T-1762

THE OPTIMIZATION OF PIPE DISTRIBUTION SYSTEMS

By

Leo A. Kinney, Jr.

A thesis submitted to the Faculty and the Board of Trustees of the Colorado School of Mines in partial fulfillment of the requirements for the degree of Master of Science (Mathematics).

Signed: Leo A. Kinney, Jr.

Leo A. Kinney, Jr.
Student

Golden, Colorado

Date: 12-8-, 1975

Approved: H. S. Swanson

Dr. H. S. Swanson
Thesis Advisor

Robert A. Walsh

Dr. Robert A. Walsh
Head, Mathematics Department

Golden, Colorado

Date: December 9, 1975

ABSTRACT

A systematic method is developed to determine the optimal pipe sizes for an irrigation pipe distribution system for the minimum total cost. A series of computer programs, based on dynamic and linear programming formulations, are used to produce the desired solution in a final report format. Two example problems are included to show the applications of the proposed algorithm.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS AND FIGURES

INTRODUCTION

The purpose of this thesis is to develop a systematic method of determining the optimal pipe sizes in a pipe distribution system for irrigation projects. Although this thesis will only deal with pipe networks for irrigation systems, the same principles could be modified for any pipe network designed to transport fluids.

There are basically two types of irrigation systems, pump and gravity. In a pump system, the total cost of pipe and the total cost of pumping must be minimized simultaneously to obtain the lowest combined total cost. In a gravity system, the beginning elevation of the water at the source is known, and the only objective is to minimize the total cost of pipe.

The two problems are similar in many aspects. For both types of systems, the geometric layout and the lengths of the pipe are known. Also, the hydraulic properties and costs of all the pipes to be considered are known or can be determined for each system. The purpose of a pipe distribution system for irrigation projects is to deliver to the farmer his desired flow of water at a required pressure. This point of delivery to the farmer will be referred to as a "delivery" in this thesis. For each system type, the required flow and pressure is known for each delivery. The only additional information needed for a pump system is the cost of power for pumping.

There have been several papers that have dealt with the problem of determining the economical pipe sizes for a pipe network. One of the earlier articles was by Camp (1) who developed a mathematical relationship that expressed the total cost of the pipe distribution system as a function of pipe size. Camp did not present a practical method that would find the pipe sizes which would make this total cost a minimum for a practical or "real world" problem. For many years the design procedure for pipeline distribution systems was based on a trial and error method. For a given network, various combinations of pipe sizes were arrived at on the basis of personal judgment and the corresponding costs were computed. Then the total cost of the network was determined for each of the different combinations of pipe sizes, and the lowest cost system was selected. This approach was very time consuming, and most engineering organizations did not have a sufficient number of designers to thoroughly investigate the numerous combinations of pipe sizes. Often the time constraint on completing a job was such that only a few trials could be made. Obviously, there was always the probability that a more economical design existed.

A paper by Mandry (2) was published in 1967, and it described a practical procedure for solving the formulation proposed by Camp. Mandry presented a straight-forward approach for obtaining the solution for the problem, but his procedure was very tedious to use on a large distribution system.

Charles A. Calhoun (3) and Leo A. Kinney, Jr., worked for Mandry in the Canals and Pipelines Section of the Bureau of Reclamation until Mandry's death in 1966. Soon thereafter, Calhoun and Kinney developed computer programs to implement Mandry's method. These computer programs were limited to accommodate a single pipeline without any branches. However, by making several computer runs, the minimum total cost could be obtained for a network with multiple branches.

The advent of the computer and operations research methods opened new doors for the designer of pipeline distribution systems. In 1968, Karmeli, Gadish, and Meyers (4) published an article on the formulation of the problem using linear programming and another article on solving the problem using dynamic programming. Calhoun's paper in 1970 presented a more detailed explanation of the linear programming formulation. Kally (6) and Liang (7) also had papers published describing dynamic programming formulations. Richard A. Simonds (8) and Kinney wrote a paper in 1972 which described the application of operations research techniques to optimize a pipe distribution system.

The two operations research methods mentioned earlier have been successfully used in the design of a pipe distribution system, but there were disadvantages in using them. Kinney developed a computer program using the dynamic programming philosophy to analyze an entire network in one run, but the output generated by this program was not in a final, usable format. Hillier and Lieberman (9) presented the following basic features which characterize a dynamic programming problem:

1. The problem can be divided into stages, with a policy decision required at each stage.

2. Each stage has a number of states associated with it.

3. The effect of the policy decision at each stage is to transform the current state into a state associated with the next stage.

4. Given the current state, an optimal policy for the remaining stages is independent of the policy adopted in previous stages.

5. The solution procedure begins by finding the optimal policy for each state of the last stage.

6. A recursive relationship is available which identifies the optimal policy for each state with n stages remaining, given the optimal policy for each state with (n-1) stages remaining.

7. Using this recursive relationship, the solution procedure moves backward stage by stage each time finding the optimal policy for each state of that stage until it finds the optimal policy when starting at the initial stage.

The computer program generates one pipe diameter for each reach of pipe, which is the distance between deliveries. This was undesirable since on some reaches of pipe there would be excess pressure. What diameter and length of a smaller pipe size would eliminate this extra pressure? The designer knew the unknown length had to be a portion of the original length between deliveries, and in most situations the unknown pipe diameter would be the next smaller pipe size. Although these facts were beneficial in solving for the location that the pipe should change from one pipe diameter to the smaller pipe diameter, there would be numerous computations for a large distribution system. The location of these pipe diameter changes are commonly known as taper locations or taper stations.

The results of the linear programming packages produced the taper locations, but additional problems arose. The linear programming formulation will be presented later in this thesis, and it will be shown that a group or series of pipe choices must be given for each reach of pipe. One method that is presently used in the preparation of data for a linear programming solution is to select for each reach one pipe size based on a velocity, say 5 feet per second. Using this pipe size as a base, the two next smaller and two next larger pipe diameters are selected, which would result in five different pipe sizes considered for each reach. The hydraulic properties of these pipe diameters are required for the constraint equations which will be formulated later in this thesis. The cost per foot of these pipes will be in the objective function which will also be later formulated. When the coefficients and right-hand sides of the objective function and constraint equations and inequalities are properly entered as input for a linear programming computer program, an optimal solution will be obtained.

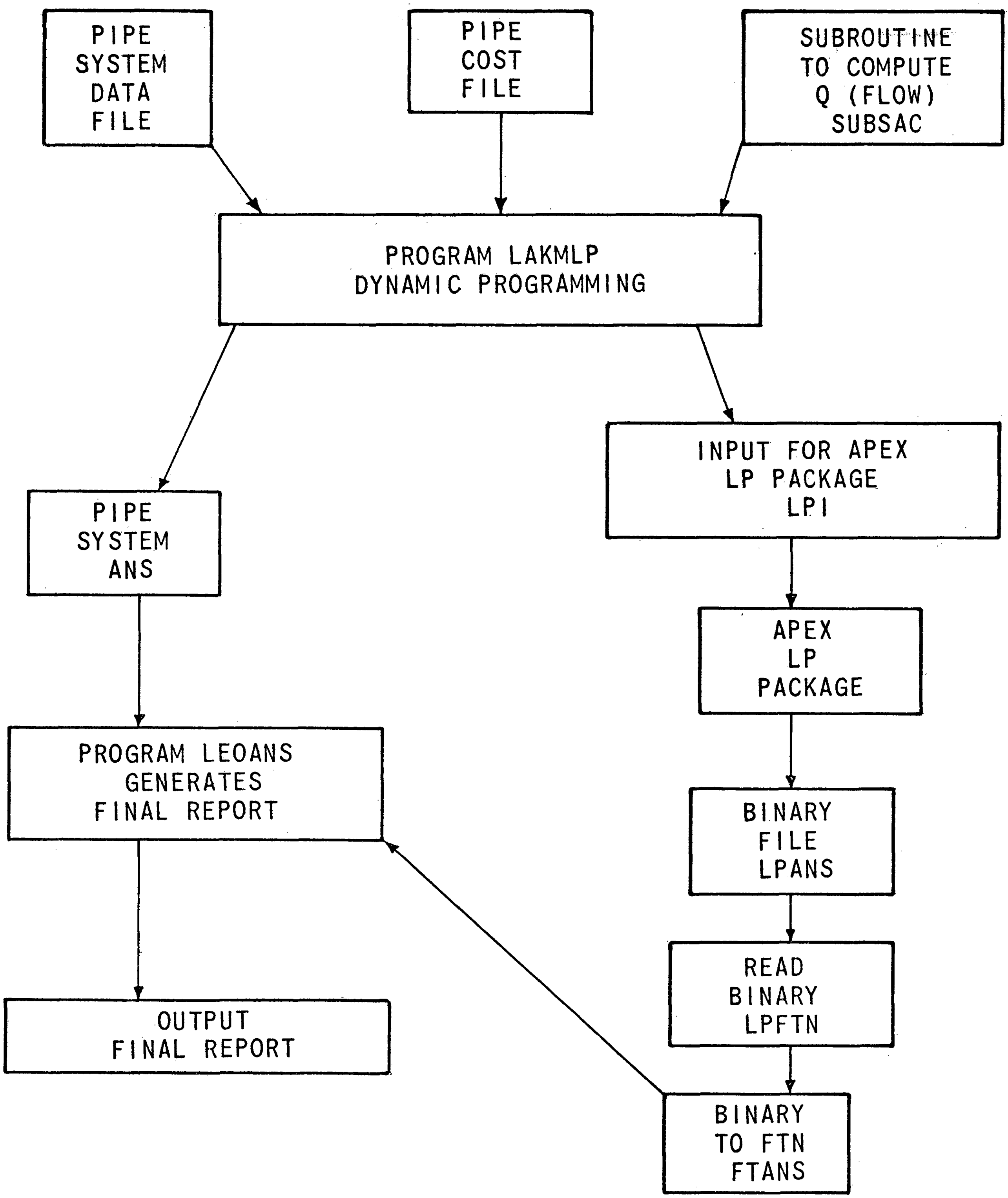
This solution is only optimal with respect to the five pipe choices per reach which were under consideration without rigorous economical justification. If the five pipe diameters had been selected by some economical method, then the linear programming solution would be optimal in a broader sense.

Another problem associated with using most linear programming computer programs is that the results generated are in tableau formats which are cumbersome to use. The costs and hydraulic properties of the five pipe choices per reach are discrete parameters and are entered into the objective function and constraint equations as mentioned earlier. Therefore, the results in the final tableau have to be manually associated with the proper pipe diameter, which is a very tedious task for a large network.

A procedure will be shown in this thesis that uses the dynamic programming philosophy in conjunction with linear programming to produce computer output in report form that will not require any further adjustments before it is used. First will be presented the dynamic programming formulation for a pipe distribution system. Then, example problems will be solved using the dynamic programming computer program named LAKMLP. This is a program that was developed to solve a pipe distribution system using dynamic programming, and at the same time, automatically generate the input file for a linear programming package. Then will be developed the linear programming formulation for the same example problems. Next, a linear programming package will be used to generate a computer file that will be later required to develop the final report.

The flow diagram on page 7 indicates the input files, programs, and output files that will be used in this thesis. Program LAKMLP requires two input files: a pipe system file and a pipe cost file. The pipe system file contains general information about the pipe distribution system, such as the geometric layout and hydraulic requirements of the network. The cost per foot of the pipes are in the pipe cost file. Since the amount of water required for deliveries varies depending on the geographic location, a variable subroutine, SUBSAC, is needed to compute the flow or

FLOW DIAGRAM



cubic feet per second in the network. Program LAKMLP generates two output files which are named ANS and LP1. ANS contains information about the pipe system that is used at a later stage to generate the final report. The other output file, LP1, is the input file for the linear programming package APEX, which is available on the CDC 6400/6600 computer model CYBER 74-28. The file LP1 consists of the coefficients and the right-hand side of the objective function and the coefficients and the right-hand sides of the constraint equations and inequalities in an acceptable format for APEX.

The standard output from APEX is in tableau form and will not be used in this thesis. An individual familiar with linear programming could use the final tableau to perform a sensitivity analysis, but most designers who need to use this are not familiar with linear programming. Although it is usually advisable to perform a sensitivity analysis to determine the effect on the optimal solution if certain parameters take on other possible values, this will not be discussed in this thesis. An optional binary file, LPANS, is also generated by APEX. The program LPFTN was written to read this binary file and to create the fortran file FTANS. The pipe system file, ANS, and the file FTANS are input files for the computer program LEOANS. The program LEOANS is used to generate the final report. This procedure has been used sucessfully to solve a network that consisted of 325 reaches which produced an initial tableau with 976 rows and 1,951 columns for the linear programming model.

GRAVITY PROBLEM

The objective for a gravity pipe distribution system is to minimize only the total cost of pipe. Camp recognized that there is only one sequence of pipe diameters satisfying the length and hydraulic constraints that will yield the minimum total pipe cost.

Problem Description

Consider the pipe system and associated data as shown in figure 1.

In most gravity systems, the source of water is a canal with a known water surface elevation, such as Elevation 1100. Each point of delivery is indicated by an arrow in the figure. At each delivery, the number of acres to be irrigated and the required hydraulic elevation are known. The distances between the delivery points and/or sublaterals are also indicated.

Generally, the lines or laterals of a pipe distribution system are given names to facilitate the bookkeeping. The longest line is considered to be the main line, and in figure 1 was arbitrarily named Main. The lines that branch off of the main line are known as subs, and have also been named. If a line comes off of a sub, the name of this line implies this as SUB 1-A.

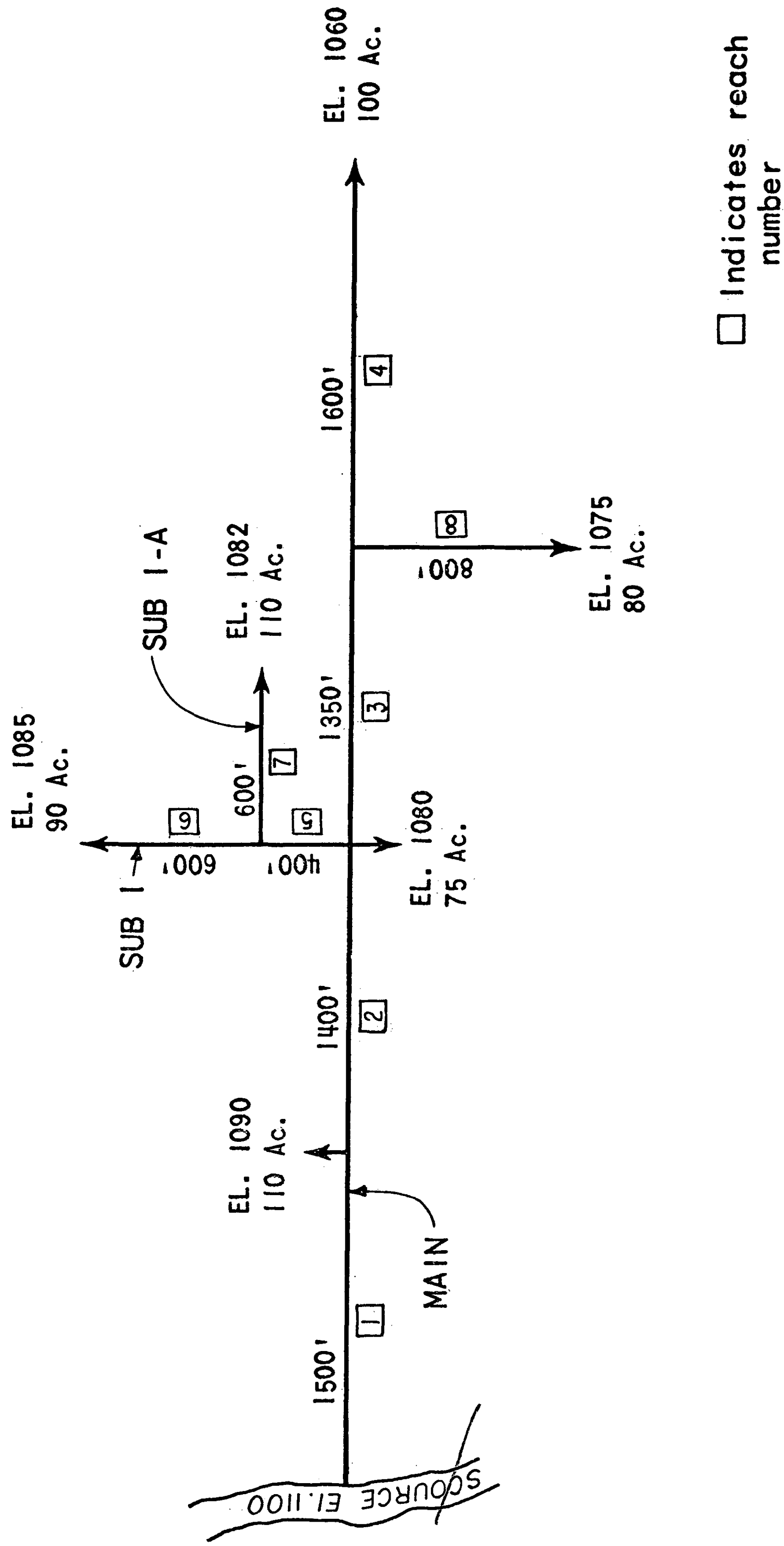

SOURCE EL.1100
1500'
1
MAIN
EL. 1090
110 Ac.
1400'
2
SUB I
600'
6
EL. 1085
90 Ac.
400'
5
EL. 1080
75 Ac.
600'
7
SUB I-A
EL. 1082
110 Ac.
1350'
3
800'
8
EL. 1075
80 Ac.
1600'
4
EL. 1060
100 Ac.
□ Indicates reach number

FIGURE I

Dynamic Programming Formulation

Mandry developed a procedure to express the total cost of a pipe distribution system as a function of pipe size using Scobey's formula. The following formulation would have to be slightly modified if one of the other well-known pipe formulas was used in computing hydraulics for a pipe distribution system.

Scobey's pipe formula is:

$$Q = 3.72\ C_s\ D^{2.625}\ H^{0.5} \qquad (1)$$

where Q is flow in cubic feet per second; C_s is coefficient of friction; D is diameter of pipe in feet; and H is head loss in feet per 1,000 feet of pipe. This formula can be rewritten as:

$$D = \left(\frac{K\ Q^2\ L}{h} \right)^{.19} \qquad (2)$$

in which K is $1/(13,840\ C_s^{\ 2})$; L is length of pipe reach; and h is the friction loss in the pipe reach. Mandry showed that the pipe costs can be expressed as:

$$C = y\ D^x \qquad (3)$$

where C is installed cost in dollars per foot of pipe length; y is cost in dollars of 1 linear foot of 12-inch-diameter pipe; and x is the slope of the cost curve. The cost curve can be obtained by plotting the cost per foot of pipe versus pipe diameter on log-log graph paper, and then values of x and y can be obtained as shown by Mandry. A least squares fit analysis could also be used to obtain the x and y values.

The total cost for the main line in figure 1 can be expressed as:

$$C = C_1L_1 + C_2L_2 + C_3L_3 + C_4L_4 \qquad (4)$$

where the subscripts refer to the number of the pipe reach starting from the source. If $y\ D_i^{\ x}$ is substituted for C_i, i = 1, 2, 3, 4; the following expression is obtained:

$$C = y\ D_1^{\ x}\ L_1 + y\ D_2^{\ x}\ L_2 + y\ D_3^{\ x}\ L_3 + y\ D_4^{\ x}\ L_4 \qquad (5)$$

Substituting $\left(\frac{K\ Q_i^{\ 2}\ L_i}{h_i}\right)^{.19}$ for D_i, i = 1, 2, 3, 4 results in:

$$C = Y\left(\frac{KQ_1^{\ 2}L_1}{h_1}\right)^{.19X} L_1 + Y\left(\frac{KQ_2^{\ 2}L_2}{h_2}\right)^{.19X} L_2 + Y\left(\frac{KQ_3^{\ 2}L_3}{h_3}\right)^{.19X} L_3 + Y\left(\frac{KQ_4^{\ 2}L_4}{h_4}\right)^{.19X} L_4$$

The cost of any two adjoining pipes in the network can be minimized by equating to zero the first derivative of C_i, cost per unit length, with respect to h_i, friction loss per unit length, in either pipe reach.

For the first two reaches, the first derivative with respect to h_1 is:

$$\frac{d\ (C_1 + C_2)}{d\ h_1} = 0 \qquad (6)$$

Mandry noted that $d\ h_1 = -d\ h_2$ and, therefore, the minimum cost for the first two pipe reaches is:

$$\frac{d\ C_1}{d\ h_1} = \frac{d\ C_2}{d\ h_2} \tag{7}$$

and the minimum cost for the main line is:

$$\frac{d\ C_1}{d\ h_1} = \frac{d\ C_2}{d\ h_2} = \frac{d\ C_3}{d\ h_3} = \frac{d\ C_4}{d\ h_4} \tag{8}$$

where:

$$\frac{dC_1}{dh_1} = \frac{.19xK^{.19X}\ Q_1^{.38X}\ L_1^{1\ +\ .19X}}{h_1^{1\ +\ .19X}} \tag{9}$$

$$\frac{dC_2}{dh_2} = \frac{.19xK^{.19X}\ Q_2^{.38X}\ L_2^{1\ +\ .19X}}{h_2^{1\ +\ .19X}} \tag{10}$$

$$\frac{dC_3}{dh_3} = \frac{.19xK^{.19X}\ Q_3^{.38X}\ L_3^{1\ +\ .19X}}{h_3^{1\ +\ .19X}} \tag{11}$$

$$\frac{dC_4}{dh_4} = \frac{.19xK^{.19X}\ Q_4^{.38X}\ L_4^{1\ +\ .19X}}{h_4^{1\ +\ .19X}} \tag{12}$$

Let $S = \frac{h}{L}$, or

$$\frac{L_i^{\,1 + .19X}}{h_i^{\,1 + .19X}} = \frac{1}{S_i^{\,1 + .19X}} \tag{13}$$

By substituting expressions, collecting terms, and simplifying, equation (8) becomes:

$$\frac{Q_1^{\,.38x}}{S_1^{\,1 + .19x}} = \frac{Q_2^{\,.38x}}{S_2^{\,1 + .19x}} = \frac{Q_3^{\,.38x}}{S_3^{\,1 + .19x}} = \frac{Q_4^{\,.38x}}{S_4^{\,1 + .19x}} \tag{14}$$

Now, each of the slopes in equation (14) can be expressed as a function of S_1.

$$S_2 = \left(\frac{Q_2^{\,.38X}}{Q_1^{\,.38X}}\right)^{\frac{1}{1 + .19X}} S_1 \tag{15}$$

$$S_3 = \left(\frac{Q_3^{\,.38X}}{Q_1^{\,.38X}}\right)^{\frac{1}{1 + .19X}} S_1 \tag{16}$$

$$S_4 = \left(\frac{Q_4^{\,.38X}}{Q_1^{\,.38X}}\right)^{\frac{1}{1 + .19X}} S_1 \tag{17}$$

As Mandry noted, this information is interesting but useless in its present form. He was able to develop a series of equations for each reach of pipe. If the source of water is point zero, and the end of reach 1 is point 1, then the allowable friction loss from point zero to point one is the difference in elevations of the two points. Let $h_{0,1}$ be this allowable friction loss. By subtracting the required elevations at each point from the elevation at the source, similar h's can be obtained. Using the same method of subscripts for the L's and Q's, the following expressions can be obtained:

Reach 1:
$$S = \frac{h_{0,1}}{L_{0,1}}$$

Reach 2:
$$S = \frac{h_{0,2}}{L_{0,1} + \left(\frac{Q_{1,2}}{Q_{0,1}}\right)^{\frac{.38X}{1 + .19X}} L_{1,2}}$$

Reach 3:
$$S = \frac{h_{0,3}}{L_{0,1} + \left(\frac{Q_{1,2}}{Q_{0,1}}\right)^{\frac{.38X}{1 + .19X}} L_{1,2} + \left(\frac{Q_{2,3}}{Q_{0,1}}\right)^{\frac{.38X}{1 + .19X}} L_{2,3}}$$

Reach 4:
$$S = \frac{h_{0,4}}{L_{0,1} + \left(\frac{Q_{1,2}}{Q_{0,1}}\right)^{\frac{.38X}{1 + .19X}} L_{1,2} + \left(\frac{Q_{2,3}}{Q_{0,1}}\right)^{\frac{.38X}{1 + .19X}} L_{2,3} + \left(\frac{Q_{3,4}}{Q_{0,1}}\right)^{\frac{.38X}{1 + .19X}} L_{3,4}}$$

Similar equations can be developed for the reaches on the sublaterals. Now the problem is to evaluate these equations, each of which will yield a different value for S. The reach with the minimum slope, S^*, is the economic control for all the pipe reaches upstream of that reach. The slopes of all the reaches down to the reach which had the minimum slope can now be determined by multiplying the minimum slope by the ratios of Q's. For example, if the minimum slope had been found in reach 3, then the following economical slopes can be found by:

$$S_1 = \left(\frac{Q_{0,1}}{Q_{0,1}} \right)^{\frac{.38X}{1 + .19X}} S^* = S^*$$

$$S_2 = \left(\frac{Q_{1,2}}{Q_{0,1}} \right)^{\frac{.38X}{1 + .19X}} S^*$$

$$S_3 = \left(\frac{Q_{2,3}}{Q_{0,1}} \right)^{\frac{.38X}{1 + .19X}} S^*$$

These economical slopes require nonstandard pipe diameters, and standard pipe diameters with slopes nearly matching these economical slopes must now be found. This completes the first stage of the dynamic programming problem. The source is now moved to the end of the reach that had the minimum slope. Then this procedure is repeated until the minimum slope is on the last reach of pipe. Thus the stages of the dynamic programming problem are not predetermined, but depend on the location of the minimum slope. The point at the end of this minimum slope is referred to as a control point. For a simple pipe system, with no sublaterals and only one control point at the end of the last reach, only one sequence of analysis is required. A large distribution system with many control points requires that this procedure be repeated as many times as the number of control points.

