GPO PRICE $ ______

CFSTI PRICE(S) $ ______

Hard copy (HC) 3.00

Microfiche (MF) .65

ff 653 July 65

N67-31357

(ACCESSION NUMBER)

(THRU) 1

265 (PAGES)

(CODE)

CR-65591 (NASA CR OR TMX OR AD NUMBER)

33 (CATEGORY)

FACILITY FORM 602

FOREWORD

This report entitled "Thermal Analyzer Computer Program for the Solution of General Heat Transfer Problems," LR 18902, was prepared by the Lockheed-California Company under NASA Contract NAS 9-3349. Although originally developed by Lockheed in 1956 and continually updated over the past several years, the Thermal Analyzer Program was extensively modified for use under this contract.

Other reports prepared under this contract are:

LR 18899 A Transient Heat Transfer and Thermodynamic Analysis of the Apollo Service Module Propulsion System - Final Report

- Vol. I - Phase I Transient Thermal Analysis
- Vol. II - Phase II Thermal Test Program

LR 18900 A Transient Heat Transfer and Thermodynamic Analysis of the Apollo Service Module Propulsion System - Summary Report

LR 18901 An Introduction to Spacecraft Thermal Control

LR 18903 Thermal Analyzer Computer Program for the Solution of Fluid Storage and Pressurization Problems

LR 18904 Computer Program for the Calculation of Incident Orbital Radiant Heat Flux

LR 18905 Computer Program for the Calculation of Three-Dimensional Configuration Factors

This report was written by Mr. H. D. Schultz of Lockheed's Thermodynamics Department. The contributions of Messrs. R. B. David, F. R. Mastroly, and J. R. Gardner, also of the Lockheed-California Company, to this report are gratefully acknowledged. Mr. David was responsible for the programming and wrote Section VII and Appendices D and E of this report. Messrs. Mastroly and Gardner wrote computer manuals for two earlier versions of the Thermal Analyzer Program. Much of the content of the present report was adapted from the previous manuals.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LOCKHEED
CALIFORNIA COMPANY

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF TABLES

PRECEDING PAGE BLANK NOT FILMED.

P..ECEDING PAGE BLANK NOT FILMED.

SUMMARY

This report discusses the application of the Thermal Analyzer Program, developed for the IBM 7040/7094 direct coupled digital computer or the IBM 7094 digital computer, to complex transient heat transfer problems. The report also discusses a separate data "debugging" program developed for the IBM 7040 digital computer, which allows examination of the program input data prior to submitting the problem for execution.

The transient heat transfer solution is obtained by converting the physical system into one consisting of lumped thermal capacities connected by thermal resistors, and then using the lumped parameter, or finite differences, approach to solve for the temperature history of the system. This solution, although discontinuous in space and time, can be obtained to any desired degree of accuracy by proper selection of lump size and computing interval within certain limitations as described herein.

The program affords direct solution of complex transient problems involving conduction, convection, radiation, heat storage, and ablation. In addition, by being able to specify any quantity as an arbitrary function of any other, it is possible to include such problems as change of state, variable thermodynamic properties, arbitrary variable boundary conditions, and other non-linear effects.

This report discusses the method used to transform the physical heat transfer problem into a resistance-capacitance (R-C) network (which is analogous to an electrical circuit), the numerical evaluation of the equivalent electrical elements, and the method of presentation (input format) required to input the problem into the computer. Several example problems illustrating most of the program features are included.

PRECEDING PAGE BLANK NOT FILMED.

I - INTRODUCTION

Transient thermal phenomena may be studied experimentally, analyzed by graphical or relaxation methods such as the Schmidt plot, or calculated by direct solution of the appropriate differential equations or finite difference equations and approximations. Such methods are usually quite tedious and difficult even for relatively simple cases, and may be inaccurate if the problem is at all intricate.

With increasing aircraft speeds, and with space exploration a reality, the need for accurate transient heat transfer analyses of complex systems has become more acute. Detailed analyses are required for accurately predicting transient structural temperature distributions during high-speed flight, component and environmental temperatures in a space vehicle, the ablation requirements for a reentry body, and in many other cases where hand computations do not suffice. Conventional aircraft, which employ many thermal systems, also require detailed temperature analyses to assure proper system design and operation.

