160 GO TO 226
ISN 0194      240 CONTINUE
ISN 0195      IC=IC-1
ISN 0196      WRITE(LT3,500) IC
ISN 0197      500 FORMAT('1- SOLUTION COMPLETED IN', I4,' ITERATIONS')
ISN 0198      RETURN
ISN 0199      END

```

/ BALALG / SIZE OF PROGRAM 056D2E HEXADECEMAL BYTES PAGE 005

NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.
A	SFA	C	R*4 001C28	F	=	C	R*4 005DC8	I	SFA	I*4	00063C	J	F	I*2	000524
P	SF	C	R*4 007088	Q	=	C	R*4 0079EE	U	SFA	R*4	00F4A4	ED	C	R*4	N.P.
DX	SFA		R*4 02C964	=F	SF	C	R*4 00672E	IA	SFA	C	I*2 00C99C	IE	SFA	I*2	000526
II	SF	I*2	000628	IU	SFA	I*2	00065E	IX	C	I*2	N.P.	JA	SFA	C	I*2 0152E4
JU	SFA	I*2	000A40	<Q	SF	I*2	00062A	KJ	SF	I*2	00062C	KZ	SF	I*2	00062E
LO	SF	I*2	000630	MA	F	XR	I*2 00000C	NN	SFA	C	I*2 017964	OM	SF	C	R*4 008348
RA	SF	R*4	000640	X1		C	R*4 N.R.	X2	C	R*4	N.R.	XZ	C	R*4	N.R.
X4		C	R*4 N.R.	DX1	SFA	R*4	02D134	IPN	C	I*2	N.R.	IPZ	C	I*2	N.R.
IUN	F	C	I*2 011A14	1XR		C	I*2 N.R.	LT1	C	I*4	N.P.	LTZ	F	C	I*4 000004

MMP	C	I*2	N.R.	NFN	F	C	I*4	000008	NN1	F	C	I*2	017968	NN? SF		I*4	000644			
NN? SF		I*4	000648	NTN	F	C	I*4	000968	NUP	F	C	I*2	01796A	NUP	F	C	I*2	01796C		
PNX SF		R*4	00064C	RHS SFA		R*4	020904	TOL	C	R*4	N.R.	ACCE SF		R*4	000650	THOW SFA		I*4	02E8A4	
ACEL SF		R*4	02E0D4	CONP	C	R*4	N.R.	DELV	C	R*4	N.R.	IPIM	F	C	I*2	0182D0	ITOL SFA		I*2	0006?4
INDF F	C	I*2	00CE4C	IRIM	C	I*2	N.R.	ITAB S		I*2	0006?2	NEUN F	C	I*2	017004	ELIMA SF	XF	R*4	000000	
IQIM F	C	I*2	011564	NCEL F	C	I*2	016854	NCLL	C	I*2	N.R.	LNTYP SF		I*2	000638	RACSUB SF	XF	R*4	000000	
MARK SFA		I*2	000636	NODE F	C	I*4	0012C8	NPUN F	C	I*2	0174B4	IPIN	C	I*2	N.R.					
NNCE	C	I*2	N.R.	IDENT	C	I*4	N.R.	JACOB SF	XF	I*2	000000									
ELIMB SF	XF	R*4	0000D0	NUNCE F	C	I*2	01796E	FRXPR=	XF	R*4	000000									
MASIX SFA		I*2	00063A	ELMPIO SF	XF	R*4	000000	IBCOM=	F	XF	I*4	000000								
BALALG		R*4	000654	NCETYP F	XR	I*2	000000	SETQAF SF	XF	R*4	000000									
KEYJUN		I*4	N.R.																	

***** COMMON INFORMATION *****

NAME OF COMMON BLOCK *				* SIZE OF BLOCK				018C30 HEXADECIMAL BYTES							
VAR. NAME	TYPE	REL. ADDR.		VAR. NAME	TYPE	REL. ADDR.		VAR. NAME	TYPE	REL. ADDR.		VAR. NAME	TYPE	REL. ADDR.	
LT1	I*4	N.R.		LT3	I*4	000004		NFN	I*4	000008		NTN	I*4	000968	
NODE	I*4	0012C8		A	R*4	001C28		CO	R*4	N.R.		DELV	R*4	N.R.	
F	R*4	005DC8		FF	R*4	006728		P	R*4	007088		Q	R*4	0079E8	
QN	R*4	008348		TDL	R*4	N.R.		X?	R*4	N.R.		X?	R*4	N.R.	
X3	R*4	N.R.		X4	R*4	N.R.		IDENT	I*4	N.R.		CONP	R*4	N.R.	
MMP	I*2	N.R.		IA	I*2	00C99C		INDF	I*2	00CE4C		IPN	I*2	N.R.	
IPT	I*2	N.R.		IPTIPN	I*2	N.R.		IRIM	I*2	N.R.		IQIM	I*2	011564	
IUN	I*2	011A14		IX	I*2	N.R.		IXR	I*2	N.R.		JA	I*2	015?E4	
NCEL	I*2	016854		NFUN	I*2	017004		NPUN	I*2	0174B4		NN	I*2	017964	
NNCE	I*2	N.R.		NN1	I*2	017968		NUP	I*2	01796A		NUP	I*2	01796C	
NUNCE	I*2	01796E		INDP	I*2	017970		INDR	I*2	017E20		IPIM	I*2	0182D0	
NCLL	I*2	N.R.													

LABEL	ADDR	LABEL	ADDR	LABEL	ADDR	LABEL	ADDR	PAGE 006
226	055C18	10	055C48	11	055C6A	282	055C9C	
230	055DC6	120	055EFA	505	055F7E	242	0560?2	
232	056096	5	056C80	6	0560CA	7	0560F6	
8	056102	9	056118	20	05613E	30	05615C	
40	05616E	41	056190	42	0561A4	45	0562B6	
43	056318	50	056338	60	056356	61	056370	
62	056378	243	0563BC	250	0563D8 NR	350	0563F2	
351	056460	253	0564E8	269	05651A	268	056534	
252	05655E	254	056596	256	0565C2	2560	056604	
251	056618	257	056632	258	05663A	262	056684	
263	05668C	261	0566BA	231	0566BA NR	81	056766	
84	056798	87	0567CA	90	0567FC	234	056828	
301	056842	238	056906	311	056920	212	05694C	

314 0569DA	143 0569EC	148 056A8C	149 056B14
166 056B2C	145 056B84	150 056BAE	155 056BE0
160 056CB8	240 056CBE		

OPTIONS IN EFFECT NAME= MAIN,OPT=00,LINECNT=60,SIZE=0000K,

OPTIONS IN EFFECT SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF

STATISTICS SOURCE STATEMENTS = 198 ,PROGRAM SIZE = 355646

STATISTICS NO DIAGNOSTICS GENERATED

***** END OF COMPILATION *****

91K BYTES OF CORE NOT USED

LEVEL 20.1 (AUG 71)

DS/360 FORTRAN H

DATE 72.237/08. 9.10

COMPILER OPTIONS - NAME= MAIN,OPT=00,LINECNT=60,SIZE=0000K,
SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF

ISN 0002	SUBROUTINE INDX1	1
	C	2
	C ** REVISED VERSION OF ISAP FOR WATER SYSTEM ANALYSIS **	3
	C INDX1 SETS UP THE NODE CORRESPONDENCE TABLE AND THE IX,IXR,IPT	4
	C INDEX VECTORS	5
	C	6
ISN 0003	IMPLICIT INTEGER*2(I=N)	7
ISN 0004	INTEGER*4 LT1,LT3,NFN,NTN,NODE,IDENT,KEYJUN	
ISN 0005	COMMON LT1,LT3, NFN(600),NTN(600),NODE(600),A(3000),CD(600)	
	1,DELV(600),F(600),FF(600),P(600),O(600),ON(600),TOL,X1(600),X2(600)	
	2),X3(500),X4(600),IDENT(600),CONP(600),MMP(600),IA(600),INDF(600),	
	3IPN(700),IPT(600),IPTIPN(300),IRIM(600),IQIM(600),IUN(600),IX(180	
	40),IXR(5000),JA(3000),NCEL(600),NFCN(600),NPUN(600),NN,NNCE,NNI,NU	
	5F,NUP,NUNCE,INBP(600),INDR(600),IPIM(600),NCLL(600)	
ISN 0006	DATA IQUES/'*'	14
	C	15
	C DETERMINE THE NODE CORRESPONDENCE TABLE	16
ISN 0007	1 NN=1	17
ISN 0008	NODE(1)=NFN(1)	18
ISN 0009	IF(NNCE-1) 12,13,12	19
ISN 0010	12 DO 17 I=2, NNCE	20
ISN 0011	DO 16 J=1,NN	21
ISN 0012	IF(NFN(I).EQ.NODE(J)) GO TO 17	22
ISN 0014	16 CONTINUE	23
ISN 0015	NN=NN+1	24
ISN 0016	NODE(NN)=NFN(1)	25
ISN 0017	17 CONTINUE	26
ISN 0018	13 DO 14 I=1, NNCE	27
ISN 0019	DO 15 J=1,NN	28
ISN 0020	IF(NTN(I).EQ.NODE(J)) GO TO 14	29

ISN 0022	15 CONTINUE	30
ISN 0023	NN=NN+1	31
ISN 0024	NODE(NN)=NTN(I)	32
ISN 0025	14 CONTINUE	33
ISN 0026	NN=NN+1	34
	C	35
	C FILL IN THE IX VECTOR	36
ISN 0027	DO 26 I=1,NNCE	37
ISN 0028	IX(3*I-2)=I	38
ISN 0029	DO 23 J=1,NN	39
ISN 0030	IF(NODE(J).NE.NFN(I)) GO TO 23	40
ISN 0032	IX(3*I-1)=J	41
ISN 0033	GO TO 24	42
ISN 0034	23 CONTINUE	43
ISN 0035	24 DO 27 J=1,NN	44
ISN 0036	IF (NODE(J).NE.NTN(I)) GO TO 27	45
ISN 0038	IX(3*I)=J	46
ISN 0039	GO TO 26	47
ISN 0040	27 CONTINUE	48
ISN 0041	26 CONTINUE	49
	C	50
	C DETERMINE IPT AND IXP VECTORS.	51
ISN 0042	II=1	52
ISN 0043	DO 20 I=1,NN	53
ISN 0044	IPT(I)=II	54
ISN 0045	IXR(II)=I	55
ISN 0046	III=II+2	56
ISN 0047	DO 18 J=1,NNCE	57
ISN 0048	IF (NODE(I).NE.NFN(J)) GO TO 18	58
	C	59
	C NCE NUMBER IS PLUS IF NODE(I) IS A FROM NODE.	60
ISN 0050	IXR(III)=J	61
ISN 0051	IXR(III+1)=IX(3*J)	62
ISN 0052	III=III+2	63
ISN 0053	18 CONTINUE	64
ISN 0054	DO 19 J=1,NNCE	65
ISN 0055	IF (NODE(I).NE.NTN(J)) GO TO 19	66
	C	67
	C NCE NUMBER IS MINUS IF NODE(I) IS A TO NODE.	68
ISN 0057	IXR(III)=-J	69
ISN 0058	IXR(III+1)=IX(3*J-1)	70
ISN 0059	III=III+2	71
ISN 0060	19 CONTINUE	72
ISN 0061	IXR(II+1)=(III-II-2)/2	73
ISN 0062	20 II=III	74
ISN 0063	RETURN	75
	C	76
ISN 0064	ENTRY INDx2	77
	C	78
	C SET UP THE IPN INDEX VECTOR. IT DESCRIBES HOW TO GET THE JACOB	79
	C IAN BY GIVING INFORMATION ON THE UNKNOWN NODE FLOWS,NODE PRESS	80
	C URES , AND NCE PARAMETERS.	81
	C	82
ISN 0065	J=0	83
ISN 0066	DO 180 I=1,NN	84

ISN 0067	J=J+1	85
ISN 0068	IPN(J)=I	86
ISN 0069	IPN(I)=J	87
	C	88
	C IS NODE I FLOW UNKNOWN	89
ISN 0070	IT=0	90
ISN 0071	IF(NFUX(I).EQ.IGUES) IT=1	91
ISN 0072	J=J+1	92
ISN 0073	IPN(J)=IT	93
	C	94
	C IS NODE I PRESSURE UNKNOWN	95
ISN 0075	IT=0	96
ISN 0076	IF(NPUM(I).EQ.IGUES) IT=1	97
ISN 0078	J=J+1	98
ISN 0079	IPN(J)=IT	99
	C	100
	C DETERMINE THE NUMBER OF UNKNOWN PRESSURES AND NCE PARAMETERS	101
	C ADJACENT TO NODE I.	102
ISN 0080	II2=0	103
ISN 0081	N=IPT(1)+1	104
ISN 0082	NO=IXR(N)	105
	C	106
	C DETERMINE AND INDEX THE ADJACENT UNKNOWN PRESSURES.	107
ISN 0083	DO 190 JJ=1,NO	108
ISN 0084	N1=N+2*JJ	109
ISN 0085	J2=IXR(N1-1)	110
ISN 0086	IF(J2.LT.0) J2=-J2	111
ISN 0088	KJJ=IXR(N1)	112
ISN 0089	IF(NPUM(KJJ)-IGUES) 190,197,190	113
ISN 0090	197 II2=II2+1	114
ISN 0091	IT=J+1-2*II2	115
ISN 0092	IPN(IT-1)=J2	116
ISN 0093	IPN(IT)=KJJ	117
ISN 0094	190 CONTINUE	118
	C	119
	C PROVIDE FOR MORE THAN ONE NODE CONNECTING ELEMENT BETWEEN THE	120
	C SAME TWO NODES.	121
ISN 0095	KD=0	122
ISN 0096	J=J+1	123
ISN 0097	II2M1=II2-1	124
ISN 0098	IF(II2M1) 101,101,102	125
ISN 0099	102 DO 170 <K=1,II2M1	126
ISN 0100	K1=J+2*<K	127
ISN 0101	KK1=KK+1	128
ISN 0102	DO 170 JJ=KK1,II2	129
ISN 0103	K2=J+2*JJ	130
ISN 0104	IF(IPN(K1)-IPN(K2)) 170,105,170	131
ISN 0105	105 IPN(K1)=-IPN(K1)	132
ISN 0106	KD=KD+1	133
ISN 0107	KK=KK+1	134
ISN 0108	K1=J+2*KK	135
ISN 0109	IT1=IPN(K1)	136
ISN 0110	IT2=IPN(K1-1)	137
ISN 0111	IPN(K1)=IPN(K2)	138
ISN 0112	IPN(K1-1)=IPN(K2-1)	139

ISN 0113	IPN(K2)=IT1	140
ISN 0114	IPN(K2-1)=IT2	141
ISN 0115	170 CONTINUE	142
ISN 0116	101 IPN(J)=IT2-KO	143
ISN 0117	J=J+2*IT2	144
	C	145
	C INDEX ADJACENT UNKNOWN NCE PARAMETERS.	146
ISN 0118	II2=0	147
ISN 0119	DO 210 JJ=1,N0	148
ISN 0120	N1=N+2*JJ	149
ISN 0121	J2=IXR(N1-1)	150
ISN 0122	IF(J2.LT.0) J2=-J2	151
ISN 0124	KJJ=IXR(N1)	152
ISN 0125	IF(IUN(J2)-IQUES) 210,213,210	153
ISN 0126	213 II2=II2+1	154
ISN 0127	IT=J+1+2*II2	155
ISN 0128	IPN(IT-1)=J2	156
ISN 0129	IPN(IT)=KJJ	157
ISN 0130	210 CONTINUE	158
ISN 0131	J=J+1	159
ISN 0132	IPN(J)=II2	160
ISN 0133	J=J+2*II2	161
ISN 0134	180 CONTINUE	162
ISN 0135	IPTIPN(NN1)=J+1	163
	C	164
	C SET UP VECTORS TO ASSIGN A COLUMN OF THE MATRIX FOR EACH UNKNOW	165
	C	166
ISN 0136	IF(NUNCE.EQ.0) GO TO 223	167
ISN 0138	DO 220 I=1,NUNCE	168
ISN 0139	220 IRIM(INDR(I))=I	169
ISN 0140	223 I11=NUNCE	170
ISN 0141	IF(NUP.EQ.0) GO TO 224	171
ISN 0143	DO 221 I=1,NUP	172
ISN 0144	221 IPIM(INDP(I))=I11+I	173
ISN 0145	224 I11=NUNCE+NUP	174
ISN 0146	IF(NUF.EQ.0) GO TO 225	175
ISN 0148	DO 222 I=1,NUF	176
ISN 0149	222 IQIM(INDF(I))=I11+I	177
ISN 0150	225 CONTINUE	178
	C	179
	C USING THE INFORMATION IN IPN, THE VECTORS IA AND JA CAN BE	180
	C DETERMINED.	181
	C	182
ISN 0151	II=1	183
ISN 0152	DO 45 J=1,NN	184
ISN 0153	IA(J)=II	185
ISN 0154	IP=IPTIPN(J)	186
ISN 0155	IF(IPN(IP+1))30,31,30	187
ISN 0156	30 JA(II)=IQIM(J)	188
ISN 0157	II=II+1	189
ISN 0158	31 IF(IPN(IP+2))32,33,32	190
ISN 0159	22 JA(II)=IPIM(J)	191
ISN 0160	II=II+1	192
ISN 0161	33 IT=IPN(IP+3)	193
ISN 0162	IS=IP+3	194

ISN 0163	IF(IT) 34,40,34	195
ISN 0164	34 DO 36 I=1,IT	196
ISN 0165	38 ISS=IS+2*I	197
ISN 0166	IST=IPN(ISS)	198
ISN 0167	IF(IST) 37,37,39	199
ISN 0168	37 IS=IS+2	200
ISN 0169	GO TO 33	201
ISN 0170	39 JA(II)=EPIM(IST)	202
ISN 0171	36 II=II+1	203
ISN 0172	40 ISS=IS+2*IT+1	204
ISN 0173	IT=IPN(ISS)	205
ISN 0174	IF(IT) 41,45,41	206
ISN 0175	41 DO 44 I=1,IT	207
ISN 0176	IS=ISS+2*I-1	208
ISN 0177	IST=IPN(ISS)	209
ISN 0178	JA(II)=IRIM(IST)	210
ISN 0179	44 II=II+1	211
ISN 0180	45 CONTINUE	212
ISN 0181	IA(NN+1)=II	213
	C	214
	C REORDER THE ELEMENTS OF JA SO THAT THEY ARE IN ASCENDING ORDER	215
ISN 0182	DO 70 I=1,NN	216
ISN 0183	K1=IA(I)	217
ISN 0184	K2=IA(I-1)-1	218
ISN 0185	NO=K2-K1	219
ISN 0186	IF(NO) 70,70,48	220
ISN 0187	48 J=K1+1	221
ISN 0188	DO 60 J=J,K2	222
ISN 0189	IF(JA(J-1)-JA(J)) 60,50,50	223
ISN 0190	50 IT=JA(J-1)	224
ISN 0191	JA(J-1)=JA(J)	225
ISN 0192	JA(J)=IT	226
ISN 0193	JJ=J	227
ISN 0194	55 JJ=JJ-1	228
ISN 0195	IF(JJ-K1) 60,60,56	229
ISN 0196	56 IF(JA(JJ-1)-JA(JJ)) 60,57,57	230
ISN 0197	57 IT=JA(JJ-1)	231
ISN 0198	JA(JJ-1)=JA(JJ)	232
ISN 0199	JA(JJ)=IT	233
ISN 0200	GO TO 55	234
ISN 0201	60 CONTINUE	235
ISN 0202	70 CONTINUE	236
ISN 0203	RETURN	237
ISN 0204	END	238

NAME	TAG	TYPE	ADD.
A	C	R*4	N.R.
N SF		I*2	000094
FF	C	R*4	N.R.
IS SF		I*2	00009A
JJ SF		I*2	00009E
K1 SF		I*2	0000A6
N1 SF		I*2	0000AC
X3	C	R*4	N.R.
IPN SF	C	I*2	0002FC
IT1 SF		I*2	0000B6
KJJ SF		I*2	00008A
MMP	C	I*2	N.R.
NUF	F	C	I*2 01796A
DELV	C	R*4	N.R.
IPIM SF	C	I*2	018200
NCLL	C	I*2	N.R.
NPUN	C	I*2	017484
INDX2		I*2	0000C2
KEYJUN		I*4	N.R.

NAME	TAG	TYPE	ADD.
F	C	R*4	N.R.
P	C	R*4	N.R.
IA SF	C	I*2	00C99C
IT SF		I*2	00009C
J2 SF		I*2	0000A0
K2 SF		I*2	0000A8
QN	C	R*4	N.R.
X4	C	R*4	N.R.
IPT SF	C	I*2	0109AC
IT2 SF		I*2	0000B8
KK1 SF		I*2	00008C
NFN	F	C	I*4 000008
NUP	F	C	I*2 01796C
INDF	F	C	I*2 00CE4C
IQIM SF	C	I*2	011564
NFUN	C	I*2	017004
IDENT	C	I*4	N.R.
IQUES		I*2	0000C4

NAME	TAG	TYPE	ADD.
I SF		I*2	000090
Q	C	R*4	N.R.
II SF		I*2	000096
IX SF	C	I*2	011EC4
KK SF		I*2	0000A2
NN SF	C	I*2	017964
X1	C	R*4	N.R.
III SF		I*2	0000AE
ISS SF		I*2	0000B2
IUN	C	I*2	011A14
LT1	C	I*4	N.P.
NN1 SF	C	I*2	017968
TOL	C	R*4	N.R.
INDP	F	C	I*2 017970
IRIM SF	C	I*2	0110B4
NNCE	F	C	I*2 017966
II2M1 SF		I*2	0000BE
NUNCE	F	C	I*2 01796E

NAME	TAG	TYPE	ADD.
J SF		I*2	000092
CO	C	R*4	N.R.
IP SF		I*2	000098
JA SF	C	I*2	0152E4
KO SF		I*2	0000A4
NO SF		I*2	0000AA
X2	C	R*4	N.R.
II2 SF		I*2	0000B0
IST SF		I*2	0000B4
IXR SF	C	I*2	012CD4
LT3	C	I*4	N.P.
NTN	F	C	I*4 000968
CONP	C	R*4	N.R.
INDR	F	C	I*2 017E20
NCEL	C	I*2	N.R.
NODE	S	C	I*4 0012C8
INDX1		I*2	0000C0
IPTIPN SF	C	I*2	010E5C

***** COMMON INFORMATION *****

NAME OF COMMON BLOCK * * SIZE OF BLOCK 018C20 HEXADECIMAL BYTES

VAR. NAME	TYPE	REL. ADDR.
LT1	I*4	N.R.
NODE	I*4	0012CB
F	R*4	N.R.
QN	R*4	N.P.
X3	R*4	N.R.
MMP	I*2	N.R.
IPT	I*2	0109AC
IUN	I*2	011A14
NCEL	I*2	N.R.
NNCE	I*2	017966
NUNCE	I*2	01796E
NCLL	I*2	N.R.

VAR. NAME	TYPE	REL. ADDR.
LT2	I*4	N.R.
A	R*4	N.R.
FF	R*4	N.R.
TOL	R*4	N.R.
X4	R*4	N.R.
IA	I*2	00C99C
IPTIPN	I*2	010E5C
IX	I*2	011EC4
NFUN	I*2	017004
NN1	I*2	017968
INDP	I*2	017970

VAR. NAME	TYPE	REL. ADDR.
NFN	I*4	000008
CO	R*4	N.R.
P	R*4	N.R.
X1	R*4	N.R.
IDENT	I*4	N.R.
INDF	I*2	00CE4C
IRIM	I*2	0110B4
IXR	I*2	012CD4
NPUN	I*2	017484
NUF	I*2	01796A
INDP	I*2	017E20

VAR. NAME	TYPE	REL. ADDR.
NTN	I*4	000968
DELV	P*4	N.R.
Q	P*4	N.R.
X2	P*4	N.R.
CONP	P*4	N.P.
IPN	I*2	0002FC
IQIM	I*2	011564
JA	I*2	0152F4
NN	I*2	017964
NUP	I*2	01796C
IPIM	I*2	018200

LABEL	ADDR
1	0002B8
13	00038C
24	0004D4
19	0006C0

LABEL	ADDR
12	0002E6
15	0003CA
27	00052A
20	000706

LABEL	ADDR
16	000324
14	000418
26	000544
197	00089E

LABEL	ADDR
17	000372
22	0004BA
18	000620
190	0008F8

PAGE 007

102 000926	105 000938	170 000A50	101 000A7C
213 000B22	210 000B6C	180 000BAC	220 000C02
223 000C3A	22 000C5A	224 000CA0	222 000CC4
225 000C0A	30 000D54	31 000D88	32 000DAC
33 000ED4	39 000E0C	38 000E08	37 000E22
39 000F44	36 000E6C	40 000E8E	41 000ERC
44 000F12	43 000F34	48 000FA4	50 000FE2
55 001036	56 00104C	57 00107A	60 001GCC
70 0010E2			

OPTIONS IN EFFECT NAME= MAIN,OPT=00,LINECNT=60,SIZE=0000K,

OPTIONS IN EFFECT SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF

STATISTICS SOURCE STATEMENTS = 203 ,PROGRAM SIZE = 4458

STATISTICS NO DIAGNOSTICS GENERATED

***** END OF COMPILATION *****

99K BYTES OF CORE NOT USED

LEVEL 20.1 (AUG 71)

OS/360 FORTRAN H

DATE 72.227/08.19.16

COMPILER OPTIONS - NAME= MAIN,OPT=00,LINECNT=60,SIZE=0000K,
SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
SUBROUTINE SETQAF(ITOL,IC,MASIX)

ISN 0002

C
C ** REVISED VERSION OF ISAP FOR WATER SYSTEM ANALYSIS **
C SETQAF CALCULATES THE NCE FLOWS AND DETERMINES THE NODE
C IMBALANCE F

1
2
3
4
5
6
7

ISN 0003

IMPLICIT INTEGER*2(I-N)

ISN 0004

INTEGER*4 LT1,LT2,NFN,NTN,NODE,IDENT,KEYJUN

ISN 0005

COMMON LT1,LT2, NFN(600),NTN(600),NODE(600),A(2000), CO(600)
1,DELV(600),F(600),FF(600),P(600),Q(600),QN(600),TOL,X1(600),X2(600)
2,X3(600),X4(600),IDENT(600),CONP(600),MMP(600),TA(600),INDF(600),
3IPN(7000),IPT(600),IPTIPN(300),IRIM(600),IQIM(600),IUN(600),IX(180
40),IXR(5000),JA(2000),NCEL(600),NFUN(600),NPUN(600),NN,NNCE,NNI,NU
5F,NUP,NUNCE,INDP(600),INDR(600),IPIM(600),NCLL(600)
ISN 0006 ITOL=0

14
15
16
17
18
19
20
21
22

ISN 0006

C
C CALCULATE ALL NODE CONNECTING ELEMENT FLOWS

ISN 0007

1 DO 150 J=1,NNCE

ISN 0008

KU=IX(3*J-1)

ISN 0009

KD=IX(3*J)

ISN 0010

LNTYP=NCEL(J)

ISN 0011

GO TO(10,40,70) , LNTYP

C

	C	PIPELINE FLOW	23
	C	'P1' TYPE	24
ISN 0012		10 S=1.	25
ISN 0013		DP=P(KU)-P(KD)-DELV(J)	26
ISN 0014		IF(DP.LT.0.) S=-1.	27
ISN 0016		Q(J)=S*FF(J)**2.63*CD(J)*(S*DP)**.54	28
ISN 0017		GO TO 150	29
	C		30
	C	SPRINKLER FLOW	
	C	'S1' TYPE	
ISN 0018		40 Q(J)=FF(J)*SORT(P(KU))	40
ISN 0019		GO TO 150	45
	C		46
	C	CHECK VALVE ANALYSIS	
	C	'CV' TYPE	
ISN 0020		70 IF(IC.LE.MASIX) GO TO 10	
ISN 0022		DP=P(KU)-P(KD)-DELV(J)	
ISN 0023		IF(X1(J).LT.0.) DP=-DP	
ISN 0025		IF(DP.GT.0.) GO TO 10	
ISN 0027		Q(J)=0.0	
ISN 0028		150 CONTINUE	
	C		72
	C	SUM UP ALL NCE FLOWS AND THE NODE FLOW TO DETERMINE F(NC), THE	72
	C	FLOW IMBALANCE AT NODE NC.	74
ISN 0029		DO 1300 NC=1,NN	75
ISN 0030		N=IPT(NC)+1	76
ISN 0031		NO=IXR(N)	77
ISN 0032		SU=0.	78
ISN 0033		DO 1200 I=1,NO	79
ISN 0034		N1=N+2*I-1	80
ISN 0035		J1=IXR(N1)	81
ISN 0036		J2=J1	82
ISN 0037		IF(J2.LT.0) J2=-J2	83
ISN 0039		IF(J1) 1400,1400,1500	84
ISN 0040		1400 SU=SU+Q(J2)	85
ISN 0041		GO TO 1200	86
ISN 0042		1500 SU=SU-Q(J2)	87
ISN 0043		1200 CONTINUE	88
ISN 0044		F(NC)=SU+QN(NC)	89
	C		90
	C	IF AT ANY ONE NODE, F EXCEEDS THE TOLERANCE , TOL,	91
	C	SET THE FLAG ITOL=1	92
ISN 0045		IF(ABS(F(NC)).GT.TOL) ITOL =1	93
ISN 0047		1300 CONTINUE	94
ISN 0048		RETURN	95
	C		96
ISN 0049		ENTRY JACOB	97
	C		98
	C	CALCULATE THE JACOBIAN(COEFFICIENT MATRIX OF THE SIMULTANEOUS	99
	C	LINEAR EQUATIONS).	100
ISN 0050		JJ=IA(NN1)	101
	C		102
	C	ZERO ALL OF A.	103
ISN 0051		DO 295 J=1,JJ	104
ISN 0052		295 A(J)=0.	105

ISN 0053	DO 500 NC=1,NN	106
	C	107
	C SET THE DERIVATIVE OF F(NC) WITH RESPECT TO THE UNKNOWN NODE	108
	C FLOWS QN(NC).	109
ISN 0054	N=IPTIPN(NC)	110
ISN 0055	K1=IA(NC)	111
ISN 0056	K2=IA(NC+1)-1	112
ISN 0057	IF(IPN(N+1)) 510,520,510	113
ISN 0058	510 J1=IQIM(NC)	114
ISN 0059	DO 515 J=K1,K2	115
ISN 0060	IF(J1-JA(J)) 515,516,515	116
ISN 0061	515 CONTINUE	117
ISN 0062	516 A(J)=1.	118
	C	119
	C DETERMINE THE DERIVATIVE OF F(NC) WITH RESPECT TO THE UNKNOWN	120
	C PRESSURE P(NC). (SUM OF ALL Q DERIVATIVES)	121
ISN 0063	520 IF(IPN(N+1)) 530,540,530	122
ISN 0064	530 NS=IPT(NC)+1	123
ISN 0065	NO=IXR(NS)	124
ISN 0066	SU=0.	125
ISN 0067	DO 550 I=1,NO	126
ISN 0068	N1=NS+2*I	127
ISN 0069	J2=IXR(N1)-1	128
ISN 0070	IF(J2.LT.0) J2=-J2	129
ISN 0072	KA=IXR(N1)	130
ISN 0073	SU=SU+PRSDER(KA,J2,NC)	131
ISN 0074	550 CONTINUE	132
ISN 0075	J1=IPIM(NC)	133
ISN 0076	DO 555 J=K1,K2	134
ISN 0077	IF(J1-JA(J)) 555,556,555	135
ISN 0078	555 CONTINUE	136
ISN 0079	556 A(J)=-SU	137
ISN 0080	540 CONTINUE	138
	C	139
	C DETERMINE THE DERIVATIVE OF F(NC) WITH RESPECT TO THE ADJACENT	140
	C UNKNOWN PRESSURES.	141
ISN 0081	NS=N+3	142
ISN 0082	NPSN=IPN(NS)	143
ISN 0083	IF(NPSN) 600,610,600	144
ISN 0084	600 DO 650 I=1,NPSN	145
ISN 0085	SU=0.	146
ISN 0086	605 N2=NS+2*I	147
ISN 0087	J2=IPN(N2-1)	148
ISN 0088	KA=IPN(N2)	149
ISN 0089	IF(KA) 506,620,620	150
ISN 0090	606 KA=-KA	151
ISN 0091	SU=SU+PRSDER(NC,J2,KA)	152
ISN 0092	NS=NS+2	153
ISN 0093	GO TO 625	154
ISN 0094	620 J1=IPIM(KA)	155
ISN 0095	DO 645 J=K1,K2	156
ISN 0096	IF(J1-JA(J)) 645,646,645	157
ISN 0097	645 CONTINUE	158
ISN 0098	646 A(J)=SU*PRSDER(NC,J2,KA)	159
ISN 0099	650 CONTINUE	160