Problem Solution

The computer program LAKMLP was used to solve the problem described in figure 1. The appendix contains the listing of this program which was written to solve a pipe distribution system using dynamic programming. The pipe prices and subroutine SACQ are also in the appendix.

The pipe prices are the total installed cost of the pipe. Included in this cost are excavation of pipe trench, furnishing and laying the pipe, and backfilling the trench. These costs vary with time and geographic location. Obtaining accurate pipe prices for a new pipe project might be the most difficult data to collect. Manufacturers of pipe are reluctant in giving out prices since they will be bidding for the pipe on the new project. However, the cost of pipe recently installed in the same vicinity gives an indication of what the pipe costs might be.

Subroutine SACQ was used to determine the flow for the pipe distribution system in figure 1. This is a variable subroutine and has to be modified for each new project, because water requirements for projects vary with types of crops and geographic location. The flow for the pipe distribution system in figure 1 is 6.1 gallons per minute per acre which is converted to cubic feet per second per acre in SACQ.

The pipe system file, DATA1, is in the appendix, and it contains information which describes the pipe system in figure 1. The first two lines contain the elevation at the source, pipe cost factors, x and y, and other miscellaneous information. The name of the main lateral is on the next line, and then there are five columns of numbers. The first column indicates if the lateral is the main or one of the subs. The next column is the station at which a reach of pipe begins. Stations are used in this file instead of lengths because surveyors stake pipe distribution systems using the concept of stations. In the sixth column there is a number, -250., which denotes the value of an equation. An equation is often used by surveyors to join two lines together. The concepts of stations and equations are mentioned only to point out that the usual method of describing reaches of pipe does not have to be modified for the input file, DATA1. The third column denotes the number of acres to be irrigated, and the required elevations are in column four. The fifth column contains the natural ground elevations at each delivery and at each point that a sub begins.

The solution to this dynamic programming problem is in the appendix and has each lateral on a separate page. The name of the lateral is at the top of the page, and the pipe class design elevation is also indicated. This is the maximum pressure elevation that will occur on the pipes in the different reaches. This pressure is used in the design of the pipe and is called head class, which is denoted by HC in the last column. STATION and Q have been defined earlier. The hydraulic grade line is HGL and DIA is the pipe diameter in inches. V is the velocity and HV is the velocity head. The SLOPE, S, has been defined earlier, and LOSS was defined as h or the friction loss. AC is acres, and SAC is the summation of acres. The flow for each delivery is DQ, and the size of the delivery in inches is DEL. HGR is the required elevation, and NGS is the natural ground surface elevation. The value of the equation is denoted by EQ on lateral SUB2.

It should be noted at this point that the HGL is always greater than the required elevation, HGR. The dynamic programming formulation presented earlier does not have the capability of determining more than one pipe size per reach. Linear programming does have this capability, and its formulation will be presented next.

Linear Programming Formulation

Using the format of Hillier and Lieberman, a linear programming problem is one which can be expressed as: minimize (or maximize) a linear function

$$Z = C_1 X_1 + C_2 X_2 + \quad + C_n X_n,$$

subject to the constraints,

$$a_{11} X_1 + a_{12} X_2 + \quad + a_{1n} X_n \; (\leq, =, \geq) \; b_1$$

$$a_{21} X_1 + a_{22} X_2 + \quad + a_{2n} X_n \; (\leq, =, \geq) \; b_2$$

$$a_{m1} X_1 + a_{m2} X_2 + \quad + a_{mn} X_n \; (\leq, =, \geq) \; b_m$$

and

$$X_1 \geq 0, \; X_2 \geq 0, \; \ldots \; X_n \geq 0$$

The objective function for a pipe distribution system should always be minimized, since the objective is to minimize the total cost for the system. For given sizes and types of pipe, the costs per foot of pipe (C_1, C_2, ..., C_n) are known. The problem is to find the unknown lengths (X_1, X_2, ..., X_n) of the various sizes of pipe which will minimize the objective function subject to the constraint equations and inequalities. For a pipe distribution system, there are three types of constraints. The length constraint requires that the summation of unknown lengths of various sizes of pipe equal the length of the reach under consideration. The hydraulic constraints are expressions relating the friction slopes and lengths of a pipe system to the allowable friction losses in the reaches. The third type of constraint requires that the values of the unknown lengths (X_1, X_2, ..., X_n) be greater than or equal to zero. There are five variables and one each of the three types of constraints associated with each reach of pipe.

In the linear programming formulation for a pipe distribution system, a problem arises as to the proper choice of pipe diameters for each reach of pipe. The linear programming computer program will produce an optimal solution with respect to the pipe diameters and costs that are considered. The importance of obtaining accurate pipe costs has already been discussed in the dynamic programming formulation and is equally important in the linear programming formulation. To obtain the minimum total cost for a pipe distribution system, the choice of pipe sizes should be based on some type of economical analysis. The dynamic programming computer program presented earlier has already optimally selected one pipe diameter for each reach of pipe. To insure that the linear programming computer program selects the optimal pipe diameters for each reach, five pipe diameters should be considered for each reach. Therefore, the pipe choices for each reach should consist of the pipe diameter selected by the dynamic programming computer program, the next two smaller pipe diameters, and the next two larger pipe diameters. It should be noted, that on most pipe distribution systems, there is a minimum pipe diameter that is allowed for any reach. The reasons for this limitation are varied, and will not be discussed in this thesis. This size limitation means that, for some reaches, the five pipe choices will be the smallest allowable five pipe diameters. For the distribution system shown in figure 1, the smallest allowable pipe diameter is a 6-inch-diameter pipe.

Using the data that are on figure 1 and the information that was generated by the program LAKMLP, the following two tables can be obtained:

TABLE I

DATA FOR PIPE SYSTEM IN FIGURE I

REACH NO.	LENGTH OF REACH IN FEET	FLOW, IN CUBIC FEET PER SECOND	ELEVATION REQUIRED, IN FEET
1	1500.	7.68	1090.
2	1400.	6.19	1080.
3	1350.	2.45	1071.
4	1600.	1.36	1060.
5	400.	2.72	1079.
6	600.	1.22	1085.
7	600.	1.50	1082.
8	800.	1.09	1075.

TABLE 2

POSSIBLE PIPE CHOICES

REACH	UNKNOWN VARIABLE	PIPE DIAMETER IN INCHES	FRICTION SLOPE IN FEET PER FOOT	COST, IN DOLLARS PER LINEAR FOOT
	X_1	27	.000443	19.10
	X_2	24	.000823	16.38
1	X_3	21	.001907	13.70
	X_4	18	.004285	10.45
	X_5	15	.011158	8.31
	X_6	24	.000533	16.38
	X_7	21	.001237	13.70
2	X_8	18	.002779	10.45
	X_9	15	.007237	8.31
	X_{10}	12	.023351	6.77
	X_{11}	15	.001133	8.31
	X_{12}	12	.003655	6.77
3	X_{13}	10	.009518	5.73
	X_{14}	8	.030712	4.65
	X_{15}	6	.139071	3.65
	X_{16}	15	.000350	8.31
	X_{17}	12	.001128	6.77
4	X_{18}	10	.002938	5.73
	X_{19}	8	.009479	4.65
	X_{20}	6	.042923	3.65
	X_{21}	18	.000537	10.45
	X_{22}	15	.001398	8.31
5	X_{23}	12	.004512	6.77
	X_{24}	10	.011750	5.73
	X_{25}	8	.037916	4.65
	X_{26}	15	.000283	8.31
	X_{27}	12	.000914	6.77
6	X_{28}	10	.002379	5.73
	X_{29}	8	.007678	4.65
	X_{30}	6	.034768	3.65
	X_{31}	15	.000423	8.31
	X_{32}	12	.001365	6.77
7	X_{33}	10	.003554	5.73
	X_{34}	8	.011470	4.65
	X_{35}	6	.051937	3.65
	X_{36}	15	.000224	8.31
	X_{37}	12	.000722	6.77
8	X_{38}	10	.001880	5.73
	X_{39}	8	.006067	4.65
	X_{40}	6	.027471	3.65

The objective function and constraint equations and inequalities can now be generated using the information in TABLES 1 and 2. The objective function would be:

$$Z = 19.10\ X_1 + 16.38\ X_2 + 13.70\ X_3 + 10.45\ X_4 + 8.31\ X_5 +$$

$$16.38\ X_6 + 13.70\ X_7 + 10.45\ X_8 + 8.31\ X_9 + 6.77\ X_{10} +$$

$$8.31\ X_{11} + 6.77\ X_{12} + 5.73\ X_{13} + 4.65\ X_{14} + 3.65\ X_{15} +$$

$$8.31\ X_{16} + 6.77\ X_{17} + 5.73\ X_{18} + 4.65\ X_{19} + 3.65\ X_{20} +$$

$$10.45\ X_{21} + 8.31\ X_{22} + 6.77\ X_{23} + 5.73\ X_{24} + 4.65\ X_{25} +$$

$$8.31\ X_{26} + 6.77\ X_{27} + 5.73\ X_{28} + 4.65\ X_{29} + 3.65\ X_{30} +$$

$$8.31\ X_{31} + 6.77\ X_{32} + 5.73\ X_{33} + 4.65\ X_{34} + 3.65\ X_{35} +$$

$$8.31\ X_{36} + 6.77\ X_{37} + 5.73\ X_{38} + 4.65\ X_{39} + 3.65\ X_{40}.$$

The length constraints would be:

$$X_1 + X_2 + X_3 + X_4 + X_5 = 1{,}500.$$

$$X_6 + X_7 + X_8 + X_9 + X_{10} = 1{,}400.$$

$$X_{11} + X_{12} + X_{13} + X_{14} + X_{15} = 1{,}350.$$

$$X_{16} + X_{17} + X_{18} + X_{19} + X_{20} = 1,600.$$

$$X_{21} + X_{22} + X_{23} + X_{24} + X_{25} = 400.$$

$$X_{26} + X_{27} + X_{28} + X_{29} + X_{30} = 600.$$

$$X_{31} + X_{32} + X_{33} + X_{34} + X_{35} = 600.$$

$$X_{36} + X_{37} + X_{38} + X_{39} + X_{40} = 800.$$

The first hydraulic constraint would be:

$$1100. \quad .000443\ X_1 \quad .000823\ X_2 \quad .001907\ X_3 - .004285\ X_4 \quad .011158\ X_5 \geq 1,090.$$

This constraint may be written as:

$$.000443\ X_1 + .00823\ X_2 + .001907\ X_3 + .004285\ X_4 + .01158\ X_5 \leq 10.$$

Similarily, the other hydraulic constraints are as follows:

to the second reach

$$.000443\ X_1 + .000823\ X_2 + .001907\ X_3 + .004285\ X_4 + .011158\ X_5 +$$

$$.000533\ X_6 + .001237\ X_7 + .002779\ X_8 + .007237\ X_9 + .023351\ X_{10} \leq 20.$$

to the third reach

$$.000443\ X_1 + .000823\ X_2 + .001907\ X_3 + .004285\ X_4 + .011158\ X_5 +$$

$$.000533\ X_6 + .001237\ X_7 + .002779\ X_8 + .007237\ X_9 + .023351\ X_{10} +$$

$$.001133\ X_{11} + .003655\ X_{12} + .009518\ X_{13} + .030712\ X_{14} + .139071\ X_{15} \leq 29.$$

to the fourth reach

$$.000433\ X_1 + .000823\ X_2 + .001907\ X_3 + .004285\ X_4 + .011158\ X_5 +$$

$$.000533\ X_6 + .001237\ X_7 + .002779\ X_8 + .007237\ X_9 + .023351\ X_{10} +$$

$$.001133\ X_{11} + .003655\ X_{12} + .009518\ X_{13} + .030712\ X_{14} + .139071\ X_{15} +$$

$$.000350\ X_{16} + .001128\ X_{17} + .002938\ X_{18} + .009479\ X_{19} + .042923\ X_{20} \leq 40.$$

to the fifth reach

$$.000433\ X_1 + .000823\ X_2 + .001907\ X_3 + .004285\ X_4 + .011158\ X_5 +$$

$$.000533\ X_6 + .001237\ X_7 + .002779\ X_8 + .007237\ X_9 + .023351\ X_{10} +$$

$$.000537\ X_{21} + .001398\ X_{22} + .004512\ X_{23} + .011750\ X_{24} + .037916\ X_{25} \leq 21.$$

to the sixth reach

$$.000433\ X_1 + .000823\ X_2 + .001907\ X_3 + .004285\ X_4 + .011158\ X_5 +$$

$$.000533\ X_6 + .001237\ X_7 + .002779\ X_8 + .007237\ X_9 + .023351\ X_{10} +$$

$$.00537\ X_{21} + .001398\ X_{22} + .004512\ X_{23} + .011750\ X_{24} + .037916\ X_{25} +$$

$$.000283\ X_{26} + .000914\ X_{27} + .002379\ X_{28} + .007678\ X_{29} + .034768\ X_{30} \leq 13.5.$$

to the seventh reach

$$.000433\ X_{1} + .000823\ X_{2} + .001907\ X_{3} + .004285\ X_{4} + .011158\ X_{5} +$$

$$.000533\ X_{6} + .001237\ X_{7} + .002779\ X_{8} + .007237\ X_{9} + .023351\ X_{10} +$$

$$.000537\ X_{21} + .001398\ X_{22} + .004512\ X_{23} + .011750\ X_{24} + .037916\ X_{25} +$$

$$.000423\ X_{31} + .001365\ X_{32} + .003554\ X_{33} + .011470\ X_{34} + .051937\ X_{35} \leq 15.$$

to the eighth reach

$$.000433\ X_{1} + .000823\ X_{2} + .001907\ X_{3} + .004285\ X_{4} + .011158\ X_{5} +$$

$$.000533\ X_{6} + .001237\ X_{7} + .002779\ X_{8} + .007237\ X_{9} + .023351\ X_{10} +$$

$$.001133\ X_{11} + .003655\ X_{12} + .009518\ X_{13} + .030712\ X_{14} + .139071\ X_{15} +$$

$$.000224\ X_{36} + .000722\ X_{37} + .001880\ X_{38} + .006067\ X_{39} + .027471\ X_{40} \leq 23.5.$$

The hydraulic constraints that contain the reaches in the subs have 1.5 feet subtracted from the right-hand sides. This is to compensate for the tee loss and valve loss that is at the beginning of the sub. Most subs are connected to the main lateral or another sub by means of a tee. A hydraulic loss occurs when the water changes direction to go to the sub. Since the pipe sizes are unknown, the tee loss and valve loss cannot be computed at this time, but are estimated to be 1.5 feet. Experience shows this to be an adequate approximation for this type problem. Also, the unknown variables X_i, i = 1 to 40, must be greater than or equal to zero.

Problem Solution

Formulating the objective function and the constraint equations and inequalities for the eight reaches of the pipe distribution system in figure 1 might be considered a fairly easy task, but the formulation of a large distribution system with hundreds of reaches would be very cumbersome. After the dynamic programming solution for a pipe distribution is obtained, program LAKMLP generates the linear programming formulation for the problem and creates a data file, LP1. This data file is the input file for the linear programming routine APEX, which is available on the Control Data Corporation 6400/6600 computer, model CYBER 74-28. Most of the computer manufacturers have linear programming routines available as software accessories to their computers. Since APEX was readily accessible, it was used in solving this linear programming problem.

The APEX routine has a number of optional forms in which the output may appear. The form of output that was selected is a binary file, LPANS, with the answers printed in a format that can be easily read by another computer program. The program LPFTN, which is listed in the appendix, was written to read this binary file and create another file, FTANS.

FTANS contains the value of the objective function and the values of each of the unknown variables. At the time the program LAKMLP executed, it created a file, ANS, which contains the geometric layout and properties of the pipe distribution system in figure 1.

Program LEOANS also appears in the appendix and was written to read the two files FTANS and ANS, and then generate output in a final report format.

This final report is in the appendix and in many ways is similar to the output that was generated by the dynamic programming solution. However, there are some differences such as the additional two stations 3671.11 and 5665.00 on the Main lateral. These are known as taper stations, since two pipe diameters were chosen for a single reach between the two original stations.

On each lateral there is at least one HGL that is equal the corresponding HGR at the same station. These occurrences have been underlined to emphasize that these two are equal. At the beginning of each sub, the loss for the velocity head, tee loss, and valve loss have been subtracted.

These final report sheets are as the name implies, and require no further manipulation except for some special cases. The adjustments for any special problem that arises could be manually accomplished with very little effort.

The total cost of pipe for the pipe distribution system shown in figure 1 is the last printed line on the sheet titled SUB2. This total cost was obtained from the linear programming solution, and is the value of the objective function.

PUMP PROBLEM

The total cost of pipe and the cost of pumping must be minimized simultaneously to obtain the minimum total cost of a pumped pipe distribution system. In other words, the pump problem is the same as the gravity problem with the addition of a term describing the total pump cost.

Problem Description

Consider the pipe system and associated data as shown in figure 2. The pump pipe distribution system resembles the gravity system with the exception of the pumping plant and tank. The pumping plant is required to pump water into the tank at sufficient elevation to serve all of the deliveries. Now the problem is to solve for this elevation and at the same time minimize the cost of pumping and the total cost of pipe.

Dynamic Programming Formulation

Mandry developed an expression that relates the cost of pumping and the cost of pipe to the pipe diameter. He took the partial derivative of the present worth to find the economical pipe diameter, as follows:

$$W = a_n N \, \Psi \, \phi \, Q \frac{Kq^2}{D^{5.25}} + yD^x \quad (18)$$

where: W = present worth

a_n = present worth factor based on some number of years at a specified interest rate

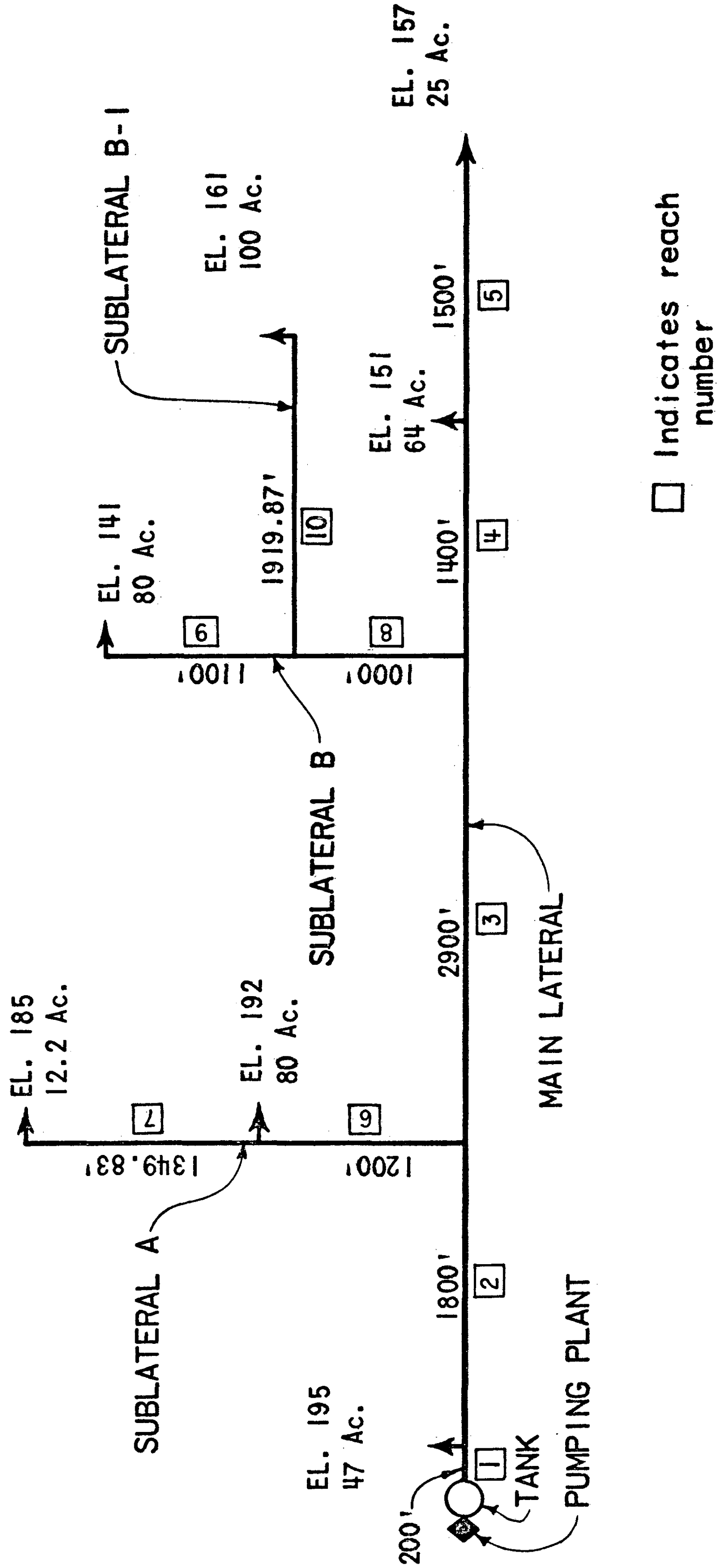

EL. 157
25 Ac.
SUBLATERAL B-1
EL. 161
100 Ac.
1500'
5
EL. 151
64 Ac.
Indicates reach number
EL. 141
80 Ac.
1919.87'
10
1400'
4
9
8
1100'
1000'
SUBLATERAL B
MAIN LATERAL
2900'
3
EL. 185
12.2 Ac.
EL. 192
80 Ac.
7
6
1349.83'
1200'
SUBLATERAL A
1800'
2
EL. 195
47 Ac.
1
TANK
PUMPING PLANT
200'

FIGURE 2

N = number of kilowatt-hours of energy necessary to lift 1 cubic foot per second 1 foot

Ψ = equivalent number of peak periods per year

φ = cost of power in mills per kilowatt-hour

K = $1/(13{,}840Cs^2)$ based on Scobey's formula

Q = total flow in cubic feet per second that is to be pumped

q = flow in a particular reach

x = slope of the pipe cost curve

y = cost of 1 linear foot of 12-inch-diameter pipe

D = pipe diameter in feet

For any particular pipe distribution system, $C = a_n N \Psi \phi Q Kq^2$ is a constant.

