

2

NAVAL POSTGRADUATE SCHOOL
Monterey, California

AD-A268 826



DTIC
ELECTE
SEP 0 1 1993
S B D

THESIS

**PRELIMINARY DESIGN OF A WATER COOLED
AVIONICS COOLING RACK**

by

Colleen Laverna Ellis

March, 1993

Thesis Advisor:

Allan D. Kraus

Approved for public release; distribution is unlimited.

93-20283



93 8 31 003

Unclassified

Security Classification of this page

REPORT DOCUMENTATION PAGE

1a Report Security Classification: Unclassified			1b Restrictive Markings		
2a Security Classification Authority			3 Distribution/Availability of Report		
2b Declassification/Downgrading Schedule			Approved for public release; distribution is unlimited.		
4 Performing Organization Report Number(s)			5 Monitoring Organization Report Number(s)		
6a Name of Performing Organization Naval Postgraduate School		6b Office Symbol (if applicable) 31	7a Name of Monitoring Organization Naval Postgraduate School		
6c Address (city, state, and ZIP code) Monterey CA 93943-5000			7b Address (city, state, and ZIP code) Monterey CA 93943-5000		
8a Name of Funding/Sponsoring Organization		8b Office Symbol (if applicable)	9 Procurement Instrument Identification Number		
Address (city, state, and ZIP code)			10 Source of Funding Numbers		
			Program Element No	Project No	Task No
			Work Unit Accession No		
11 Title (include security classification) Preliminary Design of a Water Cooled Avionics Cooling Rack					
12 Personal Author(s) Ellis, Colleen L.					
13a Type of Report Master's Thesis		13b Time Covered From To	14 Date of Report (year, month, day) 1993, March, 25	15 Page Count 47	
16 Supplementary Notation The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.					
17 Cosati Codes			18 Subject Terms (continue on reverse if necessary and identify by block number)		
Field	Group	Subgroup	Avionics, Cooling Rack, Water Cooled Avionics		
19 Abstract (continue on reverse if necessary and identify by block number)					
Military electronics are frequently operated in excessively confined spaces aboard ships and aircraft. This limited space impacts significantly on the space available for cooling equipment. The optimal solution is the development of one universal, modular rack for shipboard and aviation use. With a modular design, upgrades to equipment could also be accompanied by an upgrade to the cooling rack itself with very little additional cost or difficulty. A water cooled avionics rack can provide sufficient cooling for any piece or combination of avionics equipment if enough water flow paths are used, the water is at the appropriate temperature and the water is properly distributed within the passages. To determine if the cooling medium, water is properly distributed within a modular cooling rack, an analysis of the flow and pressure distribution of the coolant is required. This thesis presents a computer code that has been developed as an initial step in the total design of a modular cooling rack for avionics equipment. In itself, the code details a specific design technique and allows for the determination of whether a proposed configuration, including source location, characteristics of the cooling water, and the size and shape of the proposed flow passages will indeed provide a proper distribution of the coolant.					
20 Distribution/Availability of Abstract ☒ Unclassified/unlimited ☐ same as report ☐ DTIC users			21 Abstract Security Classification Unclassified		
22a Name of Responsible Individual Allan D. Kraus			22b Telephone (include Area Code) (408) 646-2730	22c Office Symbol ECKS	

DD FORM 1473, 84 MAR

83 APR edition may be used until exhausted

security classification of this page

All other editions are obsolete

Unclassified

Approved for public release; distribution is unlimited.

PRELIMINARY DESIGN OF A WATER COOLED
AVIONICS RACK

by

Colleen L. Ellis
Lieutenant Commander, United States Navy
B.S., United States Naval Academy, 1982

Submitted in partial fulfillment
of the requirements for the degree of

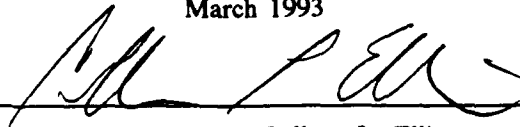
MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL

March 1993

Author:

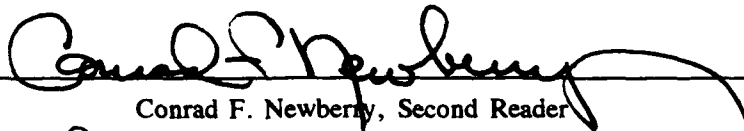


Colleen L. Ellis

Approved by:



Allan D. Kraus, Thesis Advisor



Conrad F. Newberry, Second Reader



Daniel J. Collins, Chairman

Department of Aeronautics and Astronautics

ABSTRACT

Military electronics are frequently operated in excessively confined spaces aboard ships and aircraft. This limited space impacts significantly on the space available for cooling equipment. The optimal solution is the development of one universal, modular rack for shipboard or aviation use. With a modular design, upgrades to equipment could also be accompanied by an upgrade to the cooling rack itself with very little additional cost or difficulty. A water cooled avionics rack can provide sufficient cooling for any piece or combination of avionics equipment if enough water flow paths are used, the water is at the appropriated temperature and the water is properly distributed within the passages. To determine if the cooling medium, water, is sufficiently distributed within a modular cooling rack, an analysis of the flow and pressure distribution of the coolant is required. This thesis presents a computer code that has been developed as an initial step in the total design of a modular cooling rack for avionics equipment. In itself, the code details a specific design technique and allows for the determination of whether a proposed configuration, including source location, characteristics of the cooling water, and the size and shape of the proposed flow passages will indeed provided a proper distribution of the coolant.

Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By _____	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

TABLE OF CONTENTS

I.	INTRODUCTION	1
	A. GENERAL	1
	B. BACKGROUND	1
	C. BASIC THEORY AND APPROACH	3
	1. Node Analysis	3
	2. Laminar Flow	4
	D. SCOPE	4
II.	AVIONICS COOLING RACK PROBLEM FORMULATION	5
	A. GENERAL	5
	1. COOLING RACK BASIC STRUCTURE	5
	B. VARIABLES IN DESIGN	6
	C. DEVELOPMENT AND LINEARIZATION OF FLOW EQUATIONS	6
	1. Laminar Equations	6
	D. NUMERICAL SOLUTION SCHEME	11
III.	DESCRIPTION OF THE COMPUTER CODE	19
	A. PROGRAM STRUCTURES AND CAPABILITIES	19
	1. Restrictions and Limitations	19
	2. Structure of the Main Computer Code	20
	B. DESCRIPTION OF SUBROUTINES	21
	1. Subroutine MATMUL	21
	2. Subroutine DECOMP	21
	3. Subroutine CHOLSKY	21

IV.	RESULTS AND DISCUSSION OF CASE RUNS	22
	A. CASE RUN ONE	22
	B. CASE RUN TWO	23
V.	CONCLUSION	24
	A. GENERAL COMMENTS	24
	B. ENHANCING THE MAIN PROGRAM CAPABILITY	24
	REFERENCES	25
	APPENDIX A COMPUTER CODE	26
	APPENDIX B CASE RUN RESULTS	39
	INITIAL DISTRIBUTION LIST	40

ACKNOWLEDGMENTS

A special thanks to Professor Allan D. Kraus. His guidance, patience and humor have made this thesis a very positive experience. He is a remarkable teacher who's first priority is clearly the student. I thank him for letting me work with him and all that he taught me.

I also acknowledge my mother, Roberta Fay Ellis. Her love and patience is one of the few constants in my life. Thank you for always being there.

I. INTRODUCTION

A. GENERAL

This thesis describes the development of a computer code to analyze the flow and pressure distribution of water in an avionics cooling rack. This code allows for variable rack design and flow sources. The code is written in *FORTRAN 77*. In addition, the code can be run on any IBM or IBM compatible personal computer.

B. BACKGROUND

Extremely high temperatures are a primary cause of avionics equipment unreliability. The origin of the thermal problem is in the continuous effort to develop lighter and smaller components. These smaller, more densely packed components, by virtue of the heat dissipation within a small volume, frequently operate at excessively high temperatures. These high temperatures result in component derating, performance degradation and accelerated failure. [Ref 1]

Successful thermal management of electronic systems, under development in the latter part of the 1990s will require the removal of as much as 500 W from a single chip, at heat fluxes between 50 to 100 W/cm². Volumetric heat release rates can also be expected to increase dramatically and are likely to exceed 10 W/cm³. Silicon chips, with embedded bipolar circuits, have traditionally been maintained at temperatures ranging from 65° C, for commercial computers, to between 110° C and 125° C for military equipment. [Ref 2]

Due to necessity, military electronics are frequently operated in confined spaces aboard ships and aircraft. This limited space impacts significantly on the space available for extensive cooling equipment. Air-cooled avionics systems are

currently the most frequently encountered cooling systems used in military applications. However, it is common practice with military avionics to upgrade the operational capability of a ship or aircraft by upgrading only one or two pieces of avionics equipment at a time. With an air cooled system, this is very ineffective. By replacing equipment in one of these systems with equipment of a different size and shape, the cooling airflow of the original design is disturbed and the resulting airflow around the new (as well as the old) equipment is less than optimal. The original cooling system was designed for equipment configured in a certain manner and little attention is paid to the reoptimization of the cooling system when changes are made. Moreover, cost restrictions are also an important factor in the design of military cooling systems.