```

ISN 0100      61 C NS=NS+2*NPSN+1                                161
ISN 0101      NPSN=IPN(NS)                                       162
ISN 0102      IF(NPSN) 700,710,700                                163
C                                                       164
C      DETERMINE THE DERIVATIVE OF F(NC) WITH RESPECT TO ADJACENT 165
C      UNKNOWN NCE PARAMETERS.                                   166
ISN 0103      700 DO 750 I=1,NPSN                                167
ISN 0104      N2=NS+2*I                                           168
ISN 0105      J2=IPN(N2-1)                                         169
ISN 0106      IF(J2.LT.0) J2=-J2                                  170
ISN 0108      KA=IPN(N2)                                           171
ISN 0109      J1=IRIM(J2)                                          172
ISN 0110      DO 745 J=K1,K2                                       173
ISN 0111      IF(J1-JA(J)) 745,746,745                             174
ISN 0112      745 CONTINUE                                         175
ISN 0113      746 A(J)=FCEDER(NC,J2,KA)                           176
ISN 0114      750 CONTINUE                                         177
ISN 0115      710 CONTINUE                                         178
ISN 0116      500 CONTINUE                                         179
ISN 0117      RETURN                                              180
ISN 0118      END                                                  181

```

/ SETQAF / SIZE OF PROGRAM 000B5E HEXADECEMAL BYTES PAGE 004

NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.
A	S	C	R*4 001C28	F	S	A	C R*4 005DC8	I	SF	I*2	000004	J	SF	I*2	0000D6
N	SF	I*2	0000D8	P	FA	C	R*4 007088	Q	SF	C	R*4 0079E8	S	SF	R*4	000000
CO	F	C	R*4 004808	DP	SF		R*4 000104	FF	F	C	R*4 006728	TA	F	C	I*2 00C99C
IC		I*2	0000DA	IX	F	C	I*2 011EC4	JA	C	I*2	0153E4	JJ	SF	I*2	0000DC
J1	SF	I*2	0000DE	J2	SFA	I*2	0000E0	KA	SFA	I*2	0000E2	KD	SF	I*2	0000E4
KU	SFA	I*2	0000E6	K1	SF	I*2	0000E8	K2	SF	I*2	0000EA	NC	SFA	I*2	0000EC
NN	F	C	I*2 017964	NO	SF	I*2	0000EE	NS	SF	I*2	0000F0	N1	SF	I*2	0000F2
N2	SF	I*2	0000F4	QN	F	C	R*4 008248	SU	SF		R*4 000108	X1		C	R*4 008CAC
X2		C	R*4 N.R.	X3		C	R*4 N.R.	X4		C	R*4 N.R.	IPN	F	C	I*2 00D2FC
IPN	F	C	I*2 0109AC	IUN		C	I*2 N.R.	IXR	F	C	I*2 012C04	LT1		C	I*4 N.R.
LT3		C	I*4 N.R.	MMP		C	I*2 N.R.	NFN		C	I*4 N.R.	NN1	F	C	I*2 017968
NTN		C	I*4 N.R.	NUF		C	I*2 N.R.	NUP		C	I*2 N.R.	TOL		C	R*4 008FA8
CONP		C	R*4 N.R.	DELV	F	C	R*4 005468	INDF		C	I*2 N.R.	INDP		C	I*2 N.R.
INDR		C	I*2 N.R.	IPIM	F	C	I*2 0182D0	IQIM	F	C	I*2 011564	IRIM	F	C	I*2 011084
ITDL	S	I*2	0000F6	NCEL	F	C	I*2 016B54	NCLL		C	I*2 N.R.	NFUN		C	I*2 N.R.
NNCE	F	C	I*2 017966	NODE		C	I*4 N.R.	NPSN	SF	I*2	0000F8	NPUN		C	I*2 N.R.
IDENT		C	I*4 N.R.	JACOB		I*2	0000FA	LNTYP	SF	I*2	0000FC	MASIX		I*2	0000FE
NUNCE		C	I*2 N.R.	SHORT	XF	R*4	000000	FRXPR		XF	R*4 000000	FCEDER	F	XF	R*4 000000
IPN	F	C	I*2 010E5C	KEYJUN		I*4	N.R.	PRSDER	F	XF	R*4 000000	SETQAF		R*4	00010C

***** COMMON INFORMATION *****

NAME OF COMMON BLOCK * * SIZE OF BLOCK 018C30 HEXADECIMAL BYTES

VAR. NAME	TYPE	REL. ADDR.	VAR. NAME	TYPE	REL. ADDR.	VAR. NAME	TYPE	REL. ADDR.	VAR. NAME	TYPE	REL. ADDR.
LT1	I*4	N.R.	LT2	I*4	N.R.	NFN	I*4	N.R.	NTN	I*4	N.R.
NODE	I*4	N.R.	A	F*4	001C28	CO	R*4	004B08	DELV	R*4	005468
F	R*4	005DC8	FF	F*4	006728	P	R*4	007088	Q	P*4	0079E8
QN	R*4	008348	TOL	R*4	008CA8	X1	R*4	008CAC	X2	R*4	N.R.
X3	R*4	N.R.	X4	F*4	N.R.	IDENT	I*4	N.R.	COMP	R*4	N.R.
MMP	I*2	N.R.	IA	I*2	00C99C	INDF	I*2	N.R.	IPN	I*2	00D2FC
IPT	I*2	0109AC	IPNIPN	I*2	010E5C	IRIM	I*2	0110B4	IQIM	I*2	011564
IUN	I*2	N.R.	IX	I*2	011EC4	IXP	I*2	012CD4	JA	I*2	0153E4
NCEL	I*2	016B54	NFUN	I*2	N.R.	NPUN	I*2	N.R.	NN	I*2	017964
NNCE	I*2	017966	NNI	I*2	017968	NUF	I*2	N.R.	NUP	I*2	N.R.
NUNCE	I*2	N.R.	ENDP	I*2	N.R.	INDP	I*2	N.R.	IPIM	I*2	0182D0
NCLL	I*2	N.R.									

LABEL	ADDR	LABEL	ADDR	LABEL	ADDR	LABEL	ADDR	PAGE 005
1	000288 NR	10	0002EC	40	0003D2	70	000432	
150	0004C2	1400	000566	1500	000584	1200	00059C	
1300	000602	295	000644	510	0006CA	515	00070C	
516	000716	520	00072A	530	000748	550	0007EC	
555	000338	556	000E4E	540	000864	600	00088A	
605	00039A	606	000ED4	620	00090E	645	000944	
646	00095A	650	00098C	610	0009A2	700	0009DC	
745	000A56	746	000A6C	750	000A9A	710	000AB0	
500	000AB0							

OPTIONS IN EFFECT NAME= MAIN,OPT=06,LINFCNT=60,SIZE=0000K,

OPTIONS IN EFFECT SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF

STATISTICS SOURCE STATEMENTS = 117 ,PROGRAM SIZE = 2910

STATISTICS NO DIAGNOSTICS GENERATED

***** END OF COMPILATION *****

111K BYTES OF CORE NOT USED

```

      COMPILER OPTIONS - NAME= MAIN,OPT=00,LINECNT=60,SIZE=0000K,
                        SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,TD,NOXREF
ISN 0002      FUNCTION PRSDER(N,J,K)
C
C      ** REVISED VERSION OF ISAP FOR WATER SYSTEM ANALYSIS **
C      PRSDER DETERMINES THE DERIVATIVE OF THE EQUATION FOR Q OF THE
C      ELEMENT BETWEEN NODE N AND K WITH RESPECT TO THE PRESSURE AT
C      NODE K.
C
ISN 0003      IMPLICIT INTEGER*2(I-N)
ISN 0004      INTEGER*4 LT1,LT3,NFN,NTN,NODE,IDENT,KEYJUN
ISN 0005      COMMON LT1,LT3, NFN(600),NTN(600),NODE(600),A(3000), CO(600)
1,DELV(600),F(600),FF(600),P(600),Q(600),QN(600),TOL,X1(600),X2(600)
2,X3(600),X4(600),IDENT(600),CONP(600),MMP(600),IA(600),INDF(600),
3IPN(7000),IPT(600),IPTPN(300),IRIM(600),IQIM(600),IUN(600),IX(180
40),IXR(5000),JA(3000),NCEL(600),NFUN(600),NPUN(600),NN,NNCE,NNI,NU
5F,NUP,NUNCE,INDP(600),INDR(600),IPIM(600),NCLL(600)
ISN 0006      1 NC=N
ISN 0007      J2=J
ISN 0008      KA=K
ISN 0009      LNTYP=NCEL(J2)
ISN 0010      GO TO(10,40,70) , LNTYP
C
C      NODE CONNECTING ELEMENT IS A PIPE
C      'P1' TYPE
ISN 0011      10 DP=ABS(P(KA)-P(1NC))-ABS(DELV(J2))
ISN 0012      IF(DP.EQ.0.)DP=10.
ISN 0014      S=1.
ISN 0015      IF(DP.LT.0.) S=-1.
ISN 0017      PRSDER=.54*FF(J2)**2.63*CO(J2)/(S*DP)**.46
ISN 0018      RETURN
C
C      NODE CONNECTING ELEMENT IS A SPRINKLER HEAD
C      'S1' TYPE
ISN 0019      40 IF(P(KA).GT..01) GO TO 45
C      NODE KA IS A SPRINKLER DISCHARGE NODE
ISN 0021      PRSDER=0.0
ISN 0022      RETURN
C
C      NODE KA IS A SPRINKLER SUPPLY NODE
ISN 0023      45 PRSDER=FF(J2)/(2*SQRT(P(KA)))
ISN 0024      RETURN
C
C      NODE CONNECTING ELEMENT IS A CHECK VALVE
C      'CV' TYPE
ISN 0025      70 IF(Q(J2).NE.0.) GO TO 10
ISN 0027      PRSDER=0.0
ISN 0028      RETURN
C
ISN 0029      ENTRY FCEDER(N,J,K)
C
C      FCEDER CALCULATES THE DERIVATIVE OF Q WITH RESPECT TO THE
C      APPLICABLE NODE CONNECTING ELEMENT PARAMETER.
C

```

182
183
184
185
186
187
188
189
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
220
265
266
267
268
269
270