Because of the increased need for detailed thermal analyses, artificial methods are often substituted for the exact solutions of the proper differential equations as a means of obtaining a solution. The method selected for this program is one of many such methods in use and employs the electrical resistance capacitance analog. There are two reasons for this choice:

1. The equations describing any general heat transfer problem are of the same form as those describing an equivalent electrical R-C network. The electrical network equations are simple to set up in finite difference form, and consequently the heat transfer problem may be solved to any desired degree of accuracy.

2. The network setup is easy to visualize in relation to a schematic diagram of the physical problem.

To facilitate the solution of such an analogous network, the Lockheed-California Company has developed the Thermal Analyzer Program. The purpose of this report is to enable a heat transfer engineer who is unfamiliar with the program to use it successfully in solving his problem. A basic familiarity with heat transfer laws is assumed, and hence primary emphasis is placed on the conversion of the physical problem into one that can be interpreted by the computer. Examples demonstrating the program features are included.

II - PROGRAM DESCRIPTION AND CAPABILITIES

The Thermal Analyzer Program is written in FORTRAN IV for the IBM 7040/7094 direct coupled digital computer or the IBM 7094 digital computer. It computes transient temperature distributions in configurations of arbitary complexity, using the electrical resistance-capacity analogy. Solutions are obtained by converting the physical system into one consisting of lumped thermal capacities connected by thermal resistors, and then using the lumped-parameter, or finite-differences approach to solve for the temperature history of a system.

The program permits direct solution of complex transient problems involving conduction, convection, radiation and heat storage. Furthermore, since it is possible to specify any quantity as an arbitrary function of any other, it is also possible to solve such problems as change of state, variable thermodynamic properties, arbitrary variable boundary conditions, and other nonlinear effects.

In developing the Thermal Analyzer Program, a primary objective has been to maximize input flexibility, and hence to keep the program as general as possible. Input format has not been restricted to any particular geometry, rather it is such that resistors and capacitors can be connected in the same manner as could the actual equipment components. Additions or other changes to the network can easily be made by adding or removing cards in the program input deck.

An outstanding feature of the program is its ability to accept various subroutines, or functions, as required by the particular problem. Currently available are various general-purpose and special functions which permit numerous mathematical operations beyond solution of the electrical network itself. These functions are discussed in Section IV.

Standard FORTRAN statements are accepted, allowing the user to add his own subroutines as required. More important, this flexibility in handling subroutines allows new ones to be added without altering the basic program.

The program has the capability to run consecutively several cases which are basically similar, but differ in the value of certain parameters. An example of this is a parametric study of the effect of varying the surface emissivity and absorptivity of a space vehicle. A second restart feature is available in which several cases are run consecutively with the results of the first used as inputs to the second, and so on. The temperatures at the end of each case are recorded on tape and used as the initial temperatures for the subsequent case.

The program also allows the user to specify the format for printing the answers. In addition to the tabulated answers, an option provides for machine plotting of the results.

The steps required to solve a problem using the Thermal Analyzer Program are as follows:

1. The physical problem is set up and defined.

2. The physical problem is specified in terms of time, temperature, and a thermal network analogous to an electrical network consisting of resistances and capacities. This re-definition and re-description of the problem is known as lumping, and puts the problem in the only form which the computer is able to solve.

3. The network is described in detail in a form which allows it to be accepted by the computer program and solved. This involves writing up the program in a standard format.

4. This description of the network is transferred to punched cards which are then put into the computer.

5. The computer program solves the problem and provides the answers.

6. The program then provides for printing these answers in a format which can be prescribed by the user.

7. The answers thus printed are then interpreted by the user to provide the desired solution for the original problem.

Step 1 It is assumed that anyone using this program will be able to describe and specify his problem. The user must have a detailed knowledge of the configuration, including conduction paths and surface properties if conduction and radiation are important heat transfer modes. In addition, it is assumed that boundary conditions, such as external heat inputs and adiabatic interfaces, are known.