One solution for the military applications problem is the development of a universal rack for shipboard or aviation use. If this rack were modular, it would allow for a great amount of flexibility in design and provide significant cost savings. With a modular design, upgrades to equipment could also be accompanied by an upgrade to the cooling rack itself with very little additional cost or difficulty. By developing one universal rack for all military applications, a significant improvement in cooling systems for updated designs can also be realized. This could be accompanied by additional cost savings through the elimination of unique cooling systems for every different avionics suite.

The rack would be a structure that could accomodate the placement of several modular electronic packages. One might imagine a "*shoebox*" configuration with, say, 16 "*holes*" in which 16 electronic packages could be inserted. Each of the packages is presumed to contain electronic components mounted in a variety of ways (i.e printed circuit boards or on brackets attached to the walls of the package). Such a structure would possess at least 64 flow passages exclusive of flow passages that are part of

the fluid source or pump. The objective is to assure that each flow passage carries enough fluid to absorb the dissipated heat that is somehow conducted to the rack structure. Of course, a heat exchanger may be required to transfer the heat to the ultimate heat sink which may be the environmental air or sea water.

Thus, the basic assumption is made that a modular, water cooled avionics rack can provide sufficient cooling for any piece or combination of avionics equipment if enough water flow paths are used, the water is at the appropriate temperature and the water is properly distributed within the passages. To determine if the cooling medium, water, is being sufficiently distributed within a modular cooling rack, an analysis of the flow and pressure distribution of the water is required. This can be accomplished using a computer code utilizing node analysis. Such a computer code, allowing for a variable number of branches and junctions, is presented here as a first step in the development of a universal cooling rack for military applications.

C. BASIC THEORY AND APPROACH

1. Node Analysis

The analysis of the cooling rack is based on the stipulation that any size (diameter and length) of passage may be used in the construction of the rack. Variables in the program include the density and viscosity of the water, the number and location of flow sources, the number of water paths entering each junction, the shape of the flow passages (circular or rectangular) and the length and equivalent diameter of each passage. The purpose of this degree of flexibility is to allow for easy redesign of a rack in the event that it does not meet the requirements which are the proper distribution of cooling water in each section of pipe. By varying either the rack configuration or the state of the water via computer input, a rack that provides the proper flow distribution can eventually be proposed.

The calculations for the flow in the passages employs a matrix oriented procedure used in network analysis. The network analysis approach can be tailored to flow in passages by proposing an analogy of the "current" to fluid flow and the "voltage" to the fluid pressure. The "resistance" is then attributed to the friction in each length of pipe. Therefore, each length of pipe will have a resistance associated with it, and possibly a pressure source as well, depending on the design of the cooling rack.

2. Laminar Flow

The computer code is designed to calculate a laminar flow distribution. If the flow is in transition or turbulent, there is a significant increase in the amount of frictional resistance. For Reynolds numbers (Re) less than or equal to 2100, the flow is considered laminar. For Reynolds number between 2100 and 10000, the flow is in transition and the flow is turbulent if the Reynolds number exceeds 10000. The computer code calculates the Reynolds number for each flow passage and provides a warning if the Reynolds number exceeds 2100.

D. SCOPE

- Chapter II explains and details the basic code required to analyze the flow and pressure distribution of water in a proposed cooling rack design.
- Chapter III describes the computer code, its essential capabilities and limitations, associated subroutines, input requirements and final output.
- Chapter IV presents the results from several case runs that exhibit the flexibility and capability of the method.
- Chapter V concludes with future development efforts and the application potential of this computer code.

II. AVIONICS COOLING RACK PROBLEM FORMULATION

A. GENERAL

The flow distribution of the fluid (in all branches) of the cooling rack is determined using a matrix oriented solution technique. The general equations and the matrix solution are presented in what follows.

1. COOLING RACK BASIC STRUCTURE

The basic structure of the cooling rack is a series of fluid passages fitted together using standard junction components. An almost unlimited number of passages may be fitted together. An example of one possible simple combination of passages with one flow source is presented in Figure 2.1.

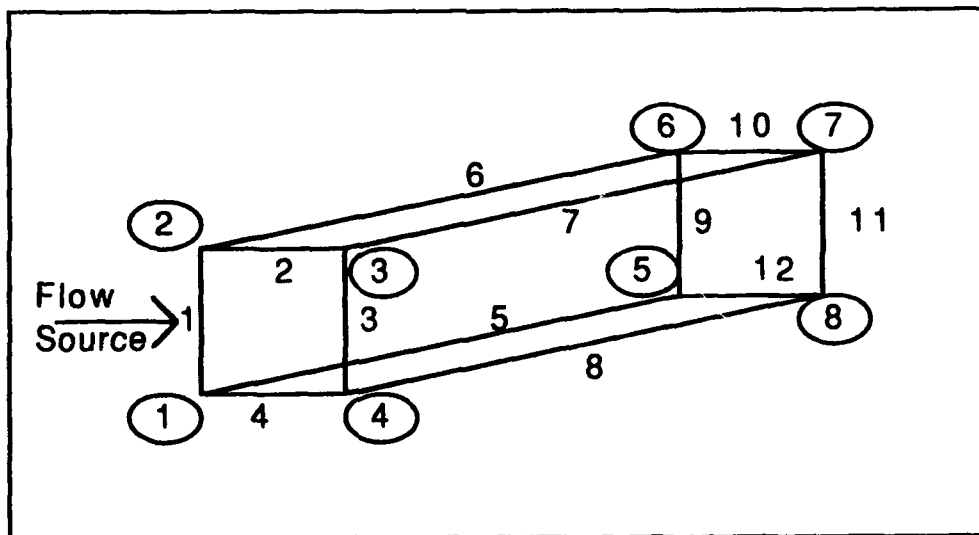


Figure 2.1: Example of cooling rack design.

This example illustrates how variable sized flow passages may be used. The rack is composed of b straight passages called branches and n_j junctions called

nodes. The branches are numbered because the computer code requires that basic information for each flow passage branch be entered separately. The nodes are also numbered and those are shown in circles. Note also that there is a pressure source located at branch-1. For convenience, the numbering sequence in Figure 2.1, begins with a 1 on the upper left side and proceeds to the right and then down. Although this numbering sequence is arbitrary, maintaining a consistent technique facilitates later adaptations to the rack and comparison with other rack configurations.

B. VARIABLES IN DESIGN

As stated in a previous section, the length, shape and the diameters of the flow passages may vary. The code is written for either circular or rectangular passages. For rectangular passages an effective diameter is calculated. Although the actual diameter or effective diameter may vary within one rack design, it is assumed that either all circular or all rectangular passages are used for any one rack configuration. All length and diameter size information is entered in the initial part of the program as $b \times 1$ matrices named $ELL(IB)$ and $D(IB)$, respectively. The density and the viscosity of the cooling water are input as the variables RHO and MU . The other input and output variables used in the computer code are summarized, for the reader's convenience, in Table 1.

C. DEVELOPMENT AND LINEARIZATION OF FLOW EQUATIONS

1. Laminar Equations

The basic equation used to determine the change in pressure or pressure loss within a fluid passage is derived from the D'Arcy equation

$$h_f = \frac{fLV^2}{2gd_e} \quad (2.1)$$

TABLE 2.1: INPUT AND OUTPUT VARIABLES (FORTRAN)

Variable	Explanation
A1	Cross sectional height
B	Number of branches
B1	Cross sectional width
D	Branch diameter vector
ELL	Branch length vector
IB	Counter for branches, IB = 1 to B
MU	Viscosity
N	Number of nodes
NF	Node "from" array
NT	Node "to" array
PS	Pressure source vector
RHO	Density