```

ISN 0030      NC=N                                271
ISN 0031      J2=J                                272
ISN 0032      KA=K                                273
ISN 0033      LNTYP=NCEL(J2)                      274
ISN 0034      GO TO (20,50,80) , LNTYP            275
C                                                     276
C      NODE CONNECTING ELEMENT IS A PIPE          277
C      'PI' TYPE                                  278
ISN 0035      20 DD=ABS(P(KA)-P(NC))-ABS(DELV(J2)) 279
ISN 0036      IF(DD.EQ.0.1DD=1.0.                280
ISN 0037      S=1.                                281
ISN 0038      IF(DD.LT.0.)S=-1.                   282
ISN 0039      FCEDER=S*2.63**F(J2)**1.63*CO(J2)*IS*DD)**.54 283
ISN 0040      RETURN                              284
ISN 0041      RETURN                              285
C
C      NODE CONNECTING ELEMENT IS A SPRINKLER HEAD
C      'SI' TYPE
ISN 0042      50 IF(P(NC).GT..08) GO TO 55
C      NODE NC IS A SPRINKLER DISCHARGE NODE
ISN 0043      FCEDER= SQRT(P(KA))
ISN 0044      RETURN
C      NODE NC IS A SPRINKLER SUPPLY NODE
ISN 0045      55 FCEDER=-SQRT(P(NC))
ISN 0046      RETURN
C
C      NODE CONNECTING ELEMENT IS A CHECK VALVE
C      'CV' TYPE
ISN 0047      80 IF(Q(J2).NE.0.) GO TO20
ISN 0048      FCEDER=0.0
ISN 0049      RETURN
ISN 0050      END
ISN 0051
ISN 0052
ISN 0053

```

/ PRSDER / SIZE OF PROGRAM 000642 HEXADECIMAL BYTES PAGE 003.

NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	
A	C	R*4	N.R.	F	C	R*4	N.R.	J	F	I*2	0000E8	K	F	I*2	0000EA	
N	F	I*2	0000EC	P	FA	C	R*4	007088	Q	C	R*4	0079E8	S	SF	R*4	0000F8
CO	F	C	R*4	DD	SF	R*4	0000FC	DP	SF	R*4	000100	FF	F	C	R*4	006728
IA	C	I*2	N.R.	IX	C	I*2	N.R.	JA	C	I*2	N.R.	J2	SFA	I*2	0000FE	
KA	SFA	I*2	0000F0	NC	SFA	I*2	0000F2	NN	C	I*2	N.R.	GN	C	R*4	N.R.	
X1	C	R*4	N.R.	X2	C	R*4	N.R.	X3	C	R*4	N.R.	X4	C	R*4	N.R.	
IPN	C	I*2	N.R.	IPN	C	I*2	N.R.	IUN	C	I*2	N.R.	TYR	C	I*2	N.R.	
LTJ	C	I*4	N.R.	LT3	C	I*4	N.R.	MMP	C	I*2	N.R.	NFN	C	I*4	N.R.	
NN1	C	I*2	N.R.	NTN	C	I*4	N.R.	NUF	C	I*2	N.R.	NLP	C	I*2	N.R.	
TOL	C	R*4	N.R.	CONP	C	R*4	N.R.	DELV	FA	C	R*4	005468	INCF	C	I*2	N.R.
INDP	C	I*2	N.R.	INDR	C	I*2	N.R.	IPIM	C	I*2	N.R.	IQIM	C	I*2	N.R.	
IRIM	C	I*2	N.R.	NCEL	F	C	I*2	006354	NCLL	C	I*2	N.R.	NELN	C	I*2	N.R.
NNCE	C	I*2	N.R.	NODE	C	I*4	N.R.	NPUN	C	I*2	N.R.	ICENT	C	I*4	N.R.	
LNTYP	SF	I*2	0000F4	NUNCE	C	I*2	N.R.	SOPT	XF	R*4	000000	FR>PR=	XF	P*4	000000	
FCEDER	S	R*4	000108	IPITPN	C	I*2	N.R.	KFYJUN	I*4	N.R.	PRSDER	S	R*4	000108		

***** COMMON INFORMATION *****

NAME OF COMMON BLOCK * * SIZE OF BLOCK 018C30 HEXADECIMAL BYTES

VAR. NAME	TYPE	REL. ADDR.	VAR. NAME	TYPE	REL. ADDR.	VAR. NAME	TYPE	REL. ADDR.	VAR. NAME	TYPE	REL. ADDR.
LT1	I*4	N.F.	LT3	I*4	N.R.	NFN	I*4	N.R.	MTN	I*4	N.R.
NODE	I*4	N.F.	A	R*4	N.R.	CO	R*4	004808	DELV	P*4	005468
F	R*4	N.R.	FF	R*4	006728	P	P*4	007088	Q	R*4	0079E8
QN	R*4	N.R.	TOL	R*4	N.R.	X1	R*4	N.R.	X2	P*4	N.R.
X3	R*4	N.F.	X4	R*4	N.R.	IDENT	I*4	N.R.	CONP	R*4	N.R.
HMP	I*2	N.F.	IA	I*2	N.R.	INDF	I*2	N.R.	IPN	I*2	N.R.
IPT	I*2	N.R.	PTIPN	I*2	N.R.	IRIM	I*2	N.R.	IQIM	I*2	N.R.
IUN	I*2	N.F.	JX	I*2	N.R.	IXR	I*2	N.R.	JA	I*2	N.R.
NCEL	I*2	016B54	NFUN	I*2	N.R.	NPUN	I*2	N.R.	NN	I*2	N.R.
NNCE	I*2	N.F.	NNI	I*2	N.R.	NUF	I*2	N.R.	NUP	I*2	N.R.
NUNCE	I*2	N.F.	INOP	I*2	N.R.	INOR	I*2	N.R.	IPIM	I*2	N.R.
NCLL	I*2	N.R.									

LABEL	ADDR	LABEL	ADDR	LABEL	ADDR	LABEL	ADDR	PAGE
1	0001C4	10	000208	40	0002F6	45	000320	004
70	00037E	20	0003EA	50	0004E4	55	000538	
80	000574							

OPTIONS IN EFFECT NAME= MAIN,OPT=00,LINECNT=60,SIZE=0000K,

OPTIONS IN EFFECT SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF

STATISTICS SOURCE STATEMENTS = 52 ,PROGRAM SIZE = 1602

STATISTICS NO DIAGNOSTICS GENERATED

***** END OF COMPILATION *****

127K BYTES OF CORE NOT USED

LEVEL 20.1 (AUG 71)

OS/360 FORTRAN H

DATE 72.237/08.19.23

COMPILER OPTIONS - NAME= MAIN,OPT=00,LINECNT=60,SIZE=0000K,
SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
SUBROUTINE ELMPIV(N,IA,JA,A,IU,JU,U,IHQW,KS)

ISN 0002

C

1
2

	C	FOR ALL REMAINING ROWS FIND THE ROWS WITH NON-ZERO PIVOT COLUMN	53
	C	ELEMENTS AND THE POSITION OF THE PIVOT COLUMN IN THE ROW.	54
	C		55
ISN 0031		DO 25 JJ=II,N	56
ISN 0032		I=IHOW(JJ)	57
ISN 0033		L1=IU(I)	58
ISN 0034		L2=IU(I+1)-1	59
ISN 0035		DO 20 LL=L1,L2	60
ISN 0036		IF(I1-JU(LL)) 25,21,20	61
ISN 0037	20	CONTINUE	62
ISN 0038		GO TO 25	63
ISN 0039	21	IT=IT+1	64
ISN 0040		K=III+2*IT+1	65
ISN 0041		IHOW(K)=I	66
ISN 0042		IHOW(K-1)=LL-L1+1	67
ISN 0043	25	CONTINUE	68
ISN 0044		IF(IT) 26,101,26	69
ISN 0045	101	KS=2	70
ISN 0046		WRITE(LT3,310) II,III	70
ISN 0047	310	FORMAT('ZERO COLUMN DETECTED, COLUMN NO.',I4,' MAXIMUM INDEX OF	70
		I IHOW AT THIS TIME=',I6)	70
ISN 0048		WRITE(LT3,315) (IHOW(I),I=1,III)	70
ISN 0049	315	FORMAT('O IHOW(') TO IHOW(III). A COLUMN OF ZEROS HAS BEEN FOUND' /	70
		1/(2015))	70
ISN 0050		WRITE(LT3,325)	70
ISN 0051	325	FORMAT('NON-ZERO TERMS OF PARTIALLY TRIANGULARIZED MATRIX U')	70
ISN 0052		DO 230 I=1,N	70
ISN 0053		K1=IU(I)	70
ISN 0054		K2=IU(I+1)-1	70
ISN 0055		WRITE(LT3,335) I	70
ISN 0056		WRITE(LT3,340) (U(J),J=K1,K2)	70
ISN 0057	330	WRITE(LT3,345) (JU(J),J=K1,K2)	70
ISN 0058	335	FORMAT(2X,'NODE',I5)	70
ISN 0059	340	FORMAT(6X,8F15.4)	70
ISN 0060	345	FORMAT(2X,8I15)	70
	C		71
	C	NO NON-ZERO ELEMENT LEFT IN THE COLUMN.	72
ISN 0061		RETURN	73
ISN 0062	26	IS=2*IT	74
	C		75
	C	CHOOSE THE LARGEST NON-ZERO ELEMENT AS THE PIVOT ELEMENT	76
ISN 0063		AMAX=1.E-15	77
ISN 0064		IMIN=0	78
ISN 0065		IIZ=III+1	79
ISN 0066		DO 30 I=2,IS,2	80
ISN 0067		ISS=IIZ+1	81
ISN 0068		IJJ=IHOW(ISS)	82
ISN 0069		IOX=IU(IJJ)+IHOW(ISS-1)-1	83
ISN 0070		IF(ABS(U(IOX))-AMAX) 30,30,27	84
ISN 0071	27	IMIN=ISS	85
ISN 0072		AMAX=ABS(U(IOX))	86
ISN 0073	30	CONTINUE	87
ISN 0074		IF(IMIN) 32,31,32	88
ISN 0075	31	KS=3	89
	C		90

	C	A PREVIOUSLY NON-ZERO VALUE HAS BEEN MADE ZERO AND AN ATTEMPT	91
	C	MADE TO USE IT AS A PIVCT.	92
ISN 0076		RETURN	93
ISN 0077	32	IP=IHOW(IMIN)	94
ISN 0078		IP=IHOW(IMIN-1)	95
ISN 0079		IHOW(IMIN)=IHOW(I1+3)	96
ISN 0080		IHOW(IMIN-1)=IHOW(I1+2)	97
ISN 0081		IHOW(I1+1)=IP	98
ISN 0082		IHOW(I1+2)=IPT	99
ISN 0083		IT=IT-1	100
ISN 0084		IHOW(I1+3)=IT	101
	C		102
	C	INTERCHANGE ROWS IN IHOW(1---N)	103
ISN 0085	55	DO 60 JJ=I1,N	104
ISN 0086		IF(IP-IOW(JJ)) 50,61,50	105
ISN 0087	60	CONTINUE	106
ISN 0088	61	IHOW(JJ)=IHOW(I1)	107
ISN 0089		IHOW(I1)=IF	108
	C		109
	C	RECOMPUTE JU'S FOR EACH ROW THAT MUST BE CHANGED(NEED TO	110
	C	CONSIDER ONLY THOSE COLUMNS BEYOND I1).	111
ISN 0090		IF(IT) 301,301,120	112
	C		113
	C	IF NO ROWS NEED CHANGED, NO CHANGES ARE NECESSARY IN IU OR JU	114
	C	(GO TO 301)	115
ISN 0091	120	IS=I1+3	116
ISN 0092		L1=IPT+IU(IP)	117
ISN 0093		L2=IU(IP+1)-1	118
ISN 0094		IF(L2-L1) 301,130,130	119
ISN 0095	130	CONTINUE	120
	C		121
	C	COMPUTE NUMBER OF POSITIONS TO SHIFT JU AND U FOR EACH POW	122
ISN 0096		DO 300 J=1,IT	123
ISN 0097		ISS=IS+2*J	124
ISN 0098		L1=IPT+IU(IP)	125
ISN 0099		L2=IU(IP+1)-1	126
ISN 0100		NR=IHOW(ISS)	127
ISN 0101		K1=IU(NR)+IHOW(ISS-1)	128
ISN 0102		K2=IU(NR+1)-1	129
ISN 0103		ITOT=0	130
ISN 0104		IF(K2-K1) 150,160,160	131
ISN 0105	150	ITOT=L2-L1+1	132
ISN 0106		GO TO 201	133
ISN 0107	160	DO 200 JJ=L1,L2	134
ISN 0108		ITES=JU(JJ)	135
ISN 0109		DO 190 KK=K1,K2	136
ISN 0110		IF(ITES-JU(KK)) 191,200,190	137
ISN 0111	190	CONTINUE	138
ISN 0112	191	ITOT=ITOT+1	139
ISN 0113	200	CONTINUE	140
ISN 0114		IF(ITOT) 201,240,201	141
	C		142
	C	SHIFT JU'S AND U'S WITH INDICES FROM K1 TO IU(NN1)-1. ALSO	143
	C	CHANGE THE IU'S WITH INDICES NR+1 TO NN1	144
ISN 0115	201	KT=IU(NN1)-1	145

ISN 0116	KMAX=K1-K1+1	146
ISN 0117	DO 210 KK=1,KMAX	147
ISN 0118	KM=KT-KK+1	148
ISN 0119	KU=KM+ITOT	149
ISN 0120	U(KU)=U(KM)	150
ISN 0121	210 JU(KU)=JU(KM)	151
	C	152
	C CHANGE APPROPRIATE IU'S	153
ISN 0122	KM=NR+1	154
ISN 0123	DO 220 KK=KM,NN1	155
ISN 0124	220 IU(KK)=IU(KK)+ITOT	156
ISN 0125	KM=K1-1	157
	C	158
	C SHUFFLE PIVOT ROW(IP) INTO THE ROW BEING WORKED ON (NR).	159
ISN 0126	K1=K1+ITOT	160
ISN 0127	K2=K2+ITOT	161
ISN 0128	L1=IP+IU(IP)	162
ISN 0129	L2=IU(IP+1)-1	163
ISN 0130	JJ=L1	164
ISN 0131	KK=K1	165
ISN 0132	IF(K2-K1) 238,231,231	166
ISN 0133	231 KM=KM+1	167
ISN 0134	IF(JU(JJ)-JU(KK)) 232,233,236	168
ISN 0135	232 JU(KM)=JU(JJ)	169
ISN 0136	U(KM)=0.	170
ISN 0137	JJ=JJ+1	171
ISN 0138	IF(JJ-L2) 231,231,240	172
ISN 0139	233 JU(KM)=JU(JJ)	173
ISN 0140	U(KM)=U(KK)	174
ISN 0141	JJ=JJ+1	175
ISN 0142	IF(JJ-L2) 234,234,240	176
ISN 0143	234 KK=KK+1	177
ISN 0144	IF(KK-K2) 231,231,238	178
ISN 0145	236 JU(KM)=JU(KK)	179
ISN 0146	U(KM)=U(KK)	180
ISN 0147	KK=KK+1	181
ISN 0148	IF(KK-K2) 231,231,238	182
ISN 0149	238 DO 239 JJ=JJ,L2	183
ISN 0150	KM=KM+1	184
ISN 0151	U(KM)=0.	185
ISN 0152	239 JU(KM)=JU(JJ)	186
ISN 0153	240 CONTINUE	187
	C	188
	C PERFORM THE ROW OPERATIONS ON ROW NR.	189
ISN 0154	LO=L1-1	190
ISN 0155	KO=IU(NR)+IHOW(ISS-1)-1	191
ISN 0156	AM=-U(KO)/U(LO)	192
ISN 0157	DO 250 M=L1,L2	193
ISN 0158	242 KO=KO+1	194
ISN 0159	IF(JU(M)-JU(KO)) 243,243,242	195
ISN 0160	243 U(KO)=U(KO)+AM*U(M)	196
ISN 0161	250 CONTINUE	197
ISN 0162	300 CONTINUE	198
	C	199
	C SET THE POINTER IN IHOW(N+1)...IHOW(2N).	200

```

ISN 0163      301 LMF=N+11
ISN 0164      HOW(LMF)=111+1
ISN 0165      111=111+3+2*11
ISN 0166      500 CONTINUE
ISN 0167      RETURN
ISN 0168      END

```

```

201
202
203
204
205
206

```

/ ELMPIV / SIZE OF PROGRAM 00152 HEXADECIMAL BYTES PAGE 005

NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.
A	F	XR	R*4 000000	I	SF	I*4 00013C		J	SF	I*4 000190		K	SF	I*4 000194	
M	SF	I*4	000198	N	SF	I*2 000138		U	SFA XR	R*4 000000		AM	SF	R*4 00019C	
IA	F	XR	I*2 000000	II	SF	I*4 0001A0		IP	SF	I*4 0001A4		IS	SF	I*4 0001A8	
IT	SF	I*4	0001AC	IU	SF XR	I*2 000000		JA	F XR	I*2 000000		IJ	SF	I*4 0001B0	
JU	SF XR	I*2	000000	KK	SF	I*4 0001B4		KM	SF	I*4 0001B8		KD	SF	I*4 0001BC	
KS	S	I*2	00018A	KT	SF	I*4 0001C0		KU	SF	I*4 0001C4		KL	SF	I*4 0001C8	
K2	SF	I*4	0001CC	LL	SF	I*4 0001D0		LD	SF	I*4 0001D4		LI	SF	I*4 0001D8	
L2	SF	I*4	0001DC	NR	SF	I*4 0001E0		IIJ	SF	I*4 0001F4		IIZ	SF	I*4 0001E8	
IJJ	SF	I*4	0001EC	IPT	SF	I*4 0001F0		IQX	SFA	I*4 0001F4		ISS	SF	I*4 0001F8	
LMF	SF	I*4	0001FC	LT3	SF	I*4 000200		NMI	SF	I*4 000204		AMAX	S	R*4 000208	
HOW	SF XR	I*4	000000	IMEN	SF	I*4 00020C		ITES	S	I*4 000210		ITCT	SF	I*4 000214	
KMAX	SF	I*4	000218	ERRDR	SF XF	R*4 000000		ELMPIV		R*4 00021C		IRCON	F XF	I*4 000000	

LABEL	ADDR	LABEL	ADDR	LABEL	ADDR	LABEL	ADDR	PAGE
1	0003D4	3	000430	2	000440	5	0004C0	
8	000532	1C	0C0570	20	000634	21	000650	
25	0006A4	101	0C06F6	30	0007E2	26	000848	
27	0008F2	3C	0C0900	31	000922	32	000932	
55	0009DA NR	6C	0C09FC	61	000A12	120	000A52	
130	000A94	15C	0C0B48	160	000A5E	190	000B22	
191	000B38	20C	0C0BC4	201	000B56	210	000C4A	
220	000C94	231	000C3E	232	000D7A	233	000DCE	
234	000E2C	236	000E48	238	000EA6	239	000FCC	
240	000FD6	242	000F76	243	000FAC	250	000FE8	
300	000FE	301	001014	500	001058			

OPTIONS IN EFFECT NAME= MAIN,DPT=00,LINECNT=60,SIZE=0000K,
 OPTIONS IN EFFECT SOURCE,EBCDIC,NOLIST,NODECK,LOAD.MAP,NOEDIT,LD,NQXREF

STATISTICS SOURCE STATEMENTS = 161, PROGRAM SIZE = 4434

STATISTICS NO DIAGNOSTICS GENERATED

***** END OF COMPILATION *****

107K BYTES OF COPE NOT USED

```

      COMPILER OPTIONS - NAME= MAIN,OPT=00,LINECNT=60,SIZE=0000K,
                        SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
ISN 0002      SUBROUTINE ELIMA(N,IA,JA,A,IU,JU,U,IHOW,KS)
C
C      THIS ROUTINE TRIANGULARIZES THE MATRIX A(STORED IN VECTOR FORM
C      GIVEN BY IA AND JA) BY PROCESSING THE INDEX VECTOR IHOW.
C      THE RESULTING MATRIX U IS STORED IN VECTOR FORM GIVEN BY IU&JU.
ISN 0003      INTEGER*2 N,IA,JA,IU,JU,KS
ISN 0004      DIMENSION IU(1),JU(1),U(1),IHOW(1),IA(1),JA(1),A(1)
ISN 0005      1 KS=0
C
C      PUT VECTOR A INTO VECTOR U
ISN 0006      JJ=IU(N+1)
ISN 0007      DO 79 J=1,JJ
ISN 0008      79 U(J)=0.
ISN 0009      DO 80 I=1,N
ISN 0010      K1=IA(I)
ISN 0011      K2=IA(I+1)-1
ISN 0012      IF(K2-K1) 80,55,55
ISN 0013      55 L1=IU(I)
ISN 0014      L2=IU(I+1)-1
ISN 0015      LL=-1
ISN 0016      DO 80 KK=K1,K2
ISN 0017      56 LL=LL+1
ISN 0018      L3=L1+LL
ISN 0019      IF(JU(L3)-JA(KK)) 56,57,56
ISN 0020      57 U(L3)=A(KK)
ISN 0021      80 CONTINUE
C
C      PERFORM ROW OPERATIONS ON U.
ISN 0022      NNM1=N-1
ISN 0023      III=2*N
ISN 0024      DO 50 II=1,NNM1
ISN 0025      IT=IHOW(III+2)
ISN 0026      IF(IT) 10,48,10
ISN 0027      10 IP=IHOW(III+1)
ISN 0028      L1=IU(IP)+IHOW(III+2)
ISN 0029      L0=L1-1
ISN 0030      IF(ABS(U(L0))-1.E-15) 25,25,22
ISN 0031      25 KS=4
C
C      A PREVIOUSLY NON-ZERO VALUE HAS BEEN MADE ZERO AND AN ATTEMPT
C      MADE TO USE IT AS A PIVOT.
ISN 0032      RETURN
ISN 0033      22 L2=IU(IP+1)-1
ISN 0034      IF(L2-L1) 48,20,20
ISN 0035      20 DO 45 J=1,IT
ISN 0036      ISS=III+2*J+2
ISN 0037      NR=IHOW(ISS)
ISN 0038      K1=IU(NR)+IHOW(ISS-1)
ISN 0039      K0=K1-1
ISN 0040      AM=-U(K0)/U(L0)
ISN 0041      DO 40 M=L1,L2

```

207
208
209
210
211
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257

```

      C      LINE UP APPROPRIATE COLUMNS.
ISN 0042      30 KO=KC+1
ISN 0043      IF(JL:M)-JU(KO)35,35,30
ISN 0044      35 U(KO)=U(KO)+AM*U(M)
ISN 0045      40 CONTINUE
ISN 0046      45 CONTINUE
ISN 0047      48 III=III+2*IT+3
ISN 0048      50 CONTINUE
ISN 0049      RETURN
ISN 0050      END

```

```

258
259
260
261
262
263
264
265
266
267
268

```

/ ELIMA / SIZE OF PROGRAM 0005F4 HEXADECEIMAL BYTES PAGE 003

NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.
A	F XR	R*4	000000	I	SF	I*4	000C94	J	SF	I*4	000098	M	SF	I*4	00009C
N	F	I*2	000090	U	SFA XR	R*4	000000	AM	SF	R*4	0000A0	IA	F XR	I*2	000000
II	SF	I*4	0000A4	IP	SF	I*4	0000A8	IT	SF	I*4	0000AC	IU	F XR	I*2	000000
JA	XR	I*2	000000	JJ	SF	I*4	0000B0	JU	XR	I*2	000000	KK	SF	I*4	0000B4
KO	SF	I*4	0000B8	KS	S	I*2	000092	KI	SF	I*4	0000BC	KZ	SF	I*4	0000C0
LL	SF	I*4	0000C4	LD	S=A	I*4	0000C8	LI	SF	I*4	0000CC	LZ	SF	I*4	0000D0
L?	SF	I*4	0000D4	NR	S=	I*4	0000D8	III	SF	I*4	0000DC	ISS	SF	I*4	0000E0
IHOW	F XR	I*4	000000	NMM1	S=	I*4	0000E4	ELIMA		R*4	0000E8				

LABEL	ADDR	LABEL	ADDR	LABEL	ADDR	LABEL	ADDR	PAGE	004
1	000190	7E	0001B4	55	000218	56	000232		
57	000292	8C	0C02PA	10	000322	25	000332		
22	000392	2C	0C03B4	20	000448	35	00047E		
40	0004BA	4E	0C04B0	48	0004E6	50	0004FA		

```

*OPTIONS IN EFFECT*      NAME= MAIN,OPT=00,LINECNT=60,SIZE=0000K,
*OPTIONS IN EFFECT*      SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,IO,NOXPEF
*STATISTICS*      SOURCE STATEMENTS =      45 ,PROGRAM SIZE =      1524
*STATISTICS*      NO DIAGNOSTICS GENERATED
***** END OF COMPILATION *****

```

122K BYTES OF CORE NOT USED

LEVEL 20.1 (AUG 71)

OS/360 FORTRAN H

DATE 72.227/08.19.30

COMPILER OPTIONS - NAME= MAIN,OPT=00,LINECNT=60,SIZE=0000K,
SOURCE,E3CDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF

```

ISN 0002      SUBROUTINE ELIMB(N,IU,JU,IHOW,U,B)                269
C                                                     270
C      THIS ROUTINE APPLIES THE PIVOTING OPERATION TO THE RIGHT HAND 271
C      SIDE,B, THAT HAS ALREADY BEEN PERFORMED ON U.                272
ISN 0003      INTEGER*2 N,IU,JU                                274
ISN 0004      DIMENSION IU(1),JU(1),IHOW(1),U(1),B(1)           275
ISN 0005      1 NNM1=N-1                                         276
ISN 0006      III=2*N                                           277
ISN 0007      DO 75 II=1,NNM1                                    278
ISN 0008      IT=IHOW(III+2)                                     279
ISN 0009      IF(IT) 55,70,55                                    280
ISN 0010      55 IP=IHOW(III+1)                                  281
ISN 0011      LO=IU(IP)+IHOW(III+2)-1                            282
ISN 0012      DO 60 J=1,IT                                       283
ISN 0013      ISS=III+2*J+3                                       284
ISN 0014      NR=IHOW(ISS)                                       285
ISN 0015      KO=IU(NR)+IHOW(ISS-1)-1                            286
ISN 0016      60 B(NR)=B(NR)-U(KO)*B(IP)/U(LO)                  287
ISN 0017      70 III=III+2*IT+3                                   288
ISN 0018      75 CONTINUE                                       289
ISN 0019      RETURN                                           290
ISN 0020      END

```

/ ELIMB / SIZE OF PROGRAM 00034C HEXADECIMAL BYTES PAGE 002

NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.
B	SF	XR	R*4 000000	J	SF	I*4 000088		N	F	I*2 00C084		U	F	XR	R*4 000000
II	SF		I*4 00008C	IP	SF	I*4 000090		IT	SF	I*4 00C094		IU	F	XR	I*2 000000
JU		XR	I*2 000000	KO	SF	I*4 000098		LO	SF	I*4 00C09C		NR	SF		I*4 0000A0
III	SF		I*4 0000A4	ISS	SF	I*4 0000A8		IHOW	F	XR	I*4 00C000	NNM1	SF		I*4 0000AC
ELIMB			R*4 0000B0												

LABEL	ADDR	LABEL	ADDR	LABEL	ADDR	LABEL	ADDR	PAGE
1	000110	55	000158	60	0001FA	70	000282	002
75	000296							

OPTIONS IN EFFECT NAME= MAIN,OPT=00,LINECNT=50,SIZE=0000K,
 OPTIONS IN EFFECT SOURCE=EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,NO,NOXREF
 STATISTICS SOURCE STATEMENTS = 19 ,PROGRAM SIZE = 844
 STATISTICS NO DIAGNOSTICS GENERATED
 ***** END OF COMPILATION ***** 131K BYTES OF CORE NOT USED

LEVEL 20.1 (AUG 71)

OS/360 FORTRAN H

DATE 72.237/08.19.31

COMPILER OPTIONS - NAME= MAIN,OPT=00,LINECNT=60,SIZE=0000K,
 SOURCE=EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,NO,NOXREF

ISN 0002	SUBROUTINE BACSUB(N,IU,JU,IHOW,U,B,X)	291
	C	292
	C THIS ROUTINE PERFORMS THE BACK SUBSTITUTION PROCESS. THE	293
	C SOLUTION TO THE SET OF EQUATIONS IS RETURNED IN X.	294
ISN 0003	IMPLICIT INTEGER*2(I-N)	295
ISN 0004	INTEGER*4 IHOW,II,III,K1,K2,J	295
ISN 0005	DIMENSION IU(1),JU(1),IHOW(1),U(1),B(1),X(1)	296
ISN 0006	DO 50 I=1,N	297
ISN 0007	M=N-I+1	298
ISN 0008	II=N+M	299
ISN 0009	III=IHOW(III)	300
ISN 0010	NR=IHOW(III)	301
ISN 0011	X1=IU(NR)+IHOW(III+1)	302
ISN 0012	X2=(U(NR+1)-1	303
ISN 0013	SUM=0.	304
ISN 0014	IF(X2-K1) 90,30,80	305
ISN 0015	80 DO 85 J=K1,K2	306
ISN 0016	L=JU(J)	307
ISN 0017	85 SUM=SUM+X(L)*U(J)	308
ISN 0018	90 X(M)=(B(NR)-SUM)/U(K1-1)	309
ISN 0019	RETURN	310
ISN 0020	END	311

/ BACSUB / SIZE OF PROGRAM 00030E HEXADECIMAL BYTES PAGE 002

NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.
B	F	XR	R*4 0000C0	I	SF	I*2 000080		J	SF	I*4 00008C		L	SF	I*2 000082	
M	SF	I*2 0000E4		N	F	I*2 000086		U	F	XR	R*4 000000	X	SF	XR	R*4 000000
II	SF	I*4 000050		IU	F	XR	I*2 000000	JU	F	XR	I*2 000000	K	SF	I*4 000094	
K2	SF	I*4 000058		NR	SF	I*2 000088		III	SF	I*4 00009C		SUM	SF	R*4 0000A0	
IHOW	F	XR	I*4 000000	BACSUB		R*4 0000A4									

LABEL	ADDR	LABEL	ADDR	LABEL	ADDR	LABEL	ADDR	PAGE
1	000100	80	000198	85	0001B4	90	0001FA	003

OPTIONS IN EFFECT NAME= MAIN,OPT=00,LINECNT=60,SIZE=0000K,
 OPTIONS IN EFFECT SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
 STATISTICS SOURCE STATEMENTS = 19 ,PROGRAM SIZE = 782
 STATISTICS NO DIAGNOSTICS GENERATED
 ***** END OF COMPILATION ***** 131K BYTES OF CORE NOT USED

LEVEL 20.1 (AUG 71)

OS/360 FORTRAN H

DATE 72.237/08.19.32

COMPILER OPTIONS - NAME= MAIN,OPT=00,LINECNT=60,SIZE=0000K,
 SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
 1SN 0002 SUBROUTINE DENSITY(KV,DSITYA,DSITYB,QNA,QNB,ALOWQA,ALOWQB,LOWIA,LOW
 1IB,ENLOWA,DNLOWB)
 1SN 0003 IMPLICIT INTEGER*2(I-N)
 1SN 0004 INTEGER*4 LT1,LT3,NFN,NTN,NODE,IDENT,KEYJUN,MAIN
 1SN 0005 COMMON LT1,LT3, NFN(600),NTN(600),NODE(600),A(2000), CO(600)
 1,DELV(600),F(600),FF(600),P(600),Q(600),QN(600),TOL,X1(600),X2(600)
 2,X3(600),X4(600),IDENT(600),CONP(600),MNP(600),IA(600),INDF(600),
 3IPN(7000),IPT(600),IPTPN(300),IRIM(600),IQIM(600),IUN(600),IX(180
 40),IXR(5000),JA(3000),NCEL(600),NFUN(600),NPUN(600),NN,NNCE,NN1,NU
 5F,NUP,NUNCE,INDP(600),INDR(600),IPIM(600),NCLL(600)

```

ISN 0006      KV=KV+1
ISN 0007      IF(KV .GE. 2) GO TO 25
ISN 0009      READ(LT1, :00) MAIN, TOTSQF, HDSQFT
ISN 0010      100 FORMAT(A4, I4, F5.0, I4, F5.1)
ISN 0011      IF(TOTSQF .EQ. 0.0) GO TO 98
ISN 0013      ALLOWQA=1000.0
ISN 0014      LOWIA=1
ISN 0015      DO 10 I=1, NNCE
ISN 0016      IF(MAIN .EQ. IDENT(I)) IMAIN=I
ISN 0018      IF(NCE1(I) .NE. 2) GO TO 10
ISN 0020      KD1=I*(3*I)
ISN 0021      IF(ALOWQA .LE. ABS(QN(KD1))) GO TO 10
ISN 0023      ALLOWQA=ABS(QN(KD1))
ISN 0024      LOWIA=I
ISN 0025      10 CONTINUE
ISN 0026      KD=IX(3*IMAIN)
ISN 0027      QNA=ABS(QN(KD))
ISN 0028      RETURN
ISN 0029      25 LOWIB=1
ISN 0030      ALLOWQB=1000.0
ISN 0031      IF(TOTSQF .EQ. 0.0) GO TO 98
ISN 0033      DO 40 I=1, NNCE
ISN 0034      IF(NCE1(I) .NE. 2) GO TO 40
ISN 0036      KD2=IX(3*I)
ISN 0037      IF(ALOWQB .LE. ABS(QN(KD2))) GO TO 40
ISN 0039      ALLOWQB=ABS(QN(KD2))
ISN 0040      LOWIB=I
ISN 0041      40 CONTINUE
ISN 0042      QNB=ABS(QN(KD))
ISN 0043      DSITYA=QNA/TOTSQF
ISN 0044      DSITYB=QNB/TOTSQF
ISN 0045      DNLOWB=ALLOWQA/HDSQFT
ISN 0046      DNLOWB=ALLOWQB/HDSQFT
ISN 0047      RETURN
ISN 0048      98 QNA=0.0
ISN 0049      99 RETURN
ISN 0050      END

```

/ DENSITY /

SIZE OF PROGRAM 0004C0 HEXADECEMAL BYTES PAGE 002

NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.
A	C	R*4	N.R.	F	C	R*4	N.R.	I SF	I*2	00009C	N.P.	P	C	R*4	N.P.
Q	C	R*4	N.P.	CO	C	R*4	N.R.	FF	C	R*4	N.P.	IA	C	I*2	N.R.
IX	F	C	I*2 011EC4	JA	C	I*2	N.R.	KD SFA	I*2	00009E	N.P.	KV SF	I*2	0000A0	N.R.
NN	C	I*2	N.R.	QN	FA	C	R*4 008343	XJ	C	R*4	N.P.	Z2	C	R*4	N.R.
X3	C	R*4	N.R.	X4	C	R*4	N.R.	IPN	C	I*2	N.R.	IPT	C	I*2	N.R.
IUN	C	I*2	N.R.	IXR	C	I*2	N.R.	KD1 SFA	I*2	0000A2	N.P.	KD2 SFA	I*2	0000A4	N.R.
LT1	F	C	I*4 000000	LT3	C	I*4	N.R.	MMP	C	I*2	N.P.	NFN	C	I*4	N.P.
NN1	C	I*2	N.R.	MTN	C	I*4	N.R.	MUF	C	I*2	N.R.	MUP	C	I*2	N.R.

QNA SF	R*4	0000AC	CNB SF	R*4	0000B0	TOL	C	R*4	N.F.	CONP	C	R*4	N.R.		
DELV	C	R*4	N.R.	INDF	C	I*2	N.R.	INOP	C	I*2	N.F.	INDR	C	I*2	N.R.
IPIM	C	I*2	N.R.	IQIM	C	I*2	N.R.	IRIM	C	I*2	N.F.	MAIN S	I*4	0000B4	
INCEL	C	I*2	016B54	NCLL	C	I*2	N.R.	NFUN	C	I*2	N.F.	NNCE F	C	I*2	017966
INODE	C	I*4	N.R.	NPUN	C	I*2	N.R.	IDENT	C	I*4	00B2EC	IMAIN SF	I*2	0000A6	
LOWIA S	I*2	0000A8	LOWIB S	I*2	0000AA	NUNCE	C	I*2	N.F.	ALLOWA SF	R*4	0000B8			
ALLOWB SF	R*4	00009C	DNLOWA S	R*4	0000C0	DNLOWB S	R*4	0000C4	DNSITY	F	R*4	0000C8			
DSITYA S	R*4	0000CC	DSITYB S	R*4	0000D0	HDSQFT SF	R*4	0000E4	IRCOM=	F	XF	I*4	000000		
IPTIPN	C	I*2	N.R.	KEYJUN	I*4	N.R.	TOTSQF SF	R*4	0000F8						

***** COMMON INFORMATION *****

NAME OF COMMON BLOCK *			* SIZE OF BLOCK			018C30 HEXADECIMAL BYTES					
VAR. NAME	TYPE	REL. ADDR.	VAR. NAME	TYPE	REL. ADDR.	VAR. NAME	TYPE	REL. ADDR.	VAR. NAME	TYPE	REL. ADDR.
LT1	I*4	000030	LT3	I*4	N.R.	NFN	I*4	N.R.	NTN	I*4	N.P.
NODE	I*4	N.R.	A	R*4	N.R.	CO	R*4	N.R.	DELV	R*4	N.R.
F	R*4	N.R.	FF	R*4	N.R.	P	R*4	N.R.	Q	R*4	N.R.
QN	R*4	008348	TOL	R*4	N.R.	X1	R*4	N.R.	X2	R*4	N.R.
X3	R*4	N.R.	X4	R*4	N.R.	IDENT	I*4	00B2EC	CONP	R*4	N.R.
MMP	I*2	N.R.	IA	I*2	N.R.	INDF	I*2	N.R.	IPN	I*2	N.R.
IPT	I*2	N.R.	IPTIPN	I*2	N.R.	IRIM	I*2	N.R.	IQIM	I*2	N.R.
IUN	I*2	N.R.	IX	I*2	011FC4	IXP	I*2	N.R.	JA	I*2	N.R.
NCEL	I*2	016B54	NFUN	I*2	N.R.	NPUN	I*2	N.R.	NN	I*2	N.P.
NNCE	I*2	017956	NNI	I*2	N.R.	NUF	I*2	N.R.	NUP	I*2	N.R.
NUNCE	I*2	N.R.	INOP	I*2	N.R.	INDR	I*2	N.R.	IPIM	I*2	N.P.
NCLL	I*2	N.R.									

LABEL	ADDR	LABEL	ADDR	LABEL	ADDR	LABEL	ADDR	PAGE	003
10	000266	25	0002B6	40	000346	98	0003AF		
99	0003B6 NR								

OPTIONS IN EFFECT NAME= MAIN,OPT=00,LINECNT=50,SIZE=0000K,

OPTIONS IN EFFECT SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXPEF

STATISTICS SOURCE STATEMENTS = 49 ,PROGRAM SIZE = 1216

STATISTICS NO DIAGNOSTICS GENERATED

***** END OF COMPILE *****

127K BYTES OF CORE NOT USED

LEVEL 20.1 (AUG 71)

OS/360 FORTRAN H

DATE 72.237/08.19.32

```

      COMPILER OPTIONS - NAME= MAIN,OPT=00,LINECNT=60,SIZE=0000K,
                        SOURCE,EBCDIC,NGLIST,NODECK,LOAD,MAP,NOEDIT,IO,NOXPEF
      ISN 0002      SUBROUTINE RERJN(INEW,IOLD,*,*)
      ISN 0003      IMPLICIT INTEGER*2(I-N)
      ISN 0004      INTEGER*4 LT1,LT3,NFN,NTN,NODE,IDENT,KEYJUN,TOLNPR,TOLNSC
      ISN 0005      DATA IQES/'*'/,NIOES/' '/
      ISN 0006      COMMON LT1,LT3, NFN(600),NTN(600),NODE(600),A(3000), CO(600)
      1,DELV(600),F(600),FF(600),P(600),Q(600),QN(600),TOL,X1(600),X2(600)
      2),X3(600),X4(600),IDENT(600),CONP(600),MMP(600),IA(600),INDF(600),
      3IPN(7000),IPT(500),IPTIPN(300),IRIM(600),IQIM(600),IUN(600),IX(180
      40),IXR(5000),JA(3000),NCEL(600),NFUN(600),NFUN(600),NN,NNCE,NN1,NU
      5F,NUN,NUNCE,INDP(600),INDR(600),IPIM(600),NCLL(600)
      ABSOPR=20.0
      ISN 0007      READ(LT1,100) TOLNPR,PRESPR,TOLNSC
      ISN 0008      100 FORMAT(A4,1X,F4.1,1X,A4)
      ISN 0009      IF(PRESPR.EQ. 0.0) GO TO 99
      ISN 0010      DO 10 I=1,NNCE
      ISN 0011      IF(TOLNPR.EQ. IDENT(I))INEW=1
      ISN 0012      IF(TOLNSC.EQ. IDENT(I))IOLD=1
      ISN 0013      10 CONTINUE
      ISN 0014      KD=IX(3*INEW)
      ISN 0015      IF(ABS(P(KD)-PRESPR).GT. ABSOPR) GO TO 98
      ISN 0016      MMP(INEW)=NIOES
      ISN 0017      MMP(IOLD)=IQES
      ISN 0018      CONP(INEW)=PRESPR
      ISN 0019      CONP(IOLD)=0.0
      ISN 0020      RETURN 1
      ISN 0021      98 WRITE(LT3,105) NTN(INEW),ABSOPR
      ISN 0022      105 FORMAT(' ',////,' THE ABSOLUTE DIFFERENCE BETWEEN THE PRESSURE NOW
      1 REQUIRED AND THE PRESSURE ASKED AT NODE ',A4,1X,' HAS EXCEEDED',F6
      2.1,' PSIG',////)
      ISN 0023      99 RETURN 2
      ISN 0024      END

```

/ REPUN / SIZE OF PROGRAM 000370 HEXADECIMAL BYTES PAGE 002

NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.			
A	C	R*4	N.R.	F	C	R*4	N.R.	I SF	I*2	000124		P	A	C	R*4	007088		
Q	C	R*4	N.R.	CO	C	R*4	N.R.	FF	C	P*4	N.R.	IA	C	I*2	N.R.			
IX	F	C	I*2	JA	C	I*2	N.R.	KD S A	I*2	000126		NN	C	I*2	N.R.			
QN	C	R*4	N.R.	X1	C	R*4	N.R.	X?	C	R*4	N.R.	X3	C	R*4	N.R.			
X4	C	R*4	N.R.	IPN	C	I*2	N.R.	IPT	C	I*2	N.R.	TLN	C	I*2	N.R.			
IXR	C	I*2	N.R.	LT1	F	C	I*4	000000	LT3	F	C	I*4	000004	MMP S	C	I*2	00045C	
NFN	C	I*4	N.R.	NN1	C	I*2	N.R.	NTN	F	C	I*4	000968	NUN	C	I*2	N.R.		
NUP	C	I*2	N.R.	TOL	C	R*4	N.R.	CONP S	C	R*4	00088C	DELV	C	R*4	N.R.			
INDF	C	I*2	N.R.	INDP	C	I*2	N.R.	INDR	C	I*2	N.R.	INEW S	I*2		000128			
IOLD SF	I*2		000124	IPIM	C	I*2	N.R.	IQES	F	I*2	00012C	IPIM	C	I*2	N.R.			
IRIM	C	I*2	N.R.	NCEL	C	I*2	N.R.	NCLL	C	I*2	N.R.	NFLN	C	I*2	N.R.			

NNCE	F	C	I*2	017966	NODE	C	I*4	N.R.	NPUN	C	I*2	N.R.	IDENT	C	I*4	00B22C
NIJES	F		I*2	00012E	NUNCE	C	I*2	N.R.	REPUN	R*4	000130		ABSDPR	SF	P*4	000134
IRCDM=	F	XF	R*4	00000D	IPTIPN	C	I*2	N.R.	KEYJUN	I*4	N.R.		PRESPR	SFA	R*4	000138
TOLNPR	S		I*4	00013C	TOLNSC	S	I*4	000140								

***** COMMON INFORMATION *****

NAME OF COMMON BLOCK *				* SIZE OF BLOCK				018C30 HEXADECIMAL BYTES							
VAR. NAME	TYPE	REL. ADDR.		VAR. NAME	TYPE	REL. ADDR.		VAR. NAME	TYPE	REL. ADDR.		VAR. NAME	TYPE	REL. ADDR.	
LT1	I*4	000000		LT3	I*4	000004		NFN	I*4	N.R.		NTN	I*4	000968	
NODE	I*4	N.R.		A	R*4	N.R.		C0	R*4	N.R.		DELV	R*4	N.R.	
F	R*4	N.R.		FF	R*4	N.R.		P	P*4	007088		Q	R*4	N.R.	
QN	R*4	N.R.		TDL	R*4	N.R.		X1	R*4	N.R.		X2	P*4	N.R.	
X3	R*4	N.R.		X4	R*4	N.R.		IDENT	I*4	00B22C		COMP	R*4	00BB8C	
MMP	I*2	00C4EC		IA	I*2	N.R.		INDF	I*2	N.R.		JPN	I*2	N.R.	
IPT	I*2	N.R.		IPTIPN	I*2	N.R.		IRIM	I*2	N.R.		IQIM	I*2	N.R.	
IUN	I*2	N.R.		IX	I*2	017EC4		IXP	I*2	N.R.		JA	I*2	N.R.	
NCEL	I*2	N.R.		NFUN	I*2	N.R.		NPUN	I*2	N.R.		NN	I*2	N.R.	
NNCE	I*2	017966		NN1	I*2	N.R.		NUF	I*2	N.R.		NUP	I*2	N.R.	
NUNCE	I*2	N.P.		INDP	I*2	N.R.		INDR	I*2	N.R.		IPIM	I*2	N.R.	
NCLL	I*2	N.R.													

LABEL	ADDR	LABEL	ADDR	LABEL	ADDR	LABEL	ADDR	PAGE 003
10	00024C	98	0002EC	99	00031C			

OPTIONS IN EFFECT NAME= MAIN,OPT=00,LINECNT=60,SIZE=0000K,
 OPTIONS IN EFFECT SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
 STATISTICS SOURCE STATEMENTS = 23 ,PROGRAM SIZE = 880
 STATISTICS NO DIAGNOSTICS GENERATED
 ***** END OF COMPILATION *****

127K BYTES OF CORE NOT USED

LEVEL 20.1 (AUG 71)

OS/360 FCRTAN H

DATE 72.237/08.19.34