Then,

$$W = CD^{-5.25} + yD^x \tag{19}$$

Taking the partial derivative of present worth, W, with respect to D results in:

$$\frac{\partial W}{\partial D} = -5.25\ C\ D^{-6.25} + x\ yD^{x-1} \tag{20}$$

Setting $\frac{\partial W}{\partial D} = 0$, for minimum present worth yields

$$5.25\ C\ D^{-6.25} = x\ yD^{x-1} \tag{21}$$

$$5.25\ C = x\ yD^{x+5.25}$$

and

$$D\ (\text{economical}) = \left(\frac{5.25\ C}{x\ y} \right)^{\frac{1}{x + 5.25}} \qquad (22)$$

Beveridge and Schechter (10) presented another method, geometric programming, to obtain this relationship.

$$\text{Min } W(D) = C\ D^{-5.25} + y\ D^{x} \qquad (23)$$

$$\delta_1 + \delta_2 = 1$$

$$(D) \quad -5.25\ \delta_1 + x\ \delta_2 = 0$$

$$-5.25\ (1 - \delta_2) + x\ \delta_2 = 0$$

$$-5.25 + 5.25\ \delta_2 + x\ \delta_2 = 0$$

$$\delta_2\ (5.25 + X) = 5.25$$

$$\delta_2 = \frac{5.25}{5.25 + x}$$

$$\delta_1 = 1 \quad \frac{5.25}{5.25 + x} = \frac{x}{5.25 + x}$$

This predual at optimality is:

$$\frac{C\ D^{-5.25}}{\frac{x}{5.25 + x}} = \frac{y\ D^{x}}{\frac{5.25}{5.25 + x}} \qquad (24)$$

$$D^{x} + 5.25 = \frac{5.25\ C}{x\ y}$$

$$D\ (\text{economical}) = \left(\frac{5.25\ C}{x\ y}\right)^{\frac{1}{x + 5.25}} \qquad (25)$$

This expression for the economical pipe diameter is the same as obtained earlier in equation (22).

The geometric programming solution was presented at this time to provide some additional valuable information. For the pipe distribution system in figure 2, say that x = 1.53.

Then,

$$\delta_1 = \frac{x}{5.25 + x} = .23$$

$$\delta_2 = \frac{5.25}{5.25 + x} = .77$$

This means that the term $C\ D^{-5.25}$, the cost of pumping, should be 23 percent of the total cost, and the term $y\ D^{x}$, the cost of pipe, should be 77 percent of the total cost of the pipe distribution system. These percentages are for an optimal design for a pumped pipe distribution system, and comparisons of the actual cost for pumping and pipe should be made after the project has been constructed.

Now, to determine the unknown water surface in the tank, the formula for the economical pipe must be used in a systematic manner. For each of the reaches, the pipe diameter can be determined by the formula that was derived earlier, since for a specific project, the formula

$$D\ (\text{economical}) = \left(\frac{5.25\ C}{x\ y} \right)^{\frac{1}{x\ +\ 5.25}}$$

has only the pipe diameter unknown.

First the pipe diameter in the last reach of the main is computed. Then the friction loss is calculated for this reach using Scobey's formula. Next, the hydraulic grade line for the beginning of the reach is obtained by adding the friction loss to the elevation required at the end of the reach. This procedure is repeated until a junction point is reached where a sublateral joins the main line. Then the same procedure that was used on the main to obtain the hydraulic grade line at this junction should be applied to the sublateral. Next, the hydraulic grade line computed on the sub is compared to the hydraulic grade line on the main. The larger of these two hydraulic grade lines is then selected at the next starting point and the process continues until the tank is reached. At that point, the last hydraulic grade line found is the required water surface in the tank.

This method proposed by Mandry, proceeds through a pipe network until the tank water surface has been obtained, but in doing so, has not economically sized the pipe on the reaches that had the smaller hydraulic grade lines. But after the water surface in the tank has been established, the pump problem becomes a gravity problem with a known water surface equal to the water surface elevation in the tank. Then the gravity formulation for dynamic programming presented earlier in this thesis can be applied to the problem.

Problem Solution

The pumped pipe distribution problem in figure 2 was solved using the computer program LAKMLP. In Mandry's paper, the determination of the various pumping cost factors was presented and will not be reproduced in this thesis. The following values were used to obtain the solution for the example problem in figure 2:

$$x = 1.53 \qquad \Psi = 2.3675$$

$$y = 7.1 \qquad \phi = 0.006$$

$$N = 45.0 \qquad a_n = 30.525$$

The data file, DATA2, for the pumped pipe problem is in the appendix.

The pumped problem solution also appears in the appendix and is very similar in appearance to the solution for the gravity problem. The major difference is the tank water surface El. 231 which is printed near the top of the main line page. The remaining headings and symbols have the same meanings as for the gravity problem.

Linear Programming Formulation

Once the tank water surface elevation is known for a pumped pipe distribution system, determining the pipe sizes to obtain the minimum total pipe cost becomes a gravity problem. Since the linear programming formulation for a gravity problem has been presented earlier in this thesis, it will not be repeated.

Problem Solution

Program LAKMLP was used to determine the tank water surface elevation 231.00 for the pumped pipe distribution system in figure 2. As was done earlier for the gravity problem, program LAKMLP generated the input file for the linear programming package APEX. The binary file created by APEX was read by the program LPFTN, which produced the file FTANS. The file ANS created earlier by LAKMLP and the file FTANS were used as input files by the program LEOANS to generate the final report.

The final report for the pumped pipe distribution system is in the appendix, and is very similar to the final report for the gravity distribution system. The Main Lateral is printed first and the tank water surface is near the top of the page. Taper stations were computed between stations 200 and 2000 and between stations 4900 and 6300. It is interesting to note that on Sublateral B there is not an HGL (hydraulic grade line) that is equal to an HGR (required elevation). This is because Sublateral B-1 requires a higher hydraulic grade line at station 1000, and the minimum allowable pipe size of 6 inches was placed in the last reach of Sublateral B. This output is in final form, except for special cases, and does not require any further modifications.

CONCLUSION

It has been shown that the optimal pipe sizes for two types of pipe distribution systems can be obtained and printed in a final report format. This final report is usable to those familiar with civil engineering terminology, but unfamiliar with linear programming and dynamic programming techniques, and overcomes the major obstacle to the use of those techniques. Using a dynamic programming computer program to generate the input file for a linear programming package is a new concept, and should be considered in obtaining the solution to other optimization problems.

The design of other types of pipe systems, such as "loop" systems, requires modifying the techniques described in this thesis. A loop system, as the name implies, is a pipe network which consists of piping that permits the water to flow in more than one direction. Normally, these loop systems are not encountered in irrigation pipe distribution systems, but are often found in municipal water networks. A problem of this type has been optimized with a revised version of the techniques presented in this thesis.

APPENDIX

**** PROGRAM LAKMLP ****

```
      PROGRAM LAKMLP(HDATA,OUTPUT,PICOST,TAPE5=HDATA,TAPE6=OUTPUT,
     1          TAPE3=PICOST,TAPE1,TAPE7,TAPE2)
C     PROGRAM NAME  LAKMLP
C
C     PROGRAM COMPUTES ECONOMIC PIPE SIZES FOR A GRAVITY OR A PUMP
C
C      PIPELINE DISTRIBUTION SYSTEM AND THE ASSOCIATED QUANTITIES.
C
C      ECONOMIC COMPUTATIONS ARE BASED ON THE ASCE PAPER - DESIGN OF
C
C      PIPE DISTRIBUTION SYSTEMS FOR SPRINKLER PROJECTS - BY JAMES E.
C
C      MANDRY  SEPTEMBER 1967 ,
C
C      AND THE APEX I LP PACKAGE (CDC CYBER 74-28).
C
C     AUTHOR  KINNEY    CODE  224
C
C         TAPE1      INPUT FOR APEX
C         TAPE2      DUMMY FILE
C         TAPE3      PIPE COST DATA FILE
C         TAPE5      JOB DATA FILE
C         TAPE6      OUTPUT FOR LINE PRINTER
C         TAPE7      INPUT FOR FINAL ANSWER
C
C
      COMMON NN,MDIA( 650),GL( 650),MHC( 650),SEXCAV(300),SBKFL(300),
     1       ZG,PIPCOV,HVH,PIPMIN,NO,GDEL( 600),JHC( 600),LASAV,TKHSEL,
     2       CSS,CSB,SJEXC,SJBKF,MDEL,JW
      COMMON /LPPACK/  DP(2000),VP(2000),HP(2000),HVP(2000),
     1          COSTPI(2000),C(30,40),RL(400),GZ(400,16),RHS(400),
     2           HCC(2000),IC
      DIMENSION LA1( 36),STA1( 36),AC1( 36),L1( 36),SAC1( 36),Q1( 36),
     1 HGR1( 36),HL1( 36),DIA1( 36),HF1( 36),HGL1( 36),GS1( 36),
     1 LAT1(3),DQ1( 36),DEL1( 36),EQ1( 36),IR1( 36),
     2          LA2( 93),STA2( 93),AC2( 93),L2( 93),SAC2( 93),Q2( 93),
     2 HGR2( 93),HL2( 93),DIA2( 93),HF2( 93),HGL2( 93),GS2( 93),
     2 LAT2( 36,2),DQ2( 93),DEL2( 93),EQ2( 93),IR2( 93),
     3          LA3(180),STA3(180),AC3(180),L3(180),SAC3(180),Q3(180),
     3 HGR3(180),HL3(180),DIA3(180),HF3(180),HGL3(180),GS3(180),
     3 LAT3( 93,2),DQ3(180),DEL3(180),EQ3(180),IR3(180),
     4          LA4(150),STA4(150),AC4(150),L4(150),SAC4(150),Q4(150),
     4 HGR4(150),HL4(150),DIA4(150),HF4(150),HGL4(150),GS4(150),
     4 LAT4(180,2),DQ4(150),DEL4(150),EQ4(150),IR4(150),
     5          LA5( 50),STA5( 50),AC5( 50),L5( 50),SAC5( 50),Q5( 50),
     5 HGR5( 50),HL5( 50),DIA5( 50),HF5( 50),HGL5( 50),GS5( 50),
     5 LAT5(150,2),DQ5( 50),DEL5( 50),EQ5( 50),IR5( 50),
     6 TEELOS2(93),TEELOS3(180),TEELOS4(150),TEELOS5(50),
```

**** PROGRAM LAKMLP ****

```
      6 TELOS2(93),TELOS3(180),TELOS4(150),TELOS5(50),
      6 IVALV2(93),IVALV3(180),IVALV4(150),IVALV5(50)
       REAL L1,L2,L3,L4,L5
       INTEGER SENTRY
       CALL DATE(TODAY)
       WRITE (6,1)
     1 FORMAT (1H1)
       IMAX=36
       JMAX=93
       KMAX=180
       LMAX=150
       MMAX=50
       PI=3.14159
       JW=2
    10 READ (5,20) TEELOS,PIPCOV,DWSADD,X,Y,CSS,CSB,PIPMIN,ITAP
    20 FORMAT (2F5.2,6F10.2,I10)
C
C         TEELOS - TEE LOSS FOR SUB - IF BLANK,TEELOS=2.0
C         PIPCOV - AVE. EARTH COVER OVER PIPE
C         DWSADD - AMOUNT TO ADD TO OBTAIN DESIRED DELIVERY WATER SURFACE
C         X AND Y - PIPE COST FACTORS
C         CSS - SCOBEY CS FOR SMALL PIPE (LESS THAN 23 INCHES)
C         CSB - SCOBEY CS FOR BIG PIPE (GREATER THAN 23 INCHES)
C         PIPMIN - MINIMUM PIPE SIZE - USUALLY 4-INCH
C
       IF (ITAP.NE.0)  JW=6
       K2 = 0
       K3 = 0
       K4 = 0
       K5 = 0
       NO=0
       NN=0
       SJEXC=0.
       SJBKF=0.
       IF (TEELOS.NE.0.0) GO TO 30
       TEELOS=2.0
    30 READ (5,40) MDEL,SENTRY,ITYPE,TQP,CP,AVH1,HVH1,TKWSEL,GDIFF,INPUOT
    40 FORMAT (2I5,I10,5F10.2,F5.2,I5)
C
C         MDEL - 1 TO READ IN DELIVERY SIZE - OTHERWISE BLANK OR ZERO
C         SENTRY - 9 TO START JOB OR LATERAL -- 3 TO COMPUTE QUANTITIES
C         ITYPE - TYPE OF SYSTEM
C             1 - GRAVITY FROM CANAL
C             2 - PUMP INTO ELEVATED TANK NEAR PUMPING PLANT
C             3 - PUMP INTO RES. OR REG. TANK WITH KNOWN WATER SURFACE
C             4 - PUMP INTO RES. OR REG. TANK WITH UNKNOWN WATER SURFACE
C         TQP - TOTAL Q FOR EACH LATERAL SYSTEM
C         CP - PUMP FACTOR
```

**** PROGRAM LAKMLP ****

```
C         AVH1 - AVAILABLE HEAD FOR MAIN LATERAL
C         HVH1 - MAXIMUM ELEVATION FOR PIPE CLASS WITHOUT WATERHAMMER
C         TKWSEL - KNOWN WATER SURFACE IN AFTERBAY TANK OR RESERVOIR
C         GDIFF - GRAVITY DIFFERENCE - IF BLANK,GDIFF=1.0
C         INPUOT - 1-FOR INTERMEDIATE PUMP CONTROLS   BLANK OR 0 FOR NONE
C
      WRITE (6,1)
      ZP=1./(X+5.25)
      ZG=0.38*X /(1.+0.19*X)
      LASAV=0
      ICOUNT=0
      JCOUNT=0
      KCOUNT=0
      J=0
      JJ=0
      K=0
      KK=0
      L=0
      LL=0
      M=0
      MM=0
      IG=0
      IF (SENTRY.NE.9 .AND. SENTRY.NE.8) GO TO 9999
      IF (SENTRY.EQ.8) GO TO 9999
      IF (CP.NE. 0.0) GO TO 50
      CP=.48
   50 QO=(TQP**ZP)*CP*12.
      IF (GDIFF.NE.0.0) GO TO 1000
      GDIFF=1.0
C
C         READ IN MAIN LATERAL
C                SPACES 2 THROUGH 30
C
 1000 READ (5,1010) (LAT1(II),II=1,3),IDUM
 1010 FORMAT (3A10,I10)
      IDUM=IDUM+1
      I=0
 1020 I=I+1
      READ (5,1030) DEL1(I),LA1(I),STA1(I),AC1(I),HGR1(I),GS1(I),EQ1(I)
                    HGR1(I)=HGR1(I) + DWSADD
 1030 FORMAT (F5.2,I5,5F10.3)
C
C         DEL1(I) - DELIVERY SIZE AT STA1(I)
C         LA1(I) - SIGNAL  1  FOR STATION ON MAIN EXCEPT SUB,THEN 2
C         STA1(I) - ACTUAL STATION ON MAIN
C         AC1(I) - ACRES FOR TURNOUT OFF OF MAIN
C         HGR1(I) - HYDRAULIC GRADE LINE REQUIRED FOR TURNOUT
C                ENTER .09 FOR HGR1(I) TO LOCATE A TANK AT STA1(I)
```

**** PROGRAM LAKMLP ****

```
C          GS1(I) - NATURAL GROUND SURFACE ON MAIN
C          EQ1(I) - NUMERICAL VALUE OF EQUATION DOWNSTREAM OF STA1(I)
C
        IF (HGR1(I).NE.(DWSADD+.09)) GO TO 1035
        LASAV=I
        STASAV=STA1(I)
        GO TO (1035,1035,1032,1032) ,ITYPE
 1032   HGR1(LASAV)=TKWSEL
 1035   IF (I.GT.IMAX) GO TO 7100
        IF (LA1(I).EQ.2) GO TO 2000
        IF (LA1(I).EQ.0) GO TO 1040
        GO TO 1020
 1040   N1=I-1
        N1L=N1+1
        SAC1(N1+1)=0.0
        DO 1050 I=1,N1
        L1(I)=STA1(I+1)-STA1(I)+EQ1(I)
        N1I=N1+1-I
        SAC1(N1I)=SAC1(N1I+1)+AC1(N1I+1)
 1050   CONTINUE
        CALL SACQ   (1,SAC1,AC1,Q1,DQ1,N1)
C       IF (ABS (Q1(1)-TQP) .GE. 1.0) GO TO 7800
        IF (ITYPE.NE.1) GO TO 1120
        IF (IG.EQ.0) GO TO 1120
        I=N1
 1090   IG=I+1
 1100   DO 1110 I=IG,N1
        IF (LA1(I).EQ.2) GO TO 2060
 1110   CONTINUE
 1120   DO 1190 I=1,N1
        D1=QQ * (Q1(I)**(ZP*2.))
        IF (D1.GT.12.) GO TO 1130
        JX=(D1/2.) +1.
        DIA1(I)=JX*2
        GO TO 1150
 1130   IF (D1.GT.54.) GO TO 1140
        JX=(D1/3.) +1.
        DIA1(I)=JX*3
        GO TO 1150
 1140   JX=(D1/6.) +1.
        DIA1(I)=JX*6
 1150   IF (DIA1(I)-22.) 1160,1160,1170
 1160   CS=CSS
        GO TO 1180
 1170   CS=CSB
 1180   HL1(I)=(((576.*Q1(I)) / (PI*CS*DIA1(I)**2.625))**2.) /1000.
        HF1(I)=L1(I)*HL1(I)
 1190   CONTINUE
```

**** PROGRAM LAKMLP ****

```
      HGL1(N1+1)=HGR1(N1+1)
      DO 1200 I=1,N1
      N1I=N1+1-I
      HGL1(N1I)=HF1(N1I) + HGL1(N1I+1)
      IF (HGL1(N1I) .GE. HGR1(N1I)) GO TO 1200
      HGL1(N1I)=HGR1(N1I)
      IF (ITYPE.NE.4) GO TO 1200
      IF (HGR1(N1I).LE.TKWSEL) GO TO 1200
      TKWSEL=HGR1(N1I)
 1200 CONTINUE
      IF (ITYPE.NE.4) GO TO 1205
      IF (HGL1(LASAV).GT.TKWSEL) GO TO 1201
      IF (TKWSEL.NE.0.0) GO TO 1202
 1201 TKWSEL=IFIX(HGL1(LASAV)+1.0)
      GO TO 1205
 1202 TKWSEL=IFIX(TKWSEL +1.5)
 1205 IF (INPUOT .NE. 1) GO TO 1240
      WRITE (6,1210)
 1210 FORMAT (1H1,100H   STATION      ACRES      HGL          HGR       LENGTH
     1   SAC              Q          DIA          HL           HF    /)
      WRITE (6,1010) (LAT1(II),II=1,3)
      DO 1230 I=1,N1
      WRITE (6,1220) STA1(I),AC1(I),HGL1(I),HGR1(I),L1(I),SAC1(I),Q1(I),
     1  DIA1(I),HL1(I),HF1(I)
 1220 FORMAT (4F10.2 / 40X,4F10.2,F10.6,F10.2)
 1230 CONTINUE
      WRITE (6,1220) STA1(N1+1),AC1(N1+1),HGL1(N1+1),HGR1(N1+1)
 1240 CONTINUE
      N2=0
      N3=0
      N4=0
      N5=0
      JY=1
      KY=1
      LY=1
      MY=1
      K2=0
      K3=0
      K4=0
      K5=0
      PZC=.00007226 / CSB**2.
      GO TO (1300,1400,1500,1500) , ITYPE
C
C        ITYPE = 1
C
 1300 WRITE (JW,1310) CP,HGL1(1),AVH1,ICOUNT,JCOUNT,KCOUNT
 1310 FORMAT ( 4H CP=,F10.4,10X, 10H HGL1(1)= ,F10.2,10X,  7H AVH1= ,
     1 F10.2,3I10)
```

**** PROGRAM LAKMLP ****

```
      HVH=HVH1
      IF (ABS (HGL1(1)-AVH1) .LE. GDIFF) GO TO 1510
      IF (KCOUNT.EQ.2) GO TO 1510
      IF (AVH1.LT.HGL1(1)) GO TO 1320
      ICOUNT=ICOUNT + 1
      IF (ICOUNT.EQ.11) GO TO 7600
      CP=CP-.01
      IF (JCOUNT.GT.0) GO TO 1340
      GO TO 1350
1320  JCOUNT=JCOUNT + 1
      IF (JCOUNT.EQ.11) GO TO 7700
      CP=CP+.01
      IF (ICOUNT.GT.0) GO TO 1330
      GO TO 1350
1330  CP=CP-.01
1340  KCOUNT=KCOUNT + 1
      CP=CP / (AVH1 / HGL1(1))
1350  QQ=(TQP **ZP) *CP *12.
      I=0
      JG=1
      KG=1
      LG=1
      MG=1
      GO TO 1090
C
C
C        ITYPE = 2
C
1400  AVH1=IFIX(HGL1(1)+1.0)
      HVH =AVH1 + 5.0
      GO TO 1510
C
C        ITYPE = 3   AND   ITYPE = 4
C
1500  HGL1(LASAV)=TKWSEL
      DO 1505 I=2,LASAV
      NNI=LASAV + 1 - I
      HGL1(NNI)=HF1(NNI) + HGL1(NNI+1)
1505  CONTINUE
      AVH1=HGL1(1)
      HVH =0.0
1510  WRITE (JW,1520) (LAT1(II),II=1,3),TODAY
1520  FORMAT (1H1,35H    ECONOMICAL PIPE HYDRAULIC STUDY,30X,3A10,30X
     1,A10//)
      WRITE (7,1521) (LAT1(II),II=1,3),ITYPE,STA1(N1L),TEELOS,AVH1
1521  FORMAT (3A10,I10,3F10.2)
      GO TO (1530,1530,1550,1550) ,ITYPE
1530  WRITE (JW,1540) HVH
```

**** PROGRAM LAKMLP ****

```
1540  FORMAT (48X,30H PIPE CLASS DESIGN ELEVATION =,F10.2/)
      WRITE (7,1541) HVH,IDUM
1541  FORMAT (F10.2,I10)
      GO TO 1570
1550  WRITE (JW,1560)
1560  FORMAT(38X,50HPIPE CLASS DESIGN ELEVATION = HYDRAULIC GRADE LINE/
      WRITE (7,1565) IDUM
1565  FORMAT (I10)
1570  GO TO  (1650,1640,1580,1600) ,ITYPE
1580  WRITE (JW,1590) TKWSEL,STASAV
1590  FORMAT (28X,39H DOWN TO THE KNOWN TANK WATER SURFACE (,F10.2,
     1 9H ) AT STA,F10.2/)
      WRITE (7,1591) TKWSEL,STASAV
1591  FORMAT (2F10.2)
      GO TO 1620
1600  WRITE (JW,1610) TKWSEL,STASAV
1610  FORMAT (26X,42H DOWN TO THE COMPUTED TANK WATER SURFACE (,F10.2,
     1 9H ) AT STA,F10.2/)
      WRITE (7,1591) TKWSEL,STASAV
1620  IF (STASAV.EQ.STA1(NIL)) GO TO 1650
      WRITE (JW,1630) STASAV,TKWSEL
1630  FORMAT (25X,10H AFTER STA,F10.2,36H ,THE PIPE CLASS DESIGN ELEVATI
     1ON = ,F10.2, 10H + 5.0 FT./)
      GO TO 1660
1640  WRITE (JW,1650) AVH1
1650  FORMAT (28X,19H WATER SURFACE IS (,F10.2,37H ) IN THE TANK NEAR TH
     1E PUMPING PLANT/)
1660  WRITE (JW,1670)
1670  FORMAT  (* STATION            Q   HGL    DIA     V      HV     SLOPE
     1 LENGTH   LOSS  SUBLATERAL NAME      AC       SAC    DQ   DEL     HGR
     2  NGS       HC*//)
C
      CALL PIPSIZ (LA1,STA1,AC1,L1,SAC1,Q1,HGR1,GS1,AVH1,1,N1,LAT2,IMAX,
     1          IR1,DQ1,HGL1,DEL1,EQ1)
      SJEXC=SJEXC + SEXCAV(N1)
      SJBKF=SJBKF + SBKFL(N1)
      I=0
      HVH=HVH1
1680  I=I+1
      IF (I.GT.N1) GO TO 8000
      IF (LA1(I).NE.2) GO TO 1680
C
C        SUB
C
      K2=K2+1
      WRITE (JW,1690) (LAT2(I,II),II=1,2),TODAY
1690  FORMAT (1H1,35H     ECONOMICAL PIPE HYDRAULIC STUDY,30X,2A10,40X,
     2 A10//)
```