Step 2 The most crucial and time-consuming step is the conversion of the physical problem into an equivalent resistance-capacitance network. The user must divide the physical geometry into sections called "lumps" and then calculate the resistance and capacity of these lumps. The capacity of each lump is the thermal capacity (mass times specific heat) of that portion of the physical problem which the lump represents. The use of the lumping process implies that a given portion of the actual structure is at a uniform average temperature. In lumping a problem there are many factors which influence the size, shape, and number of lumps to be used. Among these are the nature of the physical problem, the amount of detailed information desired, and the anticipated transient response rates and temperature gradients. Some of the considerations involved in problem lumping are discussed in Section III, where several examples are presented. Once the method of lumping has been established, each capacitor and resistor is assigned an integer designation number. Although the designation numbers are arbitrary, a systematic numbering scheme is usually employed for convenience. The user then computes the network resistor and capacitor values, following the general procedures outlined in Section III.

Step 3 After computing the network parameters, the problem must be described on input data sheets (General Purpose Data Sheets or FORTRAN Coding Sheets) in a prescribed manner. The input format is described in detail in Section V. The problem description is divided into five distinct blocks and two subroutines. The first three blocks define the network and give the initial values of the temperatures, resistors, and capacitors. The fourth block is a list of sub-blocks of data required for the problem solution. An example of the type of data that might appear here is the point-by-point description of a curve which is to be used by the functions (described below) for interpolation. Each data sub-block is assigned an arbitrary designation number so that it may be referenced later in the program. The fifth block of data lists the printing interval, the final time of the case, and the initial time. The latter two times correspond to the real time of the physical problem.

The user must then prepare a standard FORTRAN subroutine (the FUNCT subroutine) in which he specifies the miscellaneous functions, or operations, which the program must perform. In general, the FUNCT subroutine specifies all operations necessary to solve the problem with the exception of the actual heat balance. An example of a function is the interpolation of a curve described in block 4, to perhaps specify the external heating rate to a portion of the network. The FUNCT subroutine may also be used to call in special subroutines such as aerodynamic heating, ablation, curve plotting, and several others. A complete list of the available functions and special subroutines is given in Section IV.

The last item which the user must code on input sheets is the PRINT subroutine, in which he specifies the quantities that are to be printed out at each printing time, and the desired output format. The program has been set up so that essentially every quantity of interest has an addressable storage location and may therefore be printed out. Section V describes the format used in writing the PRINT subroutine.

Although the FUNCT and PRINT subroutines are ordinary FORTRAN subroutines, standard input formats are presented in Section V, and the user need not have a knowledge of FORTRAN to prepare these routines.

<u>Steps 4 and 5</u> The information on the input sheets is transferred to punched cards which are then input to the computer, and the problem is solved. The actual running of the program is described in Appendix D.

<u>Step 6</u> The answers are printed on a line printer using the format prescribed by the user in the PRINT subroutine. As mentioned above, machine plotting of the results is optional.

<u>Step 7</u> Interpretation of the answers to provide the desired solution for the original problem should present no difficulties.

III - PROBLEM SETUP

BASIC THERMAL SYSTEM AND ELECTRICAL ANALOG

The Thermal Analyzer Program requires that the problem be described as an equivalent network using resistance, capacity, and temperature to define the heat transfer situation.

Thermal resistance refers to resistance to heat flow, analogous to electrical resistance which refers to resistance to current flow.

In any case involving heat transfer between two points, at temperatures T_i and T_k, the heat flow is given by an equation (analogous to Ohm's electrical law) as follows:

$$q = \frac{T_i - T_k}{R} \qquad (3\text{-}1)$$

Some simple examples might be given here:

For conduction, $q = kA\frac{\Delta T}{\Delta x} \Rightarrow R = \frac{\Delta x}{kA}$

For convection, $q = hA\Delta T \Rightarrow R = \frac{1}{hA}$

If q is in Btu/hr or Btu/sec, R must be in hr.°F/Btu or sec.°F/Btu, respectively.