In this equation, h_f is the pressure loss due to friction and d_e is the equivalent diameter. The D'Arcy equation is valid for steady flow within passages running full of liquid. In eq (2.1), f is a friction factor. If the flow is laminar, the friction factor, f , as shown in Figure 2.2, can be represented by the equation

$$f = \frac{64}{Re} \quad (2.2)$$

The Reynolds number, by definition is

$$Re \equiv \frac{\rho V d_e}{\mu} \quad (2.3)$$

Therefore, f , is seen to have an alternate form

$$f = \frac{64\mu}{\rho V d_e} \quad (2.4)$$

Substitution of eq (2.4) into eq (2.1) yields

$$h_f = \frac{32LV\mu}{\rho g d_e^2} \quad (2.5)$$

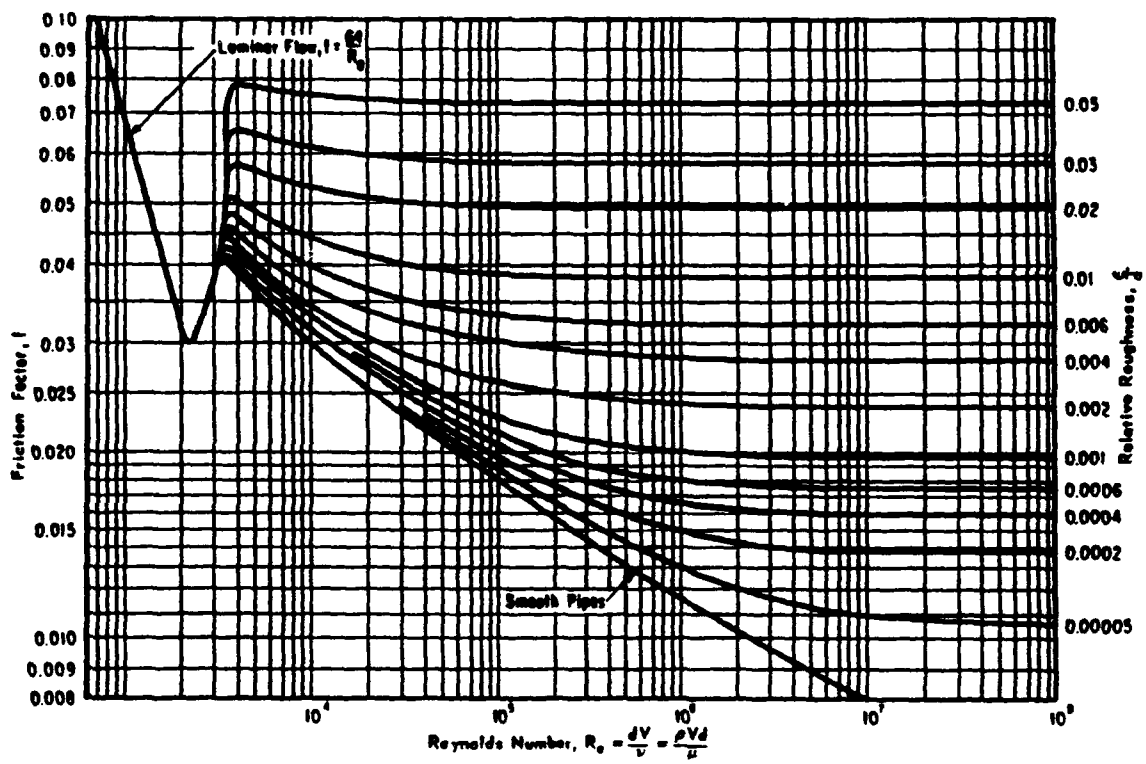


Figure 2.2: Moody's chart for the friction factor.

By definition, the flow rate, Q , is

$$Q = VA \quad (2.6)$$

The area, A , of a circular flow passage is, of course, given by

$$A = \pi \frac{d^2}{4} \quad (2.7a)$$

but for rectangular passages of side dimensions a and b , it is

$$A = ab \quad (2.7b)$$

and for square passages where $a = b$, it is

$$A = a^2 \quad (2.7c)$$

For circular passages, d_e , is simply the passage diameter, d , and for rectangular passages, d_e is defined as

$$d_e \equiv \frac{CA}{P}$$

where A is the passage flow area and P , for channels flowing full, is the passage wetted perimeter. In order to make the equivalent diameter for a circular passage equal to the actual diameter, d , $C = 4$ so that for a rectangular passage with side dimensions, a and b , the equivalent diameter becomes

$$d_e \equiv \frac{2ab}{a+b} \quad (2.8a)$$

In the event that the passage is square ($a = b$), the equivalent diameter becomes

$$d_e \equiv a \quad (2.8b)$$

Substitution of eqs (2.7) and (2.8) into equation (2.5) yields for the circular passage

$$h_f = \frac{128LQ\mu}{g\pi d^4\rho} \quad (2.9a)$$

for the rectangular passage

$$h_f = \frac{8L\mu Q(a+b)^2}{\rho g(ab)^3} \quad (2.9b)$$

and for the square passage

$$h_f = \frac{32L\mu Q}{\rho g a^4} \quad (2.9c)$$

To simplify eq (2.9a) further, let

$$h_f = P$$

and

$$R = \frac{128L\mu}{g\pi d^4 \rho} \quad (2.10)$$

This yields the equation

$$P = RQ$$

which is of the same form as

$$V = RI$$

Similar adjustments can be made to eqs (2.9b) and (2.9c). When these adjustments are made, it is seen that for a rectangular passage, eq (2.9b) gives

$$R = \frac{8L\mu(a+b)^2}{\rho g(ab)^3} \quad (2.11)$$

and that for a square passage eq (2.9c) provides

$$R = \frac{32L\mu}{\rho g a^4}$$

Additional losses (other than passage friction already discussed) occur in flow passage systems and cannot be disregarded without appreciable error. Compensation for entrance and exit losses must be considered in the cooling rack system by adding an equivalent length of straight pipe. The equivalent length used for square edged entries is 20 diameters (or equivalent diameters) and for exits, the equivalent length is 40 diameters (or equivalent diameters) [Ref. 2]. Therefore, the additional length of 60 passage diameters accounts for both exit and entrance losses.

D. NUMERICAL SOLUTION SCHEME

We first consider a general branch (the k^{th} branch) as shown in Figure 2.3. The branch can contain a pump with pressure head, p_{sk} , with an associated section of pipe of resistance, R_k . The branch carries a flow, q_k , and possesses a total head loss, p_k . Notice that the orientation of the head loss is in the direction of the flow (from "+" to "-").

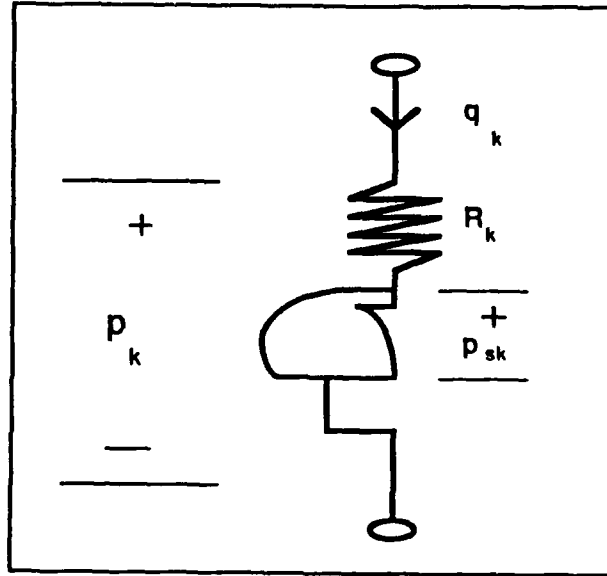


Figure 2.3: Branch with pressure source.

Also observe the orientation of the flow, q_k , and notice that by the analogy to Ohm's law and because of compatibility (the sum of the losses must match the head available)

$$p_k = R_k q_k + p_{sk}$$

This may be written for all k branches in matrix form

$$\mathbf{P} = \mathbf{RQ} + \mathbf{P}_s$$

Consider a pump and a length of discharge piping. The pump can be represented as a flow source or as a pressure source as indicated in Figure 2.4.

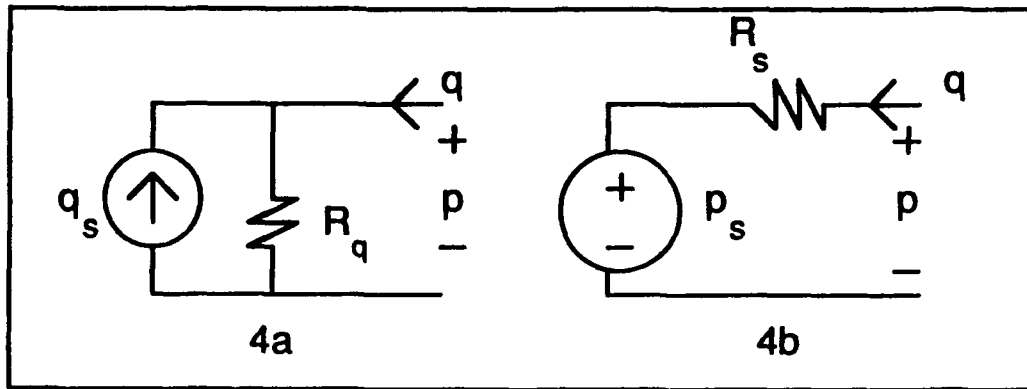


Figure 2.4: Alternative source arrangements for the development of the Flow source ↔ Pressure source transformation.

In these figures, the R 's represent the resistance to flow of the discharge piping and, in the case of the pressure source, the $+$ and $-$ indicate the discharge (+) and suction (-). In Figure 2.4a, continuity dictates that,

$$q = \frac{p}{R_q} - q_s$$

or

$$p = R_q q + R_q q_s$$

In Figure 2.4b, compatability (the pressure drops around the simple fluid loop must match)

$$p = R_p p_s$$

If there is to be an equivalence between the two (Figures 2.4a and 2.4b) then

$$R = R_p = R_q$$

and

$$p_s = R q_s$$

or indeed

$$q_s = \frac{p_s}{R}$$

Thus, when we represent a pump as a pressure source with a resistance, we can immediately transform this to an equivalent flow source. The computer code is written for the source input to be a flow source.