**** PROGRAM LAKMLP ****

```
      WRITE (7,1691) (LAT2(I,II),II=1,2),IVALV2(K2)
1691  FORMAT (2A10,I10)
      GO TO (1710,1700,1720,1720) ,ITYPE
1700  HVH=AVH1+5.0
1710  AVH2=HGL1(I)-TELOS2(K2)
      GO TO 1740
1720  IF (STA1(I).GT.STASAV) GO TO 1730
      AVH2=TKWSEL
      HVH=HGL1(I)
      GO TO 1740
1730  HVH=TKWSEL + 5.0
      AVH2=HGL1(I) -TELOS2(K2)
1740  WRITE (JW,1540) HVH
      WRITE (7,1541) HVH
      WRITE (JW,1670)
      JY=N2+1
      CALL PIPSIZ (LA2,STA2,AC2,L2,SAC2,Q2,HGR2,GS2,AVH2,JY,N2,LAT3,JMAX
     2        ,IR2,DQ2,HGL2,DEL2,EQ2)
      SJBKF=SJBKF + SBKFL(N2)
      SJEXC=SJEXC + SEXCAV(N2)
      N2=N2+1
      JB=0
1750  J=JB+JY
      JB=JB+1
      IF (J.GT.N2) GO TO 1680
      IF (LA2(J).NE.3) GO TO 1750
      K3=K3+1
      WRITE (JW,1690) (LAT3(J,II),II=1,2),TODAY
      WRITE (7,1691) (LAT3(J,II),II=1,2),IVALV3(K3)
      WRITE (JW,1540) HVH
      WRITE (7,1541) HVH
      WRITE (JW,1670)
      AVH3=HGL2(J)-TELOS3(K3)
      KY=N3+1
      CALL PIPSIZ (LA3,STA3,AC3,L3,SAC3,Q3,HGR3,GS3,AVH3,KY,N3,LAT4,KMAX
     3        ,IR3,DQ3,HGL3,DEL3,EQ3)
      SJBKF=SJBKF + SBKFL(N3)
      SJEXC=SJEXC + SEXCAV(N3)
      N3=N3+1
      KB=0
1760  K=KB+KY
      KB=KB+1
      IF (K.GT.N3) GO TO 1750
      IF (LA3(K).NE.4) GO TO 1760
      K4=K4+1
      WRITE (JW,1690) (LAT4(K,II),II=1,2),TODAY
      WRITE (7,1691) (LAT4(K,II),II=1,2),IVALV4(K4)
      WRITE (JW,1540) HVH
```

**** PROGRAM LAKMLP ****

```
      WRITE (7,1541) HVH
      WRITE (JW,1670)
      AVH4=HGL3(K)-TELOS4(K4)
      LY=N4+1
      CALL PIPSIZ (LA4,STA4,AC4,L4,SAC4,Q4,HGR4,GS4,AVH4,LY,N4,LAT5,LMAX
     4         ,IR4,DQ4,HGL4,DEL4,EQ4)
      SJBKF=SJBKF + SBKFL(N4)
      SJEXC=SJEXC + SEXCAV(N4)
      N4=N4+1
      LB=0
 1770 L=LB+LY
      LB=LB+1
      IF (L.GT.N4) GO TO 1760
      IF (LA4(L).NE.5) GO TO 1770
      K5=K5+1
      WRITE (JW,1690) (LAT5(L,II),II=1,2),TODAY
      WRITE (7,1691) (LAT5(L,II),II=1,2),IVALV5(K5)
      WRITE (JW,1540) HVH
      WRITE (7,1541) HVH
      WRITE (JW,1670)
      AVH5=HGL4(L)-TELOS5(K5)
      MY=N5+1
      CALL PIPSIZ (LA5,STA5,AC5,L5,SAC5,Q5,HGR5,GS5,AVH5,MY,N5,LAT6,MMAX
     5         ,IR5,DQ5,HGL5,DEL5,EQ5)
      SJBKF=SJBKF + SBKFL(N5)
      SJEXC=SJEXC + SEXCAV(N5)
      N5=N5+1
      GO TO 1770
C
C        READ IN SUB LATERAL DATA
C             SPACES 2 THROUGH 20
C
 2000 READ (5,2010) (LAT2(I,II),II=1,2),IVALVE,TLOSS,TOSS
 2010 FORMAT (2A10,I10,2F8.2)
      K2 = K2+1
      IVALV2(K2) = IVALVE
      IF (TLOSS.EQ.0.) GO TO 2015
      TEELOS2(K2) = TLOSS
      TELOS2(K2) = TOSS
      GO TO 2016
 2015 TEELOS2(K2) = TEELOS
      TELOS2(K2) = TEELOS
 2016 JJ=J+1
 2020 J=J+1
      READ (5,1030) DEL2(J),LA2(J),STA2(J),AC2(J),HGR2(J),GS2(J),EQ2(J)
                    HGR2(J)=HGR2(J) + DWSADD
C
C        LA2(J) - SIGNAL  2  FOR STATION ON SUB EXCEPT SUB SUB,THEN 3
```

**** PROGRAM LAKMLP ****

```
C
      IF (J.GT.JMAX) GO TO 7200
      IF (LA2(J).EQ.3) GO TO 3000
      IF (LA2(J).EQ.0) GO TO 2040
      GO TO 2020
2040  N2=J-1
      SAC2(N2+1)=0.0
      DO 2050 J=JJ,N2
      L2(J)=STA2(J+1)-STA2(J)+EQ2(J)
      N2J=N2+JJ-J
      SAC2(N2J)=SAC2(N2J+1)+AC2(N2J+1)
2050  CONTINUE
      CALL SACQ   (JJ,SAC2,AC2,Q2,DQ2,N2)
      IF (ITYPE.NE.1) GO TO 2120
      IF (IG.EQ.0) GO TO 2120
2060  K2=K2 +1
      DO 2070 J=JG,JMAX
      IF (LA2(J).EQ.0) GO TO 2080
2070  CONTINUE
      J=JMAX
2080  N2=J-1
      JGSAV=JG
      GO TO 2100
2090  JG=J+1
2100  DO 2110 J=JG,N2
      IF (LA2(J).EQ.3) GO TO 3060
2110  CONTINUE
      JJ=JGSAV
2120  DO 2130 J=JJ,N2
      D2=Q0 * (Q2(J)**(ZP*2.))
      IF (D2.GT.12.) GO TO 2130
      JX=(D2/2.) +1.
      DIA2(J)=JX*2
      GO TO 2150
2130  IF (D2.GT.54.) GO TO 2140
      JX=(D2/3.) +1.
      DIA2(J)=JX *3
      GO TO 2150
2140  JX=(D2/6.) +1.
      DIA2(J)=JX * 6
2150  IF (DIA2(J)-22.) 2160,2160,2170
2160  CS=CSS
      GO TO 2180
2170  CS=CSB
2180  HL2(J)=(((576.*Q2(J)) / (PI*CS*DIA2(J)**2.625))**2.) /1000.
      HF2(J)=L2(J)*HL2(J)
2190  CONTINUE
      HGL2(N2+1)=HGR2(N2+1)
```

**** PROGRAM LAKMLP ****

```
      DO 2200 J=JJ,N2
      N2J=N2+JJ-J
      HGL2(N2J)=HF2(N2J) + HGL2(N2J+1)
      IF (HGL2(N2J) .GE. HGR2(N2J)) GO TO 2200
      HGL2(N2J)=HGR2(N2J)
 2200 CONTINUE
      IF (INPUOT  .NE. 1) GO TO 2240
      WRITE (6,2210)
 2210 FORMAT ( // 100H    STATION      ACRES      HGL        HGR      LENGTH
     1   SAC          Q         DIA        HL         HF    /)
      WRITE (6,2010) (LAT2(I,II),II=1,2)
      DO 2230 J=JJ,N2
      WRITE (6,1220) STA2(J),AC2(J),HGL2(J),HGR2(J),L2(J),SAC2(J),Q2(J),
     2  DIA2(J),HL2(J),HF2(J)
 2230 CONTINUE
      WRITE (6,1220) STA2(N2+1),AC2(N2+1),HGL2(N2+1),HGR2(N2+1)
 2240 CONTINUE
C
C        COMPUTE TANK WATER SURFACE ELEVATION
C
      IF (ITYPE.NE.4) GO TO 2260
      IF (LASAV.EQ.0) GO TO 2250
      IF (I.GE.LASAV) GO TO 2260
 2250 IF (HGL2(JJ).LT.TKWSEL) GO TO 2260
      TKWSEL = HGL2(JJ) + TELOS2(K2)
 2260 HGR1(I)=HGL2(JJ) + TELOS2(K2)
      AC1(I)=SAC2(JJ)
      IF (ITYPE.NE.1) GO TO 1020
      IF (IG.EQ. 0) GO TO 1020
      JG=N2+2
      GO TO 1090
C
C        READ IN SUB SUB LATERAL DATA
C             SPACES 2 THROUGH 20
C
 3000 READ (5,2010) (LAT3(J,II),II=1,2),IVALVE,TLOSS,TOSS
      K3 = K3+1
      IVALV3(K3) = IVALVE
      IF (TLOSS.EQ.0.) GO TO 3015
      TEELOS3(K3) = TLOSS
      TELOS3(K3) = TOSS
      GO TO 3016
 3015 TEELOS3(K3) = 2.*TEELOS
      TELOS3(K3) = TEELOS
 3016 KK=K+1
 3020 K=K+1
      READ (5,1030) DEL3(K),LA3(K),STA3(K),AC3(K),HGR3(K),GS3(K),EQ3(K)
                          HGR3(K)=HGR3(K) + DWSADD
```

**** PROGRAM LAKMLP ****

```
C
C        LA3(K) - SIGNAL  3  FOR STATION ON SUB SUB EXCEPT SSS,THEN 4
C
      IF (K.GT.KMAX) GO TO 7300
      IF (LA3(K).EQ.4) GO TO 4000
      IF (LA3(K).EQ.0) GO TO 3040
      GO TO 3020
3040  N3=K-1
      SAC3(N3+1)=0.0
      DO 3050 K=KK,N3
      L3(K)=STA3(K+1)-STA3(K)+EQ3(K)
      N3K=N3+KK-K
      SAC3(N3K)=SAC3(N3K+1) + AC3(N3K+1)
3050  CONTINUE
      CALL SACQ   (KK,SAC3,AC3,Q3,DQ3,N3)
      IF (ITYPE.NE.1) GO TO 3120
      IF (IG.EQ.0) GO TO 3120
3060  K3=K3+1
      DO 3070 K=KG,KMAX
      IF (LA3(K).EQ.0) GO TO 3080
3070  CONTINUE
      K=KMAX
3080  N3=K-1
      KGSAV=KG
      GO TO 3100
3090  KG=K+1
3100  DO 3110 K=KG,N3
      IF (LA3(K).EQ.4) GO TO 4060
3110  CONTINUE
      KK=KGSAV
3120  DO 3190 K=KK,N3
      D3=DQ * (Q3(K)**(ZP*2.))
      IF (D3.GT.12.) GO TO 3130
      JX=(D3/2.) +1.
      DIA3(K)=JX*2
      GO TO 3150
3130  IF (D3.GT.54.) GO TO 3140
      JX=(D3/3.) +1.
      DIA3(K)=JX*3
      GO TO 3150
3140  JX=(D3/6.) +1.
      DIA3(K)=JX*6
3150  IF (DIA3(K)-22.) 3160,3160,3170
3160  CS=CSS
      GO TO 3180
3170  CS=CSB
3180  HL3(K)=(((576.*Q3(K)) / (PI*CS*DIA3(K)**2.625))**2.) /1000.
      HF3(K)=L3(K)*HL3(K)
```

**** PROGRAM LAKMLP ****

```
 3190   CONTINUE
        HGL3(N3+1)=HGR3(N3+1)
        DO 3200 K=KK,N3
        N3K=N3+KK-K
        HGL3(N3K)=HF3(N3K) + HGL3(N3K+1)
        IF (HGL3(N3K).GE. HGR3(N3K)) GO TO 3200
        HGL3(N3K)=HGR3(N3K)
 3200   CONTINUE
        IF (INPUOT .NE. 1) GO TO 3240
        WRITE (6,2210)
        WRITE (6,2010) (LAT3(J,II),II=1,2)
        DO 3230 K=KK,N3
        WRITE (6,1220) STA3(K),AC3(K),HGL3(K),HGR3(K),L3(K),SAC3(K),Q3(K),
       3 DIA3(K),HL3(K),HF3(K)
 3230   CONTINUE
        WRITE (6,1220) STA3(N3+1),AC3(N3+1),HGL3(N3+1),HGR3(N3+1)
 3240   CONTINUE
        HGR2(J)=HGL3(KK) +TELOS3(K3)
        AC2(J)=SAC3(KK)
        IF (ITYPE.NE.1) GO TO 2020
        IF (IG.EQ. 0) GO TO 2020
        KG=N3+2
        GO TO 2090
 C
 C         READ IN SUB SUB SUB LATERAL DATA
 C                SPACES 2 THROUGH 20
 C
 4000   READ (5,2010) (LAT4(K,II),II=1,2),IVALVE,TLOSS,TOSS
        K4 = K4+1
        IVALV4(K4) = IVALVE
        IF (TLOSS.EQ.0.) GO TO 4015
        TEELOS4(K4) = TLOSS
        TELOS4(K4) = TOSS
        GO TO 4016
 4015   TEELOS4(K4) = 3.*TEELOS
        TELOS4(K4) = TEELOS
 4016   LL=L+1
 4020   L=L+1
        READ (5,1030) DEL4(L),LA4(L),STA4(L),AC4(L),HGR4(L),GS4(L),EQ4(L)
                      HGR4(L)=HGR4(L) + DWSADD
 C
 C         LA4(L) - SIGNAL  4  FOR STATION ON SUBSUBSUB EXCEPT SSSS,THEN 5
 C
        IF (L.GT.LMAX) GO TO 7400
        IF (LA4(L).EQ.5) GO TO 5000
        IF (LA4(L).EQ.0) GO TO 4040
        GO TO 4020
 4040   N4=L-1
```

**** PROGRAM LAKMLP ****

```
      SAC4(N4+1)=0.0
      DO 4050 L=LL,N4
      L4(L)=STA4(L+1)-STA4(L)+EQ4(L)
      N4L=N4+LL-L
      SAC4(N4L)=SAC4(N4L+1) + AC4(N4L+1)
4050  CONTINUE
      CALL SACQ   (LL,SAC4,AC4,Q4,DQ4,N4)
      IF (ITYPE.NE.1) GO TO 4120
      IF (IG.EQ.0) GO TO 4120
4060  K4=K4+1
      DO 4070 L=LG,LMAX
      IF (LA4(L).EQ.0) GO TO 4080
4070  CONTINUE
      L=LMAX
4080  N4=L-1
      LGSAV=LG
      GO TO 4100
4090  LG=L+1
4100  DO 4110 L=LG,N4
      IF (LA4(L).EQ.5) GO TO 5060
4110  CONTINUE
      LL=LGSAV
4120  DO 4190 L=LL,N4
      D4=DQ * (Q4(L)**(ZP*2.))
      IF (D4.GT.12.) GO TO 4130
      JX=(D4/2.) +1.
      DIA4(L)=JX*2
      GO TO 4150
4130  IF (D4.GT.54.) GO TO 4140
      JX=(D4/3.) +1.
      DIA4(L)=JX*3
      GO TO 4150
4140  JX=(D4/6.) +1.
      DIA4(L)=JX*6
4150  IF (DIA4(L)-22.) 4160,4160,4170
4160  CS=CSS
      GO TO 4180
4170  CS=CS3
4180  HL4(L)=(((576.*Q4(L)) / (PI*CS*DIA4(L)**2.625))**2.) /1000.
      HF4(L)=L4(L)*HL4(L)
4190  CONTINUE
      HGL4(N4+1)=HGR4(N4+1)
      DO 4200 L=LL,N4
      N4L=N4+LL-L
      HGL4(N4L)=HF4(N4L) + HGL4(N4L+1)
      IF (HGL4(N4L).GE. HGR4(N4L)) GO TO 4200
      HGL4(N4L)=HGR4(N4L)
4200  CONTINUE
```

**** PROGRAM LAKMLP ****

```
      IF (INPUOT .NE. 1) GO TO 4240
      WRITE (6,2210)
      WRITE (6,2010) (LAT4(K,II),II=1,2)
      DO 4230 L=LL,N4
      WRITE (6,1220) STA4(L),AC4(L),HGL4(L),HGR4(L),L4(L),SAC4(L),Q4(L),
     4  DIA4(L),HL4(L),HF4(L)
4230  CONTINUE
      WRITE (6,1220) STA4(N4+1),AC4(N4+1),HGL4(N4+1),HGR4(N4+1)
4240  CONTINUE
      HGR3(K)=HGL4(LL) +TELOS4(K4)
      AC3(K)=SAC4(LL)
      IF (ITYPE.NE.1) GO TO 3020
      IF (IG.EQ. 0) GO TO 3020
      LG=N4+2
      GO TO 3090
C
C        READ IN SUB SUB SUB SUB LATERAL DATA
C               SPACES 2 THROUGH 20
C
5000  READ (5,2010) (LAT5(L,II),II=1,2),IVALVE,TLOSS,TOSS
      K5 = K5+1
      IVALV5(K5) = IVALVE
      IF (TLOSS.EQ.0.) GO TO 5015
      TEELOS5(K5) = TLOSS
      TELOS5(K5) = TOSS
      GO TO 5016
5015  TEELOS5(K5) = 4.*TEELOS
      TELOS5(K5) = TEELOS
5016  MM=M+1
5020  M=M+1
      READ (5,1030) DEL5(M),LA5(M),STA5(M),AC5(M),HGR5(M),GS5(M),EQ5(M)
                         HGR5(M)=HGR5(M) + DWSADD
C
C        LA5(M) - SIGNAL  5  FOR STATION ON SUB SUB SUB SUB
C
      IF (M.GT.MMAX) GO TO 7500
      IF (LA5(M).EQ.0) GO TO 5040
      GO TO 5020
5040  N5=M-1
      SAC5(N5+1)=0.0
      DO 5050 M=MM,N5
      L5(M)=STA5(M+1)-STA5(M)+EQ5(M)
      N5M=N5+MM-M
      SAC5(N5M)=SAC5(N5M+1) + AC5(N5M+1)
5050  CONTINUE
      CALL SACQ   (MM,SAC5,AC5,Q5,DQ5,N5)
      IF (ITYPE.NE.1) GO TO 5120
      IF (IG.EQ.0) GO TO 5120
```

**** PROGRAM LAKMLP ****

```
5060  K5=K5+1
      DO 5070 M=MG,MMAX
      IF (LA5(M).EQ.0) GO TO 5080
5070  CONTINUE
      M=MMAX
5080  N5=M-1
      MGSAV=MG
      MM=MGSAV
5120  DO 5190 M=MM,N5
      D5=Q0 * (Q5(M)**(ZP*2.))
      IF (D5.GT.12.) GO TO 5130
      JX=(D5/2.) +1.
      DIA5(M)=JX*2
      GO TO 5150
5130  IF (D5.GT.54.) GO TO 5140
      JX=(D5/3.) +1.
      DIA5(M)=JX*3
      GO TO 5150
5140  JX=(D5/6.) +1.
      DIA5(M)=JX*6
5150  IF (DIA5(M)-22.) 5160,5160,5170
5160  CS=CSS
      GO TO 5180
5170  CS=CSB
5180  HL5(M)=(((576.*Q5(M)) / (PI*CS*DIA5(M)**2.625))**2.) /1000.
      HF5(M)=L5(M)*HL5(M)
5190  CONTINUE
      HGL5(N5+1)=HGR5(N5+1)
      DO 5200 M=MM,N5
      N5M=N5+MM-M
      HGL5(N5M)=HF5(N5M) + HGL5(N5M+1)
      IF (HGL5(N5M).GE. HGR5(N5M)) GO TO 5200
      HGL5(N5M)=HGR5(N5M)
5200  CONTINUE
      IF (INPUOT .NE. 1) GO TO 5240
      WRITE (6,2210)
      WRITE (6,2310) (LAT5(L,II),II=1,2)
      DO 5230 M=MM,N5
      WRITE (6,1220) STA5(M),AC5(M),HGL5(M),HGR5(M),L5(M),SAC5(M),Q5(M),
     5  DIA5(M),HL5(M),HF5(M)
5230  CONTINUE
      WRITE (6,1220) STA5(N5+1),AC5(N5+1),HGL5(N5+1),HGR5(N5+1)
5240  CONTINUE
      HGR4(L)=HGL5(MM) +TELOS5(K5)
      AC4(L)=SAC5(MM)
      IF (ITYPE.NE.1) GO TO 4020
      IF (IG.EQ. 0) GO TO 4020
      MG=N5+2
```

**** PROGRAM LAKMLP ****

```
      GO TO 4090
C
C        WRITE INPUT ERROR MESSAGES
C
 7100 IO=I-1
      WRITE (6,7110) (LAT1(IT),II=1,3),IO,STA1(IO),I
 7110 FORMAT (1H1,48X,3A10// 6H STA1(,I3, 3H) =,F10.3/// 10X, 4H I =,I3/
     1// 85H DIMENSION AND IMAX MUST BE INCREASED TO THE NUMBER OF STATI
     1ONS ON THE MAIN (1) LINE.)
      GO TO 30
 7200 JO=J-1
      WRITE (6,7210) (LAT2(I,II),II=1,2),JO,STA2(JO),J
 7210 FORMAT (1H1,48X,2A10/// 6H STA2(,I3, 3H) =,F10.3/// 10X, 4H J =,I3
     2/// 88H DIMENSION AND JMAX MUST BE INCREASED TO THE NUMBER OF STAT
     2IONS ON THE SUB (2) LATERALS.)
      GO TO 30
 7300 KO=K-1
      WRITE (6,7310) (LAT3(J,II),II=1,2),KO,STA3(KO),K
 7310 FORMAT (1H1,48X,2A10/// 6H STA3(,I3, 3H) =,F10.3/// 10X, 4H K =,I3
     3// 92H DIMENSION AND KMAX MUST BE INCREASED TO THE NUMBER OF STATI
     3ONS ON THE SUB SUB (3) LATERALS.)
      GO TO 30
 7400 LO=L-1
      WRITE (6,7410) (LAT4(K,II),II=1,2),LO,STA4(LO),L
 7410 FORMAT (1H1,48X,2A10/// 6H STA4(,I3, 3H) =,F10.3/// 10X, 4H L =,I3
     4// 95H DIMENSION AND LMAX MUST BE INCREASED TO THE NUMBER OF STATI
     4ONS ON THE SUB SUB SUB (4) LATERALS.)
      GO TO 30
 7500 MO=M-1
      WRITE (6,7510) (LAT5(L,II),II=1,2),MO,STA5(MO),M
 7510 FORMAT (1H1,48X,2A10/// 6H STA5(,I3, 3H) =,F10.3/// 10X, 4H M =,I3
     5// 100H DIMENSION AND MMAX MUST BE INCREASED TO THE NUMBER OF STAT
     5IONS ON THE SUB SUB SUB SUB (5) LATERALS.)
      GO TO 30
 7600 WRITE (6,7610) CP
 7610 FORMAT (1H1//////10X, 25H CP MUST BE DECREASED TO ,F10.3)
      GO TO 30
 7700 WRITE (6,7710) CP
 7710 FORMAT (1H1//////10X, 25H CP MUST BE INCREASED TO ,F10.3)
      GO TO 30
 8000 I=0
      IF (ITYPE.GT.2) GO TO 8005
      TKWSEL=AVH1
 8005 J=0
      II=0
      IF (L1(1).EQ.0) GO TO 9000
 8010 I=I+1
      IP=IP1(I)
```