Transient analyses differ from steady-state analyses in that heat storage in a material undergoing a heating or cooling process is accounted for, thus causing a time lag in the temperature response of the material. The quantity of heat thus stored, and the description of the temperature response, will depend on the properties of the material itself. These properties determine the quantity called "thermal capacity," which behaves in the thermal network in the same manner as electrical capacity behaves in an electrical network. Thermal capacity must be in the units of heat quantity per degree of temperature (e.g., Btu/°F) and is a function of the material's density, specific heat and volume. Physically, the thermal capacity of a material represents the amount of heat stored in a given volume for each degree of temperature rise experienced by the material.

For most materials, property values such as thermal conductivity, emissivity, and specific heat will be a function of temperature. In those cases where this variation is significant, it may be taken into consideration in the program through the use of curves as discussed in Section IV.

PROBLEM LUMPING

To transform the physical problem into a form suitable for the computer, it is necessary to convert it into an equivalent resistance-capacitance (R-C) network. This is accomplished by dividing the physical system into sections called "lumps" and calculating the resistance and capacity of these lumps. A "lump," then, is any portion of the physical problem which (though not necessarily physically disconnected) will not be connected to any other portion of the problem except by resistors. The discussions to follow outline some of the considerations involved in problem lumping.

Location of Lumps

Although the lumps may take any size or shape, they should bear a simple relationship to the physical geometry. As a general rule, the nodes (the points where the lump capacities are assumed to be concentrated) should be located at those points where temperature data are desired, and these in turn are dictated by the nature of the problem itself. This is illustrated by the following examples. In each instance, the node locations are determined first and the lump boundaries located afterwards.

Example 1, Re-entry Structure - Figure 3-1 shows a re-entry structure consisting of an inner and outer skin separated by a z-section, with the outer skin protected by a ceramic coating. It is assumed that the section is not influenced by other such sections and that the problem is two-dimensional,