Note also that continuity dictates that for the k^{th} branch

$$q_k = Y_k p_k + q_{sk}$$

And for all k branches this can be written in matrix form as

$$\mathbf{Q} = \mathbf{Y}\mathbf{P} + \mathbf{Q}_s$$

Next consider a rack containing b branches and n_t nodes. The n_t^{th} node is the datum node (the node at the suction end of the pump), and we may set down an $n_t \times b$ augmented node-branch incidence matrix $\mathbf{A}^a$ with elements

$$a_{ij} = \begin{cases} +1 & \text{if branch } j \text{ leaves node } i \\ -1 & \text{if branch } j \text{ enters node } i \\ 0 & \text{if branch } j \text{ is not incident} \\ & \text{upon node } i \end{cases} \quad (2.10)$$

For example in the network displayed in Figure 2.5 with its oriented graph shown in Figure 2.6, there are $n_t = 5$ nodes and $b = 6$ branches. For this network, the augmented node branch incidence matrix $\mathbf{A}^a$ is

$$\mathbf{A}^a = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & -1 & 1 \\ -1 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

Note that for each column, which represents a single branch, there should only be two non-zero entries, a 1 and a -1. These entries correspond to the nodes

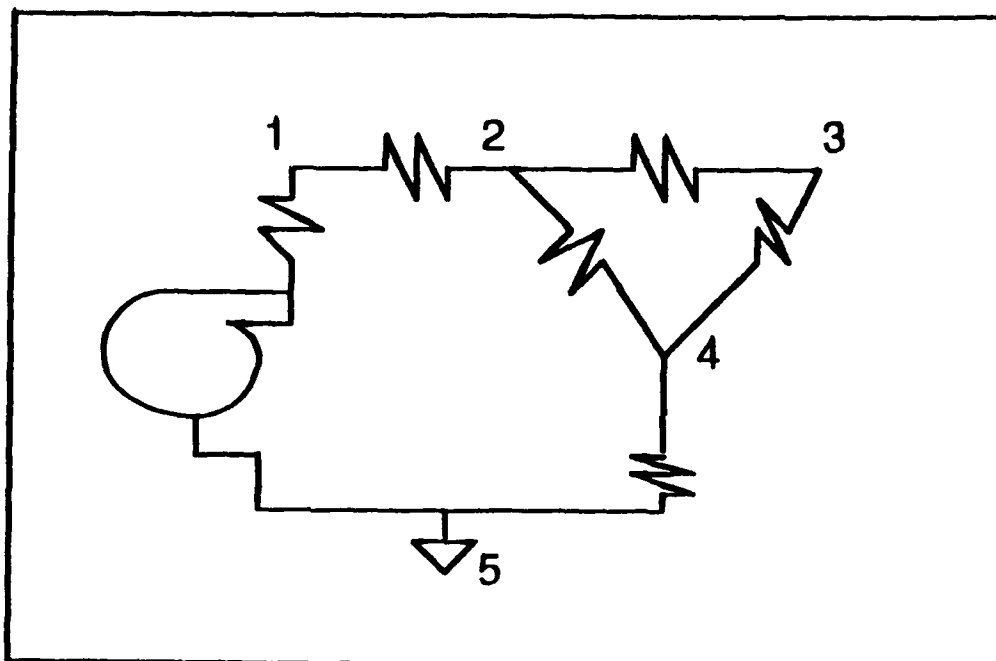


Figure 2.5: Network for example.

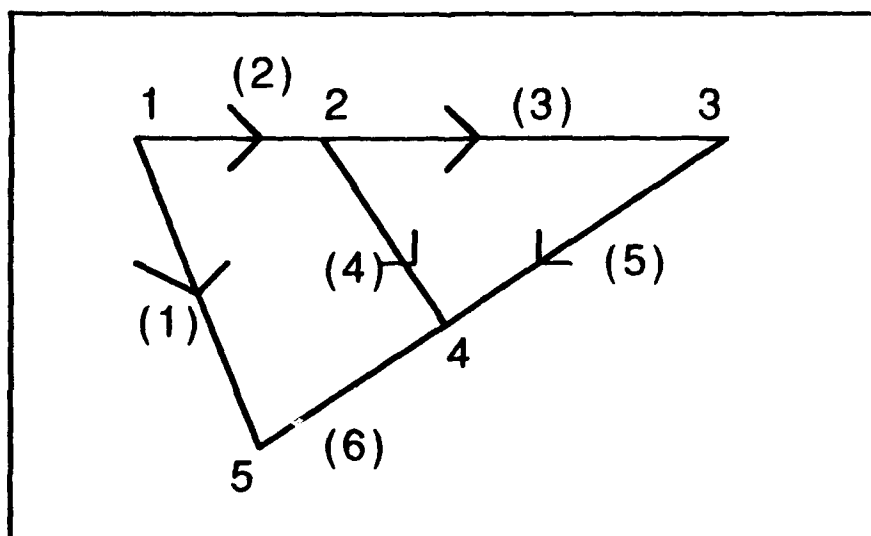


Figure 2.6: Oriented graph for network in Figure 2.5.

at the extremities of the branch. A branch that leaves a node shows a +1 while a branch that leaves a node shows a -1.

Next, we form a node-branch incidence matrix which is $n \times b$. This is done by deleting the row in A^a that corresponds to the datum node. In our example, row-5 is the datum node and thus

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & -1 & 1 \end{bmatrix}$$

The vector Q is defined as a $b \times 1$ vector representing the flow in all k branches and its elements are designated as q_k . The product of A with Q , AQ , represents a series of equations, one equation for each node. These equations represent the sum of all flow in and out of the respective node. Continuity demands that product, AQ be null

$$AQ = 0$$

and this simple matrix equation is a statement of continuity for the whole system.

To illustrate

$$AQ = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \end{bmatrix} = \begin{bmatrix} (q_1 + q_2) \\ (-q_2 + q_3 + q_4) \\ (-q_3 + q_5) \\ (-q_4 - q_5 + q_6) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

The pressures or heads at the nodes are represented by an $n \times 1$ vector designated by H and the branch pressure drops are represented by an $n \times 1$ vector designated by P . Reference to Figure 2.6 shows that

$$p_1 = h_1$$

$$p_2 = h_1 - h_2$$

$$p_3 = h_2 - h_3$$

$$p_4 = h_2 - h_4$$

$$p_5 = h_3 - h_4$$

$$p_6 = h_4$$

The matrices **P** and **H** can be related by a matrix **C**

$$\mathbf{P} = \mathbf{CH} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{bmatrix}$$

and it can be observed that

$$\mathbf{C} = \mathbf{A}^T$$

Refer now to the general branch in Figure 2.3 and the vector representation for the flow in all k branches. The $b \times 1$ vector, **Q** which represents the branch flow rates has been shown to be

$$\mathbf{Q} = \mathbf{Y}\mathbf{P} + \mathbf{Q}_s \quad (2.11)$$

Again, we carefully note the orientation of the flow source. Here **Y** is an $n \times n$ source admittance matrix. Premultiply eq (2.11) by **A** to obtain

$$\mathbf{A}\mathbf{Q} = \mathbf{A}\mathbf{Y}\mathbf{P} + \mathbf{A}\mathbf{Q}_s$$

But, $\mathbf{P} = \mathbf{A}^T\mathbf{H}$ and $\mathbf{A}\mathbf{Q} = 0$, thus

$$\mathbf{A}\mathbf{Y}\mathbf{A}^T\mathbf{H} + \mathbf{A}\mathbf{Q}_s = 0$$

The node equations are then formulated

$$\mathbf{Y}_n\mathbf{H} = \tilde{\mathbf{Q}}$$

where

$$Y_n = AYA^T$$

and

$$\tilde{Q} = -AQ_s$$

Therefore, the node pressures H , can now be determine by

$$H = Y_n^{-1}\tilde{Q}$$

Although many techniques exist for the solution of large sets of linear simultaneous algebraic equations, the most efficient computationally, appears to be the Cholesky reduction followed by back substitution that is employed in the Gauss elimination method. The only restriction on the use of Cholesky reduction is that its use is confined to symmetric, positive definite matrices. Here it is fortunate that Y_n is always positive definite.

The Cholesky reduction is based on the premise that there is a unique lower triangular matrix that permits a factorization in the form of $A = LL^T$ if the matrix A is symmetric and positive definite. Computationally, the Cholesky reduction is a very efficient technique in that it only requires $n(n+1)/2$ storage locations for L , rather than the usual n^2 locations required by other methods. [Ref. 3]. The number of operations using the Cholesky reduction is approximately $n^3/6$ rather than the usual $n^3/3$ required for most other decompositions [Ref. 3].