**** PROGRAM LAKMLP ****

```
      IF(LA1(I).EQ.0) GO TO 8200
      IF(IP.EQ.0) GO TO 8010
      RL(IP)=L1(I)
      II=1
      GZ(IP,1)=IP+1
      IF(L1(I+1).EQ.0.0) GO TO 8025
      IF(LA1(I+1).EQ.2) GO TO 8100
8020  RHS(IP)=TKWSEL-HGR1(I+1)
      GO TO 8010
8025  IF(LA1(I+1).EQ.0) GO TO 8075
      IF(LA1(I+1).EQ.2) GO TO 8040
      IF(LA1(I+1).NE.LA1(I+2)) GO TO 8030
      IF(HGR1(I+1).GT.HGR1(I+2)) GO TO 8020
      RHS(IP)=TKWSEL-HGR1(I+2)
      GO TO 8010
8030  IF(LA1(I+2).EQ.0) GO TO 8035
      RHS(IP)=TKWSEL-HGR1(I+1)
      GO TO 8105
8035  IF(HGR1(I+1).GT.HGR1(I+2)) GO TO 8080
      RHS(IP)=TKWSEL-HGR1(I+2)
      GO TO 8085
8040  IF(LA1(I+1).NE.LA1(I+2)) GO TO 8070
      RHS(IP)=TKWSEL-GS1(I+1)
8045  J=J+1
      II=II+1
      GZ(IP,II)=IR2(J)
8050  J=J+1
      IF(LA2(J).EQ.0) GO TO 8060
      GO TO 8050
8060  IF(II.EQ.3) GO TO 8010
      GO TO 8045
8070  RHS(IP)=TKWSEL-HGR1(I+2)
      GO TO 8105
8075  IF(L1(I).EQ.0.0) GO TO 8090
8080  RHS(IP)=TKWSEL-HGR1(I+1)
8085  GZ(IP,1)=0
      GO TO 8010
8090  IP=IR1(I-1)
      GO TO 8085
8100  RHS(IP)=TKWSEL-GS1(I+1)
8105  J=J+1
      II=II+1
      GZ(IP,II)=IR2(J)
8120  J=J+1
      IF(LA2(J).EQ.0) GO TO 801
      GO TO 8120
8200  JA=0
      K2=1
```

**** PROGRAM LAKMLP ****

```
      K=0
      II=0
      JY=1
      IF (L2(1).EQ.0) GO TO 9000
8210  J=JA+JY
      JA=JA+1
      IP=IR2(J)
      IF(LA2(J).EQ.0) GO TO 8295
      IF(IP.EQ.0) GO TO 8210
      RL(IP)=L2(J)
      II=1
      GZ(IP,1)=IP+1
      IF(L2(J+1).EQ.0.0) GO TO 8225
      IF(LA2(J+1).EQ.3) GO TO 8300
8220  RHS(IP)=TKWSEL-HGR2(J+1)-TEELOS2(K2)
      GO TO 8210
8225  IF(LA2(J+1).EQ.0) GO TO 8275
      IF(LA2(J+1).EQ.3) GO TO 8240
      IF(LA2(J+1).NE.LA2(J+2)) GO TO 8230
      IF(HGR2(J+1).GT.HGR2(J+2)) GO TO 8220
      RHS(IP)=TKWSEL-HGR2(J+2)-TEELOS2(K2)
      GO TO 8210
8230  IF(LA2(J+2).EQ.0) GO TO 8235
      RHS(IP)=TKWSEL-HGR2(J+1)-TEELOS2(K2)
      GO TO 8305
8235  IF(HGR2(J+1).GT.HGR2(J+2)) GO TO 8280
      RHS(IP)=TKWSEL-HGR2(J+2)-TEELOS2(K2)
      GO TO 8285
8240  IF(LA2(J+1).NE.LA2(J+2)) GO TO 8270
      RHS(IP)=TKWSEL-GS2(J+1)
8245  K=K+1
      II=II+1
      GZ(IP,II)=IR3(K)
8250  K=K+1
      IF(LA3(K).EQ.0) GO TO 8260
      GO TO 8250
8260  IF(II.EQ.3) GO TO 8210
      GO TO 8245
8270  RHS(IP)=TKWSEL-HGR2(J+2)-TEELOS2(K2)
      GO TO 8305
8275  IF(L2(J).EQ.0.0) GO TO 8290
8280  RHS(IP)=TKWSEL-HGR2(J+1)-TEELOS2(K2)
8285  GZ(IP,1)=0
      GO TO 8210
8290  IP=IR2(J-1)
      GO TO 8285
8295  JY=J+1
      JA=0
```

**** PROGRAM LAKMLP ****

```
      K2=K2+1
      IF (LA2(J+1).EQ.0) GO TO 8400
      GO TO 8210
8300  RHS(IP)=TKWSEL-GS2(J+1)
8305  K=K+1
      II=II+1
      GZ(IP,II)=IR3(K)
8320  K=K+1
      IF(LA3(K).EQ.0) GO TO 8210
      GO TO 8320
8400  KA=0
      K3=1
      L=0
      II=0
      KY=1
      IF (L3(1).EQ.0) GO TO 9000
8410  K=KA+KY
      KA=KA+1
      IP=IR3(K)
      IF(LA3(K).EQ.0) GO TO 8495
      IF(IP.EQ.0) GO TO 8410
      RL(IP)=L3(K)
      II=1
      GZ(IP,1)=IP+1
      IF(L3(K+1).EQ.0.0) GO TO 8425
      IF(LA3(K+1).EQ.4) GO TO 8500
8420  RHS(IP)=TKWSEL-HGR3(K+1)-TEELOS3(K3)
      GO TO 8410
8425  IF(LA3(K+1).EQ.0) GO TO 8475
      IF(LA3(K+1).EQ.4) GO TO 8440
      IF(LA3(K+1).NE.LA3(K+2)) GO TO 8430
      IF(HGR3(K+1).GT.HGR3(K+2)) GO TO 8420
      RHS(IP)=TKWSEL-HGR3(K+2)-TEELOS3(K3)
      GO TO 8410
8430  IF(LA3(K+2).EQ.0) GO TO 8435
      RHS(IP)=TKWSEL-HGR3(K+1)-TEELOS3(K3)
      GO TO 8505
8435  IF(HGR3(K+1).GT.HGR3(K+2)) GO TO 8480
      RHS(IP)=TKWSEL-HGR3(K+2)-TEELOS3(K3)
      GO TO 8485
8440  IF(LA3(K+1).NE.LA3(K+2)) GO TO 8470
      RHS(IP)=TKWSEL-GS3(K+1)
8445  L=L+1
      II=II+1
      GZ(IP,II)=IR4(L)
8450  L=L+1
      IF(LA4(L).EQ.0) GO TO 8460
      GO TO 8450
```

**** PROGRAM LAKMLP ****

```
8460  IF(II.EQ.3) GO TO 8410
      GO TO 8445
8470  RHS(IP)=TKWSEL-HGR3(K+2)-TEELOS3(K3)
      GO TO 8505
8475  IF(L3(K).EQ.0.0) GO TO 8490
8480  RHS(IP)=TKWSEL-HGR3(K+1)-TEELOS3(K3)
8485  GZ(IP,1)=0
      GO TO 8410
8490  IP=IR3(K-1)
      GO TO 8485
8495  KY=K+1
      KA=0
      K3=K3+1
      IF (LA3(K+1).EQ.0) GO TO 8600
      GO TO 8410
8500  RHS(IP)=TKWSEL-GS3(K+1)
8505  L=L+1
      II=II+1
      GZ(IP,II)=IR4(L)
8520  L=L+1
      IF(LA4(L).EQ.0) GO TO 8410
      GO TO 8520
8600  LA=0
      K4=1
      M=0
      II=0
      LY=1
      IF (L4(1).EQ.0) GO TO 9000
8610  L=LA+LY
      LA=LA+1
      IP=IR4(L)
      IF(LA4(L).EQ.0) GO TO 8695
      IF(IP.EQ.0) GO TO 8610
      RL(IP)=L4(L)
      II=1
      GZ(IP,1)=IP+1
      IF(L4(L+1).EQ.0.0) GO TO 8625
      IF(LA4(L+1).EQ.5) GO TO 8700
8620  RHS(IP)=TKWSEL-HGR4(L+1)-TEELOS4(K4)
      GO TO 861
8625  IF(LA4(L+1).EQ.0) GO TO 8675
      IF(LA4(L+1).EQ.5) GO TO 8640
      IF(LA4(L+1).NE.LA4(L+2)) GO TO 8630
      IF(HGR4(L+1).GT.HGR4(L+2)) GO TO 8620
      RHS(IP)=TKWSEL-HGR4(L+2)-TEELOS4(K4)
      GO TO 8610
8630  IF(LA4(L+2).EQ.0) GO TO 8635
      RHS(IP)=TKWSEL-HGR4(L+1)-TEELOS4(K4)
```

**** PROGRAM LAKMLP ****

```
      GO TO 8705
8635  IF(HGR4(L+1).GT.HGR4(L+2)) GO TO 8680
      RHS(IP)=TKWSEL-HGR4(L+2)-TEELOS4(K4)
      GO TO 8685
8640  IF(LA4(L+1).NE.LA4(L+2)) GO TO 8670
      RHS(IP)=TKWSEL-GS4(L+1)
8645  M=M+1
      II=II+1
      GZ(IP,II)=IR5(M)
8650  M=M+1
      IF(LA5(M).EQ.0) GO TO 8660
      GO TO 8650
8660  IF(II.EQ.3) GO TO 8610
      GO TO 8645
8670  RHS(IP)=TKWSEL-HGR4(L+2)-TEELOS4(K4)
      GO TO 8705
8675  IF(L4(L).EQ.0.0) GO TO 8690
8680  RHS(IP)=TKWSEL-HGR4(L+1)-TEELOS4(K4)
8685  GZ(IP,1)=0
      GO TO 8610
8690  IP=IR4(L-1)
      GO TO 8685
8695  LY=L+1
      LA=0
      K4=K4+1
      IF (LA4(L+1).EQ.0) GO TO 8800
      GO TO 8610
8700  RHS(IP)=TKWSEL-GS4(L+1)
8705  M=M+1
      II=II+1
      GZ(IP,II)=IR5(M)
8720  M=M+1
      IF(LA5(M).EQ.0) GO TO 8610
      GO TO 8720
8800  MA=0
      K5=1
      II=0
      MY=1
      IF (L5(1).EQ.0) GO TO 9000
8810  M=MA+MY
      MA=MA+1
      IP=IR5(M)
      IF(LA5(M).EQ.0) GO TO 8895
      IF(IP.EQ.0) GO TO 8810
      RL(IP)=L5(M)
      II=1
      GZ(IP,1)=IP+1
      IF(L5(M+1).EQ.0.0) GO TO 8825
```

**** PROGRAM LAKMLP ****

```
8820  RHS(IP)=TKWSEL-HGR5(M+1)-TEELOS5(K5)
      GO TO 8810
8825  IF(LA5(M+1).EQ.0) GO TO 8875
      IF(LA5(M+1).NE.LA5(M+2)) GO TO 8835
      IF(HGR5(M+1).GT.HGR5(M+2)) GO TO 8820
      RHS(IP)=TKWSEL-HGR5(M+2)-TEELOS5(K5)
      GO TO 8810
8835  IF(HGR5(M+1).GT.HGR5(M+2)) GO TO 8880
      RHS(IP)=TKWSEL-HGR5(M+2)-TEELOS5(K5)
      GO TO 8885
8875  IF(L5(M).EQ.0.0) GO TO 8890
8880  RHS(IP)=TKWSEL-HGR5(M+1)-TEELOS5(K5)
8885  GZ(IP,1)=0
      GO TO 8810
8890  IP=IR5(M-1)
      GO TO 8885
8895  MY=M+1
      MA=0
      K5=K5+1
      IF (LA5(M+1).EQ.0) GO TO 9000
      GO TO 8810
C
C          MODIFICATION FO TYPES 3 AND 4
C
9000  IF (ITYPE.LE.2) GO TO 9001
      RHS(1)=HF1(1)
      GZ(IR1(LASAV-1),1) = 0
9001  WRITE (JW,9010) LAT1
9010  FORMAT (1H1,1X,14HNAME              ,3A10)
      WRITE (1,9011)
9011  FORMAT (24HNAME            LEO KINNEY)
      WRITE (JW,9020)
9020  FORMAT (1X,4HROWS)
      WRITE (1,9021)
9021  FORMAT (4HROWS)
      WRITE (JW,9030)
9030  FORMAT (1X,5H  N C)
      WRITE (1,9031)
9031  FORMAT (5H  N C)
C
C          IC/5   IS NO. OF REACHES
      IC=IC/5
C
      WRITE (JW,9045) (I,I=1,IC)
9045  FORMAT (10(1X,5H  E L,I4))
      DO 9050 I=1,IC
      WRITE (1,9046) I
9046  FORMAT (5H  E L,I4)
```

**** PROGRAM LAKMLP ****

```
9050  CONTINUE
      WRITE (JW,9055) (I,I=1,IC)
9055  FORMAT (10(1X,5H   E H,I4))
      DO 9060 I=1,IC
      WRITE (1,9056) I
9056  FORMAT (5H   E H,I4)
9060  CONTINUE
      WRITE (JW,9065) (I,I=1,IC)
9065  FORMAT (10(1X,5H   L S,I4))
      DO 9070 I=1,IC
      WRITE (1,9066) I
9066  FORMAT (5H   L S,I4)
9070  CONTINUE
      WRITE (JW,9080)
9080  FORMAT (1X,7HCOLUMNS)
      WRITE (1,9081)
9081  FORMAT (7HCOLUMNS)
      DO 9090 J=1,IC
      I1=5*J-4
      I2=I1+4
      K1=0
      ICOST=0
      DO 9090 I=I1,I2
      K1=K1+1
      K=J*10 + K1
       IDP=DP(I)
       IHC=HCC(I)
      IF (COSTPI(I).NE.0.0) GO TO 9084
      WRITE (JW,9083) K
9083  FORMAT (1X,*COST OF PIPE *,I10,* = 0.0*)
      ICOST=ICOST+1
9084  WRITE (JW,9085) K,COSTPI(I),IDP,VP(I),IHC,J,J,HP(I)
9085  FORMAT (1X,5H     X,I4,6H      C,3X,F10.2,3X,I4,*-INCH   VEL=*,
     1         F5.1,*  H*,I5,
     2                5H      L,I3,6X,3H 1.,
     3                6H      H,I3,6X,F10.6)
      WRITE (1,9086) K,COSTPI(I),K,J,K,J,HP(I)
9086  FORMAT (    5H     X,I4,6H      C,9X,F10.2/
     1            5H     X,I4,6H      L,I3,6X,3H 1./
     2            5H     X,I4,6H      H,I3,6X,F10.6)
9090  CONTINUE
      IF (ICOST.GE.1) CALL EXIT
      DO 9100 I=1,IC
      WRITE (JW,9093) I,I,I,I
9093  FORMAT (1X,5H     F,I3,7H      S,I3,6X,3H 1.,
     1        1X,5H     F,I3,7H      H,I3,6X,3H-1.)
      WRITE (1,9094) I,I,I,I
9094  FORMAT (5H     F,I3,7H      S,I3,6X,3H 1./
```

**** PROGRAM LAKMLP ****

```
      1          5H    F,I3,7H        H,I3,6X,3H-1.)
         DO 9100 J=1,16
         IGZ=GZ(I,J)
         IF (GZ(I,J).EQ.0) GO TO 9100
         WRITE (JW,9097) I,IGZ
 9097    FORMAT (1X,5H    F,I3,7H       H,I3,6X,3H 1.)
         WRITE (1,9098) I,IGZ
 9098    FORMAT (    5H    F,I3,7H       H,I3,6X,3H 1.)
 9100    CONTINUE
C
C            WRITE RIGHT HAND SIDE
C
         WRITE (JW,9110)
 9110    FORMAT (1X,3HRHS)
         WRITE (1,9111)
 9111    FORMAT (    3HRHS)
         DO 9120 I=1,IC
         WRITE (JW,9115) I,RL(I)
 9115    FORMAT (1X,15H    RHS         L,I3,6X,F10.2)
         WRITE (1,9116) I,RL(I)
 9116    FORMAT (15H    RHS         L,I3,6X,F10.2)
 9120    CONTINUE
         DO 9130 I=1,IC
         WRITE (JW,9125) I,RHS(I)
 9125    FORMAT (1X,15H    RHS         S,I3,6X,F10.4)
         WRITE (1,9126) I,RHS(I)
 9126    FORMAT (    15H    RHS         S,I3,6X,F10.4)
 9130    CONTINUE
         WRITE (JW,9140)
 9140    FORMAT (1X,6HENDATA)
         WRITE (1,9141)
 9141    FORMAT (6HENDATA)
 9999    CALL EXIT
         END
         SUBROUTINE PIPSIZ (LA,STA,AC,L,SAC,Q,HGR,GS,AVH,IS,N,LAT,LATDIM   ,
        1 IR,DQ,HGL,DEL,EQ)
         COMMON NN,MDIA( 650),GL( 650),MHC( 650),SEXCAV(300),SBKFL(300),
        1        ZG,PIPCOV,HVH,PIPMIN,ND,GDEL( 600),JHC( 600),LASAV,TKWSEL,
        2        CSS,CSB,SJEXC,SJBKF,MDEL,JW
         COMMON /LPPACK/  DP(2000),VP(2000),HP(2000),HVP(2000),
        1           COSTPI(2000),C(30,40),RL(400),GZ(400,16),RHS(400),
        2            HCC(2000),IC
         DIMENSION PIPARR(30),IR( 1 )
         DIMENSION LA( 1 ),STA( 1 ),AC( 1 ),L( 1 ),SAC( 1 ),Q( 1 ),
        1          HGR( 1 ),GS( 1 ),LAT(LATDIM,1),DQ( 1 ),HGL( 1 ),DEL( 1 ),
        2          EQ( 1 ),
        2           H(300),DIA(300),HF(300),DZG(300),R(300),V(300),HV(300),
        3           SL(300),HC(300),DIAOD(300),TWIDTH(300),SP(300)
```

**** PROGRAM LAKMLP ****

```
      REAL L
C
C        DATA ARRAY IS FOR PIPE 4" TO 120"
C
      DATA (PIPARR(I),I=1,30) /4.,6.,8.,10.,12.,15.,18.,21.,24.,27.,30.,
     1 33.,36.,39.,42.,45.,48.,51.,54.,60.,66.,72.,78.,84.,90.,96.,102.,
     2 108.,114.,120./
      IF (LA(IS).NE.1) GO TO 5
      IP =PIPMIN/2.-1.
      IRC=0
      IC=0
      IL=0
    5 IPEMAX=650
      NDEMAX=600
      JCOUNT=0
      HVHCK=0.0
      JA=0
      NNI=0
   10 I=JA+IS
      JA=JA+1
      IF (LA(I).NE.0) GO TO 10
      N=I-1
      NL=N+1
      SC=10.
      IY=IS
   20 DO 30 I=IY,N
      IF (HGR(I).GE.AVH) GO TO 2000
C     IF (STA(I+1).LE.STA(I)) GO TO 2100
   30 CONTINUE
      IF (JCOUNT.GT.0) GO TO 9999
      DO 60 I=IS,N
      QZG(I)=Q(I)**ZG
      R(I)=QZG(I) / QZG(IS)
      IF (I.EQ.IS) GO TO 40
      SL(I)=SL(I-1) + R(I) * L(I)
      IF(SL(I).EQ.0.0) GO TO 60
      GO TO 50
   40 SL(I)=R(I) * L(I)
      IF(SL(I).EQ.0.0) GO TO 60
   50 S=(AVH-HGR(I+1)) / SL(I)
      IF (S.GE.SC) GO TO 60
      SC=S
      ISAV=I
   60 CONTINUE
      HGL(IS)=AVH
      XP=1. /2.625
      CA=3.14159 * CSS
      CB=3.14159 * CSB
```

**** PROGRAM LAKMLP ****

```
      DO 100 J=IS,ISAV
      SP(J)=SC * R(J)
      D=((576.*Q(J)) / (CB * SQRT(1000.*SP(J)))) ** XP
      IF (D.GE.24.0) GO TO 80
      D=((576.*Q(J)) / (CA * SQRT(1000.*SP(J)))) ** XP
      IF (D.GT.PIPMIN) GO TO 70
      DIA(J)=PIPMIN
      GO TO 100
   70 IF (D.GE.12.0) GO TO 80
      JX=D/2.0
      DIA(J)=JX*2
      GO TO 100
   80 IF (D.GE.54.0) GO TO 90
      JX=D/3.0
      DIA(J)=JX*3
      GO TO 100
   90 JX=D/6.0
      DIA(J)=JX*6
  100 CONTINUE
      II=IS
  110 DO 150 I=II,ISAV
      IF (DIA(I)-22.0) 120,120,130
  120 CX=CA
      GO TO 140
  130 CX=CB
  140 H(I)=(331.776 * Q(I)*Q(I)) / (CX * CX * DIA(I)**5.25)
      HF(I)=L(I)*H(I)
      HGL(I+1)=HGL(I)-HF(I)
      IF (HGL(I+1).GE.HGR(I+1)) GO TO 150
      GO TO 160
  150 CONTINUE
      GO TO 210
  160 SH=H(IS)-SP(IS)
      DO 170 J=IS,I
      IF ((H(J)-SP(J)).LT.SH) GO TO 170
      SH=H(J)-SP(J)
      JKEEP=J
  170 CONTINUE
      J=JKEEP
      IF (DIA(J).GE.12.0) GO TO 180
      DIA(J)=DIA(J)+ 2.0
      IF (J.EQ.IS) GO TO 200
      II=J-1
      GO TO 110
  180 IF (DIA(J).GE.54.0) GO TO 190
      DIA(J)=DIA(J)+ 3.0
      IF (J.EQ.IS) GO TO 200
      II=J-1
```

**** PROGRAM LAKMLP ****

```
      GO TO 110
 190  DIA(J)=DIA(J)+ 6.0
      IF (J.EQ.IS) GO TO 200
      II=J-1
      GO TO 110
 200  II=J
      GO TO 110
 210  IF (ISAV.EQ.N) GO TO 2900
1000  ISAV=ISAV + 1
      SO=10.
      SL(ISAV-1)=0.0
      DO 1060 K=ISAV,N
      IF(QZG(K).EQ.0.0) GO TO 1060
      R(K)=QZG(K) / QZG(ISAV)
      SL(K)=SL(K-1) + R(K) * L(K)
      IF(SL(K).EQ.0.0) GO TO 1060
      S=(HGL(ISAV)-HGR(K+1)) / SL(K)
      IF (S.GE.SO) GO TO 1060
      SO=S
      KSAV=K
1060  CONTINUE
      K=N
      IF (SO.LT. 10.) GO TO 1065
      DO 1063 J=ISAV,N
      IF (DIA(J-1).EQ.0.0) GO TO 1063
      DIA(J)=DIA(J-1)
      SP(J)=S*R(J)
1063  CONTINUE
      KSAV=K
      GO TO 1105
1065  DO 1100 K=ISAV,KSAV
      SP(K)=SO * R(K)
      D=((576.*Q(K)) / (CB * SQRT(1000.*SP(K)))) ** XP
      IF (D.GE.24.0) GO TO 1080
      D=((576.*Q(K)) / (CA * SQRT(1000.*SP(K)))) ** XP
      IF (D.GT.PIPMIN) GO TO 1070
      DIA(K)=PIPMIN
      GO TO 1100
1070  IF (D.GE.12.0) GO TO 1080
      JX=D/2.0
      DIA(K)=JX*2
      GO TO 1100
1080  IF (D.GE.54.0) GO TO 1090
      JX=D/3.0
      DIA(K)=JX*3
      GO TO 1100
1090  JX=D/6.0
      DIA(K)=JX*6
```