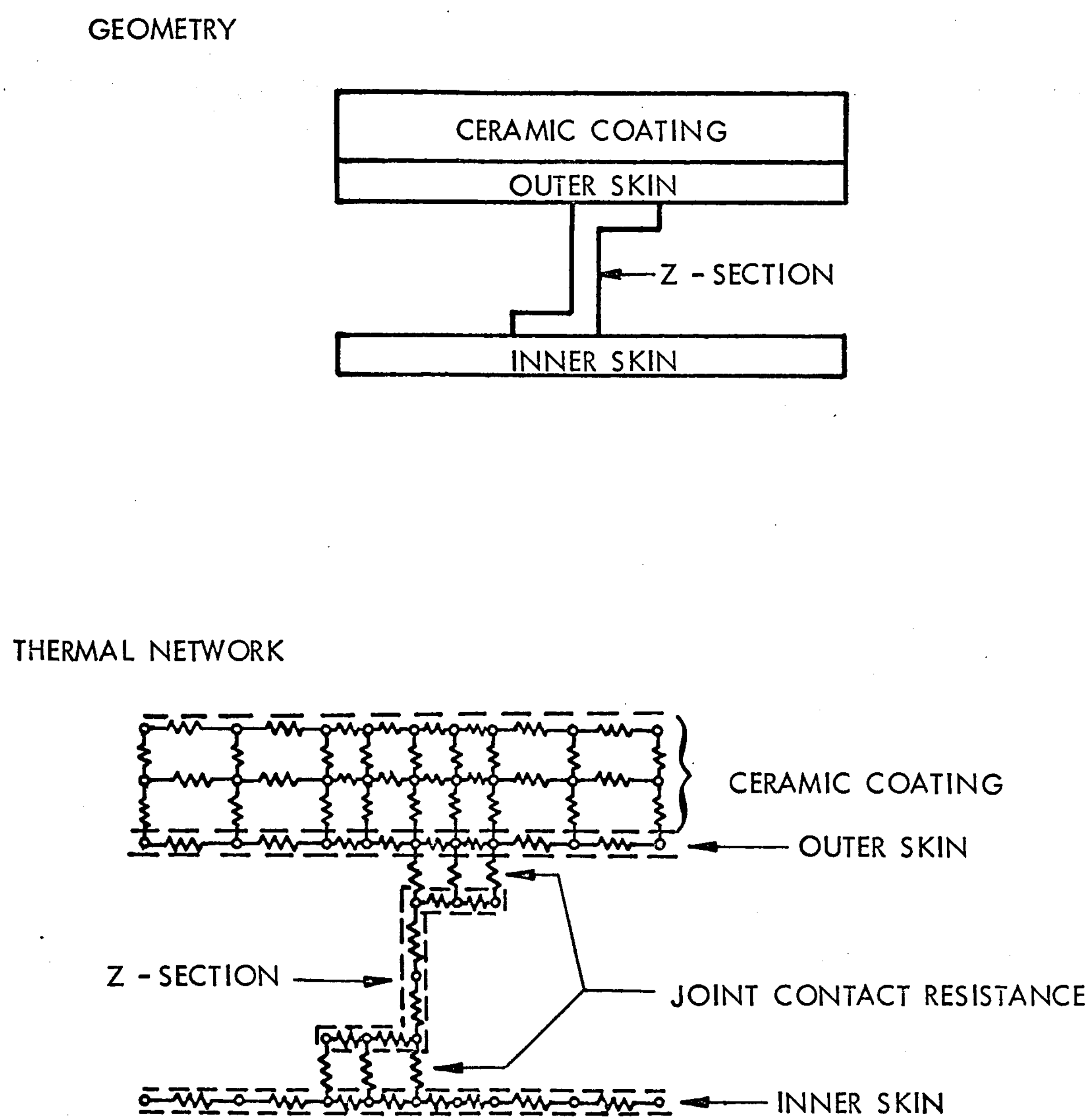


Figure 3-1. Physical Geometry and Corresponding Thermal Network for a Re-Entry Structure

i.e., no heat flow in and out of the plane of the drawing. However, it is a simple matter to connect many such sections into a complex three-dimensional problem. For this example, the temperature of the ceramic surface and the underlying structure is of primary importance. Also, it is assumed that large lateral temperature gradients exist near the z-section, with smaller gradients further out. With these points in mind, the resulting network is as shown in the lower sketch in Figure 3-1.

With regard to lump boundaries, the usual procedure is to place them so that the nodes are approximately in the center of the lumps except, of course, at the boundaries of the various layers. The problem of lump boundaries and the calculation of resistors and capacitors are discussed in detail in subsequent sections.

Example 2, Electronic Equipment Rack - Figure 3-2 shows an electronic equipment rack consisting of intersecting webs on which heat dissipating components are mounted. The corresponding thermal network is shown in the lower sketch. Since component temperatures are of primary interest here, the various capacities are assumed to be concentrated at points corresponding to equipment locations. However, this places the nodes inside the web boundaries as shown at the free ends of the two webs. At the juncture between the two webs, a string of zero capacitance nodes (designated by ⊗ in Figure 3-2 and often referred to as "dummy" nodes) is required to effect a connection between webs. This technique is particularly useful in a complex network where many such interconnecting webs are involved, since it allows each web to be treated separately and then connected to other such webs at the various dummy nodes.

Example 3, Spacecraft Window - A section of a spacecraft window exposed to convection and radiation on both surfaces is shown in Figure 3-3. For this problem, one-dimensional heat flow is assumed. Three lumps have been arbitrarily assumed for the conduction network through the window, with nodes appearing at the boundaries to properly account for convection and radiation, both of which depend on the surface temperature.

The preceding illustrations are of but a few of the many lumping situations which arise. Probably the most important factor in problem lumping