```

COMPILER OPTIONS - NAME= MAIN,CPT=00,LINECNT=60,SIZE=C000K,
SOURCE,ERCDIC,NCLIST,NODECH,LJAD,MAP,NOEDIT,ID,NOXREF

ISN 0002      SUBROUTINE FRCTN
ISN 0003      IMPLICIT INTEGER*2(I-N)
ISN 0004      INTEGER*4 LT1,LT2,NFN,NTN,NODE,IDENT,<EYJUN
ISN 0005      DIMENSION FRCLDS(3000),LNNUMB(3000),RECDIA(600)
ISN 0006      COMMON LT1,LT2,NFN(600),NTN(600),NODE(600),A(3000),CO(600)
1,DELV(600),F(600),FF(600),P(600),Q(600),QN(600),TOL,X1(600),X2(600)
2),X3(600),X4(600),IDENT(600),CONP(600),MMP(600),IA(600),INDF(600),
3IPN(7000),IPT(600),IPTIPN(300),IRIN(600),IQIM(600),IUN(600),IX(180
40),IXR(5000),JA(3000),NCEL(600),NFUN(600),NPMN(600),NN,NNCE,NNI,NU
5F,NUP,NUNCF,INBP(600),INDR(600),IPIM(600),NCLL(600)
EQUIVALENCE (A(1),FRCLDS(1)),(X3(1),RECDIA(1)),(JA(1),LNNUMB(1))
ISN 0007      AMAXFR=3.0
ISN 0008      ICOUNT=0
ISN 0009      DO 40 I=1,NNCE
ISN 0010      KU=IX(3*I-1)
ISN 0011      KD=IX(3*I)
ISN 0012      IF(NCEL(I).EQ. 2) GO TO 40
ISN 0013      DP=P(KU)-P(KD)-DELV(I)
ISN 0015      IF(ABS(DP).LT. AMAXFR) GO TO 40
ISN 0016      S=1.0
ISN 0017      IF(DP .LT. 0.0) S=-1.0
ISN 0018      FF(I)=ABS((S*G(I)**.38)/((CO(I)**.38)*(AMAXFR **.705)))
ISN 0019      IF(FF(I) .GE. 0.924 .AND. FF(I) .LT. 1.149) GO TO 1
ISN 0020      IF(FF(I) .GE. 1.149 .AND. FF(I) .LT. 1.480) GO TO 2
ISN 0021      IF(FF(I) .GE. 1.480 .AND. FF(I) .LT. 1.710) GO TO 3
ISN 0022      IF(FF(I) .GE. 1.710 .AND. FF(I) .LT. 2.167) GO TO 4
ISN 0023      IF(FF(I) .GE. 2.167 .AND. FF(I) .LT. 2.569) GO TO 5
ISN 0024      IF(FF(I) .GE. 2.569 .AND. FF(I) .LT. 3.168) GO TO 6
ISN 0025      IF(FF(I) .GE. 3.168 .AND. FF(I) .LT. 3.648) GO TO 7
ISN 0026      IF(FF(I) .GE. 3.648 .AND. FF(I) .LT. 4.126) GO TO 8
ISN 0027      IF(FF(I) .GE. 4.126 .AND. FF(I) .LT. 5.147) GO TO 9
ISN 0028      IF(FF(I) .GE. 5.147 .AND. FF(I) .LT. 6.165) GO TO 10
ISN 0029      IF(FF(I) .GE. 6.165 .AND. FF(I) .LT. 8.081) GO TO 11
ISN 0030      IF(FF(I) .GE. 8.081 .AND. FF(I) .LT. 10.120) GO TO 12
ISN 0031      IF(FF(I) .GE. 10.120 .AND. FF(I) .LT. 12.100) GO TO 13
ISN 0032      IF(FF(I) .GE. 12.100 .AND. FF(I) .LT. 13.350) GO TO 14
ISN 0033      IF(FF(I) .GE. 13.350 .AND. FF(I) .LT. 15.350) GO TO 15
ISN 0034      IF(FF(I) .GE. 15.350 .AND. FF(I) .LT. 17.280) GO TO 16
ISN 0035      IF(FF(I) .GE. 17.280 .AND. FF(I) .LT. 19.280) GO TO 17
ISN 0036      IF(FF(I) .GE. 19.280 .AND. FF(I) .LT. 24.100) GO TO 18
ISN 0037      IF(FF(I) .GE. 24.100 .AND. FF(I) .LT. 30.100) GO TO 19
ISN 0038      IF(FF(I) .GE. 30.100 .AND. FF(I) .LT. 36.100) GO TO 20
ISN 0039      GO TO 40
ISN 0040      1 ICOUNT=ICOUNT+1
ISN 0041      RECDIA(ICOUNT)= 1.049
ISN 0042      LNNUMB(ICOUNT)=1
ISN 0043      GO TO 40
ISN 0044      2 ICOUNT=ICOUNT+1
ISN 0045      RECDIA(ICOUNT)= 1.3E0
ISN 0046      LNNUMB(ICOUNT)=1
ISN 0047      GO TO 40
ISN 0048
ISN 0049
ISN 0050
ISN 0051
ISN 0052
ISN 0053
ISN 0054
ISN 0055
ISN 0056
ISN 0057
ISN 0058
ISN 0059
ISN 0060
ISN 0061
ISN 0062
ISN 0063
ISN 0064
ISN 0065
ISN 0066
ISN 0067
ISN 0068
ISN 0069
ISN 0070

```

2

ISN 0071	3	ICOUNT=ICOUNT+1
ISN 0072		REC DIA(ICOUNT)= 1.610
ISN 0073		LNNUMR(ICOUNT)=I
ISN 0074		GO TO 40
ISN 0075	4	ICOUNT=ICOUNT+1
ISN 0076		REC DIA(ICOUNT)= 2.067
ISN 0077		LNNUMR(ICOUNT)=I
ISN 0078		GO TO 40
ISN 0079	5	ICOUNT=ICOUNT+1
ISN 0080		REC DIA(ICOUNT)= 2.469
ISN 0081		LNNUMB(ICOUNT)=I
ISN 0082		GO TO 40
ISN 0083	6	ICOUNT=ICOUNT+1
ISN 0084		REC DIA(ICOUNT)= 3.068
ISN 0085		LNNUMB(ICOUNT)=I
ISN 0086		GO TO 40
ISN 0087	7	ICOUNT=ICOUNT+1
ISN 0088		REC DIA(ICOUNT)= 3.548
ISN 0089		LNNUMB(ICOUNT)=I
ISN 0090		GO TO 40
ISN 0091	8	ICOUNT=ICOUNT+1
ISN 0092		REC DIA(ICOUNT)= 4.026
ISN 0093		LNNUMB(ICOUNT)=I
ISN 0094		GO TO 40
ISN 0095	9	ICOUNT=ICOUNT+1
ISN 0096		REC DIA(ICOUNT)= 5.047
ISN 0097		LNNUMB(ICOUNT)=I
ISN 0098		GO TO 40
ISN 0099	10	ICOUNT=ICOUNT+1
ISN 0100		REC DIA(ICOUNT)= 6.065
ISN 0101		LNNUMB(ICOUNT)=I
ISN 0102		GO TO 40
ISN 0103	11	ICOUNT=ICOUNT+1
ISN 0104		REC DIA(ICOUNT)= 7.981
ISN 0105		LNNUMB(ICOUNT)=I
ISN 0106		GO TO 40
ISN 0107	12	ICOUNT=ICOUNT+1
ISN 0108		REC DIA(ICOUNT)=10.020
ISN 0109		LNNUMB(ICOUNT)=I
ISN 0110		GO TO 40
ISN 0111	13	ICOUNT=ICOUNT+1
ISN 0112		REC DIA(ICOUNT)=12.000
ISN 0113		LNNUMR(ICOUNT)=I
ISN 0114		GO TO 40
ISN 0115	14	ICOUNT=ICOUNT+1
ISN 0116		REC DIA(ICOUNT)=13.250
ISN 0117		LNNUMB(ICOUNT)=I
ISN 0118		GO TO 40
ISN 0119	15	ICOUNT=ICOUNT+1
ISN 0120		REC DIA(ICOUNT)=15.250
ISN 0121		LNNUMB(ICOUNT)=I
ISN 0122		GO TO 40
ISN 0123	16	ICOUNT=ICOUNT+1
ISN 0124		REC DIA(ICOUNT)=17.180
ISN 0125		LNNUMB(ICOUNT)=I

```

ISN 0126      GO TO 40
ISN 0127      17 ICOUNT=ICOUNT+1
ISN 0128      RECDIA(ICOUNT)=19.180
ISN 0129      LNNUMB(ICOUNT)=1
ISN 0130      GO TO 40
ISN 0131      18 ICOUNT=ICOUNT+1
ISN 0132      RECDIA(ICOUNT)=24.060
ISN 0133      LNNUMB(ICOUNT)=1
ISN 0134      GO TO 40
ISN 0135      19 ICOUNT=ICOUNT+1
ISN 0136      RECDIA(ICOUNT)=30.060
ISN 0137      LNNUMB(ICOUNT)=1
ISN 0138      GO TO 40
ISN 0139      20 ICOUNT=ICOUNT+1
ISN 0140      RECDIA(ICOUNT)=36.060
ISN 0141      LNNUMB(ICOUNT)=1
ISN 0142      40 CONTINUE
ISN 0143      45 IF(ICOUNT.EQ. 0) GO TO 99
ISN 0144      WRITE(LT3,100)
ISN 0145      100 FORMAT(' ',8X,'LINE',12X,'RECOMMENDED',10X,'FRICTION',/, ' ',26X,'DIAMETER',10X,'LOSS (PSIG)',//)
ISN 0146      DO 80 I=1,ICOUNT
ISN 0147      DP=((ABS(Q(LNNUMB(I))))**1.852)/((RECDIA(I)**4.870)*(CO(LNNUMB(I))
ISN 0148      1**1.852))
ISN 0149      WRITE(LT3,101) IDENT(LNNUMB(I)),RECDIA(I),DP
ISN 0150      101 FORMAT(' ',8X,15.15X,F6.3,14X,F6.3,/,)
ISN 0151      80 CONTINUE
ISN 0152      99 RETURN
ISN 0153      END

```

/ FRICTN / SIZE OF PROGRAM 000EAA HEXADECEIMAL BYTES PAGE 004

NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.		
A	CE	R*4	001C28	F	C	R*4	N.F.	I SFA	I*2	000108	P	F	C	R*4	007088		
Q	FA	C	R*4	S SFA	R*4	0001F0	CO	FA	C	R*4	004B08	BP	SFA	R*4	0001E4		
FF S	C	R*4	006728	IA	C	I*2	N.F.	IX	F	C	I*2	011EC4	JA	CE	I*2	0153F4	
KD SF	I*2	0001DA		KU SF	I*2	0001DC		NN	C	I*2	N.R.	QN	C	R*4	N.R.		
X1	C	R*4	N.R.	X2	C	R*4	N.F.	X2	CE	R*4	009F6C	XA	C	R*4	N.R.		
IPN	C	I*2	N.R.	IPT	C	I*2	N.R.	IUN	C	I*2	N.R.	IXE	C	I*2	N.R.		
LT1	C	I*4	N.R.	LT3	F	C	I*4	0000C4	MMP	C	I*2	N.R.	NEH	C	I*4	N.P.	
NN1	C	I*2	N.R.	NTN	C	I*4	N.F.	NUF	C	I*2	N.R.	NJP	C	I*2	N.R.		
TOL	C	R*4	N.R.	CNP	C	R*4	N.R.	DELV	F	C	R*4	005468	INDF	C	I*2	N.P.	
INDP	C	I*2	N.R.	INDR	C	I*2	N.R.	IPIM	C	I*2	N.P.	IDEN	C	I*2	N.R.		
IRIM	C	I*2	N.R.	NCEL	C	I*2	016854	NCLL	C	I*2	N.R.	NEUN	C	I*2	N.R.		
NNCE	F	C	I*2	NODE	C	I*4	N.F.	NPUN	C	I*2	N.R.	IDENT	F	C	I*4	00B22C	
NUNCE	C	I*2	N.R.	FRXPR=	XF	R*4	0000G0	AMAXFR	SFA	R*4	0001E8	FRCLDS	CE	R*4	001C28		
FRICTN	R*4	0001EC		IBCDM=	F	XF	I*4	0000G0	ICOUNT	SF	I*2	0001DE	IPTIPN	C	I*2	N.R.	
KEYJUN	I*4	N.P.		LNNUMB	SFA	CE	I*2	0153F4	RECDIA	SF	CE	R*4	009F6C				

***** COMMON INFORMATION *****

NAME OF COMMON BLOCK * * SIZE OF BLOCK 018C30 HEXADECIMAL BYTES

VAR. NAME	TYPE	REL. ADDR.	VAR. NAME	TYPE	REL. ADDR.	VAR. NAME	TYPE	REL. ADDR.	VAR. NAME	TYPE	REL. ADDR.
LT1	I*4	N.F.	LT3	I*4	000004	NFN	I*4	N.R.	NTN	I*4	N.R.
NODE	I*4	N.R.	A	R*4	001C28	CO	R*4	004B08	DELV	P*4	005468
F	R*4	N.R.	FF	R*4	006728	P	R*4	007088	Q	P*4	0079E8
QN	R*4	N.R.	TOL	R*4	N.R.	X1	R*4	N.R.	X2	P*4	N.R.
X3	R*4	009F6C	X4	R*4	N.R.	IDENT	I*4	00B22C	CONP	R*4	N.R.
MMP	I*2	N.R.	IA	I*2	N.R.	INDF	I*2	N.R.	IPN	I*2	N.R.
IPT	I*2	N.R.	IPTIPN	I*2	N.R.	IRIM	I*2	N.R.	IQIM	I*2	N.R.
IUN	I*2	N.R.	IX	I*2	011EC4	IXR	I*2	N.R.	JA	I*2	0153E4
NEEL	I*2	016A54	NFUN	I*2	N.R.	NPUN	I*2	N.R.	NN	I*2	N.R.
NVCE	I*2	017966	NN1	I*2	N.R.	NUP	I*2	N.R.	NUP	I*2	N.R.
NUNCE	I*2	N.R.	INDP	I*2	N.R.	INDR	I*2	N.R.	IPIM	I*2	N.R.
NLL	I*2	N.P.									

EQUIVALENCED VARIABLES WITHIN THIS COMMON BLOCK

VARIABLE	OFFSET	VARIABLE	OFFSET	VARIABLE	OFFSET
FRCLOS	001C28	RECDIA	009F6C	LNNUM8	0153E4

LABEL	ADDR	LABEL	ADDR	LABEL	ADDR	LABEL	ADDR	PAGE
1	0008D6	2	00090E	3	000946	4	00097E	005
5	0009B6	6	0009EE	7	000A26	8	000A5E	
9	000A96	10	000ACE	11	000B06	12	000B3E	
13	000B76	14	000BAE	15	000BE6	16	000C1E	
17	000C56	18	000CBE	19	000CC6	20	000CFE	
40	000D30	45	000D4A	80	000E6E	99	000F84	

OPTIONS IN EFFECT NAME= MAIN,OPT=00,LINECNT=50,SIZE=0000K,

OPTIONS IN EFFECT SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,LD,NOXREF

STATISTICS SOURCE STATEMENTS = 152 ,PROGRAM SIZE = 3754

STATISTICS NO DIAGNOSTICS GENERATED

***** END OF COMPILATION *****

107K BYTES OF CORE NOT USED

STATISTICS NO DIAGNOSTICS THIS STEP

NAME	ORIGIN	LENGTH
COREDUMP*	65550	6A4
IHCTRCH *	65BF8	2
DOUTRL *	65C00	398
DOSTUF *	65F98	408
DOERMD *	663A0	190
DOSETD *	66530	424
DOCCDS *	66958	20
DODUMP *	66988	208
DOGADS *	66A98	294
IHCFFEXIT*	66F30	1C
\$BLANKCOM	66E50	18C30

NAME	LOCATION	NAME	LOCATION	NAME	LOCATION	NAME	LOCATION
CDUMP	65550						
IHCUTRL	65C00						
DOSAVE	65F98	DOORSTR	65FAA	SETCHK	65FE0	DPSEQT8L	66030
UNITTRLE	662C0	DOOSYSN	66368	SFTFAULT	66394		
SETFAU	66530						
IOKLOC	66C30	DOOTIME	66D02				
EXIT	66E30						

ENTRY ADDRESS 00
TOTAL LENGTH 7FA80

***NAMEX DOES NOT EXIST BUT HAS BEEN ADDED TO DATA SET

IHC002I STOP 0

LIMITATIONS

The size of the network must be such that none of the following units are exceeded:

Number of Node-Connecting Elements	400
Number of Nodes	500

These dimensions could be increased by making changes in the program.

This program does not take into consideration pressure losses due to velocity in making its calculations. The assumption is that friction pressure drops would not be allowed to become large enough in designing the sprinkler system for the velocity pressure drops to become significant.

SAMPLE PROBLEM

The sample that was chosen represents a wet sprinkler system for a manufacturing building with no high racks. The criterion specifies that the system be capable of providing a density of 0.30 gpm of water per square foot over the most remote 4000 square feet of floor space.

As will be noted from the flow diagram, design drawing, and data sheet (Figures 2 - 4), the system requires a total of 1145 gpm of water at 75.7 psig at the yard main. The computer recommends changing the pipe size on two lines.

SYSTEM VARIABLES AND EXECUTION CONTROLS

&DIN
 TOL= 10.000000 ,MA= 50. 1. 50. 20. 2. 10. 3. 1. 0. 0
 &END

SYSTEM DESCRIPTION DATA FOR CPL CCL'S A-D61-7

NCE NUM	FROM NODE	TO NODE	NCE TYPE	NCE STATUS	PRESS. STATUS	FLOW (GPM)	FLOW STATUS	DIAM. (IN.)	LENGTH (FT)	C FACTO ²	ELEV. (FT)	CONSTANT PRESSURE
L0	2	1	P1			1200.00	*	0.0	10.00	120.	788.50	100.00
L1	1	2	P1		*	0.0		7.98	235.00	120.	794.50	0.0
L2	2	3	P1		*	0.0		6.07	37.00	120.	804.50	0.0
L3	3	4	P1		*	0.0		6.07	17.50	120.	816.50	0.0
L4	4	5	P1		*	0.0		6.07	69.00	120.	816.50	0.0
L5	5	6	P1		*	0.0		3.55	63.00	120.	816.50	0.0
L6	6	7	P1		*	0.0		3.55	22.00	120.	816.50	0.0
L7	7	8	P1		*	0.0		3.55	40.00	120.	816.50	0.0
L55	5	15	P1		*	0.0		3.55	215.00	120.	816.50	0.0
L8	8	9	P1		*	0.0		3.55	10.00	120.	816.50	0.0
L9	9	S1	P1		*	0.0		2.07	22.00	120.	819.50	0.0
L10	S1	S2	P1		*	0.0		2.07	10.00	120.	819.50	0.0
L11	S2	S3	P1		*	0.0		1.61	10.00	120.	819.50	0.0
L12	S3	S4	P1		*	0.0		1.61	10.00	120.	819.50	0.0
L13	9	10	P1		*	0.0		3.55	10.00	120.	816.50	0.0
L14	10	S5	P1		*	0.0		2.07	18.00	120.	819.50	0.0
L15	S5	S6	P1		*	0.0		2.07	10.00	120.	819.50	0.0
L16	S6	S7	P1		*	0.0		1.61	10.00	120.	819.50	0.0
L17	S7	S8	P1		*	0.0		1.61	10.00	120.	819.50	0.0
L18	S8	S9	P1		*	0.0		1.38	10.00	120.	819.50	0.0
L19	S9	S10	P1		*	0.0		1.05	10.00	120.	819.50	0.0
L20	10	11	P1		*	0.0		3.55	10.00	120.	816.50	0.0
L21	11	S11	P1		*	0.0		2.07	22.00	120.	819.50	0.0
L22	S11	S12	P1		*	0.0		2.07	10.00	120.	819.50	0.0
L23	S12	S13	P1		*	0.0		1.61	10.00	120.	819.50	0.0
L24	S13	S14	P1		*	0.0		1.61	10.00	120.	819.50	0.0
L25	S14	S15	P1		*	0.0		1.38	10.00	120.	819.50	0.0
L26	S15	S16	P1		*	0.0		1.05	10.00	120.	819.50	0.0
L27	11	12	P1		*	0.0		3.55	10.00	120.	816.50	0.0
L28	12	S17	P1		*	0.0		2.07	22.00	120.	819.50	0.0
L29	S17	S18	P1		*	0.0		2.07	10.00	120.	819.50	0.0
L30	S18	S19	P1		*	0.0		1.61	10.00	120.	819.50	0.0
L31	S19	S20	P1		*	0.0		1.61	10.00	120.	819.50	0.0
L32	S20	S21	P1		*	0.0		1.38	10.00	120.	819.50	0.0
L33	S21	S22	P1		*	0.0		1.05	10.00	120.	819.50	0.0
L34	12	13	P1		*	0.0		3.55	10.00	120.	816.50	0.0
L35	13	S23	P1		*	0.0		2.07	22.00	120.	819.50	0.0
L36	S23	S24	P1		*	0.0		2.07	10.00	120.	819.50	0.0
L37	S24	S25	P1		*	0.0		1.61	10.00	120.	819.50	0.0
L38	S25	S26	P1		*	0.0		1.61	10.00	120.	819.50	0.0
L39	S26	S27	P1		*	0.0		1.38	10.00	120.	819.50	0.0
L40	S27	S28	P1		*	0.0		1.05	10.00	120.	819.50	0.0
L41	14	13	P1		*	0.0		3.55	10.00	120.	816.50	0.0
L42	14	S29	P1		*	0.0		2.07	22.00	120.	819.50	0.0
L43	S29	S30	P1		*	0.0		2.07	10.00	120.	819.50	0.0
L44	S30	S31	P1		*	0.0		1.61	10.00	120.	819.50	0.0
L45	S31	S32	P1		*	0.0		1.61	10.00	120.	819.50	0.0
L46	S32	S33	P1		*	0.0		1.38	10.00	120.	819.50	0.0
L47	S33	S34	P1		*	0.0		1.05	10.00	120.	819.50	0.0
L48	15	14	P1		*	0.0		3.55	10.00	120.	816.50	0.0
L49	15	S35	P1		*	0.0		2.07	22.00	120.	819.50	0.0
L50	S35	S36	P1		*	0.0		2.07	10.00	120.	819.50	0.0
L51	S36	S37	P1		*	0.0		1.61	10.00	120.	819.50	0.0
L52	S37	S38	P1		*	0.0		1.61	10.00	120.	819.50	0.0
L53	S38	S39	P1		*	0.0		1.38	10.00	120.	819.50	0.0
L54	S39	S40	P1		*	0.0		1.05	10.00	120.	819.50	0.0
K1	S1	H1	S1			-30.00	*	5.56	0.0	0.	0.0	0.0
K2	S2	H2	S1			-30.00	*	5.56	0.0	0.	0.0	0.0
K3	S3	H3	S1			-30.00	*	5.56	0.0	0.	0.0	0.0
K4	S4	H4	S1			-30.00	*	5.56	0.0	0.	0.0	0.0
K5	S5	H5	S1			-30.00	*	5.56	0.0	0.	0.0	0.0
K6	S6	H6	S1			-30.00	*	5.56	0.0	0.	0.0	0.0
K7	S7	H7	S1			-30.00	*	5.56	0.0	0.	0.0	0.0
K8	S8	H8	S1			-30.00	*	5.56	0.0	0.	0.0	0.0
K9	S9	H9	S1			-30.00	*	5.56	0.0	0.	0.0	0.0
K10	S10	H10	S1			-30.00	*	5.56	0.0	0.	0.0	0.0
K11	S11	H11	S1			-30.00	*	5.56	0.0	0.	0.0	0.0
K12	S12	H12	S1			-30.00	*	5.56	0.0	0.	0.0	0.0
K13	S13	H13	S1			-30.00	*	5.56	0.0	0.	0.0	0.0
K14	S14	H14	S1			-30.00	*	5.56	0.0	0.	0.0	0.0
K15	S15	H15	S1			-30.00	*	5.56	0.0	0.	0.0	0.0
K16	S16	H16	S1			-30.00	*	5.56	0.0	0.	0.0	0.0
K17	S17	H17	S1			-30.00	*	5.56	0.0	0.	0.0	0.0
K18	S18	H18	S1			-30.00	*	5.56	0.0	0.	0.0	0.0
K19	S19	H19	S1			-30.00	*	5.56	0.0	0.	0.0	0.0
K20	S20	H20	S1			-30.00	*	5.56	0.0	0.	0.0	0.0
K21	S21	H21	S1			-30.00	*	5.56	0.0	0.	0.0	0.0
K22	S22	H22	S1			-30.00	*	5.56	0.0	0.	0.0	0.0
K23	S23	H23	S1			-30.00	*	5.56	0.0	0.	0.0	0.0

K24	S24	H24	S1	-30.00	*	5.56	0.0	0.	0.0	0.0
K25	S25	H25	S1	-30.00	*	5.56	0.0	0.	0.0	0.0
K26	S26	H26	S1	-30.00	*	5.56	0.0	0.	0.0	0.0
K27	S27	H27	S1	-30.00	*	5.56	0.0	0.	0.0	0.0
K28	S28	H28	S1	-30.00	*	5.56	0.0	0.	0.0	0.0
K29	S29	H29	S1	-30.00	*	5.56	0.0	0.	0.0	0.0
K30	S30	H30	S1	-30.00	*	5.56	0.0	0.	0.0	0.0
K31	S31	H31	S1	-30.00	*	5.56	0.0	0.	0.0	0.0
K32	S32	H32	S1	-30.00	*	5.56	0.0	0.	0.0	0.0
K33	S33	H33	S1	-30.00	*	5.56	0.0	0.	0.0	0.0
K34	S34	H34	S1	-30.00	*	5.56	0.0	0.	0.0	0.0
K35	S35	H35	S1	-30.00	*	5.56	0.0	0.	0.0	0.0
K36	S36	H36	S1	-30.00	*	5.56	0.0	0.	0.0	0.0
K37	S37	H37	S1	-30.00	*	5.56	0.0	0.	0.0	0.0
K38	S38	H38	S1	-30.00	*	5.56	0.0	0.	0.0	0.0
K39	S39	H39	S1	-30.00	*	5.56	0.0	0.	0.0	0.0
K40	S40	H40	S1	-30.00	*	5.56	0.0	0.	0.0	0.0

NODE CORRESPONDENCE TABLE AND LIST OF ADJACENT NODES

INTERNAL NODE ADJACENT NODES (BY NAME)
NODE NUMBER NAME

1	2	1	2	1
2	1	2	2	
3	3	4	2	
4	4	5	3	
5	5	6	15	4
6	6	7	2	
7	7	8	6	
8	8	9	7	
9	9	S1	10	8
10	S1	S2	H1	9
11	S2	S3	H2	S1
12	S3	S4	H3	S2
13	S4	S5	H4	9
14	S5	S6	H5	10
15	S6	S7	H6	S3
16	S7	S8	H7	S6
17	S8	S9	H8	S7
18	S9	S10	H9	S8
19	S10	S11	H10	10
20	S11	S12	H11	11
21	S12	S13	H12	S11
22	S13	S14	H13	S12
23	S14	S15	H14	S13
24	S15	S16	H15	S14
25	12	S17	11	13
26	S17	S18	H17	12
27	S18	S19	H18	S17
28	S19	S20	H19	S18
29	S20	S21	H20	S19
30	S21	S22	H21	S20
31	13	12	S23	14
32	S23	S24	H23	13
33	S24	S25	H24	S23
34	S25	S26	H25	S24
35	S26	S27	H26	S25
36	S27	S28	H27	S26
37	14	13	S29	15
38	S29	S30	H29	14
39	S30	S31	H30	S29
40	S31	S32	H31	S30
41	S32	S33	H32	S31
42	S33	S34	H33	S32
43	15	14	S35	5
44	S35	S36	H35	15
45	S36	S37	H36	S35
46	S37	S38	H37	S36
47	S38	S39	H38	S37
48	S39	S40	H39	S38
49	S40	H4	S3	
50	S10	H10	S9	
51	S16	H16	S15	
52	S22	H22	S21	
53	S28	H28	S27	
54	S34	H34	S33	
55	S40	H40	S39	
56	H1	S1		
57	H2	S2		

58	H3	S3
59	H4	S4
60	H5	S5
61	H6	S6
62	H7	S7
63	H8	S8
64	H9	S9
65	H10	S10
66	H11	S11
67	H12	S12
68	H13	S13
69	H14	S14
70	H15	S15
71	H16	S16
72	H17	S17
73	H18	S18
74	H19	S19
75	H20	S20
76	H21	S21
77	H22	S22
78	H23	S23
79	H24	S24
80	H25	S25
81	H26	S26
82	H27	S27
83	H28	S28
84	H29	S29
85	H30	S30
86	H31	S31
87	H32	S32
88	H33	S33
89	H34	S34
90	H35	S35
91	H36	S36
92	H37	S37
93	H38	S38
94	H39	S39
95	H40	S40

*** PROBLEM SUMMARY ***