The Cholesky method to solve the basic system

$$AX = B$$

is based on finding the solution to two equivalent systems:

$$L^TC = B \quad \text{and} \quad LX = C$$

The elements of C are determined by the algorithms

$$c_1 = \frac{b_1}{s_{11}}$$

and

$$c_i = \frac{b_i - \sum_{l=1}^{i-1} s_{li} c_l}{s_{ii}}, \quad i > 1$$

Once C is known, X can be found using back substitution as employed in the Gauss elimination method, [Ref. 5] that is

$$x_n = \frac{c_n}{s_{nn}}$$

and

$$x_i = \frac{c_i - \sum_{l=i+1}^n s_{il} x_l}{s_{ii}}, \quad i < n$$

III. DESCRIPTION OF THE COMPUTER CODE

A. PROGRAM STRUCTURES AND CAPABILITIES

1. Restrictions and Limitations

The computer system used is an IBM or IBM compatible personal computer. The current program fixes the maximum number of individual fluid passages allowed at 300. For proper use of this program, the computer must have a minimum storage requirement of 2 M Bytes. A detailed computing time study for the program has not been undertaken because the computing time changes exponentially with the number of nodes and branches selected in the proposed cooling rack design.

The computer code can be operated using the following information

- Laminar Flow Conditions
- Density of water varying from x to y
- Viscosity of water varying from x to y
- Up to 100 branches
- Up to 100 nodes
- Up to 40 sources
- Any diameter (or equivalent diameter) flow passages
- Branches of any length
- Metric or English units

2. Structure of the Main Computer Code

The the main computer code including the three associated subroutines are in Appendix A. The main code is essentially divided into two major sections, initialization/input and laminar flow solution with its associated output. The main function of the initialization/input section are:

1. Set up the problem formulation by reading in the necessary values (node from, node to, length and cross sectional dimensions) for each branch of the cooling rack.
2. Read in the viscosity and density values.
3. Read in location and characteristics of each pressure source.

The second section calculates the pressure and flow distribution in the rack. It includes the subroutines MATMUL, DECOMP and CHOLESKY. The main functions of this section are to

1. Develop the node-branch incidence matrix, A , using the information read in section one.
2. Calculate the effective branch lengths to account for losses at the nodes.
3. Calculate the area and effective diameter of each flow passage cross sectional area.
4. Calculate the branch admittance matrix, Y .
5. Develop the flow source matrix, Q_s , from information read in section one.
6. Calculate the transpose of the matrix A .

7. Calculate the node admittance matrix, Y_n .
8. Calculate the node flow source vector, I_s .
9. Calculate the node pressure vector, H .
10. Calculate the inverse of Y_n .
11. Calculate the branch pressure vector, P .
12. Calculate the branch flow rate vector, Q .
13. Calculate the branch Reynolds numbers.
14. Provide readouts of all branch flow rates, all node pressures and all branch Reynolds numbers.

B. DESCRIPTION OF SUBROUTINES

1. Subroutine MATMUL

This subroutine [Ref. 5] multiplies an $n \times m$ matrix by any $m \times \ell$ matrix to form an $n \times \ell$ matrix and is called whenever matrix multiplication is required.

2. Subroutine DECOMP

This subroutine [Ref. 6] performs the decomposition of the symmetric, positive definite matrix, Y_n using the Cholesky reduction. It is called by subroutine CHOLESKY.

3. Subroutine CHOLESKY

This subroutine [Ref. 7] provides the solution of a linear system of equations using the Cholesky decomposition method for positive definite matrices. This subroutine calls subroutine DECOMP.

IV. RESULTS AND DISCUSSION OF CASE RUNS

A. CASE RUN ONE

Case one was run using the configuration shown in Figure 2.1 and the following data:

1. Water density = 62.4 lb/ft^3
2. Water viscosity = $8.8 \times 10^{-4} \text{ lb/sec-ft}$
3. Basic shoebox design with 8 nodes and 12 branches
4. Length of end branches (1-4 and 9-12) = 2.2 ft
5. Length of middle branches (5-8) = 3.0 ft
6. Circular passages with a diameter of 0.3 inches
7. One flow source at branch one
8. Strength of source was 0.4 gal/min

The results from case one are in Appendix B. It should be noted that the Reynolds number in each branch indicates laminar flow and that there is cooling water in all sections of the sample rack design. These results demonstrate that the analysis technique presented here, does allow for the determination of whether or not a proposed cooling rack design will indeed provide a proper distribution of cooling water. It also verifies that the laminar flow assumption is valid for some design parameters.

B. CASE RUN TWO

Case run two demonstrates the flexibility of the program through a variation in some of the input parameters. The basic configuration is still the same configuration shown in Figure 2.1.

1. Water density = 999.5 kg/m^3
2. Water viscosity = $13.1 \times 10^{-4} \text{ kg/m-sec}$
3. Basic shoebox design with 8 nodes and 12 branches
4. Length of end branches (1-4 and 9-12) = 1.0 m
5. Length of middle branches (5-8) = 1.2 m
6. Rectangular passages with a height of 1.3 cm and width of 1.4 cm.
7. One flow source at branch one
8. Strength of source was 1.8 lit/min

The results from case two are located in Appendix B.

V. CONCLUSION

A. GENERAL COMMENTS

This computer code has been developed as an initial step in the total design of a modular cooling rack for avionics equipment. In itself, the code details a specific design technique and allows for the determination of whether one proposed configuration, including source location, characteristics of the cooling water, and the size and shape of the proposed flow passages will indeed provide a proper distribution of the cooling water. Without proper coolant distribution, the cooling rack will be inefficient and perhaps, totally ineffective.

B. ENHANCING THE MAIN PROGRAM CAPABILITY

The next step in the development of a computer code to provide more flexibility and range in the rack design is the inclusion of the turbulent flow regime within the coolant passages. After turbulent flow is effectively incorporated, the final phase in the total rack design is a modification for heat input in the individual coolant flow passages.

REFERENCES

1. Kraus, A. D., *Kraus' Cooling Electronic Equipment*, p.214, College of Engineering, University of South Florida, Tampa, Florida
2. Bar-Cohen, Avram, "State-of-the-Art and Trends in the Thermal Packaging of Electronic Equipment," *Journal of Electronic Packaging*, 114, pp. 264-265, September 1992.
3. Baumeister, T., editor, *Mechanical Engineers' Handbook*, p. 3-78, McGraw Hill Book Company, Inc., 1958.
4. Atkinson, K. E., *An Introduction to Numerical Analysis*, pp. 452- 453, John Wiley & Sons, Inc., 1978.
5. Brebbia, C. A. and Ferrante, A. J., *Computational Methods for the Solution of Engineering Problems*, pp. 57-59, Pentech Press Limited, 1978.
6. Brebbia, C. A. and Ferrante, A. J., *Computational Methods for the Solution of Engineering Problems*, p. 26, Pentech Press Limited, 1978.
7. Brebbia, C. A. and Ferrante, A. J., *Computational Methods for the Solution of Engineering Problems*, pp. 59-60, Pentech Press Limited, 1978.
8. Brebbia, C. A. and Ferrante, A. J., *Computational Methods for the Solution of Engineering Problems*, p. 61, Pentech Press Limited, 1978.

APPENDIX A

PROGRAM COOL

Cooling Rack Calculations Using Node Analysis

INITIALIZATION

MU: fluid viscosity (lb/ft-sec)
 RHO: fluid density (lb/ft³)
 N: number of junctions in system
 N1: number of junction minus the datum node
 B: number of branches in system
 G: acceleration of gravity (ft/sec²)
 PI: pi

INTEGER N,N1,B,I,J,K,QB,S
 INTEGER NT(40),NF(40),X,IUNITS
 REAL MU,RHO,G,PI,A(40,40),D(40),ELL(40),R(40),MU1
 REAL QS(40,1),AT(40,40),QS1,I1(40,1),LENGTH
 REAL E(40,1),P(40,1),Y(40,40),YNI(40,40)
 REAL YP(40,1),Q(40,1),AY(40,40),YN(40,40)
 REAL V(40,1),RE(40,1),a1(40),b1(40),AREA(40)
 REAL IS(40,1),ISI(40,1),RHO1
 CHARACTER ANS,ANS1,ANS2

G=32.174
 PI=3.1415926
 IUNITS=0

 THIS PROGRAM DETERMINES THE FLOW RATE AND PRESSURE DISTRIBUTION
 OF COOLING WATER WITHIN A VARIABLE-SIZED AVIONICS COOLING RACK.
 VARIABLES INCLUDE RACK DIMENSIONS AND WATER CHARACTERISTICS.