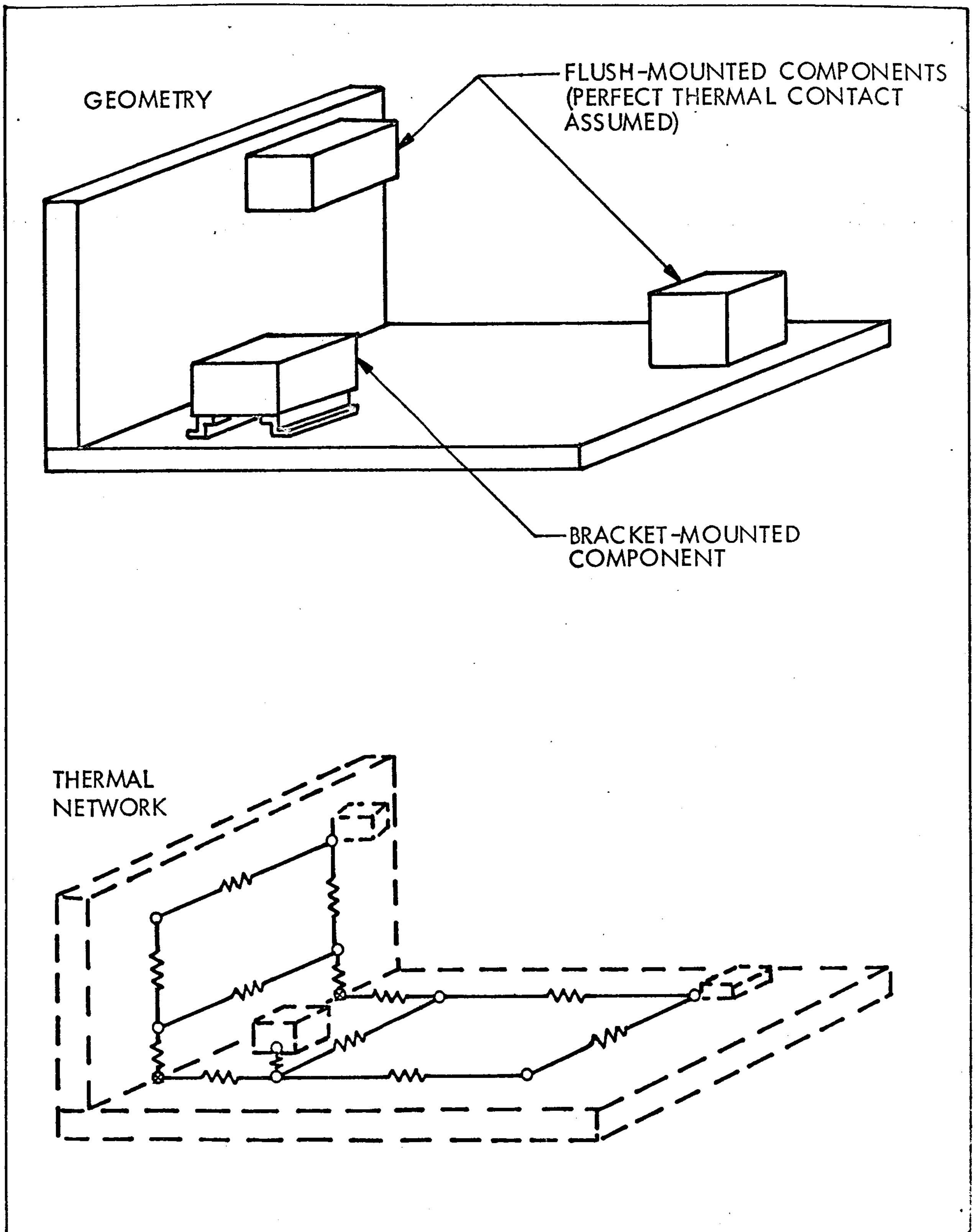


Figure 3-2. Physical Geometry and Corresponding Thermal Network for an Electronic Equipment Rack