```

95 NODES IN THE SYSTEM
96 NODE CONNECTING ELEMENTS
0 UNKNOWN NCE PARAMETERS
54 UNKNOWN PRESSURES
41 UNKNOWN FLOWS

```

LIST OF INDEX VECTORS. INTERNAL NUMBERING USED.

MAXIMUM INDICIES OF SYSTEM DEPENDENT INDEX VECTORS.

IXR	574
JA	244
IPN	775

INDR

9

INDP

1	2	4	5	6	7	8	42	9	10
11	12	49	13	14	15	16	17	18	50
19	20	21	22	23	24	51	25	26	27
28	29	30	52	32	33	34	25	36	53
31	28	39	40	41	42	54	37	44	45
46	47	48	55						

INDF

2	56	57	58	59	60	61	62	63	64
65	66	67	68	69	70	71	72	73	74
75	76	77	78	79	80	81	82	83	84
85	86	87	88	89	90	91	92	93	94
95									

IRIM

[illegible]

IPIM

1	0	2	3	4	5	6	7	9	10
11	12	14	15	16	17	18	19	21	22
23	24	25	26	28	29	30	31	32	33
41	35	36	37	38	39	48	42	43	44
45	46	8	49	50	51	52	53	13	20
27	34	40	47	54	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

IOIM

0	55	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95					

IPTIPM

1	8	17	26	35	46	55	64	73	84
93	102	111	122	131	140	149	158	167	178
187	196	205	214	223	234	243	252	261	270
279	290	299	308	317	326	335	346	355	364
373	382	391	402	411	420	429	438	447	454
461	468	475	482	489	496	503	510	517	524
531	538	545	552	559	566	573	580	587	594
601	608	615	622	629	636	643	650	657	664
671	678	685	692	699	706	713	720	727	734
741	748	755	762	769					

IPN

1	0	1	1	3	3	0	2	1	0
1	2	-1	1	1	0	3	0	1	2
4	4	3	1	0	4	0	1	2	5
5	4	3	0	5	0	1	3	6	6
9	43	5	4	0	6	0	1	2	7
7	6	5	0	7	0	1	2	8	8
7	6	0	8	0	1	2	10	9	8
7	0	9	0	1	3	11	10	15	13
10	8	0	10	0	1	2	12	11	11
9	0	11	0	1	2	13	12	12	10
0	12	0	1	2	14	49	13	11	0
13	0	1	3	16	14	22	19	15	9
0	14	0	1	2	17	15	16	13	0
15	0	1	2	18	16	17	14	0	16
0	1	2	19	17	18	15	0	17	0
1	2	20	18	19	16	0	18	0	1
2	21	20	20	17	0	19	0	1	3
23	20	29	25	22	13	0	20	0	1
2	24	21	23	19	0	21	0	1	2
25	22	24	20	0	22	0	1	2	26
24	25	21	0	23	0	1	2	27	24
26	22	0	24	0	1	2	28	51	27
23	0	25	0	1	3	30	26	29	19
36	31	0	26	0	1	2	31	27	30
25	0	27	0	1	2	32	28	31	26
0	28	0	1	2	33	29	32	27	0
29	0	1	2	34	30	33	28	0	30
0	1	2	35	35	34	29	0	31	0
1	3	36	25	37	32	43	37	0	32
0	1	2	38	33	37	31	0	33	0
1	2	39	34	30	32	0	34	0	1
2	40	35	39	33	0	35	0	1	2
41	36	40	34	0	36	0	1	2	42
53	41	35	0	37	0	1	3	43	31
44	38	50	43	0	38	0	1	2	45
39	44	37	0	39	0	1	2	46	40
45	38	0	40	0	1	2	47	41	46
39	0	41	0	1	2	48	42	47	40
0	42	0	1	2	49	54	48	44	0
43	0	1	3	50	37	51	44	9	5
0	44	0	1	2	52	45	51	43	0
45	0	1	2	53	46	52	44	0	46
0	1	2	54	47	53	45	0	47	0
1	2	55	48	54	46	0	48	0	1
2	56	55	55	47	0	49	0	1	1
14	12	0	50	0	1	1	21	18	0
51	0	1	1	28	24	0	52	0	1
1	35	30	0	53	0	1	1	42	36
0	54	0	1	1	49	42	0	55	0
1	1	56	48	0	56	1	0	1	57
10	0	57	1	0	1	58	11	0	58
1	0	1	59	12	0	59	1	0	1
60	49	0	60	1	0	1	61	14	0
61	1	0	1	62	15	0	62	1	0

1	63	16	0	63	1	0	1	64	17
0	64	1	0	1	65	18	0	65	1
0	1	66	50	0	66	1	0	1	67
20	0	67	1	0	1	68	21	0	68
1	0	1	69	22	0	69	1	0	1
70	23	0	70	1	0	1	71	24	0
71	1	0	1	72	51	0	72	1	0
1	73	26	0	73	1	0	1	74	27
0	74	1	0	1	75	28	0	75	1
0	1	76	29	0	76	1	0	1	77
20	0	77	1	0	1	78	52	0	78
1	0	1	79	22	0	79	1	0	1
80	33	0	80	1	0	1	81	34	0
81	1	0	1	82	35	0	82	1	0
1	83	26	0	83	1	0	1	84	53
0	84	1	0	1	85	38	0	85	1
0	1	86	39	0	86	1	0	1	87
40	0	87	1	0	1	88	41	0	88
1	0	1	89	42	0	89	1	0	1
90	54	0	90	1	0	1	91	44	0
91	1	0	1	92	45	0	92	1	0
1	93	46	0	93	1	0	1	94	47
0	94	1	0	1	95	48	0	95	1
0	1	96	55	0					

PIVOT ROW LIST.									
1	2	3	4	5	6	7	8	9	10
11	12	49	13	14	15	16	17	18	50
19	20	21	22	23	24	51	25	26	27
28	29	30	52	22	23	24	25	26	53
31	38	39	40	41	42	54	27	44	45
46	47	48	55	2	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95					

MAXIMUM INDEX OF JU AND U IS 1038 MAXIMUM INDEX OF IHOW IS 2069

SOLUTION COMPLETED IN 9 ITERATIONS

CPL COL'S A-D&1-7									
NCE NUM	NCE TYPE	FLOW (GPM)	FRICTION LOSS(PSI)	ELEVATION LOSS(PSI)	FROM NODE	TO NODE	PRESSURE (PSIG)	NODE FLOW (GPM)	RESIDUAL (GPM)
L0	PJ	0.0	-2.23	-2.60	2	1	100.00	1233.83	-3.626
L1	PJ	1227.45	3.23	2.60	1	2	94.17	0.0	7.890
L2	PJ	1729.56	1.91	4.33	2	3	87.93	0.0	0.004
L3	PJ	1229.56	0.90	5.20	3	4	81.84	0.0	-2.100
L4	PJ	1232.66	3.57	0.0	4	5	78.27	0.0	0.607
L5	PJ	685.30	14.98	0.0	5	6	63.28	0.0	-0.183
L6	PJ	685.48	5.24	0.0	6	7	58.05	0.0	0.107
L7	PJ	685.38	9.52	0.0	7	8	48.53	0.0	-0.255
L55	PJ	546.75	32.66	0.0	5	15	44.61	0.0	-0.034
L8	PJ	685.63	2.38	0.0	8	9	46.15	0.0	0.325
L9	PJ	139.15	2.78	1.30	9	51	41.07	0.0	-0.430
L10	PJ	101.95	1.00	0.0	51	52	40.07	0.0	0.001
L11	PJ	68.75	1.58	0.0	52	53	38.49	0.0	-0.032
L12	PJ	24.29	0.44	0.0	53	54	38.05	0.0	-0.003
L13	PJ	546.16	1.56	0.0	9	10	44.59	0.0	0.454
L14	PJ	184.97	5.24	1.30	10	55	38.05	0.0	-0.532
L15	PJ	151.21	2.00	0.0	55	56	36.05	0.0	0.007
L16	PJ	117.82	4.29	0.0	56	57	31.75	0.0	-0.043
L17	PJ	86.53	2.42	0.0	57	58	29.23	0.0	-0.026
L18	PJ	56.45	2.33	0.0	58	59	27.00	0.0	-0.027
L19	PJ	27.58	2.24	0.0	59	510	24.66	0.0	-0.029
L20	PJ	360.73	0.72	0.0	10	11	43.86	0.0	0.562
L21	PJ	180.91	6.14	1.30	11	511	36.42	0.0	-0.525
L22	PJ	147.88	1.92	0.0	511	512	34.50	0.0	0.006
L23	PJ	115.22	4.12	0.0	512	513	30.38	0.0	-0.043
L24	PJ	84.61	2.33	0.0	513	514	28.05	0.0	-0.026
L25	PJ	55.19	2.23	0.0	514	515	25.82	0.0	-0.027

L26	P1	26.97	2.24	0.0	S15	S16	23.58	0.0	-0.029
L27	P1	179.26	0.20	0.0	S1	S2	43.66	0.0	0.255
L28	P1	180.47	6.32	1.30	S12	S17	26.25	0.0	-0.522
L29	P1	147.52	1.91	0.0	S17	S18	24.23	0.0	0.005
L30	P1	114.94	4.10	0.0	S18	S19	30.23	0.0	-0.042
L31	P1	84.41	2.31	0.0	S19	S20	27.92	0.0	-0.027
L32	P1	55.06	2.22	0.0	S20	S21	25.69	0.0	-0.027
L33	P1	26.90	2.23	0.0	S21	S22	23.46	0.0	-0.029
L34	P1	1.56	0.00	0.0	S2	S2	43.66	0.0	0.255
L35	P1	180.47	6.12	1.30	S23	S23	26.25	0.0	-0.524
L36	P1	147.52	1.91	0.0	S23	S24	24.33	0.0	0.006
L37	P1	114.94	4.10	0.0	S24	S25	30.23	0.0	-0.042
L38	P1	84.41	2.31	0.0	S25	S26	27.92	0.0	-0.026
L39	P1	55.06	2.22	0.0	S26	S27	25.69	0.0	-0.027
L40	P1	26.90	2.23	0.0	S27	S28	23.46	0.0	-0.029
L41	P1	182.86	0.21	0.0	S4	S4	43.66	0.0	0.820
L42	P1	180.92	6.15	1.30	S29	S29	26.42	0.0	-0.528
L43	P1	147.89	1.92	0.0	S29	S30	24.50	0.0	0.007
L44	P1	115.23	4.12	0.0	S30	S31	30.38	0.0	-0.041
L45	P1	84.62	2.32	0.0	S31	S32	28.06	0.0	-0.026
L46	P1	55.20	2.23	0.0	S32	S33	25.82	0.0	-0.026
L47	P1	26.97	2.24	0.0	S33	S34	23.58	0.0	-0.028
L48	P1	164.26	0.74	0.0	S5	S5	43.87	0.0	0.485
L49	P1	187.52	6.75	1.30	S35	S35	27.06	0.0	-0.525
L50	P1	140.21	1.06	0.0	S35	S36	25.11	0.0	0.009
L51	P1	116.25	4.19	0.0	S36	S37	30.92	0.0	-0.041
L52	P1	85.23	2.26	0.0	S37	S38	28.56	0.0	-0.024
L53	P1	55.69	2.27	0.0	S38	S39	26.28	0.0	-0.025
L54	P1	27.21	2.28	0.0	S39	S40	24.00	0.0	-0.027
K1	S1	26.67			S1	H1	0.0	-25.62	
K2	S1	25.20			S2	H2	0.0	-25.20	
K3	S1	24.49			S3	H3	0.0	-24.49	
K4	S1	24.20			S4	H4	0.0	-24.20	
K5	S1	24.30			S5	H5	0.0	-24.30	
K6	S1	23.38			S6	H6	0.0	-23.38	
K7	S1	21.22			S7	H7	0.0	-21.22	
K8	S1	20.11			S8	H8	0.0	-20.11	
K9	S1	28.89			S9	H9	0.0	-28.89	
K10	S1	27.61			S10	H10	0.0	-27.61	
K11	S1	23.55			S11	H11	0.0	-23.55	
K12	S1	22.66			S12	H12	0.0	-22.66	
K13	S1	20.64			S13	H13	0.0	-20.65	
K14	S1	29.45			S14	H14	0.0	-29.45	
K15	S1	28.25			S15	H15	0.0	-28.25	
K16	S1	27.00			S16	H16	0.0	-27.00	
K17	S1	23.47			S17	H17	0.0	-23.48	
K18	S1	22.58			S18	H18	0.0	-22.58	
K19	S1	20.57			S19	H19	0.0	-20.57	
K20	S1	29.28			S20	H20	0.0	-29.28	
K21	S1	28.18			S21	H21	0.0	-28.19	
K22	S1	26.93			S22	H22	0.0	-26.93	
K23	S1	32.47			S23	H23	0.0	-32.48	
K24	S1	32.58			S24	H24	0.0	-32.58	
K25	S1	20.57			S25	H25	0.0	-20.57	
K26	S1	29.38			S26	H26	0.0	-29.38	
K27	S1	28.18			S27	H27	0.0	-28.19	
K28	S1	26.93			S28	H28	0.0	-26.93	
K29	S1	32.56			S29	H29	0.0	-32.56	
K30	S1	32.66			S30	H30	0.0	-32.66	
K31	S1	20.65			S31	H31	0.0	-20.65	
K32	S1	29.45			S32	H32	0.0	-29.45	
K33	S1	28.25			S33	H33	0.0	-28.26	
K34	S1	27.00			S34	H34	0.0	-27.00	
K35	S1	23.85			S35	H35	0.0	-23.85	
K36	S1	32.94			S36	H36	0.0	-32.95	
K37	S1	30.92			S37	H37	0.0	-30.92	
K38	S1	29.71			S38	H38	0.0	-29.71	
K39	S1	28.51			S39	H39	0.0	-28.51	
K40	S1	27.24			S40	H40	0.0	-27.24	

LINE

RECOMMENDED
DIAMETERFRICTION
LOSS (PSIG)

L1	10.020	1.068
L4	7.981	0.943
L5	5.047	2.704
L6	4.026	2.840
L7	5.047	1.717
L55	6.065	2.402
L9	2.469	1.603
L14	2.469	2.222
L16	2.067	1.272

L21	2.469	2.606
L23	2.067	1.220
L28	2.469	2.594
L30	2.067	1.215
L35	2.469	2.594
L37	2.067	1.215
L42	2.469	2.606
L44	2.067	1.221
L49	2.469	2.649
L51	2.067	1.241

RUN COMPLETED

REQUEST																		JOB TITLE																				WRITTEN BY															DATE																										
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
BASIC INFORMATION CARD																																																																															
NO.		KEY JUN.					KEY JUN.					KEY JUN.					I O					TITLE OF JOB																																																									
N C E'S		NO.					PRES.					ELEV.																																																																			
LINE & SPRINKLER INFORMATION CARD																																																																															
FROM		TO		TYPE		PRES.		NODE		FLOW		LINE DIA.		LINE		"C"		TO NODE		CONSTANT		LINE																																																									
NODE		NODE						FLOW		OR		K FACTOR		LENGTH		FACTOR		ELEVATION		PRESSURE		NAME																																																									
DENSITY SUBROUTINE CARD																																																																															
LINE NO.		TOTAL					SQ FT																																																																								
TO		SQ. FT.					PER																																																																								
MAIN		OF JOB					HEAD																																																																								
RERUN SUBROUTINE CARD																																																																															
LINE NO.		PRES.					LINE NO.																																																																								
TO		AT					TO																																																																								
MAIN		MAIN					KEY JUN.																																																																								

▼ DENOTES LOCATION OF ASSUMED DECIMAL POINT

Figure 4. CARD FORMAT DATA SHEET.

REFERENCES

- (1) Bugbee, Percy; *Men Against Fire*, p 1; National Fire Protection Association, Boston, Massachusetts; April 1971.
- (2) *Ibid*; p 181.
- (3) Merry and Schiffhauer; "Hydraulic Sprinkler Systems Design—A Computer Approach", *Fire Technology*, p 95; National Fire Protection Association, Boston, Massachusetts; May 1966.
- (4) *Standard for the Installation of Sprinkler Systems*, NFPA 13-1972, p 13-168; National Fire Protection Association, Boston, Massachusetts; May 1972.
- (5) Shamir, U. and Howard, C. D. D.; "Water Distribution Systems Analysis", *Journal of the Hydraulics Division of the American Society of Civil Engineers*, 94, (HY11), pp 210 233; January 1968.

APPENDIX

BASIC PROGRAM DESCRIPTION

Note: The equations used in this description are for gas flows in pipelines. The program that is the subject of this document uses the same technique described herein even though the actual equations in the program are for water flows in pipelines.

The data cards described in this Appendix were broken into NCE data cards and NODE data cards. In the program described by this paper, these two cards have been combined into one.

INTRODUCTION

The following write-up describes a program used to solve a general network balancing problem for a gas transportation system. The basic model formulation is that given in the paper, "Steady-State Analysis of Gas Production, Transmission and Distribution Systems" by M.A. Stoner, presented at the 44th Annual Fall Meeting of the Society of Petroleum Engineers of AIME, Denver, Colorado, September 29, 1969. The node-connecting element types supported by the program are similar to those used in the paper and are described in terms of program language in the section entitled "Node-Connecting Element Equations and Their Derivatives." A node-connecting element abbreviated, NCE, is a system facility such as a pipeline, compressor, valve or regulator, storage field, etc., and resides between a From node and a To node. The program being described was assembled in the Fall of 1969 to provide a basis for instruction of this new analysis method. Various versions of ISAP were in existence at this time, each giving emphasis to different capabilities, but using the same solution method.

The capabilities of the program are very flexible in terms of what network variables can be determined to balance the network flow equations. Also, as an optional feature in this program, a simple sensitivity analysis, which uses the "unknowns" of the network solution as the "free variables," can be performed after the system has been balanced.

As an aid in reading the following program description, a chart showing subroutine hierarchy and calling sequence is provided on the next page. This write-up should be read in the company of a program listing.

Fortran Module	Function
MAIN	- Input/output, some problem definition.
INDX1	- Provide internal description of the system to be balanced.
INDX2	- Set up index vectors for efficient problem solution.
BALALG	- Obtain values of unknowns that balance the system using the n-dimensional Newton-Raphson method.
SETQAF	- Calculate NCE Q's and node imbalance, F.
JACOB	- Set up the Jacobian for the Newton-Raphson linear equations.
PRSDER	- Evaluate pressure derivatives.
FCEDER	- Evaluate prime NCE parameter derivatives.
SENANL	- Performs a sensitivity analysis.
ELMPIV	- Gaussian Elimination of sparse coefficient matrix with pivoting.
ELIMA	- Gaussian Elimination of sparse coefficient matrix using previously determined pivoting.
ELIMB	- Performs row operations on right-hand side of simultaneous linear equations.
BACSUB	- Back substitution of simultaneous linear equations.

MAIN PROGRAM

The program begins by reading in some NAMELIST data which includes the gas gravity, G, the gas temperature, TEMP, the tolerance for balancing the node equations, TOL, and specific values of the balancing control variable, MA, if one doesn't want to use the predefined set. (See Subroutine BALALG.)

Next, some indices are zeroed. This is one of two places in the complete program (including subroutines) where a DO loop terminator is dimension dependent. The other place is in determining the NCE number during the read-in process. The system description cards are then read giving the NCE information. For every NCE the following information is read: A From node and To node name (variables NFN and NTN), an NCE type designation, ie, P1 for pipeline, status of the prime NCE variable, either fixed or unknown, and the values of all parameters used to define the pressure-flow equation for the element. (See the NCE equation and derivative section for more details.) Internal NCE numbers are assigned in the sequence that the NCEs are read. All values for each NCE are printed in the same sequence.

For each NCE a type number is assigned for easier internal differentiation between types. The order of these numbers is determined by their position in the DATA statement initializing the vector NCETYP. The type numbers correspond to the index of NCETYP and are stored in NCEL (I) where I refers to the internal NCE number. A "ZZZZ" in the NFN field indicates all NCEs have been read. The number of NCEs (NNCE) is now known.

Subroutine INDX1 is now called which sets up index vectors that describe the system in terms of internal NCE numbers (assigned during read-in) and internal node numbers (determined in INDX1).

Information derived from INDX1 is the following:

- NN - The number of nodes in the system.
- NN1 - Number of nodes plus 1 (NN+1).
- NODE - A vector of node names relating the node number (index of NODE) to the node name (contents of NODE).
- IX - A vector of internal NCE numbers with associated From node and To node numbers.
- IXR - A node connection vector. For each node it gives the number of adjacent nodes, and each adjacent node number and connecting NCE number.
- 1PT - A vector containing pointers to the start of each node group information in IXr.

(See Subroutine INDX1 write-up for more details.)

Next, a listing of the node correspondence table and the nodes adjacent to each of these nodes is printed. A visual inspection of this list in conjunction with a system schematic will help in making sure that all NCEs were included in the data and that the proper network structure is being modeled.

The vector IUN(I) is searched for a question mark (?) which indicates that the NCE parameter for the Ith NCE is an unknown. A list of these is set aside in INDR and the Number of Unknown NCEs is recorded in NUNCE.

At this stage of the program all possible information has been extracted from the NCE cards. The next step is to read the Node Input Data.

Node card variable names and associated information are the following:

- NODA - The node name.
- PNX - The node pressure.
- MMP - The node pressure status (question mark if unknown).
- QQ - The node flow.
- MMQ - The node flow status (question mark if unknown).

PNX and QQ represent the initial guesses of these variables if their status is unknown. If their status is known, ie, a blank in MMP or MMQ, then these values are fixed values used throughout the solution.

Node cards are read until a "ZZZZ" is encountered in the NODA field. Therefore, it is the user's responsibility to make sure that all nodes defined in the system description data have node input data cards. The user will be told of discrepancies if they occur, and the program will stop. After reading a node input card, the NODE vector is searched to find the node number (N), and the read-in variables are assigned to their proper positions in the appropriate storage vectors. The pressure is converted to absolute and assigned

to the vector P. A vector P2, standing for P squared, is also used because of the repeated use of P squared in the computations for pipeline elements. The node flow QQ is assigned to the vector QN. NPUN records the status of the node pressure and NFUN records the status of the node flow. As the node cards are read, if the status of a node pressure is unknown (?) the node number is entered into the list of node pressure unknowns, INDP, and the Number of Unknown Pressures counter, NUP, is incremented. Likewise, for a node flow the node number is entered in INDF and the Number of Unknown Flows counter, NUF, is incremented. A print-out of the node input data is provided as this data is being read.

After the "ZZZZ" is encountered, a check is made to insure that all nodes were read in and assigned values. A problem summary is then written out giving the number of nodes, number of NCEs, number of unknown NCEs, unknown pressures and unknown flows. Then, before allowing the execution of the problem to proceed, a check is made to insure that the number of unknowns matches the number of nodes. The program will stop if they are not equal.

Subroutine INDX2 is called next to set up some index vectors that will reduce list searching during the application of the N-dimensional Newton-Raphson solution. The index vectors determined in INDX2 are the following:

- IPN - A vector which stores information used to set up the Jacobian.
- IPTIPN - A vector pointing to the node groups in IPN.
- IRIM - A vector which is NNCE positions long giving the matrix column number for the Ith NCE.
- IPIIM - A vector which is NN positions long giving the matrix column number for the Ith node pressure.
- IQIM - A vector NN positions long giving the matrix column number for the Ith node flow.

Because the structure of the Jacobian (coefficient matrix) is known once IPN is determined, row and column indexing, IA and JA, can also be established in INDX2. The IA and JA indexing is necessary for the sparse matrix storage scheme. (See Gaussian Elimination write-up.)

The next section of the Main Program prints out some of the index vectors and the maximum indices of some vectors whose storage requirements are problem dependent. From the list of variables mentioned so far, IXR, JA, and IPN fall into this category. The others, which will be mentioned later, are JU, IHOW, A, and U. The maximum indices are printed out to allow for checking that the maximum dimensioning has not been exceeded.

As a last step, before balancing the system, the prime NCE parameters are placed in a storage vector FF and constant terms for pipeline elements are grouped into the storage vector CO.

The Subroutine BALALG is now called to obtain the values of the unknown variables that will provide a balanced system. Upon return from BALALG all system variables, fixed and unknown, are printed out. This print-out should be the desired results for the balanced system. The node residuals (F) are printed for easy checking of the balancing tolerance.

If it is desired to perform a sensitivity analysis on the system of equations just balanced, this can be done by reading in the system variable which is to be the denominator of the partial derivatives. This program, as written, restricts the free variables to being those that were unknowns in the balancing problem just completed. A more general sensitivity analysis program would not make this restriction but would require more input data to describe the set of free variables. From a tutorial standpoint, however, making this restriction provides a simple sensitivity analysis capability at a very low price. Three variables are read in for each denominator variable desired.

- NFND - The From node if the denominator is to be an NCE parameter; the node name if the denominator is a node pressure or node flow.
- NTND - The To node if the denominator is to be an NCE parameter; the field is blank if the denominator is a node pressure or node flow.
- LTP - The designation of the unknown type:
 - "NCE" for a node connecting element (J = 1)
 - "PRS" for a node pressure. (J = 2)
 - "FLO" for a node flow (J = 3)

A denominator type number (J) is assigned, dependent upon the contents of LTP. The NCE or node number (I) is determined by searching the appropriate lists. The values of J and I are then passed to Subroutine SENANL, which computes the right-hand side of the linear equations and obtains the solution (sensitivity values) which are left in the COMMON vector F. Upon returning from SENANL the sensitivity values are printed out. This process can be repeated for as many sensitivity denominator values that are desired. When a "ZZZZ" is encountered in the NFND field the program returns to Statement 1 to look for a new system to work upon. If no new system is encountered the program stops.

SUBROUTINE INDX1