1040 FORMAT (///' ENTER THE TOTAL NUMBER OF NODES IN THE RACK SYSTEM
 + ',2X,\)
 1042 FORMAT (BN, I3)
 1043 FORMAT (///' ENTER THE TOTAL NUMBER OF BRANCHES IN THE RACK SYSTEM
 + ',2X,\)

222 WRITE(IOT,1040)
 READ(IN,1042) N
 WRITE(IOT,1045) N
 IF(N .LT. 0 .OR. N .GT. 100) THEN
 WRITE(IOT,6565)
 GO TO 222
 END IF
 223 WRITE(IOT,1043)
 READ(IN,1042) B
 WRITE(IOT,1046) B
 IF(B .LT. 0 .OR. B .GT. 100) THEN

```

WRITE(IOT,6567)
GO TO 223
END IF

```

C
C

```

1045  FORMAT (// '    YOU ENTERED THE NUMBER OF NODES AS:',1X,I3\ )
1046  FORMAT (// '    YOU ENTERED THE NUMBER OF BRANCHES AS:',1X,I3,/, \ )
6565  FORMAT (/ '    THE NUMBER OF NODES MUST BE GREATER THAN ZERO AND LESS',
+ 1X,THAN 100.', \ )
6567  FORMAT (/ '    THE NUMBER OF BRANCHES MUST BE GREATER THAN ZERO AND LESS
+ THAN 100.', \ )

```

C
C

```

      N1=N-1
      DO 5 I=1,B
        QS(I,1)=0.
5      CONTINUE

```

C
C
C
C

```

55      CONTINUE
      WRITE(IOT,7676)
7676    FORMAT(/ ' Do you want to work in British units or SI units (B/S)?
+ ', \ )
      READ(IN,7677) ANS
7677    FORMAT(A1)
      IF(ANS .EQ. 'B' .OR. ANS .EQ. 'b') GO TO 7800
      IF(ANS .EQ. 'S' .OR. ANS .EQ. 's') GO TO 7678
      GO TO 55
7800    IUNITS = 1

```

C

```

      WRITE(IOT,1000)
1000    FORMAT (// ' Input Viscosity ( x 10^4 lbm/ft-sec)',2X, \ )
      READ(IN,1003) MU
      MU1=MU*.0001
      WRITE(IOT,1002)
1002    FORMAT (/ ' Input density (lb/ft^3)',2X, \ )
      READ(IN,1001) RHO
1001    FORMAT (BN, F8.4)
1003    FORMAT (BN, F7.6)
1004    FORMAT (BN, I3)
      GO TO 7801

```

C
C

```

7678    WRITE(IOT,7602)
7602    FORMAT (// ' Input Viscosity (Kg/m-sec x 10^4)',2X, \ )
      READ(IN,1003) MU
      WRITE(IOT,7603)
7603    FORMAT (/ ' Input Density (Kg/m^3)',2X, \ )
      READ(IN,1001) RHO1
      MU1=MU*.00006719
      RHO=RHO1*.06243

```

C
C
C
C

A: NODE-BRANCH INCIDENCE MATRIX

C INITIALIZE A MATRIX

C

7801 DO 20 I=1,N1
DO 20 J=1,B
20 A(I,J)=0.

C

DO 22 I=1,B
NF(I)=0
22 NT(I)=0

C

C

C

C

C

C

C

DETERMINATION OF PASSAGE SHAPE

8000 WRITE(IOT,8001)
8001 FORMAT(/' IF FLOW PASSAGES ARE CIRCULAR, ENTER A ZERO.',/,
+ ' IF PASSAGES ARE RECTANGULAR, ENTER A ONE.',2X,\)
READ(IN, 1042) X
IF(X .EQ. 0) THEN
GO TO 8004
END IF
IF(X .EQ. 1) THEN
GO TO 8005
END IF
8003 WRITE(IOT,8006)
8006 FORMAT(/' ERRONEOUS INPUT'\)
GO TO 8000

C

C

C

C

C

C

C

C

C

C

INITIAL DATA INPUT TO DEVELOP A,ELL AND D MATRICES FOR
RECTANGULAR PASSAGES

ELL: LENGTH
D: DIAMETER

8005 CALL CLS
IF (IUNITS .EQ. 0) THEN
DO 1493 J=1,B
WRITE(IOT,1051) J
WRITE(IOT,1492)
1492 FORMAT(/' FROM NODE, TO NODE, LENGTH(m), HEIGHT(cm), WIDTH(cm)',
+ /)
READ(IN,8202) NF(J),NT(J),ELL(J),a1(J),b1(J)
1493 CONTINUE
7272 FORMAT(/' YOU ENTERED THE FOLLOWING DATA.'\
7273 FORMAT(/' BRANCH FROM NODE TO NODE LENGTH(m) HEIGHT(cm)',2X,
+ 'WIDTH(cm)',/,)
7274 FORMAT(2X,I3,6X,I3,8X,I3,2X,F9.5,4X,F7.5,4X,F7.5)
7275 FORMAT(/' IS THIS CORRECT (Y/N)? '\
CALL CLS
16 WRITE(IOT,7272)
WRITE(IOT,7273)
DO 7277 J=1,B
WRITE(IOT,7274) J,NF(J),NT(J),ELL(J),a1(J),b1(J)

```

7277 CONTINUE
7276 WRITE(IOT,7275)
      READ(IN,7677) ANS
      IF(ANS .EQ. 'Y' .OR. ANS .EQ. 'y') GO TO 7278
      IF(ANS .EQ. 'N' .OR. ANS .EQ. 'n') GO TO 7280
      GO TO 7276
7278 DO 7279 J=1,B
      ELL(J)=ELL(J)*3.28
      a1(J)=a1(J)*.3937
      b1(J)=b1(J)*.3937
7279 CONTINUE
      GO TO 1495
7280 CONTINUE
7281 FORMAT(/' INPUT ONE BRANCH NUMBER TO CHANGE VALUES.'\ )
7284 WRITE(IOT,7281)
      READ(IN,1042) K1
7282 FORMAT(/' INPUT VALUES FOR BRANCH:',1X,I3,\ )
      WRITE(IOT,7282) K1
      WRITE(IOT,1492)
      READ(IN,8202) NF(K1),NT(K1),ELL(K1),a1(K1),b1(K1)
7283 FORMAT(/' DO YOU HAVE ANYMORE CHANGES (Y/N)?'\ )
7285 WRITE(IOT,7283)
      READ(IN,7677) ANS1
      IF(ANS1 .EQ. 'Y' .OR. ANS1 .EQ. 'y') GO TO 16
      IF(ANS1 .EQ. 'N' .OR. ANS1 .EQ. 'n') GO TO 7278
      GO TO 7285
      END IF

C
      CONTINUE

C
C
      DO 8200 I=1,B
      WRITE(IOT,1051) I
      WRITE(IOT,8201)
8201 FORMAT(/' FROM NODE, TO NODE, LENGTH(FT), HEIGHT(IN), WIDTH(IN)
+ ',/)
8202 FORMAT(2I3,F7.4,2F6.4)
      READ(IN,8202) NF(I),NT(I),ELL(I),a1(I),b1(I)
8200 CONTINUE

C
      CALL CLS
15 WRITE(IOT,7272)
      WRITE(IOT,8207)
8207 FORMAT(/' BRANCH FROM NODE TO NODE LENGTH(FT) HEIGHT(IN)',2X,
+ 'WIDTH(IN)',/)
7374 FORMAT(2X,I3,7X,I3,7X,I3,4X,F9.5,6X,F7.5,3X,F7.5)
      DO 8277 J=1,B
      WRITE(IOT,7374) J,NF(J),NT(J),ELL(J),a1(J),b1(J)
8277 CONTINUE
8276 WRITE(IOT,7275)
      READ(IN,7677) ANS
      IF(ANS .EQ. 'Y' .OR. ANS .EQ. 'y') GO TO 1495
      IF(ANS .EQ. 'N' .OR. ANS .EQ. 'n') GO TO 8280
      GO TO 8276
8280 CONTINUE
8284 WRITE(IOT,7281)

```

```

      READ(IN,1042) K1
      WRITE(IOT,7282) K1
      WRITE(IOT,8201)
      READ(IN,8202) NF(K1),NT(K1),ELL(K1),a1(K1),b1(K1)
8285  WRITE(IOT,7283)
      READ(IN,7677) ANS1
      IF(ANS1 .EQ. 'Y' .OR. ANS1 .EQ. 'y') GO TO 15
      IF(ANS1 .EQ. 'N' .OR. ANS1 .EQ. 'n') GO TO 1495
      GO TO 8285
1495  DO 1496 I=1,B
        IF (NF(I) .LT. N) THEN
          A(NF(I),I) = 1.
        END IF
        IF (NT(I) .LT. N) THEN
          A(NT(I),I) = -1.
        END IF
        a1(I)=a1(I)/12
        b1(I)=b1(I)/12
        AREA(I)=a1(I)*b1(I)
        D(I)=(2*(a1(I)*b1(I)))/(a1(I)+b1(I))
        LENGTH=ELL(I)
        ADDL=60*d(I)
        ELL(I)=LENGTH + ADDL
1496  CONTINUE
      GO TO 8300