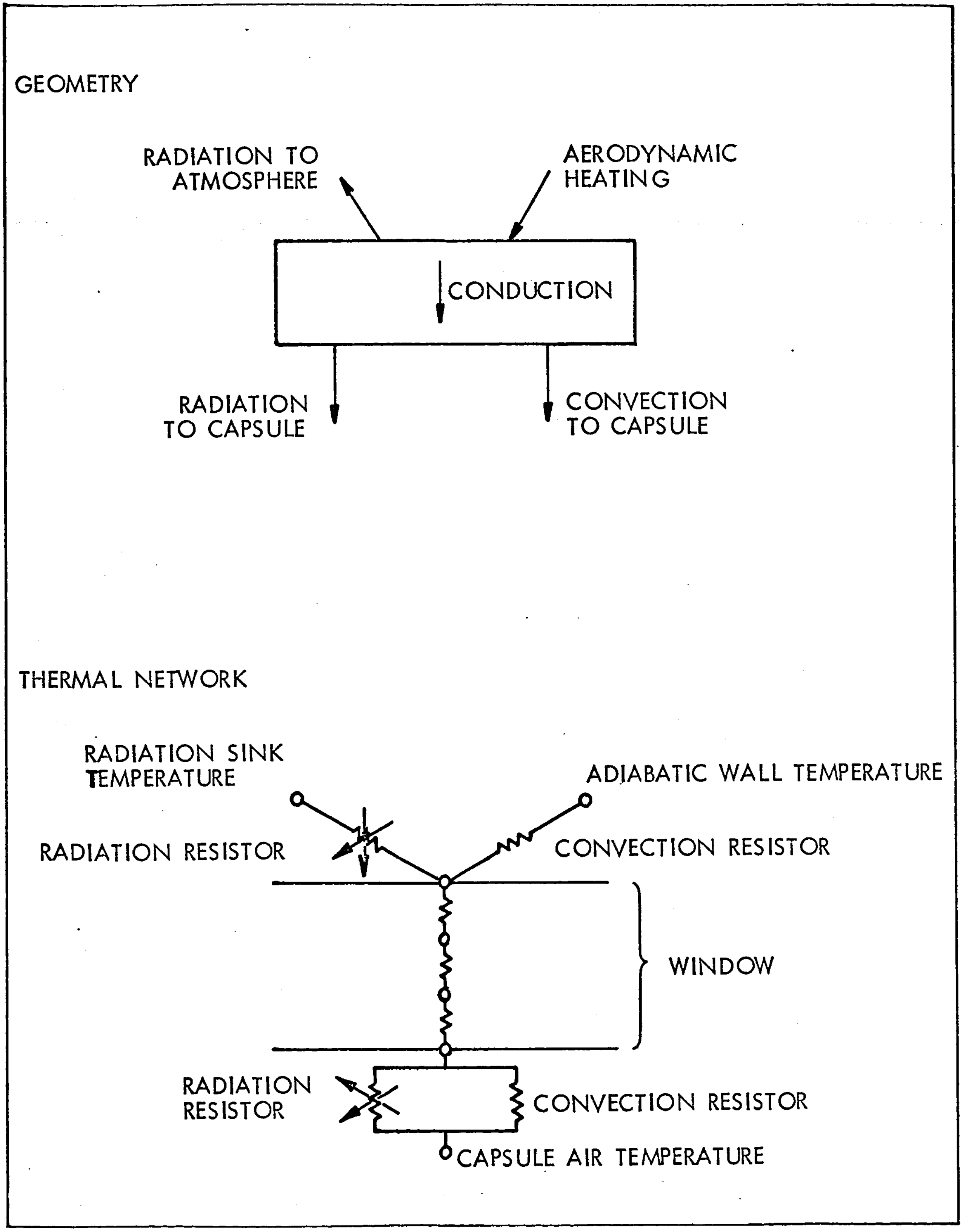


Figure 3-3. Physical Geometry and Corresponding Thermal Network for a Window Section

is experience, and this cannot be acquired merely by reading reports. Also, since no two situations are identical, it is impossible to cover all conceivable situations in a single report. It is hoped, however, that the examples presented here and in subsequent sections will provide some insight into the problems involved.

Choice of Lump Size

In selecting the optimum lump size, recourse must be made to logic, and, most of all, experience. Here again, the nature of the physical problem will dictate to a great extent the final decisions. Generally, the choice of lump size will be based upon these factors:

1. Consideration of inaccuracies introduced into the system resulting from the finite difference method of solution. These inaccuracies decrease (not necessarily linearly) as lump size decreases. About the only definite statement which can be made is that lump size should be as large as possible without causing excessive inaccuracies.

2. Anticipated temperature gradients and relative rates of transient response. Where it is suspected that large temperature gradients will occur, nodes should be placed closer together than those where these gradients are smaller. This is especially true when the thermal diffusivity of a particular layer is very small, with the resulting temperature gradients across it being highly nonlinear.

3. Convenience in visualizing the network and making calculations.

4. Program capacity. Ordinarily the capacity of the computer is not approached; on occasion, for extremely large and complex problems, this becomes an important consideration.