This subroutine is used for indexing and describing the system of node-connecting elements by an internal numbering system. Basically, for any problem there are two systems of indexing. One is the system that we use outside the program in which we relate or identify nodes and node-connecting elements by node names. The other system is the one that the computer uses internally and is based on an arbitrary numbering system, hence the reference to node-connecting element numbers and node numbers. Subroutine INDX1 sets up vectors such that the program can easily reference NCEs and nodes by its own numbering system. The index vectors used will be described in part by use of a sample system.

Assume that the sample system shown below has been read-in in the following order. The order in which NCE cards are read is arbitrary. Element types and parameters are not important for this demonstration.

From Node	To Node
A14	B6
A12	A14
A12	B6
A12	T900
T900	B6
B6	20

Note that any combination of four alphanumeric characters constitutes a valid node name. The internal node-connecting element numbers would be assigned in the order the NCEs were read in such that A14 to B6 would be NCE No 1, A12 to A14 is No 2, etc. Since there are six elements (NNCE = 6), the vectors NFN and NTN would be six positions long and would contain the following alphanumeric information.

NFN (1) = A14	NTN (1) = B6
NFN (2) = A12	NTN (2) = A14
NFN (3) = A12	NTN (3) = B6
NFN (4) = A12	NTN (4) = T900
NFN (5) = T900	NTN (5) = B6
NFN (6) = B6	NTN (6) = 20

The first step is to set up the node correspondence table which relates each node (by node name) to its internal node number. The method chosen to number the nodes is to first process the list of From nodes and then process the list of To nodes looking for node names that have not been entered in the table. When this process is finished, the number of nodes in the system has been determined (NN = 5) and the node correspondence table is established.

Internal Node No	Node Name
1	NODE (1) = A14
2	NODE (2) = A12
3	NODE (3) = T900
4	NODE (4) = B6
5	NODE (5) = 20

The index vector IX is set up next. IX gives the From node number and the To node number for each NCE in the system. Information is stored in groups of three.

- (1) NCE number
- (2) From node number
- (3) To node number

For the sample system IX is as follows.

IX																	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1	4	2	2	1	3	2	4	4	2	3	5	3	4	6	4	5

The last function INDX1 performs is to set up the node connection index, IXR, and a pointer for this index called IPT. For each node in the system IXR tells how many other NCEs are attached to the node and what the node number is at the other end of each attached NCE. The information is grouped by nodes and IPT(I) gives the position in the IXR vector where the node I information starts. The format for a node group in IXR is as follows:

If the node in question is the From node for the attached NCE, the NCE number is stored with a plus sign. If the node is the To node, the NCE number is stored with a minus sign. For the sample system, the IXR and IPT vectors are as follows:

IXR

1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	2	1	4	-2	2	2	3	2	1	3	4	4	3

15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
3	2	5	4	-4	2	4	4	6	5	-1	1	-3	2	-5	3

31	32	33	34
5	1	-6	4

IPT

1	2	3	4	5
1	7	15	21	31

It is recognized that storage of the NCE number in the IX vector and storage of the node number starting a node group in IXR could be eliminated. It is suggested that if this is ever done, different vector names be used since IX and IXR as defined above are used in several other programs within the Company.

SUBROUTINE INDX2

As written (November 1969) INDX2 is an entry in INDX1. However, it is a "stand alone" subroutine and could be made a separate subroutine by simply providing a COMMON statement. INDX2 sets up an index vector IPN and a pointer vector IPTIPN which facilitates the determination of the Jacobian for the solution of the node equations. It determines vectors IRIM, IPIM, and IQIM which give the matrix column position for each unknown. It also determines the row and column indexing (IA, JA) for the sparse matrix storage of the Jacobian.

Information contained in IPN makes the job of evaluating the Jacobian of the nodal equations very straightforward without much searching of index lists. Since the Jacobian must be evaluated several times in the N-dimensional, Newton-Raphson method, having IPN saves time during the balancing routine. Information is again grouped by nodes (equations) and IPTIPN(I) points to the position in IPN where the node I information begins. The format for a node group of information is as follows:

*(See example on Page 81 for case of more than one NCE between same nodes.)

The contents of IPN are determined by processing the IUN, NPUN, NFUN, and IXR vectors established in the main program and Subroutine INDX1. Suppose for the sample problem that the NCE parameter between A14 and B6 is to be determined along with the Node 20 flow and the pressures at Nodes A14, B6, and T900. IUN, NPUN, and NFUN, have the following contents:

	1	2	3	4	5	6
IUN	?					

	1	2	3	4	5
NPUN	?		?	?	

	1	2	3	4	5
NFUN					?

Therefore, NUNCE = 1, NUP = 3, and NUF = 1. Based on this information and assuming that the node input data occurred in the sequence A12, A14, B6, T900, 20, then INDR, INDP and INDF will have the following contents:

	1
INDR	1

	1	2	3
INDP	1	4	3

	1
INDF	5

With this set of unknowns for the sample problem, IPN and IPTIPN would be filled as follows:

IPN

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	1	1	1	4	1	1	4	2	0	0	3	2	1	3	4	4	3	0

21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38
3	0	1	1	5	4	0	4	0	1	2	1	1	5	3	1	1	1

39	40	41	42	43	44	45
5	1	0	1	6	4	0

IPTIPN

1	2	3	4	5	6
1	10	21	28	39	46

If there had been more than one NCE between the same two nodes, IPN would have to account for this fact to insure that proper derivatives are taken when the Jacobian is constructed. Suppose A12 to T900 had three NCEs, all with fixed parameters. Assume the extra NCEs are added to the end of the NCE data of the above example.

For the same set of unknowns as above, the entries in IXR and IPN for Node A12 (Node No 2) would be as follows:

IXR

7	8	9	10	11	12	13	14	15	16	17	18
2	5	2	1	3	4	4	3	7	3	8	3

IPN

10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
2	0	0	3	2	1	3	4	4	-3	7	-3	8	3	0

For this node there are still three adjacent nodes with unknown pressures. However, all three NCEs going to Node No 3 must be included in the derivative computations. This is accounted for by grouping the information in IPN for the three NCEs together and affixing a minus sign to all duplicate node numbers except the last one. The plus or minus status of a node number coming from the IPN list is then checked when evaluating the derivatives in the Subroutine JACOB. Let us return now to the original example.

The vectors IRIM, IPIM, and IQIM give the matrix column identification for each unknown variable in the system. The unknowns are assigned a column sequentially starting with the unknowns in INDR followed by the INDPs and lastly the INDFs. All nonapplicable entries in IRIM, IPIM, and IQIM are zero. IRIM is NNCE positions long and for this example contains the following values:

	1	2	3	4	5	6
IRIM	1	0	0	0	0	0

IPIM and IQIM are each NN positions long and contain the following information:

	1	2	3	4	5
IPIM	2	0	4	3	0
IQIM	0	0	0	0	5

The last function INDXX2 performs is to allocate the necessary storage and set up indexing to describe the Jacobian using sparse matrix storage (ie, storage of nonzero elements only). In the above sample problem the coefficient matrix would have the following structure:

Node Name	Node No	ΔNCE_1	ΔP_1	ΔP_4	ΔP_3	ΔQN_5
A14	1	X	X	X	0	0
A12	2	0	X	X	X	0
T900	3	0	0	X	X	0
B6	4	X	X	X	X	0
20	5	0	0	X	0	X

It is desired to store and index only the nonzero elements of this coefficient matrix. For each row, JA contains the column numbers of nonzero elements in that row. IA(I) gives the starting position in JA for the Ith row.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
JA	1	2	3	2	3	4	3	4	1	2	3	4	3	5

	1	2	3	4	5	6
IA	1	4	7	9	13	15

These vectors are determined by processing IPN with help from IRIM, IPIM, and IQIM. More information on the meaning and use of the index vectors IA and JA is given in the write-up on Gaussian Elimination using sparse matrix programming.

SUBROUTINE SETQAF (SET Q_s AND F_s)

This subroutine is called when it is desired to calculate the flow in all NCEs (vector Q) and to sum up the NCE flows and the flow entering or leaving the system at a particular node (QN) to obtain F , the node flow imbalance. This function is performed with the initial estimates of all variables and after every application of the correction values to the unknowns. The set of statements necessary to calculate a particular NCE's flow is reached by branching, based on the NCEL number for the element. This type of program organization makes it a simple matter to add other NCE types to increase or change the program's capabilities. The Fortran code for obtaining the NCE flows should be self-explanatory. The presently supported NCE types and their equations are described in the section entitled "Node Connecting Element Equations and Their Derivatives." Once the Q_s for all elements have been determined, the vector IXR is processed to sum up the flows at each node. The resultant imbalance of node flows is placed in the vector F . The sign convention used is plus for flow entering a node and minus for flow leaving a node. If the node imbalance at any one node exceeds the tolerance, TOL, the variable ITOL is set equal to 1 before returning to the calling program.

SUBROUTINE JACOB (JACOBIan)

This routine sets up the Jacobian or coefficient matrix for the simultaneous linear equations. The values are stored in the vector A and are indexed by IA and JA as described in the sparse matrix write-up. The indexing for A (ie, the content of IA and JA) was determined in INDX2. The routine begins by zeroing A . When all derivatives have been computed, there should be no zero values in A except for a possible value resulting from a sonic valve. This zero derivative may or may not give trouble depending on how the system is configured. In running the program one should watch that he doesn't end up with a sonic regulator or valve separating a pressure specified part of the system from a part without pressure specification. This might occur for instance where a sonic valve or regulator is positioned on a lateral with no pressure specified downstream of this element. In effect, two separate systems have been created by the sonic valve.

The coefficient matrix is filled row by row (equation by equation) until all NN equations have been processed. The contents of IPN tell what derivatives are nonzero and how to obtain them. In the process, two basic derivative routines are called. PRSDER evaluates the pressure derivatives and FCEDER determines the NCE parameter derivatives.

FUNCTION PRSDER (PReSsure DERivative)

This function is called to evaluate the derivative of a particular NCE flow with respect to a node pressure. In the function call, the dummy argument N refers to the node for which the equation derivative is desired. J refers to the node-connecting element number and K is the node at the opposite end of this NCE. It is the pressure at node K which is the denominator of the derivative. In other words the function evaluates the derivative of node N equation with respect to the pressure at node K . As in SETQAF a particular element type is referenced by branching based on the value of NCEL for the NCE number. It is to be noted once again that this open-ended structure provides for the easy addition of new node-connecting elements. If a sonic regulator or valve is encountered generating a zero derivative, a notation to this effect is made in the output stream.

FUNCTION FCEDER

This function evaluates the derivative of a particular NCE flow with respect to the prime NCE parameter. This function is also a stand-alone function but is made an entry in PRSDER simply for convenience. In this case the dummy argument N refers to the node whose equation is being processed. J is the node-connecting element number and K is the node number at the opposite end. In summary, FCEDER evaluates the derivative of the node N equation with respect to the prime parameter of NCE number J .

SUBROUTINE BALALG (BALancing ALGorithm)

This subroutine performs the main function of obtaining the solution of the network equations for the particular system being modeled. Prior to calling this routine, all system indexing has been performed and all variables initialized. The solution is obtained by applying the N dimensional Newton-Raphson method to the nodal equations until the flow imbalance of all node equations is less than a certain tolerance, TOL. The satisfaction of this criteria leads to a normal exit from BALALG and a return to the main program for print out of the final results. Abnormal exits are possible. (1) The linear equations which constitute the

Newton-Raphson solution turn out to be singular. Usually this means that the solution being asked for is physically impossible. (2) The solution fails to converge in a specified maximum number of iterations. The trouble here can be a significance problem, ie, adjacent node pressures are so close that round-off error causes an imbalance in the node flows. In this case the balancing criteria is not sufficient to indicate a satisfactorily balanced system. The criteria will probably be changed in the future by including a test on the size of the calculated correction terms. Another possibility is that a solution within prescribed limits is not possible. Items in this category would be that the solution "ran out of pressure" at one or more nodes or an unknown node-connecting element parameter has been driven to its lower limit.

The set of simultaneous linear equations derived from the application of the Newton-Raphson method to the nonlinear node equations has a very sparse $NN \times NN$ coefficient matrix. Due to the structure of the equations, the maximum storage required even during the solution of these equations is less than the $NN \times NN$ locations for a full coefficient matrix. Advantage of this fact has been taken by developing a set of subroutines hereafter called "sparse matrix routines" which solve a general set of simultaneous linear equations using and storing only nonzero elements of the coefficient matrix. Two definite benefits are gained: (1) Problem size capabilities are greatly expanded. (2) Solution speed is considerably increased. A description of these routines is found in the write-up "A General Gaussian Elimination Solution of a Set of Simultaneous Linear Equations Using Sparse Matrix Programming" by M. A. Stoner, October 1969. This is a write-up for in-Company use at Consumers Power Company. An understanding of this solution method and variable storage scheme is essential for following the logic of BALALG.

The solution time for a given problem is dependent on the sequence of calls on the sparse matrix routines. In light of this statement, BALALG provides the capability for the user to determine, *a priori*, set of parameters that will control the calls on the sparse matrix routines. These parameters are contained in the vector MA. The meaning and use of each position of MA are as follows:

- MA(1) - The maximum number of iterations for the solution. If the tolerance is not satisfied after this iteration number, return is made to the main program. A range for this variable is from 10 to 20 iterations.
- MA(2) - The iteration number up to and including that for which a new pivot element determination is made after evaluating the Jacobian (ie, call ELMPIV). This variable must be one or greater.
- MA(3) - The value of MA(3) gives the frequency of calls on ELMPIV after iteration No MA(2) is passed. For most problems it should not be necessary to determine pivots after the first few iterations.
- MA(4) - Up to and including the iteration number given by MA(4), a new Jacobian is always evaluated. It may or may not be triangularized with pivoting depending on the value of MA(2) and MA(3). After this iteration number, it will be possible to find a new right-hand side of the equations and use the same Jacobian as used with the previous iteration. This will perhaps be of advantage during the latter iterations when changes in the Jacobian are small.
- MA(5) - The value of MA(5) gives the iteration frequency for computing new Jacobians after iteration MA(4) has passed. For the intervening iterations, the new right-hand sides are evaluated and used with the most recent Jacobian. The term right-hand side used here refers to the node imbalance or residuals.
- MA(6) - MA(6) and MA(7) give numbers which govern the application of acceleration values to the unknown variable correction terms. MA(6) is the number of initial iterations for which the acceleration values for all unknowns are to increase linearly from 0 to 1.0. The example shown clarifies the use of MA(6).

This provides the user with control over solutions that are strongly divergent during the first few iterations. A value of $MA(6) = 0$ will, of course, let the routine apply the full correction at iteration No 1.

MA(7) - This value gives the frequency of iterations after MA(6) has passed at which the acceleration scheme described in the SPE paper is applied.

MA(8) - This position is not used at this time.

MA(9) - Print out of the Jacobian (in sparse matrix form), triangularized matrix, U, and a portion of IHOW are controlled by this variable. Print out of this information (voluminous for large problems) is provided for all iterations up to and including the iteration number given in MA(9).

MA(10) - For each iteration up to and including MA(10) a print out is provided for all node pressures and flows and the node-connecting elements that have unknown parameters. This allows the user to monitor the convergence of the solution.

A basic flow chart showing the use of the variables MA(2) to MA(5) and the resultant calls on various subroutines is provided at the end of this subroutine write-up.

BALALG begins by zeroing counters IC, KO, and LO. IC is the iteration counter. KO counts iterations between determination of new pivoting and LO counts iterations between determination of new Jacobians.

The path through the subroutine for any one iteration is dependent upon the current values of the MA parameters and the iteration number. (See the flow chart.)

Beginning with the most time-consuming path (determination of pivoting), the first step is to call Subroutine SETQAF (SET Qs And Fs). This subroutine computes the current values of Q (flow through the NCE) for all NCEs and sums up all flows into and out of each node including the QN value to determine the node imbalance F. The vector of -Fs is later used as the right-hand side of the simultaneous linear equations. If the tolerance is satisfied, SETQAF returns a value ITOL = 0. The system is considered balanced and control returns to the main program. If the tolerance is not met, a return of ITOL = 1 is made and the iteration process continues. The next step is to evaluate the coefficient matrix or Jacobian (synonymous terms) by calling Subroutine JACOB. The numbers are placed in the vector A. A set of simultaneous linear equations is now defined and a print out is made if desired. Subroutine ELMPIV (ELiMination with PIVoting) is called to perform Gaussian elimination on the coefficient matrix and at the same time establish the IHOW vector which describes the pivot elements and row operations for later use. The triangularized matrix is stored in U with row and column indices IU and JU. MARK is returned as 0 if the matrix is nonsingular and as 1, 2, or 3 if it is singular. The associated reason is printed and execution stops. A printout of the triangularized matrix and the contents of IHOW is provided if desired (MA(9) control). The next step is to apply the elimination row operations to the Right-Hand Side vector, RHS. To do this, Subroutine ELIMB is called. This is immediately followed by a call on BACSUB which performs the BACK SUBstitution operation and leaves the solution of the simultaneous linear equations in the vector DX (Delta X). This path is the one which will always be required on the first iteration.

Another possibility for any iteration after the first is to use the pivoting information stored in IHOW to perform Gaussian elimination on a newly determined coefficient matrix. In this case the program first calls SETQAF followed by JACOB if the tolerance is not satisfied. The Subroutine ELIMA is called to triangularize the coefficient matrix. Now the only possibility for a singular matrix is to encounter a pivot element which is zero but was previously nonzero when the pivoting was established. MARK is returned as 4 if this happens. Having triangularized the matrix, the solution proceeds by calling ELIMB and BACSUB to obtain the solution vector DX.

Another possibility for any iteration after that given by the contents of MA(4) is to use a previous iteration coefficient matrix with the new node residuals. The sequence of calls is now SETQAF to get the new residuals followed by ELIMB and BACSUB.

Once DX has been determined by one of the above sequences of subroutine calls, these correction values must be used to update their respective unknown variables. As described in the SPE paper and under the usage of MA(6) and MA(7) given above, an acceleration value is determined to aid in the convergence of the solution. This acceleration value, ACEL, is multiplied times the correction value DX and the product added to the current value of the variable.

Since negative values for all NCE parameters and node pressures are not valid, a lower limit is adhered to for all unknown variables except node flows. The arbitrary limits set in the program are pipe diameter 1", compression horsepower 10 units, valve conveyance .00005 and deliverability curve, number of wells is 1 well. For a node pressure the lower limit is 14.7 psia. No limit is placed on node flows. These limits may, of course, be adjusted to reflect differing usage of the method. If a solution will not balance in the maximum number of iterations, one of the most likely reasons is that there is not a physically possible solution and, consequently, one or more variables will be at a lower limit.