**** PROGRAM LAKMLP ****

```
1100  CONTINUE
1105  II=ISAV
1110  DO 1150 I=II,KSAV
      IF (DIA(I)-22.0) 1120,1120,1130
1120  CX=CA
      GO TO 1140
1130  CX=CB
1140  H(I)=(331.776 * Q(I)*Q(I)) / (CX * CX * DIA(I)**5.25)
      HF(I)=L(I)*H(I)
      HGL(I+1)=HGL(I)-HF(I)
      IF (HGL(I+1).GE.HGR(I+1)) GO TO 1150
      GO TO 1160
1150  CONTINUE
      I=KSAV
      GO TO 1210
1160  SH=H(ISAV)-SP(ISAV)
      DO 1170 J=ISAV,I
      IF ((H(J)-SP(J)).LT.SH) GO TO 1170
      SH=H(J)-SP(J)
      JKEEP=J
1170  CONTINUE
      J=JKEEP
      IF (DIA(J).GE.12.0) GO TO 1180
      DIA(J)=DIA(J)+ 2.0
      IF (J.EQ.ISAV) GO TO 1200
      II=J-1
      GO TO 1110
1180  IF (DIA(J).GE.54.0) GO TO 1190
      DIA(J)=DIA(J)+ 3.0
      IF (J.EQ.ISAV) GO TO 1200
      II=J-1
      GO TO 1110
1190  DIA(J)=DIA(J)+ 6.0
      IF (J.EQ.ISAV) GO TO 1200
      II=J-1
      GO TO 1110
1200  II=J
      GO TO 1110
1210  IF (KSAV.EQ.N) GO TO 2900
      ISAV=KSAV
      GO TO 1000
C
C        WRITE INPUT ERROR MESSAGES
C
2000  WRITE (6,2010) HGR(I),STA(I),AVH
2010  FORMAT (//10X, 4H HGR,F10.2, 7H AT STA,F10.2, 32H IS GREATER THAN
     1OR EQUAL TO AVH,F10.2)
      JCOUNT=1
```

**** PROGRAM LAKMLP ****

```
      IY=I+1
      GO TO 2^
2200  WRITE (6,2210) GS(I),STA(I),HVH
2210  FORMAT (//10X, 4H GS ,F10.2, 7H AT STA,F10.2, 20H IS GREATER THAN
     1HVH,F10.2)
      JCOUNT=1
      IY=I+1
      GO TO 3050
2220  WRITE (6,2210) GS(NL),STA(NL),HVH
      GO TO 9999
2300  WRITE (6,2310)
2310  FORMAT (/// 44H NO. OF PIPE LENGTHS ARE GREATER THAN IPEMAX)
      GO TO 9999
2400  WRITE (6,2410)
2410  FORMAT (/// 41H NO. OF DELIVERIES IS GREATER THAN NDEMAX)
      GO TO 9999
C
C        COMPUTE EARTHWORK,VELOCITY,HEAD CLASS
C              EARTHWORK IS BASED ON USBR DWG. NO.40-D-6036  MAY 7,1965
C                 PRESSURE PIPE  TYPICAL TRENCHES  REV. APRIL 20,1966
C
2900  IF (LA(IS).NE.1) GO TO 3000
      IF (TKWSEL.EQ.0.0) GO TO 3000
      IF(NNI.NE.0) GO TO 3000
      HGL (LASAV)=TKWSEL
      IB=IS+1
      DO 2910 I=IB,LASAV
      NNI=LASAV +1 -I
      HGL(NNI)=HF(NNI) + HGL(NNI+1)
2910  CONTINUE
      ISAV=LASAV-1
      IF (ISAV.EQ.N) GO TO 3000
      GO TO 1000
3000  DO 3040 I=IS,N
      IF (DIA(I).GT.6.0) GO TO 3010
      DIAOD(I)=(DIA(I)+2.0)/12.
      TWIDTH(I)=2.0
      GO TO 3040
3010  IF (DIA(I).GT.18.0) GO TO 3020
      DIAOD(I)=(DIA(I)+4.0)/12.
      TWIDTH(I)=0.083*(DIA(I)+24.0)
      GO TO 3040
3020  IF (DIA(I).GT.24.0) GO TO 3030
      DIAOD(I)=(DIA(I)+4.0)/12.
      TWIDTH(I)=0.083*(DIA(I)+40.0)
      GO TO 3040
3030  DIAOD(I)=1.167*DIA(I)/12.
      TWIDTH(I)=0.083*(1.167*DIA(I) +36.0)
```

**** PROGRAM LAKMLP ****

```
3040  CONTINUE
      IY=IS
3050  DO 3100 I=IY,N
      NN=NN+1
      IF (NN.GT.IPEMAX) GO TO 2300
      V(I)=Q(I) / ((3.14159*DIA(I)**2.0) / 576.)
      HV(I)=(V(I)*V(I))/64.32
      IF (HVH.NE.0.0) GO TO 3070
      HVHCK=0.0
      IF (I.GE.LASAV) GO TO 3060
      HVHCK=1.0
      HVH=(HGL(I)+HGL(I+1)) /2.0
      GO TO 3070
3060  HVH=TKWSEL + 5.0
3070  IF (GS(I).GT.HVH) GO TO 2200
      IW= (HVH-(GS(I)+GS(I+1))/2.0) + DIAOD(I)/2.0 +PIPCOV
      IF (HVHCK.EQ.0.0) GO TO 3080
      HVH=0.0
3080  DUMHC=IW/25
      HC(I)=DUMHC * 25.0  + 25.0
      MDIA(NN)=DIA(I)
      GL(NN)=L(I)
      MHC(NN)=HC(I)
      EXCAV=TWIDTH(I)*L(I)*((PIPCOV+DIAOD(I))/27.)
      BKFL=EXCAV-(((3.14159*DIAOD(I)**2.)/4.) *L(I)/27.)
      IF (I.EQ.IS) GO TO 3090
      SEXCAV(I)=EXCAV+SEXCAV(I-1)
      SBKFL(I)=BKFL+SBKFL(I-1)
      GO TO 3100
3090  SEXCAV(I)=EXCAV
      SBKFL(I)=BKFL
3100  CONTINUE
      IF (TKWSEL.EQ.0.0) GO TO 3130
      IF (LA(IS).NE.1) GO TO 3110
      IF (NL.EQ.LASAV) GO TO 3120
3110  IF (HVH.NE.(TKWSEL+5.)) GO TO 3130
3120  HVH=TKWSEL + 5.0
3130  IF (GS(NL).GT.HVH) GO TO 2220
C
C        SIZE DELIVERIES
C
      IB=IS+1
      DEL(IS)=0.0
      IF (MDEL.EQ.1) GO TO 4170
4000  DO 4160 I=IB,NL
      DEL(I)=0.0
      IF (LA(IS).NE.1) GO TO 4010
      IF (I.EQ.LASAV) GO TO 4160
```

**** PROGRAM LAKMLP ****

```
4010  IF (I.EQ.NL) GO TO 4020
      IF (LA(I).NE.LA(IS)) GO TO 4160
4020  IF (DQ(I).GT.0.0) GO TO 4030
      GO TO 4150
4030  IF (DQ(I).GT.0.02) GO TO 4040
      DEL(I)=1.0
      GO TO 4160
4040  IF (DQ(I).GT.0.05) GO TO 4050
      DEL(I)=1.5
      GO TO 4160
4050  IF (DQ(I).GT.0.178) GO TO 4060
      DEL(I)=2.0
      GO TO 4160
4060  IF (DQ(I).GT.0.446) GO TO 4070
      DEL(I)=3.0
      GO TO 4160
4070  IF (DQ(I).GT.0.891) GO TO 4080
      DEL(I)=4.0
      GO TO 4160
4080  IF (DQ(I).GT.2.005) GO TO 4090
      DEL(I)=6.0
      GO TO 4160
4090  IF (DQ(I).GT.2.674) GO TO 4100
      DEL(I)=8.0
      GO TO 4160
4100  IF (DQ(I).GT.3.565) GO TO 4110
      DEL(I)=10.0
      GO TO 4160
4110  IF (DQ(I).GT.5.013) GO TO 4120
      DEL(I)=12.0
      GO TO 4160
4120  IF (DQ(I).GT.6.684) GO TO 4130
      DEL(I)=14.0
      GO TO 4160
4130  IF (DQ(I).GT.8.457) GO TO 4140
      DEL(I)=16.0
      GO TO 4160
4140  IF (DQ(I).GT.10.027) GO TO 4150
      DEL(I)=18.0
      GO TO 4160
4150  DEL(I)=0.0
4160  CONTINUE
4170  DO 4190 J=IB,NL
      IF (DEL(J).EQ.0.0) GO TO 4190
      ND=ND+1
      IF (ND.GT.NDEMAX) GO TO 2400
      GDEL(ND)=DEL(J)
      IF (J.EQ.NL) GO TO 4180
```

**** PROGRAM LAKMLP ****

```
      IF (HC(J).LT.HC(J-1)) GO TO 4180
      JHC(ND)=HC(J)
      GO TO 4190
 4180 JHC(ND)=HC(J-1)
 4190 CONTINUE
      IF (LA(IS).NE.1) GO TO 7000
C
C        READ IN COST OF PIPE
 7900 REWIND 3
      READ (3,7910) MC1,MC,NC
 7910 FORMAT (3I10)
      READ (3,7920) ((C(M,N),M=1,10),N=1,NC)
 7920 FORMAT (10F8.2)
C
C        MC1 - SMALLEST PIPE DIA. FOR WHICH COST IS GIVEN - HAS TO BE
C                4,6,OR 8
C        MC  - SUBSCRIPT LIMIT FOR PIPE DIA (MUST BE 10,20,OR 30)
C        NC  - SUBSCRIPT LIMIT FOR PIPE CLASS
C        C ARRAY IS DIMENSIONED TO PROVIDE FOR DIAMETERS UP TO 120 INCH
C                AND HEAD UP TO 1000 FT.
C        COST VALUES SHOULD BE GIVEN FOR HEADS IN INCREMENTS OF 25 FEET
C
      IF (MC.GT.20) GO TO 7930
      READ (3,7920) ((C(M,N),M=11,MC),N=1,NC)
      GO TO 7000
 7930 READ (3,7920) ((C(M,N),M=11,20),N=1,NC)
      READ (3,7920) ((C(M,N),M=21,MC),N=1,NC)
C
C
 7000 N=NL-1
      RMC1=MC1
      DO 8100 J=IS,N
      IF (L(J).EQ.0.0) GO TO 8100
      IRC=IRC+1
      IC=IC+1
      HCC(IC)=HC(J)
      HCC(IC+1)=HC(J)
      HCC(IC+2)=HC(J)
      HCC(IC+3)=HC(J)
      HCC(IC+4)=HC(J)
      NH=(HC(J)+.1)/25.
      IR(J) =IRC
      IF (DIA(J).GT.12.) GO TO 7010
      ID=DIA(J)/2.-1.
      GO TO 7030
 7010 IF (DIA(J).GT.54.) GO TO 7020
      ID=DIA(J)/3. + 1.
      GO TO 7030
```

**** PROGRAM LAKMLP ****

```
7020  ID=DIA(J)/5. + 10.
7030  IF ((ID-IP).GT.2) GO TO 7050
7040  DP(IC  ) =PIPARR(IP+4)
      DP(IC+1) =PIPARR(IP+3)
      DP(IC+2) =PIPARR(IP+2)
      DP(IC+3) =PIPARR(IP+1)
      DP(IC+4) =PIPARR(IP)
      GO TO 7070
7050  IF (ID.GE.28) GO TO 7060
      DP(IC  ) =PIPARR(ID+2)
      DP(IC+1) =PIPARR(ID+1)
      DP(IC+2) =DIA(J)
      DP(IC+3) =PIPARR(ID-1)
      DP(IC+4) =PIPARR(ID-2)
      GO TO 7070
7060  DP(IC  ) =PIPARR(30)
      DP(IC+1) =PIPARR(29)
      DP(IC+2) =PIPARR(28)
      DP(IC+3) =PIPARR(27)
      DP(IC+4) =PIPARR(26)
7070  IC4=IC+4
      DO 8080 I=IC,IC4
      VP(I)=Q(J) / ((3.14159*DP(I)*DP(I)) / 576.)
      IF (DP(I).GT.22.) GO TO 7080
      CS=CA
      GO TO 7090
7080  CS=CB
7090  HP(I)=(331.776 * Q(J)*Q(J)) / (CS * CS * DP(I)**5.25)
      HVP(I)=(VP(I)*VP(I)) /64.32
C
C        LOOK UP COST OF PIPE
C
      IF (DP(I).LE.10.) GO TO 8020
      IF (DP(I).LE.54.) GO TO 8010
      M=DP(I)/6. +12.2 -(RMC1/2.)
      GO TO 8030
8010  M=DP(I)/3. +3.2 -(RMC1/2.)
      GO TO 8030
8020  M=DP(I)/2. + 1.2 -(RMC1/2.)
8030  IF (NH.LE.NC) GO TO 8050
      WRITE (6,8040) DP(I),NH
8040  FORMAT (//* HEAD CLASS FOR PIPE *,F5.0,* IN. MUST BE *,F5.0)
      ONENTR=1.0
8050  IF (M.LE.MC) GO TO 8070
      WRITE (6,8060) DP(I)
8060  FORMAT (//* PIPE DECK COST MUST INCLUDE COST FOR *,F5.0,*IN.PIPE*)
      ONENTR=1.0
8070  COSTPI(I)=C(M,NH)
```

**** PROGRAM LAKMLP ****

```
8080  CONTINUE
       IC=IC+4
8100  CONTINUE
C
C        WRITE FINAL ANSWERS
5000  DO 5090 I=IS,N
      IF (AC(I).EQ.0.0) GO TO 5040
      IF (I.EQ.IS) GO TO 5040
      IF (LA(I).NE.LA(IS)) GO TO 5020
      WRITE (JW,5010) STA(I),HGL(I),AC(I),DQ(I),DEL(I),HGR(I),GS(I)
5010  FORMAT (1X,F10.2,3X,F8.2,41X,20X,F9.2,6X,F6.2,F5.1,F8.2,F8.2)
      GO TO 5060
5020  WRITE(JW,5030) STA(I),HGL(I),(LAT(I,II),II=1,2),AC(I),HGR(I),GS(I)
5030  FORMAT (1X,F10.2,3X,F8.2,41X,2A10,F9.2,17X,,F8.2,F8.2)
      GO TO 5060
5040  WRITE (JW,5050) STA(I),HGL(I),GS(I)
5050  FORMAT (1X,F10.2,3X,F8.2,95X,F8.2)
5060  WRITE(JW,5070)Q(I),DIA(I),V(I),HV(I),H(I),L(I),HF(I),SAC(I),HC(I)
5070  FORMAT (13X,F7.2,6X,F5.0,2F6.2,F9.6,F12.2,F7.2,24X,F10.2,26X,F5.0)
      IF (EQ(I).EQ.0.0) GO TO 5090
      WRITE (JW,5080) EQ(I)
5080  FORMAT (1H+,3HEQ=,F9.2)
5090  CONTINUE
6490  WRITE (7,6500) IS,N
6500  FORMAT (2I5)
      DO 6516 I=IS,N
      WRITE (7,6510) LA(I),STA(I),(LAT(I,II),II=1,2),AC(I),EQ(I),DQ(I),
     1 DEL(I),HGR(I),GS(I),Q(I),L(I),SAC(I),HC(I)
6510  FORMAT (I5,F10.2,2A10,2F9.2,F6.2,F5.1,2F8.2,F10.5,2F10.2,F5.0)
      IF (L(I).EQ.0.0) GO TO 6516
      IL=IL+1
      IL4=IL+4
      WRITE (7,6514) (DP(II),II=IL,IL4)
6514  FORMAT (5F10.2)
      IL=IL+4
6516  CONTINUE
      WRITE (7,6520) NL
6520  FORMAT (I5)
      WRITE (7,6515) LA(NL),STA(NL),AC(NL),DQ(NL),DEL(NL),HGR(NL),GS(NL)
6515  FORMAT (I5,F10.2,20X,F9.2,F6.2,F5.1,2F8.2)
      IF (DEL(I).EQ.0.0) GO TO 6000
      WRITE(JW,5010)STA(NL),HGL(NL),AC(NL),DQ(NL),DEL(NL),HGR(NL),GS(NL)
      GO TO 9999
6000  WRITE (JW,6010) STA(NL),HGL(NL),AC(NL),HGR(NL),GS(NL)
6010  FORMAT (1X,F10.2,3X,F8.2,41X,20X,F9.2,17X,2F8.2)
9999  RETURN
      END
```

```
                         ****  INPUT FIL- DATA1  ****

1.5  4.5        0.         1.53        7.1         .345        .370        6.
        9          1       7.684                   1100.       1110.
   MAIN
        1       0.         0.          0.          1110.
        1    1500.       110.        1090.         1089.
        2    2900.         0.          0.          1078.
     SUB 1
        2       0.         0.          0.          1078.
        3     400.         0.          0.          1079.
     SUB 1-A
        3       0.         0.          0.          1079.
        0     600.       110.        1082.         1078.
        0    1000.        90.        1085.         1083.
        1    2900.        75.        1080.         1077.
        2    4250.         0.          0.          1071.
     SUB 2
        2       0.         0.          0.          1071.       -250.
        0    1050.        80.        1075.         1073.
        0    5850.       100.        1060.         1058.
        8
```

**** INPUT FILE PIPRIC ****

PIPE SIZES 4 INCHES THROUGH 27 INCHES

4 30 20 EXAMPLE PROJECT PIPE PRICES DEC.1973

2.68	3.65	4.65	5.73	6.77	8.31	10.45	13.70	16.38	19.10
2.68	3.65	4.65	5.73	6.77	8.31	10.45	13.70	16.38	19.60
2.68	3.65	4.65	5.73	6.77	8.31	10.45	13.70	16.38	20.10
2.68	3.65	4.65	5.73	6.77	8.31	10.45	13.70	16.38	21.10
2.68	3.65	4.65	5.73	6.77	8.31	10.45	13.70	16.38	21.60
2.68	3.65	4.65	5.73	6.77	8.31	10.45	13.70	16.38	22.10
2.68	3.65	4.65	5.73	6.77	8.31	10.45	13.75	16.38	25.60
2.68	3.65	4.65	5.73	6.77	8.31	10.45	14.20	16.88	25.60
2.68	3.65	4.65	5.73	6.77	8.81	11.05	14.20	16.88	25.60
2.68	3.65	4.65	5.73	6.77	8.81	11.05	14.20	16.88	25.60
2.68	3.65	4.65	5.73	6.77	8.81	12.95	16.70	20.38	26.60
2.68	3.65	4.65	5.93	7.07	9.91	12.95	16.70	20.38	26.60
2.68	3.65	4.85	5.93	7.07	9.91	12.95	16.70	20.38	26.60
2.68	3.65	4.85	6.43	7.77	10.21	13.45	17.20	21.38	26.60
2.68	3.65	5.05	6.43	7.77	10.21	13.45	17.20	21.38	27.60
2.68	3.65	5.05	6.73	8.17	10.81	13.95	18.20	22.38	27.60
2.83	3.85	5.35	6.73	8.17	10.81	13.95	18.20	22.38	27.60
2.83	3.85	5.35	6.73	9.47	12.81	16.45	21.20	26.38	27.60
2.83	4.05	5.75	7.53	9.47	12.81	16.45	21.70	26.33	28.60
2.83	4.05	5.75	7.53	9.47	12.81	16.45	21.70	26.38	28.60

**** INPUT FILE PIPRIC ****

PIPE SIZES 30 INCHES THROUGH 57 INCHES

1.82	24.06	27.29	30.54	33.80	36.07	40.34	43.62	46.92	55.52
2.32	25.06	28.29	31.54	34.80	38.07	42.34	45.62	49.92	58.52
2.82	26.06	29.29	32.54	36.80	40.07	44.34	47.62	51.92	61.52
3.82	27.06	30.29	34.54	37.80	42.07	46.34	50.62	54.92	64.52
4.82	28.06	31.29	35.54	39.80	44.07	48.34	52.62	57.92	67.52
5.82	28.06	33.29	36.54	40.80	45.07	50.34	54.62	59.92	70.52
8.82	33.06	37.29	40.54	45.80	50.07	54.34	59.62	63.92	74.52
8.82	33.06	37.29	41.54	45.80	50.07	55.34	59.62	64.92	76.52
9.82	33.06	37.29	41.54	46.80	51.07	56.34	61.62	66.92	77.52
9.82	34.06	38.29	42.54	46.80	52.07	57.34	62.62	67.92	79.52
9.82	34.06	38.29	43.54	47.80	53.07	58.34	63.62	68.92	80.52
9.82	34.06	39.29	43.54	48.80	53.07	58.34	64.62	69.92	82.52
0.82	35.06	39.29	43.54	48.80	54.07	59.34	64.62	70.92	83.52
0.82	35.06	39.29	44.54	48.80	54.07	60.34	65.62	71.92	84.52
1.82	36.06	40.29	45.54	50.80	56.07	61.34	67.62	73.92	87.52
1.82	36.06	41.29	45.54	50.80	57.07	62.34	68.62	74.92	88.52
2.82	37.06	42.29	46.54	51.80	58.07	63.34	69.62	75.92	89.52
2.82	38.06	43.29	47.54	52.80	59.07	64.34	69.62	76.92	90.52
3.82	39.06	44.29	47.54	53.80	59.07	65.34	70.62	77.92	92.52
3.82	40.06	46.29	48.54	54.80	60.07	67.34	71.62	78.92	94.52

**** INPUT FILE PIPRIC ****

PIPE SIZES 60 INCHES THROUGH 66 INCHES

4.16	72.83
7.16	77.83
1.16	81.83
5.16	86.83
9.16	91.83
3.16	95.83
6.16	98.83
8.16	101.83
0.16	103.83
2.16	105.83
4.16	106.83
6.16	111.83
7.16	111.83
9.16	116.83
2.16	116.83
4.16	121.83
6.16	121.83
1.16	126.83
1.16	126.83
6.16	131.83

**** SUBROUTINE SUBSAC ****

```
      SUBROUTINE SACQ (NS,SAC,AC,Q,DQ,NF)
      DIMENSION SAC( 1 ),AC( 1 ),Q( 1 ),DQ( 1 )
C
C         EXAMPLE PROJECT  Q=6.1 GPM/AC     11-29-73
C
      DO 500 I=NS,NF
      Q(I)=SAC(I) *0.0136
  500 CONTINUE
      NF1=NF+1
      DO 600 I=NS,NF1
      DQ(I)=AC(I)    *0.0136
  600 CONTINUE
      RETURN
      END
```

DYNAMIC PROGRAMMING SOLUTION
(GRAVITY PROBLEM)

ECONOMICAL PIPE HYDRAULIC STUDY   MAIN   75/04/13.