```

C
C
C
C
C

RECEIVE DATA FROM KEYBOARD TO DEVELOP A,D AND ELL MATRICES FOR
CIRCULAR PASSAGES

```

1050  FORMAT(1X,I3,8X,I3,5X,F7.4,4X,F6.4)
9050  FORMAT(2I3,2F7.4)
8004  CALL CLS
      IF(IUNITS .EQ. 0) THEN
        GO TO 993
      ELSE
        DO 25 I=1,B
1051  FORMAT (/ ' INPUT THE FOLLOWING DATA, SEPARATED BY COMMAS FOR,1X,
+ BRANCH:',I3,2X,\)
1053  FORMAT (/ ' FROM NODE,TO NODE,LENGTH(FT),DIAMETER(IN)',2X,/)
        WRITE(IOT,1051) I
        WRITE(IOT,1053)
        READ(IN,9050) NF(I),NT(I),ELL(I),D(I)
25    CONTINUE
        CALL CLS
18    WRITE(IOT,7272)
        WRITE(IOT,1059)
1059  FORMAT(/ ' BRANCH      FROM NODE    TO NODE    LENGTH(FT) DIAMETER(IN)',
+ /)
        DO 9277 J=1,B
        WRITE(IOT,1155) J,NF(J),NT(J),ELL(J),D(J)
1155  FORMAT(2X,I3,9X,I3,8X,I3,6X,F7.4,4X,F6.4)
1255  FORMAT(2X,I3,7X,I3,7X,I3,5X,F7.4,5X,F6.4)
9277  CONTINUE
9276  WRITE(IOT,7275)
      READ(IN,7677) ANS

```

```

IF(ANS .EQ. 'Y' .OR. ANS .EQ. 'y') GO TO 26
IF(ANS .EQ. 'N' .OR. ANS .EQ. 'n') GO TO 9280
GO TO 9276
9280 CONTINUE
9284 WRITE(IOT,7281)
      READ(IN,1042) K1
      WRITE(IOT,7282) K1
      WRITE(IOT,1053)
      READ(IN,9050) NF(K1),NT(K1),ELL(K1),D(K1)
9285 WRITE(IOT,7283)
      READ(IN,7677) ANS1
      IF(ANS1 .EQ. 'Y' .OR. ANS1 .EQ. 'y') GO TO 18
      IF(ANS1 .EQ. 'N' .OR. ANS1 .EQ. 'n') GO TO 26
      GO TO 9285
      END IF
      GO TO 26

C
C
993 DO 29 J=1,B
    WRITE(IOT,1051) J
1055 FORMAT (/ ' FROM NODE, TO NODE, LENGTH(M),DIAMETER(CM) ',/)
    WRITE(IOT,1055)
    READ(IN,9050) NF(J),NT(J),ELL(J),D(J)
29 CONTINUE
C
    CALL CLS
19 WRITE(IOT,7272)
   WRITE(IOT,1077)
1077 FORMAT(/ ' BRANCH FROM NODE TO NODE LENGTH(M) DIAMETER(CM) ',
+ /)
    DO 1277 J=1,B
    WRITE(IOT,1255) J,NF(J),NT(J),ELL(J),D(J)
1277 CONTINUE
1276 WRITE(IOT,7275)
    READ(IN,7677) ANS
    IF(ANS .EQ. 'Y' .OR. ANS .EQ. 'y') GO TO 111
    IF(ANS .EQ. 'N' .OR. ANS .EQ. 'n') GO TO 1280
    GO TO 1276
1280 CONTINUE
1284 WRITE(IOT,7281)
    READ(IN,1042) K1
    WRITE(IOT,7282) K1
    WRITE(IOT,1055)
    READ(IN,9050) NF(K1),NT(K1),ELL(K1),D(K1)
1285 WRITE(IOT,7283)
    READ(IN,7677) ANS1
    IF(ANS1 .EQ. 'Y' .OR. ANS1 .EQ. 'y') GO TO 19
    IF(ANS1 .EQ. 'N' .OR. ANS1 .EQ. 'n') GO TO 111
    GO TO 1285

C
111 DO 28 J=1,B
    ELL(J)=ELL(J)*3.28
    D(J)=D(J)*.3937
28 CONTINUE
C
26 DO 33 I=1,B

```

```

      IF (NF(I) .LT. N) THEN
        A(NF(I),I)=1.
      END IF
      IF (NT(I) .LT. N) THEN
        A(NT(I),I)=-1.
      END IF
      D(I)=D(I)/12
      AREA(I)=PI*(D(I)**2)/4
      LENGTH=ELL(I)
      ADDL=60*D(I)
      ELL(I)=LENGTH+ADDL

```

```

33      CONTINUE

```

C
C
C
C

```

8300    DO 30 K=1,B
      R(K)=(RHO*G*AREA(K)*((D(K))**2))/(32.*MU1*ELL(K))
30      CONTINUE

```

C

```

      DO 35 J=1,B
      DO 35 K=1,B
        Y(J,K)=0.

```

```

35      CONTINUE

```

C
C

```

      DO 40 I=1,B
        Y(I,I)=R(I)

```

```

40      CONTINUE

```

C
C

```

      QS: MATRIX OF FLOW SOURCES

```

C
C

```

      S: NUMBER OF FLOW SOURCES

```

C

```

      QB: SOURCE BRANCH

```

C

```

      QS1: SOURCE STRENGTH

```

C

```

1015    FORMAT (' Input the number of flow sources:',3X,\)
1016    FORMAT (' Input the branch of source',I3,3X,,\)
1017    FORMAT (' Input the strength of the source (gal/min):',3X,\)
9017    FORMAT (' Input the strength of the source (liter/min):',3X,/)
      WRITE(IOT,1015)
      READ(IN,1004) S
      DO 50 I=1,S
        WRITE(IOT,1016) I
        READ(IN,1004) QB
        IF(IUNITS .EQ. 1) THEN
          WRITE(IOT,1017)
          READ(IN,1001) QS1
        ELSE
          WRITE(IOT,9017)
          READ(IN,1001) QS1
          QS1=QS1*.2642
        END IF
        QS1=QS1/444
      DO 60 K=1,B

```

```

      IF (K .EQ. QB) THEN
      QS(K,1)=QS1
      ELSE
      QS(K,1)=0.
      END IF

```

```

60      CONTINUE
50      CONTINUE

```

```

      WRITE(IOT,3012) (QS(I,1),I=1,B)

```

```

C MATRIX MANIPULATION TO DETERMINE INDIVIDUAL BRANCH FLOW AND
C PRESSURE DISTRIBUTIONS

```

```

C AT: TRANSPOSE OF MATRIX A
C AY: PRODUCT OF MATRIX A AND MATRIX Y
C YN: PRODUCT OF MATRIX AY AND MATRIX AT (YN=AYAT)

```

```

      DO 80 I=1,N1
      DO 80 J=1,B
80      AT(J,I)=A(I,J)

```

```

      CALL MATMUL(AY,A,Y,N1,B,B)
      CALL MATMUL(YN,AY,AT,N1,B,N1)

```

```

C IS: NODE FLOW SOURCE VECTOR IS= -AQS
      CALL MATMUL(IS,A,QS,N1,B,1)

```

```

      DO 90 I=1,N1
      IS(I,1)=-IS(I,1)
      ISI(I,1)=IS(I,1)
90      CONTINUE

```

```

C YNI: MATRIX EQUAL TO MATRIX YN BEFORE CHOLESKI INVERSION. AFTER
C DECOMPOSITION IT HOLDS THE UPPER TRIANGULAR MATRIX.
C ISI: MATRIX EQUAL TO MATRIX IS BEFORE INVERSION. AFTER INVERSION,
C IT HOLDS THE SOLUTION TO E=YN(-1) IS

```

```

      DO 95 I=1,N1
      DO 95 J=1,N1
      YNI(I,J)=YN(I,J)
95      CONTINUE

```



```

      CALL CHOLSKY(YNI,ISI,N1,N1)
C
C
C      E: NODE VOLTAGE MATRIX E=YN(-1)IS
C      P: BRANCH PRESSURE P=ATE
C      Q: BRANCH FLOW RATE Q=YP+QS
C      YP: PRODUCT OF MATRIX Y AND MATRIX P (YP)
C
      DO 96 I=1,N1
        E(I,1)=ISI(I,1)
96    CONTINUE
C
C
      CALL MATMUL(P,AT,E,B,N1,1)
C
C
      CALL MATMUL(YP,Y,P,B,B,1)
C
      DO 120 I=1,B
        Q(I,1)=YP(I,1)+QS(I,1)
120    CONTINUE
      CALL CLS
C
C
      REYNOLDS NUMBER CALCULATIONS
C
      DO 5000 I=1,B
        V(I,1)=ABS(Q(I,1)/(AREA(I)))
        RE(I,1)=rho*V(I,1)*d(I)/mu1
        IF(RE(I,1) .GT. 2100) THEN
          WRITE(IOT,5001) I
          WRITE(IOT,5002)
5001      FORMAT(/'      REYNOLDS NUMBER IN BRANCH',I3,2x,'EXCEEDS',\ )
5002      FORMAT(/'      2100.  THE FLOW IS NOT LAMINAR.',2X,\ )
          ELSE
            END IF
5000    CONTINUE
C
C
1500  FORMAT(3X,I3,7X,F11.8,2X,2F15.5,1X,F10.0)
1700  FORMAT(3X,I3,5X,F11.8,3X,2F15.5,2X,F5.0)
1501  FORMAT(///'  Branch',4X,'Flow Rate (ft/sec)',4x,'P in (psf)',6x,
+ 'P out (psf)',6x,'Re',1x,/)
1601  FORMAT(///'  Branch',4x,'Flow Rate (m/sec)',5x,'P in (N/m^2)',7x,
+ 'P out (N/m^2)',6x,'Re',2x,/)
      IF(IUNITS .EQ. 1) THEN
        WRITE(IOT,1501)
        DO 7000 I=1,B
          WRITE(IOT,1500)I,Q(I,1),E((NF(I)),1),E((NT(I)),1),RE(I,1)
7000    CONTINUE
99    FORMAT(/'      WOULD YOU LIKE A PRINTOUT OF THIS TABLE (Y/N)?'\ )
98    WRITE(IOT,99)
      READ(IN,7677)ANS2
      IF(ANS2 .EQ. 'Y' .OR. ANS2 .EQ. 'y') GO TO 97
      IF(ANS2 .EQ. 'N' .OR. ANS2 .EQ. 'n') GO TO 83
      GO TO 98