The print out for each iteration is intended to be as instructive as possible in an effort to aid in understanding the convergence of the solution. All node values are printed. Pressures and flows with trailing question marks are unknown and the value given reflects the applied correction term modified by the printed acceleration value. During the iteration printout, only unknown NCE information is printed.

In conclusion of this subroutine write-up, it should be noted that the scheme used to vary the subroutine calls and the scheme used to obtain acceleration values would both have to be classified as experimental. Although they have been found to work quite well in obtaining the balance of a gas system, much experimentation remains to be done, perhaps leading to changes in this balancing algorithm.

SUBROUTINE SENANL (SENSitivity ANaLysis)

This subroutine performs the task of obtaining the sensitivity values when a sensitivity analysis is requested. It is placed as an entry in BALALG simply because the vectors IU, JU, and U are needed and are already defined there. According to the reasoning programmed into BALALG concerning the need for pivoting, there should be no need to reestablish pivoting during this simplified sensitivity analysis solution, since the structure of the coefficient matrix will not change. Furthermore, the most recent value of the Jacobian used in balancing the system can be used for the sensitivity analysis without introducing significant error. However, for completeness, the first time SENANL is entered, a new Jacobian and pivoting information is determined by calling Subroutine JACOB and ELMPIV. On succeeding calls of SENANL this step is bypassed. The right-hand side is the only part of the set of equations that will change from one call to the next (ie, for successive calls with different denominator variables). Therefore, the remainder of the subroutine involves determining the proper RHS, performing the row operations on it by calling ELIMB, and solving for the sensitivity coefficients by calling BACSUB. These values are placed in the vector F and return is made to the main program for print out.

GLOSSARY OF MAJOR VARIABLE NAMES IN THE INSTRUCTIONAL VERSION OF ISAP

COMMON VARIABLES

- A - The vector in which the compacted coefficient matrix is stored.
- CO - This vector contains the grouped effect of all nonvariable pipeline parameters, including the system gravity and temperature.
- F - This variable represents the node flow imbalances.
- FF - This vector is used to store the prime NCE parameters for all node-connecting elements.
- IA - The contents of this variable give the starting position in the Vectors A and JA of the row of A given by the index of IA.
- INDF - This vector is NUF elements long and contains the node numbers of nodes with unknown flows.
- INDP - This vector is NUP elements long and contains the node numbers of nodes with unknown pressures.
- INDR - This vector is NUNCE elements long and contains the NCE numbers of all the NCEs with unknown parameters.
- IPIM - A vector NN positions long giving the matrix column number for the Ith node pressure.
- IPN - A vector which stores information used to set up the Jacobian.
- IPT - A pointer to node groups in the IXR vector.
- IPTIPN - A pointer to node groups in the IPN vector.
- IQIM - A vector NN elements long giving the matrix column number for the Ith node flow.
- IRIM - A vector NNCE elements long giving the matrix column number for the Ith NCE.
- IUN - A vector NNCE elements long containing the unknown status (question mark or blank) of each NCE parameter.
- IX - A vector of internal NCE numbers with associated From node and To node numbers.
- IXR - A node-connection vector giving the node numbers of adjacent nodes and NCE numbers of adjacent node-connecting elements for each node in the system.
- JA - Contains column numbers of nonzero elements in the Vector A.
- IT1 - The logical unit number for input.
- LT3 - The logical unit number for output.
- NCEL - A vector NNCE elements long containing the NCE type number.
- NFN - A list of From nodes for the system.
- NFUN - A vector containing the unknown status of each node flow.
- NN - The number of nodes in the system.

NN1 - The number of nodes plus 1.
 NNCE - The number of node-connecting elements in the system.
 NODE - A vector of all the node names in the system.
 NPUN - A vector containing the unknown status of all node pressures.
 NTN - A list of all To nodes in this system.
 NUF - The number of nodes with unknown flows.
 NUNCE - The number of node-connecting elements with unknown parameters.
 NUP - The number of nodes with unknown pressures.
 P - A vector containing the pressure at every node.
 P2 - The square of the pressure at every node.
 Q - A vector NNCE elements long containing the flow in MMcf/d in each node-connecting element.
 QN - The node flow or "load" for every node in the system.
 TOL - A node imbalance tolerance in MMcf/d to which all node equations will be balanced.
 X1 - A storage vector for NCE parameters contains the diameter for a pipeline, the horsepower for a compressor, the valve constant for a valve and the number of wells for a storage field.
 X2 - Contains the length in miles for a pipeline, the factor k_1 for a compressor, the flow coefficient for a storage field and is not used for a valve.
 X3 - Contains the efficiency factor for a pipeline, k_2 for a compressor, the exponent n for a storage field and is not used for a valve.
 X4 - Contains the variable k_3 for a compressor and is not used for a pipeline, valve, or storage field.

MAIN PROGRAM

G - The gravity of the flowing gas.
 GT - A combination of the gravity and temperature variables.
 IQUES - Contains a question mark for use in comparisons.
 IS - Contains a list of letter combinations to determine sensitivity analysis denominator types.
 LNTYP - A single variable used for NCE type numbers in computed GO TO statements.
 LTP - A read-in variable for a sensitivity analysis denominator type.
 MA - A vector ten elements long containing control information for the balancing algorithm.
 MMP - A read-in variable containing the unknown status of a node pressure.
 MMQ - A read-in variable containing the unknown status of a node flow.
 NCETYP - A vector containing the NCE type designation names used for input and output identification, ie, P1, C1, R1, and F1.
 NCLL - A read-in variable containing the NCF type designator.
 NFND - For sensitivity denominator variable input; it is the node name if a node pressure or node flow, the From node if an NCE parameter.
 NODA - A read-in variable containing a node name.
 NSTOP - Contains the characters "ZZZZ" used to terminate input lists.

- NTND - For sensitivity analysis denominator input it is a node, the To node name if an NCE parameter; it is not used if the denominator is a node pressure or node flow.
- PNX - A variable used for input and output of node pressures.
- QQ - A variable used for input and output of node flows.
- TEMP - The temperature in degrees rankine of the flowing gas.

BALALG SUBROUTINE

- ACEL - A vector containing the acceleration values for the unknown variable correction terms.
- DX - A vector containing the correction terms for the NN unknown variables at any one iteration.
- DX1 - A vector containing the correction terms for the unknown variables at the previous iteration.
- IC - The iteration counter.
- IHOW - A vector which records information about the Gaussian elimination process determined in subroutine ELMPIV.
- ITOL - A switch variable indicating the satisfaction of the tolerance limit computed in subroutine SETUAT.
- IU - A vector NN elements long containing the starting position for each row of information in JU and U.
- JU - A vector giving the matrix column positions of nonzero elements in the vector U.
- KO - An iteration counter between determination of new pivoting.
- LO - An iteration counter between determination of new Jacobians.
- MARK - An argument returned from the Gaussian elimination process indicating the condition of the matrix, ie, singular or nonsingular.
- RHS - A vector used to communicate the right-hand side of a set of simultaneous linear equations to the subroutines which solve these equations.
- U - A vector containing the nonzero elements of the triangularized matrix determined during the Gaussian elimination process.

NODE-CONNECTING ELEMENT EQUATIONS AND THEIR DERIVATIVES

For this instructional version of ISAP, very simple equations are used to describe the four basic NCE types, ie, pipeline, compressor, valve, and storage field. The equations used are the same as in the SPE paper except the pipeline equation now used is the Panhandle B. Supercompressibility is not used in the program in order to maintain simplicity. Therefore, the program as it stands would probably not be useful for real-life application. It is, however, very easy to modify the equations and introduce such items as compressibility, flow dependent friction, node elevations, etc. One other difference is noted between the SPE paper and the program, that being the diameter is now used for the prime NCE parameter of a pipeline instead of the conglomerate conveyance term. The term "prime NCE parameter" refers to the one parameter of any NCE which can be an unknown.

All flows used in the program are in MMcf/d. All input and output pressures are in psig; however, the program works with pressures in psia.

For each node-connecting element there is a distinctive From and To node except in the case of more than one NCE between the same two nodes. For each NCE there is also an indication of the unknown status of the prime NCE parameter. This information is necessary for each NCE but is not the main item of interest here. This section is concerned with the correspondence between the NCE parameters and the program variables. In this version of the program, storage is provided for four NCE parameters which are X1, X2, X3, and X4. For each NCE type, the meaning of these variables is described below along with the NCE type designator for each NCE supported by the program. The system elements presently supported by the program are described below:

PIPELINE

The Panhandle B equation is used to model a pipeline.

$$Q = E 737 \left(\frac{T_b}{P_b} \right)^{1.020} \left[\frac{P_1^2 - P_2^2}{G .961 \text{ TEMP } L Z} \right]^{.510} D^{2.53}$$

Since no supercompressibility is used, Z is unity. For use in the program an orientation is given to this equation by defining positive flow from the From node to the To node with a base temperature of 520° and the base pressure of 14.7 psia, the equation becomes

$$Q_{ij} = \frac{S 0.02794 E_{ij}}{G .49 \text{ TEMP} .51 L_{ij} .51} D_{ij}^{2.53} \left[S (P_i^2 - P_j^2) \right]^{.51}$$

where S is +1. for positive flow and -1. for negative flow. The gas gravity G and temperature TEMP are input for the whole system in the NAMELIST &DIN leaving three pipeline parameters to be input on the NCE card. D, the diameter in inches, is read in the X1 field and is the prime pipeline parameter. L, the length in miles, is read in the X2 field and E, the line efficiency, is read in the X3 field. The type designation for this equation is P1. Before balancing begins, the terms which cannot vary are grouped into the variable CO.

$$CO_{ij} = \frac{0.02794 E_{ij}}{G .49 \text{ TEMP} .51 L_{ij} .51}$$

which then makes the pipeline equation:

$$Q_{ij} = S D_{ij}^{2.53} CO_{ij} \left[S (P_i^2 - P_j^2) \right]^{.51}$$

In terms approaching those used in the program, the pipeline equation becomes:

$$CO_{ij} = \frac{0.02794 X3_{ij}}{G .49 \text{ TEMP} .51 X2_{ij} .51}$$

$$Q_{ij} = S FF_{ij}^{2.53} CO_{ij} (S DP)^{.51}$$

$$\text{where } DP = P_i^2 - P_j^2$$

$$\text{and } S = +1. \quad DP \geq 0.$$

$$S = -1. \quad DP < 0.$$

FF_{ij} contains the prime NCE parameter which in this case is the diameter.

COMPRESSOR

This equation is used just as it is given in the SPE paper.

$$Q_{ij} = \frac{HP_{ij}}{k_1 \left(\frac{P_j}{P_i} \right)^{k_3} - k_2}$$

The compressor unit horsepower, HP, is read in the X1 field and is the prime NCE parameter. k₁ is read in the X2 field, k₂ is read in the X3 field, and k₃ is in the X4 field. The compressor type designation is C1.

When a pressure gain element of this type is introduced into a system, there is no longer a unique solution to the network equations unless the flow through the compressor element is restricted to being in one direction only. Therefore, to prevent the generation of an undesired solution the user must decide in advance of the simulation which way he wants the flow to go through the compressor. Then he chooses the From and To node so that flow will always be positive through the compressor. The program is presently designed to force this condition at all times. Consequently, in terms of the program variables, the flow through a compressor unit is

$$Q_{ij} = FF_{ij} / (X2_{ij} (P_j/P_i)^{X4_{ij}} - X3_{ij})$$

where FF_{ij} contains the unit horsepower.

VALVE

The valve or regulator model is the same as that given in the SPE paper. For subsonic flow

$$\left. \begin{aligned} Q_{ij} &= K_{ij} \sqrt{(P_i - P_j) P_j} & P_i > P_j \\ Q_{ij} &= -K_{ij} \sqrt{(P_j - P_i) P_i} & P_j > P_i \end{aligned} \right\} .55 \leq \frac{P_i}{P_j} \leq 1.82$$

and for sonic flow

$$Q_{ij} = .5 K_{ij} P_i \frac{P_i}{P_j} > 1.82$$

$$Q_{ij} = .6 K_{ij} P_j \frac{P_i}{P_j} < .55$$

As written there is only one valve parameter, K_{ij} , which is read in the X1 field. X2 to X4 have no meaning for a valve. K is therefore the prime NCE parameter for a valve which means also that the form of the valve equation in the computer program is very similar to those given above with the exception that K_{ij} is replaced with ΓF_{ij} . The type designation for a valve or regulator is R1.

STORAGE FIELD DELIVERABILITY CURVE

The equation used for this element is again the same as in the SPE paper.

$$Q_{ij} = S NW_{ij} CO_{ij} (S (P_i^2 - P_j^2))^n$$

NW_{ij} , the number of wells, is the prime NCE parameter and is read in the X1 field. CO_{ij} is read in the X2 field and the exponent n is read in the X3 field. The type designation for this element is F1. In terms of the program variables the equation is

$$Q_{ij} = S FF_{ij} X2_{ij} (S DP) X3_{ij}$$

where DP and S are the same as defined for the pipeline equation.

DERIVATIVES

In order to construct the Jacobian it is necessary to be able to define and evaluate various derivatives of the NCE equations. In actuality what is needed is the derivative of F, the node equation, with respect to an NCE parameter, a node pressure, or a node flow. The node flow derivative is simply 1, and is set in JACOB. For the others, function subprograms are called to evaluate the needed derivative. For an NCE parameter derivative, function FCEDER is called. The value returned is really the derivative of Q for that NCE but it is also the derivative of F. In the case of a node pressure derivative, PRSDER is called which also evaluates the derivative of a Q with respect to a node pressure. This value is either exactly the value of the desired F derivative: $\frac{\partial F_j}{\partial P_i}$

or part of a summation to get the F derivative as in the case of $\frac{\partial F_j}{\partial P_i}$. It is interesting and helpful to note that an increase in the node imbalance is produced by an increase in an adjacent node pressure for all NCE types. Likewise, for all NCE types presently supported in the program an increase in the prime NCE parameter produces an increase in the NCE flow which increases the To node F and decreases the From node F. In the following derivatives reference is made to the typical segment depicted as

It should be noted that the sign convention used is plus (+) for flow entering a node and minus (-) for flow leaving a node.

Assume all NCE parameters have the subscripts ij.

PIPELINE

$$Q_{ij} = S FF^{2.53} CO (S (P_i^2 - P_j^2))^{.51}$$

Pressure Derivatives

$$\frac{\partial F_j}{\partial P_i} = \frac{\partial Q_{ij}}{\partial P_i} = FF^{2.53} CO 1.02 P_i / (S (P_i^2 - P_j^2))^{.49}$$

$$\frac{\partial F_i}{\partial P_j} = - \frac{\partial Q_{ij}}{\partial P_j} = FF^{2.53} CO 1.02 P_j / (S (P_i^2 - P_j^2))^{.49}$$

Diameter Derivatives

$$\frac{\partial F_j}{\partial FF} = \frac{\partial Q_{ij}}{\partial FF} = S 2.53 FF^{1.53} CO (S (P_i^2 - P_j^2))^{.51}$$

$$\frac{\partial F_i}{\partial FF} = - \frac{\partial Q_{ij}}{\partial FF} = -S 2.53 FF^{1.53} CO_{ij} (S (P_i^2 - P_j^2))^{.51}$$

COMPRESSOR

$$Q_{ij} = \frac{FF}{X2 \left(\frac{P_j}{P_i} \right)^{X4} - X3} = FF \left(X2 \left(\frac{P_j}{P_i} \right)^{X4} - X3 \right)^{-1}$$

Pressure Derivative

$$\frac{\partial F_j}{\partial P_i} = \frac{\partial Q_{ij}}{\partial P_i} = \frac{X2 X4 FF P_j^{X4}}{\left(X2 \left(\frac{P_j}{P_i} \right)^{X4} - X3 \right)^2 P_i (1. + X4)}$$

$$\frac{\partial F_i}{\partial P_j} = - \frac{\partial Q_{ij}}{\partial P_j} = - \frac{X2 X4 FF (1. / P_i)^{X4}}{\left(X2 \left(\frac{P_j}{P_i} \right)^{X4} - X3 \right)^2 P_j (1. - X4)}$$

Horsepower Derivatives

$$\frac{\partial F_j}{\partial FF} = \frac{\partial Q_{ij}}{\partial FF} = 1. / \left(X2 \left(\frac{P_j}{P_i} \right)^{X4} - X3 \right)$$

$$\frac{\partial F_i}{\partial FF} = - \frac{\partial Q_{ij}}{\partial FF} = -1. / \left(X2 \left(\frac{P_j}{P_i} \right)^{X4} - X3 \right)$$

VALVE

$$\text{Subsonic Flow } (.55 \leq \frac{P_i}{P_j} \leq 1.82)$$

$$P_i > P_j$$

$$Q_{ij} = FF \sqrt{(P_i - P_j) P_j}$$

$$\frac{\partial F_j}{\partial P_i} = \frac{\partial Q_{ij}}{\partial P_i} = \frac{.5 FF P_j}{\sqrt{(P_i - P_j) P_j}}$$

$$\frac{\partial F_i}{\partial P_j} = - \frac{\partial Q_{ij}}{\partial P_j} = - \frac{.5 FF (2 P_j - P_i)}{\sqrt{(P_i - P_j) P_j}}$$

$$\frac{\partial F_j}{\partial FF} = \frac{\partial Q_{ij}}{\partial FF} = \sqrt{(P_i - P_j) P_j}$$

$$\frac{\partial F_i}{\partial FF} = - \frac{\partial Q_{ij}}{\partial FF} = - \sqrt{(P_i - P_j) P_j}$$

$$P_i < P_j$$

$$Q_{ij} = -FF \sqrt{(P_j - P_i) P_i}$$

$$\frac{\partial F_j}{\partial P_i} = \frac{\partial Q_{ij}}{\partial P_i} = \frac{.5 FF (2 P_i - P_j)}{\sqrt{(P_j - P_i) P_i}}$$

$$\frac{\partial F_i}{\partial P_j} = - \frac{\partial Q_{ij}}{\partial P_j} = \frac{.5 FF P_i}{\sqrt{(P_j - P_i) P_i}}$$

$$\frac{\partial F_j}{\partial FF} = \frac{\partial Q_{ij}}{\partial FF} = - \sqrt{(P_j - P_i) P_i}$$

$$\frac{\partial F_i}{\partial FF} = - \frac{\partial Q_{ij}}{\partial FF} = \sqrt{(P_j - P_i) P_i}$$

Sonic Flow

$$\frac{P_i}{P_j} > 1.82$$

$$Q_{ij} = .5 FF P_i$$

$$\frac{\partial F_j}{\partial P_i} = \frac{\partial Q_{ij}}{\partial P_i} = .5 FF$$

$$\frac{\partial F_i}{\partial P_j} = \frac{\partial Q_{ij}}{\partial P_j} = 0.$$

$$\frac{\partial F_j}{\partial FF} = \frac{\partial Q_{ij}}{\partial FF} = .5 P_i$$

$$\frac{\partial F_i}{\partial FF} = - \frac{\partial Q_{ij}}{\partial FF} = - .5 P_i$$

$$\frac{P_i}{P_j} < .55$$

$$Q_{ij} = -.5 FF P_j$$

$$\frac{\partial F_j}{\partial P_i} = \frac{\partial Q_{ij}}{\partial P_i} = 0.$$

$$\frac{\partial F_i}{\partial P_j} = - \frac{\partial Q_{ij}}{\partial P_i} = .5 FF$$

$$\frac{\partial F_j}{\partial FF} = \frac{\partial Q_{ij}}{\partial FF} = -.5 P_j$$

$$\frac{\partial F_i}{\partial FF} = - \frac{\partial Q_{ij}}{\partial FF} = .5 P_j$$

STORAGE FIELD DELIVERABILITY CURVE

$$Q_{ij} = S FF X2 (S (P_i^2 - P_j^2)) X3$$

Pressure Derivatives

$$\frac{\partial F_j}{\partial P_i} = \frac{\partial Q_{ij}}{\partial P_i} = 2 X3 FF X2 P_j / (S (P_i^2 - P_j^2)) (1. - X3)$$

$$\frac{\partial F_i}{\partial P_j} = - \frac{\partial Q_{ij}}{\partial P_j} = 2 X3 FF X2 P_i / (S (P_i^2 - P_j^2)) (1. - X3)$$

Well Derivative

$$\frac{\partial F_j}{\partial FF} = \frac{\partial Q_{ij}}{\partial FF} = S X2 (S (P_i^2 - P_j^2)) X3$$

$$\frac{\partial F_i}{\partial FF} = - \frac{\partial Q_{ij}}{\partial FF} = -S X2 (S (P_i^2 - P_j^2)) X3$$

In programming some of the above derivative evaluations, an artificial delta P square is used if both attached node pressures are exactly the same. If this safeguard is not used, an erroneous zero derivative can generate a singular matrix which will keep the program from solving the problem. So far this has only been a problem in starting a solution where several adjacent node pressures were given an identical initial guess. A zero pressure derivative is valid only for a sonic valve.

In conclusion of this section one is reminded that it is very easy to add, delete or modify NCE equations in the program and thus extend or contract the capabilities of the program in terms of supported NCE types.

Node Connecting Element Cards

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
NFN				NTN				NCLL				IUN				X1				X2				X3				X4																																																			
A4				A4				A2				A1				F9.3				F9.3				F9.3				F9.3																																																			

Node Input Data Cards

[illegible]

Sensitivity Analysis Cards

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80																								
NFND												NIND												LTP																																																																															
A4												A4												A3																																																																															

DISTRIBUTION**Atomic Energy Commission - ORO**

Handler, R. (3)
 Hickman, H. D.
 Milloway, J. T., Jr (2)
 Zachry, D. S., Jr

Y-12 Central Files (5)
 Y-12 Central Files (master copy)
 Y-12 Central Files (route)
 Y-12 Central Files (Y-12RC)

Goodyear Atomic - Portsmouth

Beabout, R. W.
 Buckalew, F.

Paducah Gaseous Diffusion Plant

Hughey, J. W. (6)
 Phipps, J. K.

National Fire Protection Association

Morgan, C. E.
 Stevens, R. E. (3)

Oak Ridge Gaseous Diffusion Plant

Orrin, R. C.
 Patton, F. S.
 Richardson, W. L. (3)
 Trotter, T. C.
 Ulm, R. W. (14)

Oak Ridge National Laboratory

Beeler, B. M. (6)
 Winslow, W. R.

Oak Ridge Y-12 Plant

Blackmon, J. T., Jr (50)
 DeMonbrun, J. R. (6)
 Denny, A.
 Fleischman, W. H.
 Fraser, R. J.
 Godsey, A. R.
 Keith, A.
 Parsons, J. A.
 Phillips, L. R.
 Rhew, J. E.
 Stickle, E. T. (6)
 Weathersby, W. E.

In addition, this report is distributed in accordance with the category, UC-2, **General, Miscellaneous, and Progress Reports**, as given in the *USAEC Standard Distribution Lists for Unclassified Scientific and Technical Reports*, TID-4500.