PIPE CLASS DESIGN ELEVATION = 1110.00

STATION	Q	HGL	DIA	V	HV	SLOPE	LENGTH	LOSS	SUBLATERAL NAME	AC	SAC	DQ	DEL	HGR	NGS	HC
0.00		1100.00													1110.00	
	7.68		21.	3.19	.16	.001907	1500.00	2.86			565.00					25.
1500.00		1097.14								110.00		1.50	6.0	1090.00	1089.00	
	6.19		18.	3.50	.19	.002779	1400.00	3.89			455.00					50.
2900.00		1093.25							SUB 1	200.00				1089.73	1078.00	
	3.47		10.	6.36	.63	.019101	0.00	0.00			255.00					50.
2900.00		1093.25								75.00		1.02	6.0	1080.00	1077.00	
	2.45		10.	4.49	.31	.009518	1350.00	12.85			180.00					50.
4250.00		1080.40							SUB 2	80.00				1078.00	1071.00	
	1.36		8.	3.90	.24	.009479	1600.00	15.17			100.00					75.
5850.00		1065.23								100.00		1.36	6.0	1060.00	1058.00	

DYNAMIC PROGRAMMING SOLUTION
(GRAVITY PROBLEM)

ECONOMICAL PIPE HYCRAULIC STUDY SUB 1 75/04/13.

PIPE CLASS DESIGN ELEVATION = 1110.00

STATION	Q	HGL	DIA	V	HV	SLOPE	LENGTH	LOSS	SUBLATERAL NAME	AC	SAC	DQ	DEL	HGR	NGS	HC
0.00		1091.75													1078.00	
	2.72		12.	3.46	.19	.004512	400.00	1.80			200.00					50.
400.00		1089.94							SUB 1-A	110.00				1084.32	1079.00	
	1.22		8.	3.51	.19	.007678	600.00	4.61			90.00					50.
1000.00		1085.34								90.00		1.22	6.0	1085.00	1083.00	

DYNAMIC PROGRAMMING SOLUTION
(GRAVITY PROBLEM)

ECONOMICAL PIPE HYDRAULIC STUDY SUB 1-A 75/04/13.

PIPE CLASS DESIGN ELEVATION = 1110.00

STATION	Q	HGL	DIA	V	HV	SLOPE	LENGTH	LOSS	SUBLATERAL NAME	AC	SAC	DQ	DEL	HGR	NGS	HC
0.00		1088.44													1079.00	
	1.50		10.	2.74	.12	.003554	600.00	2.13			110.00					50.
600.00		1086.31								110.00		1.50	6.0	1082.00	1078.00	

DYNAMIC PROGRAMMING SOLUTION
(GRAVITY PROBLEM)

ECONCMICAL PIPE HYCRAULIC STUDY   SUB 2   75/04/13.

PIPE CLASS DESIGN ELEVATION = 1110.00

STATION	Q	HGL	DIA	V	HV	SLOPE	LENGTH	LOSS	SUBLATERAL NAME	AC	SAC	DQ	DEL	HGR	NGS	HC
0.00		1078.90													1071.00	
EQ= -250.00	1.09		10.	1.99	.06	.001880	800.00	1.50			80.00					50.
1050.00		1077.40								80.00		1.09	6.0	1075.00	1073.00	

**** PROGRAM LPFIN ****

```
      PROGRAM ZLPANS(LPANS,FTANS,TAPE12=LPANS,TAPE8=FTANS)
      DIMENSION WORD(7)
      EQUIVALENCE(WORD(1),IWORD)
      READ(12) X,(WORD(K),K=1,7)
      WRITE (8,100) WORD(7)
100   FORMAT ( 1X,F15.2 )
110   READ (12) X,(WORD(K),K=1,7)
      IF (IWORD.NE. 3HX11) GO TO 110
      GO TO 125
120   READ (12) X,(WORD(K),K=1,7)
125   IF (IWORD.EQ. 2HF1) GO TO 999
      IF (IWORD.EQ.7H$$ENDSS) GO TO  999
      WRITE (8,130) WORD(1),WORD(2)
130   FORMAT (1X,A10,F10.2)
      GO TO 120
999   CALL EXIT
      END
```

**** PROGRAM LEOANS ****

```
      PROGRAM REPORT (ANS,FTANS,OUTPUT,INFILE,TAPE8=FTANS,
     1    TAPE7=ANS,TAPE6=OUTPUT,TAPE10=INFILE,TAPE9)
C****    PROGRAM READS FROM TWO TAPES TO GENERATE A FINAL REPORT
C****       ANS - FILE CREATED BY LAKMLP
C****     FTANS - FILE CREATED BY LPFTN
C****
      COMMON HVDUM,TEELOS,IVALVE,IDUM
      DIMENSION LA1( 50),STA1( 50),HGL1( 50),LAT1(3),HV1( 50),
     2           LA2(125),STA2(125),HGL2(125),LAT2(50,2),HV2(125),
     3           LA3(230),STA3(230),HGL3(230),LAT3(125,2),HV3(230),
     4           LA4(175),STA4(175),HGL4(175),LAT4(230,2),HV4(175),
     5           LA5( 75),STA5( 75),HGL5( 75),LAT5(175,2),HV5( 75),
     6           WHGR1( 50),WHGR2(125),WHGR3(230),WHGR4(175),WHGR5( 75),
     7           DEL1( 50),DEL2(125),DEL3(230),DEL4(175),DEL5( 75),
     8           AC1( 50),AC2(125),AC3(230),AC4(175),AC5( 75),
     9           HGR1( 50),HGR2(125),HGR3(230),HGR4(175),HGR5( 75),
     1           GS1( 50),GS2(125),GS3(230),GS4(175),GS5( 75),
     2           EQ1( 50),EQ2(125),EQ3(230),EQ4(175),EQ5( 75)
      DIMENSION TODAY(2),TLOSS(50)
      CALL DAMOYR(TODAY)
      ISP=0
      HVDUM=0.0
      READ (7,100) (LAT1(II),II=1,3),ITYPE,STA1NIL,TEELOS,AVH1
  100 FORMAT (3A10,I10,3F10.2)
      WRITE (6,110) (LAT1(II),II=1,3),TODAY
  110 FORMAT (1H1,35H    ECONOMICAL PIPE HYDRAULIC STUDY,25X,3A10,25X,
     1 2A10//)
      GO TO (120,120,150,150),ITYPE
  120 READ (7,130) HVH,IDUM
  130 FORMAT (F10.2,I10)
      WRITE (6,140) HVH
  140 FORMAT (48X,30H PIPE CLASS DESIGN ELEVATION =,F10.2/)
      GO TO 170
  150 WRITE (6,160)
  160 FORMAT(38X,50HPIPE CLASS DESIGN ELEVATION = HYDRAULIC GRADE LINE/)
      READ (7,165) IDUM
  165 FORMAT (I10)
  170 GO TO (270,250,180,210),ITYPE
  180 READ (7,190) TKWSEL,STASAV
  190 FORMAT (2F10.2)
      WRITE (6,200) TKWSEL,STASAV
  200 FORMAT (28X,39H DOWN TO THE KNOWN TANK WATER SURFACE (,F10.2,
     1 9H ) AT STA,F10.2/)
      GO TO 230
  210 READ (7,190) TKWSEL,STASAV
      WRITE (6,220) TKWSEL,STASAV
  220 FORMAT (26X,42H DOWN TO THE COMPUTED TANK WATER SURFACE (,F10.2,
     1 9H ) AT STA,F10.2/)
```

**** PROGRAM LEOANS ****

```
 230   IF (STASAV.EQ.STA1NIL) GO TO 270
       WRITE (6,240) STASAV,TKWSEL
 240   FORMAT (25X,10H AFTER STA,F10.2,36H ,THE PIPE CLASS DESIGN ELEVATI
      1ON = ,F10.2, 10H + 5.0 FT./)
       GO TO 270
 250   WRITE (6,260) AVH1
 260   FORMAT (28X,10H WATER SURFACE IS (,F10.2,37H ) IN THE TANK NEAR TH
      1E PUMPING PLANT/)
 270   WRITE (6,280)
 280   FORMAT  (*  STATION          Q    HGL    DIA    V     HV    SLOPE
      1 LENGTH   LOSS  SUBLATERAL NAME     AC      SAC    DO    DEL     HGR
      2  NGS        HC*//)
C
C****     WRITE MAIN LATERAL
C
       READ (8,300) TOTANS
 300   FORMAT (1X,F15.2)
C****  IF (ITYPE.GT.2) GO TO 305
 305   CALL WRIANS (LA1,STA1,HGL1,HV1,50,AVH1,I1,N1,SULOSS1,WHGR1)
C****  WRITE (6,306) (WHGR1(I),I=1,N1)
C*06   FORMAT (1X,F8.2,/)
       N1=N1+1
       I=0
 310   I=I+1
       IF (I.GE.N1) GO TO 8000
       IF (LA1(I).NE.2) GO TO 310
C****     WRITE SUBS
       READ (7,320) (LAT2(I,II),II=1,2),IVALVE
 320   FORMAT (2A10,I10)
       WRITE (6,330) (LAT2(I,II),II=1,2),TODAY
 330   FORMAT (1H1,35H   ECONOMICAL PIPE HYDRAULIC STUDY,30X,2A10,30X,
      1 2A10//)
       GO TO (340,340,350,350),ITYPE
 340   AVH2=HGL1(I)
       HVDUM=HV1(I-1)
       GO TO 370
 350   IF (STA1(I).GT.STASAV) GO TO 360
       AVH2=TKWSEL
       GO TO 370
 360   AVH2=HGL1(I)
       HVDUM=HV1(I-1)
 370   READ (7,130) HVH
       WRITE (6,140) HVH
       WRITE (6,280)
       CALL WRIANS (LA2,STA2,HGL2,HV2,125,AVH2,I2,N2,SULOSS2,WHGR2)
C****  WRITE (6,306) (WHGR2(J),J=I2,N2)
       ISP = ISP + 1
       TLOSS(ISP) = SULOSS2
```

**** PROGRAM LEOANS ****

```
******WRITE (6,371) I2,N2
*71     FORMAT (2I10)
        N2=N2+1
        JB=0
 380    J=JB+I2
        JB=JB+1
******WRITE(6,381)J,LA2(J),STA2(J),HGL2(J)
*81     FORMAT (2I10,2F10.2)
        IF (J.GE.N2) GO TO 310
        IF (LA2(J).NE.3) GO TO 380
        READ (7,320) (LAT3(J,II),II=1,2),TVALVE
        WRITE (6,330) (LAT3(J,II),II=1,2),TODAY
        READ (7,130) HVH
        WRITE (6,140) HVH
        WRITE (6,280)
        HVDUM=HV2(J-1)
        AVH3=HGL2(J)
        CALL WRIANS (LA3,STA3,HGL3,HV3,230,AVH3,I3,N3,SULOSS3,WHGR3)
C****   WRITE (6,306) (WHGR3(K),K=I3,N3)
        ISP = ISP + 1
        TLOSS(ISP) = SULOSS3
        N3=N3+1
        KB=0
 390    K=KB+I3
        KB=KB+1
        IF (K.GE.N3) GO TO 380
        IF (LA3(K).NE.4) GO TO 390
        READ (7,320) (LAT4(K,II),II=1,2),TVALVE
        WRITE (6,330) (LAT4(K,II),II=1,2),TODAY
        READ (7,130) HVH
        WRITE (6,140) HVH
        WRITE (6,280)
        HVDUM=HV3(K-1)
        AVH4=HGL3(K)
        CALL WRIANS (LA4,STA4,HGL4,HV4,175,AVH4,I4,N4,SULOSS4,WHGR4)
        ISP =ISP + 1
        TLOSS(ISP) = SULOSS4
        N4=N4+1
        LB=0
 400    L=LB+I4
        LB=LB+1
        IF (L.GE.N4) GO TO 390
        IF (LA4(L).NE.5) GO TO 400
        READ (7,320) (LAT5(L,II),II=1,2),TVALVE
        WRITE (6,330) (LAT5(L,II),II=1,2),TODAY
        READ (7,130) HVH
        WRITE (6,140) HVH
        WRITE (6,280)
```

**** PROGRAM LEOANS ****

```
      HVDUM=HV4(L-1)
      AVH5=HGL4(L)
      CALL WRTANS (LA5,STA5,HGL5,HV5,75,AVH5,I5,N5,SULOSS5,WHGR5)
      ISP=ISP+1
      TLOSS(ISP)=SULOSS5
      N5=N5+1
      GO TO 400
8000  WRITE (6,7999) TOTANS
7999  FORMAT (///10X,*TOTAL COST OF PIPE IS $*,F15.2)
C**** WRITE (6,425) (TLOSS(IRQ),IRQ=1,ISP)
C*25  FORMAT (1X,F8.2,/)
      READ (10,430) TEELOS,PIPCOV,DWSADD,X,Y,CSS,CSB,PIPMIN,IPCOST
 430  FORMAT (2F5.2,3F10.2,2F10.3,F10.2,I10)
      EOADD = DWSADD
      DWSADD = 0.0
      WRITE (9,430) TEELOS,PIPCOV,DWSADD,X,Y,CSS,CSB,PIPMIN,IPCOST
      READ (10,435) MDEL,SENTRY,ITYPE,TQP,CP,AVH1,HVH1,TKWSEL,GDIFF,
     1   INPUOT
 435  FORMAT (2I5,I10,5F10.2,F5.2,I5)
      WRITE (9,435) MDEL,SENTRY,ITYPE,TQP,CP,AVH1,HVH1,TKWSEL,GDIFF,
     1    INPUOT
      READ (10,440) (LAT1(II),II=1,3)
 440  FORMAT (3A10,I10)
      WRITE (9,440) (LAT1(II),II=1,3),IDUM
      J1=0
      I=0
      J=0
      K=0
      L=0
      M=0
 445  I=I+1
      READ(10,450) DEL1(I),LA1(I),STA1(I),AC1(I),HGR1(I),GS1(I),EQ1(I)
 450  FORMAT (F5.2,I5,5F10.2)
      IF (I.EQ.1) GO TO 455
      IF (ITYPE.LE.2) GO TO 452
      IF(STA1(I).EQ.STASAV) WHGR1(I)=0.99
 452  IF (LA1(I).EQ.1) GO TO 460
      IF (LA1(I).EQ.2) GO TO 465
      HGR1(I) = HGR1(I) + EOADD
      WRITE (9,450) DEL1(I),LA1(I),STA1(I),AC1(I),HGR1(I),GS1(I),EQ1(I)
      GO TO 555
 455  WRITE (9,450) DEL1(I),LA1(I),STA1(I),AC1(I),HGR1(I),GS1(I),EQ1(I)
      GO TO 445
 460  WRITE (9,450) DEL1(I),LA1(I),STA1(I),AC1(I),WHGR1(I),GS1(I),EQ1(I)
      GO TO 445
 465  WRITE (9,450) DEL1(I),LA1(I),STA1(I),AC1(I),HGR1(I),GS1(I),EQ1(I)
      READ (10,470) (LAT2(I,II),II=1,2),IVALVE
 470  FORMAT (2A10,I10)
```

**** PROGRAM LECANS ****

```
      J1 = J1+1
      WRITE (9,475) (LAT2(I,II),II=1,2),IVALVE,TLOSS(J1),TLOSS(J1)
475   FORMAT (2A1 ,I1 ,2F8.2)
      TSAV2=TLOSS(J1)
      JJ = J+1
480   J = J+1
      READ (10,450) DEL2(J),LA2(J),STA2(J),AC2(J),HGR2(J),GS2(J),EQ2(J)
      IF (J.EQ.JJ) GO TO 485
      IF (LA2(J).EQ.2) GO TO 490
      IF (LA2(J).EQ.3) GO TO 495
      HGR2(J) = HGR2(J) + EOADD
      WRITE (9,450) DEL2(J),LA2(J),STA2(J),AC2(J),HGR2(J),GS2(J),EQ2(J)
      GO TO 445
485   WRITE (9,450) DEL2(J),LA2(J),STA2(J),AC2(J),HGR2(J),GS2(J),EQ2(J)
      GO TO 480
490   WRITE (9,450) DEL2(J),LA2(J),STA2(J),AC2(J),WHGR2(J),GS2(J),EQ2(J)
      GO TO 480
495   WRITE (9,450) DEL2(J),LA2(J),STA2(J),AC2(J),HGR2(J),GS2(J),EQ2(J)
      READ (10,470) (LAT3(J,II),II=1,2),IVALVE
      J1 = J1+1
      TLOSS(J1)=TLOSS(J1)+TSAV2
      TSAV=TLOSS(J1)-TSAV2
      WRITE (9,475) (LAT3(J,II),II=1,2),IVALVE,TLOSS(J1),TSAV
      TSAV3=TLOSS(J1)
      KK = K+1
500   K = K+1
      READ (10,450) DEL3(K),LA3(K),STA3(K),AC3(K),HGR3(K),GS3(K),EQ3(K)
      IF (K.EQ.KK) GO TO 505
      IF (LA3(K).EQ.3) GO TO 510
      IF (LA3(K).EQ.4) GO TO 515
      HGR3(K) = HGR3(K) + EOADD
      WRITE (9,450) DEL3(K),LA3(K),STA3(K),AC3(K),HGR3(K),GS3(K),EQ3(K)
      GO TO 480
505   WRITE (9,450) DEL3(K),LA3(K),STA3(K),AC3(K),HGR3(K),GS3(K),EQ3(K)
      GO TO 500
510   WRITE (9,450) DEL3(K),LA3(K),STA3(K),AC3(K),WHGR3(K),GS3(K),EQ3(K)
      GO TO 500
515   WRITE (9,450) DEL3(K),LA3(K),STA3(K),AC3(K),HGR3(K),GS3(K),EQ3(K)
      READ (10,470) (LAT4(K,II),II=1,2),IVALVE
      J1 = J1+1
      TLOSS(J1)=TLOSS(J1)+TSAV3
      TSAV=TLOSS(J1)-TSAV3
      WRITE (9,475) (LAT4(K,II),II=1,2),IVALVE,TLOSS(J1),TSAV
      TSAV4=TLOSS(J1)
      LL = L+1
520   L = L+1
      READ (10,450) DEL4(L),LA4(L),STA4(L),AC4(L),HGR4(L),GS4(L),EQ4(L)
      IF (L.EQ.LL) GO TO 525
```

**** PROGRAM LEOANS ****

```
      IF (LA4(L).EQ.4) GO TO 530
      IF (LA4(L).EQ.5) GO TO 535
      HGR4(L) = HGR4(L) + EOADD
      WRITE (9,450) DEL4(L),LA4(L),STA4(L),AC4(L),HGR4(L),GS4(L),EQ4(L)
      GO TO 500
  525 WRITE (9,450) DEL4(L),LA4(L),STA4(L),AC4(L),HGR4(L),GS4(L),EQ4(L)
      GO TO 520
  530 WRITE (9,450) DEL4(L),LA4(L),STA4(L),AC4(L),WHGR4(L),GS4(L),EQ4(L)
      GO TO 520
  535 WRITE (9,450) DEL4(L),LA4(L),STA4(L),AC4(L),HGR4(L),GS4(L),EQ4(L)
      READ (10,470) (LAT5(L,II),II=1,2),IVALVE
      J1 =J1+1
      TLOSS(J1)=TLOSS(J1)+TSAV4
      TSAV=TLOSS(J1)-TSAV4
      WRITE (9,475) (LAT5(L,II),II=1,2),IVALVE,TLOSS(J1),TSAV
      MM = M+1
  540 M = M+1
      READ (10,450) DEL5(M),LA5(M),STA5(M),AC5(M),HGR5(M),GS5(M),EQ5(M)
      IF (M.EQ.MM) GO TO 545
      IF ( LA5(M).EQ.5) GO TO 550
      HGR5(M) = HGR5(M) + EOADD
      WRITE (9,450) DEL5(M),LA5(M),STA5(M),AC5(M),HGR5(M),GS5(M),EQ5(M)
      GO TO 520
  545 WRITE (9,450) DEL5(M),LA5(M),STA5(M),AC5(M),HGR5(M),GS5(M),EQ5(M)
      GO TO 540
  550 WRITE (9,450) DEL5(M),LA5(M),STA5(M),AC5(M),WHGR5(M),GS5(M),EQ5(M)
      GO TO 540
  555 WRITE (9,556)
  556 FORMAT (*          */*          *)
      REWIND 9
      CALL REPPLAC (5HTAPE9,6HOUTFIL)
      CALL EXIT
      END
      SUBROUTINE WRIANS (LA,STA,HGL,HV,ID,AVH,IS,NSAV,SULOSS,WHGR)
      COMMON HVDUM,TEELOS,IVALVE,IDUM
      DIMENSION LAD(300),STAD(300),LATD(300,2),ACD(300),EQD(300),
     1          DQD(300),DELD(300),HGRD(300),GSD(300),QD(300),LD(300),
     2          SACD(300),HCD(300),
     3          DP(300,5),LR(300,5),LRJ(300,5),DPJ(300,5)
      DIMENSION LA(ID),STA(ID),HGL(ID),HV(ID),WHGR(ID)
      DIMENSION LAT(300,2),AC(300),DQ(300),DEL(300),HGR(300),GS(300),
     1          Q(300),D(300),V(300),H(300),L(300),HF(300),
     2          SAC(300),HC(300),EQ(300)
      REAL L,LD,LRJ,LR
      UNDER1=7404740474047404B
      UNDER2=740474047404B
C
C****    READ THE TWO INPUT FILES
```

**** PROGRAM LEOANS ****

```
C
      READ (7,100)IS,N
  100 FORMAT (2I5)
      DO 150 I=IS,N
      READ (7,110) LAD(I),STAD(I),(LATD(I,II),II=1,2),ACD(I),EQD(I),
     1    DDD(I),DELD(I),HGRD(I),GSD(I),QD(I),LD(I),SACD(I),HCD(I)
  110 FORMAT (I5,F10.2,2A10,2F9.2,F6.2,F5.1,2F8.2,F10.5,2F1 .2,F5.0)
      IF (LD(I).EQ.0.0) GO TO 150
      READ (7,120) (DP(I,II),II=1,5)
  120 FORMAT (5F10.2)
      DO 140 J=1,5
      READ (8,130) LR(I,J)
  130 FORMAT (11X,F10.2)
  140 CONTINUE
  150 CONTINUE
      READ (7,160) NL
  160 FORMAT (I5)
      READ (7,170) LAD(NL),STAD(NL),ACD(NL),DDD(NL),DELD(NL),HGRD(NL),
     1    GSD(NL)
  170 FORMAT (I5,F10.2,20X,F9.2,F6.2,F5.1,2F8.2)
C
C****    ADD TWO ARRAYS TO FORM ONE (NEW SUBSCRIPTS)
C
      NSAV=N
      IC=IS
  180 DO 240 I=IS,N
      IF (LD(I).NE.0.0) GO TO 185
      JCON=1
      L(IC)=0.0
      GO TO 200
  185 JCON=0
      DO 190 J=1,5
      IF (LR(I,J).EQ.0.0) GO TO 190
      JCON=JCON + 1
      LRJ(I,JCON) = LR(I,J)
      DPJ(I,JCON) = DP(I,J)
  190 CONTINUE
  200 LA(IC)=LAD(I)
      STA(IC)=STAD(I)
      LAT(IC,1)=LATD(I,1)
      LAT(IC,2)=LATD(I,2)
      AC(IC)=ACD(I)
      EQ(IC)=EQD(I)
      DD(IC)=DDD(I)
      DEL(IC)=DELD(I)
      HGR(IC)=HGRD(I)
      GS(IC)=GSD(I)
      Q(IC)=QD(I)
```

**** PROGRAM LEOANS ****

```
      SAC(IC)=SACD(I)
      HC(IC)=HCD(I)
      IF (LD(I).EQ.0.0) GO TO 230
      GO TO (201,202,203,203,203),JCON
201   D(IC)=DPJ(I,1)
      L(IC)=LRJ(I,1)
      GO TO 222
202   D(IC)=DPJ(I,1)
      LA(IC+1)=LAD(TS)
      D(IC+1)=DPJ(I,2)
      STA(IC+1)=STA(IC) + LRJ(I,1)
       Q(IC+1)=QD(I)
      AC(IC+1)=0.0
      EQ(IC+1)=EQ(IC)
      EQ(IC)=0.0
      SAC(IC+1)=SACD(I)
      GS(IC+1)=1.0
      HCB=HCD(I)
      HCA=HCD(I+1)
      HC(IC+1)=AMAX1(HCB,HCA)
      L(IC)=LRJ(I,1)
      L(IC+1)=LRJ(I,2)
      GO TO 210
203   WRITE (6,204)
204   FORMAT (1H1,* E R R O R - ADD MORE STATEMENTS AT 203 *)
      CALL EXIT
210   J=I+1
      IG=IC+JCON
      DD 220 M=J,N
      LA(IG)=LAD(M)
      STA(IG)=STAD(M)
      LAT(IG,1)=LATD(M,1)
      LAT(IG,2)=LATD(M,2)
      AC(IG)=ACD(M)
      EQ(IG)=EQD(M)
      DQ(IG)=DQD(M)
      DEL(IG)=DELD(M)
      HGR(IG)=HGRD(M)
      GS(IG)=GSD(M)
      Q(IG)=QD(M)
      SAC(IG)=SACD(M)
      HC(IG)=HCD(M)
      IG=IG+1
220   CONTINUE
      NSAV=NSAV+JCON-1
 222  NSVL=NSAV+1
      LA(NSVL)=LAD(NL)
      STA(NSVL)=STAD(NL)
```