```

```

97    WRITE(IPR,1501)
      DO 93 I=1,B
      WRITE(IPR,1500) I,Q(I,1),E((NF(I)),1),E((NT(I)),1),RE(I,1)
93    CONTINUE
83    CONTINUE
      ELSE
      DO 7470 J=1,N1
      E(J,1)=E(J,1)*47.88
7470  CONTINUE
      WRITE(IOT,1601)
      DO 7001 I=1,B
      Q(I,1)=Q(I,1)*.3048
      WRITE(IOT,1500) I,Q(I,1),E((NF(I)),1),E((NT(I)),1),RE(I,1)
7001  CONTINUE
48    WRITE(IOT,99)
      READ(IN,7677)ANS2
      IF(ANS2 .EQ. 'Y' .OR. ANS2 .EQ. 'y') GO TO 47
      IF(ANS2 .EQ. 'N' .OR. ANS2 .EQ. 'n') GO TO 82
      GO TO 48
47    WRITE(IPR,1601)
      DO 43 I=1,B
      WRITE(IPR,1500) I,Q(I,1),E((NF(I)),1),E((NT(I)),1),RE(I,1)
43    CONTINUE
82    CONTINUE
      END IF
      END

```

```

SUBROUTINE CHOLESKY(A,B,N,NX)
C
C
C SOLUTION OF LINEAR SYSTEMS OF EQUATIONS BY THE CHOLESKI
C METHOD FOR SYMMETRIC POSITIVE DEFINITIVE MATRICES.
C
C   A: ARRAY CONTAINING THE SYSTEM MATRIX (AX=B)
C   B: ARRAY CONTAINING THE VECTOR IF INDEPENDENT COEFFICIENTS
C   C: AUXILIARY VECTOR
C   N: ORDER OF A
C   NX: ROW AND COLUMN DIMENSION OF A
C
      DOUBLE PRECISION B(NX,1)
      DIMENSION A(40,40)
C
C COMPUTE UPPER TRIANGULAR MATRIX FROM A AND STORE ALSO IN A
C
      CALL DECOMP(A,N,NX)
C
C COMPUTE THE C VECTOR AND STORE IN ARRAY B
C
      B(1,1)=B(1,1)/A(1,1)
C          WRITE(IOT,100) A(1,1)
100          FORMAT(F18.15)
      DO 10 I=2,N
C          WRITE(IOT,101) I,N
101          FORMAT(2I3)
C          WRITE(IOT,102) A(I,I),A(N,N)
102          FORMAT(2F13.11)
      D=B(I,1)
      I1=I-1
      DO 5 L=1,I1
5          D=D-A(L,I)*B(L,1)
10      B(I,1)=D/A(I,I)
      B(N,1)=B(N,1)/A(N,N)
C
C COMPUTE THE SYSTEM UNKNOWNNS AND STORE IN ARRAY B
C
      N1=N-1
      DO 30 L=1,N1
      K=N-L
C          WRITE(IOT,105) A(K,K),K
105          FORMAT(F13.11,I3)
      K1=K+1
      DO 20 J=K1,N
20      B(K,1)=B(K,1)-A(K,J)*B(J,1)
30      B(K,1)=B(K,1)/A(K,K)
      RETURN
      END

```

SUBROUTINE MATMUL(C,A,B,N,M,L)

c This subroutine computes the matrix operation $C = A * B$

c

c

N: NUMBER OF ROWS IN MATRIX A AND C

c M: NUMBER OF COLUMNS IN A AND ROWS IN B

c

L: NUMBER OF COLUMNS IN B AND C

c

DIMENSION A(40,40),B(40,40),C(40,40)

DO 20 I=1,N

DO 20 J=1,L

C(I,J)=0.0

DO 20 K=1,M

20 C(I,J)=C(I,J)+A(I,K)*B(K,J)

RETURN

END

c

```

C
      SUBROUTINE DECOMP(A,N,NX)
C
C THIS SUBROUTINE PERFORMS THE DECOMPOSITION OF A POSITIVE,
C DEFINITE, SYMMETRIC MATRIX INTO AN UPPER TRIANGULAR MATRIX.
C
C   A: ARRAY ORIGINALLY CONTAINING MATRIX TO BE DECOMPOSED.
C   AT THE END IT CONTAINS THE UPPER TRIANGULAR MATRIX.
C   N: ORDER OF A
C   NX: ROW AND COLUMN DIMENSION OF A
      INTEGER N
      DIMENSION A(40,40)
      IF (A(1,1)) 1,1,3
1     WRITE(IOT,2)
2     FORMAT (' ZERO OR NEGATIVE RADICAND')
C     WRITE(IOT,500) A(1,1)
500    FORMAT ('      A(1,1)',F13.11)
      GO TO 200
3     A(1,1)=SQRT(A(1,1))
      DO 10 J=2,N
10    A(1,J)=A(1,J)/A(1,1)
      DO 40 I=2,N
C     WRITE(IOT,600) I
600    FORMAT(I3)
      I1=I-1
      D=A(I,I)
      DO 20 L=1,I1
20    D=D-A(L,I)*A(L,I)
C     WRITE(IOT,503) I,D,A(I,I)
503    FORMAT(I3,2F13.11)
      IF (A(I,I)) 1,1,21
21    A(I,I)=SQRT(D)
      IF(I .EQ. N) THEN
        GO TO 45
      END IF
      I2=I+1
      DO 40 J=I2,N
      D=A(I,J)
      DO 30 L=1,I1
30    D=D-A(L,I)*A(L,J)
C     WRITE(IOT,503) I,D,A(I,I)
40    A(I,J)=D/A(I,I)
45    DO 50 I=2,N
      I1=I-1
      DO 50 J=1,I1
50    A(I,J)=0
200   RETURN
      END

```

APPENDIX B

Case Run One

Branch	Flow Rate (ft/sec)	Pin (psf)	Pout (psf)	Re
1	.00035966	-.03996	.05160	1299.
2	.00019242	.05160	.01905	695.
3	.00015641	.01905	-.00741	565.
4	.00019242	-.00741	-.03996	695.
5	.00016724	-.00555	-.03996	604.
6	.00016724	.05160	.01719	604.
7	.00003601	.01905	.01164	130.
8	-.00003601	-.00741	.00000	130.
9	-.00013443	-.00555	.01719	485.
10	.00003281	.01719	.01164	118.
11	.00006882	.01164	.00000	249.
12	.00003281	.00000	-.00555	118.

Case Run Two

Branch	Flow Rate (m/sec)	Pin (N/sm)	Pout (N/sm)	Re
1	.00013296	-.28579	.37451	698.
2	.00006895	.37451	.13925	362.
3	.00005561	.13925	-.05052	292.
4	.00006895	-.05052	-.28579	362.
5	.00006401	-.04323	-.28579	336.
6	.00006401	.37451	.13195	336.
7	.00001333	.13925	.08872	70.
8	-.00001333	-.05052	.00000	70.
9	-.00005134	-.04323	.13195	270.
10	.00001267	.13195	.08872	67.
11	.00002600	.08872	.00000	137.
12	.00001267	.00000	-.04323	67.

INITIAL DISTRIBUTION LIST

- | | | |
|----|--|---|
| 1. | Defense Technical Information Center
Cameron Station
Alexandria, Virginia, 22304-6145 | 2 |
| 2. | Library, Code 052
Naval Postgraduate School
Monterey, California, 93943-5002 | 2 |
| 3. | Dr. Allan D. Kraus
Department of Electrical Engineering
Naval Postgraduate School
Monterey, California, 93943-5002 | 2 |
| 4. | Dr. Conrad F. Newberry
Department of Aeronautical Engineering
Naval Postgraduate School
Monterey, California, 93943-5002 | 2 |
| 5. | Professor Avram Bar-Cohen
Department of Mechanical Engineering
University of Minnesota, Twin Cities
Minneapolis, Minnesota, 55455 | 1 |
| 6. | LCDR Colleen L. Ellis
1828 Rossmoor Way
Modesto, California, 95355 | 2 |