**** PROGRAM LEOANS ****

```
      AC(NSVL)=ACD(NL)
      DO(NSVL)=DOD(NL)
      DEL(NSVL)=DELD(NL)
      HGR(NSVL)=HGRD(NL)
      GS(NSVL)=GSD(NL)
  230 IC=IC+JCON
  240 CONTINUE
      CSS=.345
      CSB=.370
C
C****    COMPUTE PIPE HYDRAULIC PROPERTIES
C
 1000 DO 1030 I=IS,NSAV
      HF(I)=0.0
      IF (L(I).EQ.0.0) GO TO 1025
      V(I)=Q(I) / ((3.14159*D(I)*D(I)) /576.)
      IF (D(I).GT.22.) GO TO 1010
      CS=3.14159*CSS
      GO TO 1020
 1010 CS=3.14159*CSB
 1020 H(I)=(331.776*Q(I)*Q(I)) / (CS*CS*D(I)**5.25)
      HV(I)=(V(I)*V(I)) /64.32
      HF(I)=H(I)*L(I)
      GO TO 1030
 1025 HV(I)=HV(I-1)
 1030 CONTINUE
C
C****    COMPUTE HYDRAULIC GRADE LINE
C
      IF (LA(IS).EQ.1) GO TO 4098
      IF (IVALVE.EQ.0) GO TO 4000
      TRULS=1.25*HV(IS)
      WRITE (6,1060) AVH,HVDUM,TRULS
 1060 FORMAT(1X*AVAILABLE HEAD =*7X,F8.2/1X*VELOCITY HEAD =*8X,F8.2
     1/1X*TEE LOSS =*13X,F8.2)
      SULOSS=HVDUM+TRULS
      GO TO 4999
 4000 IF(D(IS).LE.10.) GO TO 4015
      IF (V(IS).LT.10.0)GO TO 4020
 4005 TRULS = 1.25 * HV(IS)
      VRULS = .4 * HV(IS)
      WRITE (6,4010) AVH,HVDUM,TRULS,D(IS),VRULS
 4010 FORMAT (1X*AVAILABLE HEAD =*7X,F8.2
     1/1X*VELOCITY HEAD =*8X,F8.2
     1/1X*TEE LOSS =*13X,F8.2
     1/1X,F3.0* INCH VALVE LOSS =*2X,F8.2  )
      TRLS2=0.0
      GO TO 4998
```

**** PROGRAM LEOANS ****

```
 4020 VSAVE = V(IS)
      DSAVE = D(IS)
      IK = 10
      IL = 20
      IM = 2
 4030 DO 4050  IO=IK,IL,IM
      DTEST = IO
      VTEST = Q(IS)/((3.14159*DTEST*DTEST)/576.)
      IF (VTEST.LT.VSAVE.OR.VTEST.GT.10.0) GO TO 4040
      VSAVE = VTEST
      DSAVE = DTEST
 4040 IF (DTEST.GE.D(IS)) GO TO 4060
 4050 CONTINUE
      IF (DTEST.EQ.72) GO TO 4060
      IK = 24
      IL = 72
      IM = 6
      GO TO 4030
 4060 HVSAVE = VSAVE*VSAVE/64.32
      TPLS2 = .15 * (HVSAVE-HV(IS))
      TRULS = 1.25 * HVSAVE
      VRULS = .4 * HVSAVE
      WRITE (6,4070) AVH,HVDUM,TRULS,
     1   DSAVE,VRULS,DSAVE,D(IS),
     1TPLS2
 4070 FORMAT (1X*AVAILABLE HEAD =*7X,F8.2
     1/1X*VELOCITY HEAD =*8X,F8.2
     1/1X*TEE LOSS =*,13X,F8.2
     1/1X,F3.0* INCH VALVE LOSS =*2X,F8.2
     1/1X*TAPER *,F3.0* TO *,F3.0* LOSS =*,F8.2)
 4998 SULOSS=HVDUM + TRULS + VRULS + TPLS2
 4999 WRITE (6,1070)
1070  FORMAT (24X,*---------*)
      WRITE (6,4080) SULOSS
4080  FORMAT (1X,*TOTAL LOSS*,13X,F8.2/)
      GO TO 5000
4998  HGL(IS)=AVH
      GO TO 5001
5000  HGL(IS)=AVH-SULOSS
5001  DO 1050 I=IS,NSAV
      HGL(I+1)=HGL(I)-HF(I)
1050  CONTINUE
      J = IS-1
      DO 5030 I=IS,NSAV
      IF (GS(I).EQ.0.) GO TO 5005
      J = J+1
5005  IF (AC(I).EQ.0.0) GO TO 5040
      IF (I.EQ.IS) GO TO 5040
```

**** PROGRAM LECANS ****

```
      IF (LA(I).NE.LA(IS)) GO TO 5020
      IF (IDUM.EQ.1) GO TO 5008
      WHGR(J) = HGR(I)
      GO TO 5009
 5008 WHGR(J) = HGR(I) + HV(I-1)
 5009 WRITE (6,5010) STA(I),HGL(I),AC(I),DQ(I),DEL(I),HGR(I),GS(I)
 5010 FORMAT (1X,F10.2,3X,F8.2,41X,20X,F9.2,6X,F6.2,F5.1,F8.2,F8.2)
      GO TO 5055
 5020 WRITE (6,5030) STA(I),HGL(I),(LAT(I,II),II=1,2),AC(I),GS(I)
 5030 FORMAT (1X,F10.2,3X,F8.2,41X,2A10,F9.2,25X,F8.2)
      HGR(I)=0.0
      GO TO 5055
 5040 WRITE (6,5050) STA(I),HGL(I),GS(I)
 5050 FORMAT (1X,F10.2,3X,F8.2,95X,F8.2)
 5055 IF (ABS(HGL(I)-HGR(I)).GT.0.1) GO TO 5058
      WRITE (6,5059) UNDER1,UNDER2,UNDER1,UNDER2
 5059 FORMAT (1H+,10X,R8,R6,88X,R8,R6)
 5058 IF (L(I).NE.0.0) GO TO 5060
      WRITE (6,5066) Q(I),SAC(I),HC(I)
 5066 FORMAT (13X,F7.2,75X,F10.2,26X,F5.0)
      GO TO 5075
 5060 WRITE (6,5070) Q(I),D(I),V(I),HV(I),H(I),L(I),HF(I),SAC(I),HC(I)
 5070 FORMAT (13X,F7.2,6X,F5.0,2F6.2,F9.6,F12.2,F7.2,24X,F10.2,26X,F5.0)
 *075 WRITE (6,5076) LA(I)
 *076 FORMAT (1H+,19X,I5)
 5075 IF (EQ(I).EQ.0.0) GO TO 5090
      WRITE (6,5080) EQ(I)
 5080 FORMAT (1H+,3HEQ=,F9.2)
 5090 CONTINUE
      NL=NSAV+1
      IF (DEL(NL).EQ.0.0) GO TO 6000
      WRITE (6,5010)STA(NL),HGL(NL),AC(NL),DQ(NL),DEL(NL),HGR(NL),GS(NL)
      GO TO 6020
 6000 WRITE (6,6010) STA(NL),HGL(NL),AC(NL),HGR(NL),GS(NL)
 6010 FORMAT (1X,F10.2,3X,F8.2,41X,20X,F9.2,17X,2F8.2)
 6020 IF (ABS(HGL(NL)-HGR(NL)). GT . 0.1) GO TO 9999
      WRITE (6,5059) UNDER1,UNDER2,UNDER1,UNDER2
 9999 RETURN
      END
      SUBROUTINE DAMOYR (TODAY)
      DIMENSION MON(12),JDATE(3),TODAY(2),CKBL(6)
      DATA (MON(I),I=1,12)/10H JANUARY  ,10H FEBRUARY ,10H MARCH    ,
     1 10H APRIL    ,10H MAY      ,10H JUNE     ,10H JULY     ,
     2 10H AUGUST   ,10H SEPTEMBER,10H OCTOBER  ,10H NOVEMBER ,
     3 10H DECEMBER /
      CALL DATE(DAY)
      DECODE (10,75,DAY) JDATE(3),JDATE(2),JDATE(1)
      J=JDATE(2)
```

**** PROGRAM LEOANS ****

```
      JDATE(2)=MON(J)
      JDATE(3)=JDATE(3) + 1900
      DECODE (10,85,JDATE(2)) CKBL
      DO 80 I=1,6
      IF (CKBL(I).EQ.1H ) GO TO 90
   80 CONTINUE
      I=7
   90 J=I+3
      ENCODE (11,95,FMT) J
      ENCODE (20,FMT,TODAY) JDATE
   75 FORMAT (3(1X,I2))
   85 FORMAT (4X,6(A1))
   95 FORMAT (5H(I2,A,I2,4H,I5))
      RETURN
      END
```

FINAL REPORT
(GRAVITY PROBLEM)

ECONOMICAL PIPE HYDRAULIC STUDY — MAIN — 13 APRIL 1975

PIPE CLASS DESIGN ELEVATION = 1110.00

STATION	Q	HGL	DIA	V	HV	SLOPE	LENGTH	LOSS	SUBLATERAL NAME	AC	SAC	DQ	DEL	HGR	NGS	HC
0.00		1100.00													1110.00	
	7.68		18.	4.35	.29	.004285	1500.00	6.43			565.00					25.
1500.00		1093.57								110.00		1.50	6.0	1090.00	1089.00	
	6.19		18.	3.50	.19	.002779	1400.00	3.89			455.00					50.
2900.00		1089.68							SUB 1	200.00					1078.00	
	3.47										255.00					50.
2900.00		1089.68								75.00		1.02	6.0	1080.00	1077.00	
	2.45		12.	3.12	.15	.003655	771.11	2.82			180.00					50.
3671.11		1086.87													0.00	
	2.45		10.	4.49	.31	.009518	578.89	5.51			180.00					75.
4250.00		1081.36							SUB 2	80.00					1071.00	
	1.36		8.	3.90	.24	.009479	1415.00	13.41			100.00					75.
5665.00		1067.94													0.00	
	1.36		6.	6.93	.75	.042923	185.00	7.94			100.00					75.
5850.00		1060.00								100.00		1.36	6.0	1060.00	1058.00	

FINAL REPORT
(GRAVITY PROBLEM)

ECONOMICAL PIPE HYDRAULIC STUDY — SUB 1 — 13 APRIL 1975

PIPE CLASS DESIGN ELEVATION = 1110.00

AVAILABLE HEAD = 1089.68
VELOCITY HEAD = .19
TEE -- VALVE LOSS = 1.31

STATION	Q	HGL	DIA	V	HV	SLOPE	LENGTH	LOSS	SUBLATERAL NAME	AC	SAC	DQ	DEL	HGR	NGS	HC
0.00		1088.18													1078.00	
	2.72		15.	2.22	.08	.001398	16.15	.02			200.00					50.
16.15		1088.16													0.00	
	2.72		12.	3.46	.19	.004512	383.85	1.73			200.00					50.
400.00		1086.43							SUB 1-A	110.00					1079.00	
	1.22		10.	2.24	.08	.002379	600.00	1.43			90.00					50.
1000.00		1085.00								90.00		1.22	6.0	1085.00	1083.00	

FINAL REPORT
(GRAVITY PROBLEM)

ECONOMICAL PIPE HYDRAULIC STUDY   SUB 1-A   13 APRIL 1975

PIPE CLASS DESIGN ELEVATION = 1110.00

AVAILABLE HEAD = 1086.43
VELOCITY HEAD = .19
TEE -- VALVE LOSS = 1.31

STATION	Q	HGL	DIA	V	HV	SLOPE	LENGTH	LOSS	SUBLATERAL NAME	AC	SAC	DQ	DEL	HGR	NGS	HC
0.00		1084.93													1079.00	
	1.50		10.	2.74	.12	.003554	499.57	1.78			110.00					50.
499.57		1083.15													0.00	
	1.50		8.	4.29	.29	.011470	100.43	1.15			110.00					50.
600.00		1082.00								110.00		1.50	6.0	1082.00	1078.00	

FINAL REPORT
(GRAVITY PROBLEM)

ECONOMICAL PIPE HYDRAULIC STUDY   SUB 2   13 APRIL 1975

PIPE CLASS DESIGN ELEVATION = 1110.00

STATION	Q	HGL	DIA	V	HV	SLOPE	LENGTH	LOSS	SUBLATERAL NAME	AC	SAC	DQ	DEL	HGR	NGS	HC
AVAILABLE HEAD =		1081.36														
VELOCITY HEAD =		.31														
TEE -- VALVE LOSS =		1.19														

0.00		1079.86													1071.00	
EQ= -250.00	1.09		8.	3.12	.15	.006067	800.00	4.85			80.00					50.
1050.00		1075.00								80.00		1.09	6.0	1075.00	1073.00	

TOTAL COST OF PIPE IS $ 59317.86

**** INPUT FILE DATA2 ****

```
1.5  4.5       0.       1.53      7.1       .345      .370      6.
         9          2     5.551       .48
 MAIN LATERAL
         1       0.       0.        0.09     110.
         1     200.      47.      195.       100.
         2    2000.       0.        0.        98.
SUB LATERAL  A
         2       0.       0.        0.        98.
         2    1200.      80.      192.        97.
         0    2549.83    12.2     185.        90.
         2    4900.       0.        0.        86.
SUBLATERAL  B
         2       0.       0.        0.        86.
         3    1000.       0.        0.        76.
SUB LATERAL  B-1
         3     100.       0.        0.        76.
         0    2019.87   100.      161.       66.
         0    2100.      80.      141.       46.
         1    6300.      64.      151.        56.
         0    7800.      25.      157.0      62.
         8
```

DYNAMIC PROGRAMMING SOLUTION
(PUMP PROBLEM)

ECONOMICAL PIPE HYDRAULIC STUDY   MAIN LATERAL   75/04/13.

PIPE CLASS DESIGN ELEVATION =   236.00

WATER SURFACE IS (231.00) IN THE TANK NEAR THE PUMPING PLANT

STATION	Q	HGL	DIA	V	HV	SLOPE	LENGTH	LOSS	SUBLATERAL NAME	AC	SAC	DQ	DEL	HGR	NGS	HC
0.00		231.00													110.00	
	5.55		15.	4.52	.32	.005824	200.00	1.16			408.20					150.
200.00		229.84								47.00		.64	4.0	195.00	100.00	
	4.91		12.	6.25	.61	.014716	1800.00	26.49			361.20					150.
2000.00		203.35							SUB LATERAL A	92.20				203.17	98.00	
	3.66		12.	4.66	.34	.008162	2900.00	23.67			269.00					150.
4900.00		179.68							SUBLATERAL B	180.00				179.16	86.00	
	1.21		8.	3.47	.19	.007508	1400.00	10.51			89.00					175.
6300.00		169.17								64.00		.87	4.0	151.00	56.00	
	.34		6.	1.73	.05	.002683	1500.00	4.02			25.00					200.
7900.00		165.14								25.00		.34	3.0	157.00	62.00	

DYNAMIC PROGRAMMING SOLUTION
(PUMP PROBLEM)

ECONOMICAL PIPE HYDRAULIC STUDY　　　SUB LATERAL A　　　75/04/13.

PIPE CLASS DESIGN ELEVATION = 236.00

STATION	Q	HGL	DIA	V	HV	SLOPE	LENGTH	LOSS	SUBLATERAL NAME	AC	SAC	DQ	DEL	HGR	NGS	HC
0.00		201.85													98.00	
	1.25		8.	3.59	.20	.008058	1200.00	9.67			92.20					150.
1200.00		192.18								80.00		1.09	6.0	192.00	97.00	
	.17		6.	.85	.01	.000639	1349.83	.86			12.20					150.
2549.83		191.32								12.20		.17	2.0	185.00	90.00	

DYNAMIC PROGRAMMING SOLUTION
(PUMP PROBLEM)

ECONOMICAL PIPE HYDRAULIC STUDY SUBLATERAL B 75/04/13.

PIPE CLASS DESIGN ELEVATION = 236.00

STATION	Q	HGL	DIA	V	HV	SLOPE	LENGTH	LOSS	SUBLATERAL NAME	AC	SAC	DQ	DEL	HGR	NGS	HC
0.00		178.18													86.00	
	2.45		10.	4.49	.31	.009518	1000.00	9.52			180.00					175.
1000.00		168.56							SUB LATERAL B-1	100.00				168.14	76.00	
	1.09		9.	3.12	.15	.006067	1100.00	6.67			80.00					200.
2100.00		161.99								80.00		1.09	6.0	141.00	46.00	

DYNAMIC PROGRAMMING SOLUTION (PUMP PROBLEM)

ECONOMICAL PIPE HYCRAULIC STUDY   SUB LATERAL B-1   75/04/13.

PIPE CLASS DESIGN ELEVATION = 236.00

STATION	Q	HGL	DIA	V	HV	SLOPE	LENGTH	LOSS	SUBLATERAL NAME	AC	SAC	DQ	DEL	HGR	NGS	HC
100.00		167.16													76.00	
	1.36		10.	2.49	.10	.002938	1919.87	5.64			100.00					175.
2019.87		161.52								100.00		1.36	6.0	161.00	66.00	

FINAL REPORT
(PUMP PROBLEM)

ECONOMICAL PIPE HYDRAULIC STUDY — MAIN LATERAL — 13 APRIL 1975

PIPE CLASS DESIGN ELEVATION = 236.00

WATER SURFACE IS (231.00) IN THE TANK NEAR THE PUMPING PLANT

STATION	Q	HGL	DIA	V	HV	SLOPE	LENGTH	LOSS	SUBLATERAL NAME	AC	SAC	DQ	DEL	HGR	NGS	HC
0.00		231.00													110.00	
	5.55		15.	4.52	.32	.005824	200.00	1.16			408.20					150.
200.00		229.84								47.00		.64	4.0	195.00	100.00	
	4.91		15.	4.00	.25	.004560	1185.49	5.41			361.20					150.
1385.49		224.43													0.00	
	4.91		12.	6.25	.61	.014716	614.51	9.04			361.20					150.
2000.00		215.39							SUB LATERAL A	92.20					98.00	
	3.66		12.	4.66	.34	.008162	2900.00	23.67			269.00					150.
4900.00		191.72							SUBLATERAL B	180.00					86.00	
	1.21		8.	3.47	.19	.007508	638.20	4.79			89.00					175.
5538.20		186.92													0.00	
	1.21		6.	6.16	.59	.033999	761.80	25.90			89.00					200.
6300.00		161.02								64.00		.87	4.0	151.00	56.00	
	.34		6.	1.73	.05	.002683	1500.00	4.02			25.00					200.
7800.00		157.00								25.00		.34	3.0	157.00	62.00	

FINAL REPORT
(PUMP PROBLEM)

ECONOMICAL PIPE HYDRAULIC STUDY   SUB LATERAL A   13 APRIL 1975

PIPE CLASS DESIGN ELEVATION = 236.00

AVAILABLE HEAD = 215.39
VELOCITY HEAD = .61
TEE -- VALVE LOSS = .89

STATION	Q	HGL	DIA	V	HV	SLOPE	LENGTH	LOSS	SUBLATERAL NAME	AC	SAC	DQ	DEL	HGR	NGS	HC
0.00		213.89													98.00	
	1.25		8.	3.59	.20	.008058	770.29	6.21			92.20					150.
770.29		207.68													0.00	
	1.25		6.	6.39	.63	.036488	429.71	15.68			92.20					150.
1200.00		192.00								80.00		1.09	6.0	192.00	97.00	
	.17		6.	.85	.01	.000639	1349.83	.86			12.20					150.
2549.83		191.14								12.20		.17	2.0	185.00	90.00	

FINAL REPORT
(PUMP PROBLEM)

ECONOMICAL PIPE HYDRAULIC STUDY   SUBLATERAL B   13 APRIL 1975

PIPE CLASS DESIGN ELEVATION = 236.00

STATION	Q	HGL	DIA	V	HV	SLOPE	LENGTH	LOSS	SUBLATERAL NAME	AC	SAC	DQ	DEL	HGR	NGS	HC
AVAILABLE HEAD =		191.72														
VELOCITY HEAD =		.34														
TEE -- VALVE LOSS =		1.16														

0.00		190.22													86.00	
	2.45		10.	4.49	.31	.009518	1000.00	9.52			180.00					175.
1000.00		180.70							SUB LATERAL B-1	100.00					76.00	
	1.09		6.	5.54	.48	.027471	1100.00	30.22			80.00					200.
2100.00		150.48								80.00		1.09	6.0	141.00	46.00	

FINAL REPORT
(PUMP PROBLEM)

ECONOMICAL PIPE HYDRAULIC STUDY — SUB LATERAL B-1 — 13 APRIL 1975

PIPE CLASS DESIGN ELEVATION = 236.00

AVAILABLE HEAD = 180.70
VELOCITY HEAD = .31
TEE -- VALVE LOSS = 1.19

STATION	Q	HGL	DIA	V	HV	SLOPE	LENGTH	LOSS	SUBLATERAL NAME	AC	SAC	DQ	DEL	HGR	NGS	HC
100.00		179.20													76.00	
	1.36		8.	3.90	.24	.009479	1919.87	18.20			100.00					175.
2019.87		161.00								100.00		1.36	6.0	161.00	66.00	

TOTAL COST OF PIPE IS $ 75279.43

BIBLIOGRAPHY

1. Camp, T. R., Economic Pipe Sizes for Water Distribution Systems: Am. Soc. Civil Engineers Trans., V. 4, pp. 194-203 (1939)

2. Mandry, J. E., Design of Pipe Distribution Systems for Sprinkler Projects: Am. Soc. Civil Engineers Trans., V. 93, pp. 243-257 (1967)

3. Calhoun, C. A. Optimization of Pipe Systems by Linear Programming: Colo. State Univ., Proc. of the Inst. on Control of Flow in Closed Conduits, 22 p. (1970)

4. Karmeli, D., Gadish, Y., and Meyers, S., Design of Optimal Water Distribution Systems: Am. Soc. Civil Engineers Trans., V. 94, pp. 1-10 (1968)

5. ____________, Design of Optimal Water Distribution Systems by Dynamic Programming: Haifa, Israel, Technion, pub. no. 51, 19 p. (1968)

6. Kally, E., Pipeline Planning by Dynamic Computer Programming; Am. Water Works Assoc. Jour., V. 61, no. 3, pp. 114-118 (1969)

7. Liang, T., Design Conduit System by Dynamic Programming: Am. Soc. Civil Engineers Trans., V. 97, pp. 383-393 (1971)

BIBLIOGRAPHY Continued

8. Simonds, R. A., and Kinney, L. A., Jr., Application of Operations Research Methods for USBR Pipe Distribution Systems: Atlanta, Ga., Am. Soc. Civil Engineers, 27 p. (1972)

9. Hillier, F. S., and Lieberman, G. J., Introduction to Operations Research: San Francisco, Calif., Holden-Day, Inc., 639 p. (1970)

10. Beveridge, G.S.G., and Schechter, R. S., Optimization: Theory and Practice, San Francisco, Calif., McGraw-Hill Book Co., pp. 340-346 (1970)