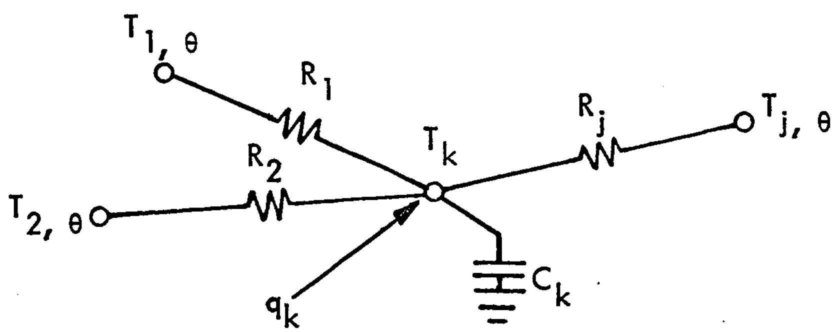
5. Consideration of machine time, which costs money. Not only do small lumps increase the number of nodes to be computed, but also they result in a smaller computing interval (difference in real time between successive steps), thus greatly increasing machine time.

METHOD OF SOLUTION

As previously indicated, the Thermal Analyzer Program solves equations in finite difference form by means of an R-C electrical analog finite difference method. The comparable values in the two systems may be noted as follows:

THERMAL	ELECTRICAL
Temperature	Voltage
Heat Flux	Current
Resistance	Resistance
Capacity	Capacity

At a given node point k,



the solution is obtained by applying Kirchhoff's Law at a point, or

$$\sum_j \frac{T_{j,\theta} - T_{k,\theta}}{R_j} + q_k = C_k \frac{dT_k}{d\theta} \qquad (3\text{-}2)$$

where

$T_{j,\theta}$ = Temperature at time θ of any arbitrary node j connected to node k by a resistor R_j

R_j = Resistor connecting nodes j and k

$T_{k,\theta}$ = Temperature of node k at time θ

$T_{k,\theta+\Delta\theta}$ = Temperature of node k after time increment $\Delta\theta$

C_k = Capacity of node k

q_k = Arbitrary heat input into node k

By making the assumption that the surrounding temperatures, T_j, remain constant over a time interval $\Delta\theta$, it is possible to integrate equation 3-2 directly. However, as a result of a comparison study, it was found that better results were obtained by using the equation which results from assuming

$$\frac{dT_k}{d\theta} \approx \frac{T_{k,\theta+\Delta\theta} - T_{k,\theta}}{\Delta\theta} \tag{3-3}$$

and solving for $T_{k,\theta+\Delta\theta}$ directly than by using the integrated equation. This comparison was made by running the same problem using both equations and varying the computing interval $\Delta\theta$. It was found that the linear equation, i.e., that obtained by using equation 3-3, was far less sensitive to $\Delta\theta$, and that the results obtained using the integrated equation approached the linear results as $\Delta\theta \rightarrow 0$. Although the integrated equation is "exact," it is suspected that the linear approximation, which "leads" the exact solution (it predicts higher temperatures in a warming system, lower temperatures in a cooling system), tends to anticipate the results. As a result, the equation used by the computer to solve the heat balance at a node was derived by combining equations 3-2 and 3-3 to obtain

$$T_{k,\theta+\Delta\theta} = \frac{\Delta\theta}{C_k}\left[\sum_j \frac{T_{j,\theta} - T_{k,\theta}}{R_j} + q_k\right] + T_{k,\theta} \tag{3-4}$$

If the value of capacity C_k is zero, e.g., in a steady state problem, $T_{k,\theta}$ in Equation 3-2 is replaced by $T_{k,\theta+\Delta\theta}$ to give

$$T_{k,\theta+\Delta\theta} = \frac{\sum_j \frac{T_{j,\theta}}{R_j} + q_k}{\sum_j \frac{1}{R_j}} \tag{3-5}$$

If no capacitor is attached to node k, i.e., C_k unspecified, no heat balance is computed at node k and T_k remains unchanged.

It is to be noted that the new temperature at a node is based upon the temperatures at the previous time point. To make the new temperatures independent of the order in which they are computed, each node is provided with two temperature storages, one for the "old" temperature, $T_{k,\theta}$, and one for the "new" temperature, $T_{k,\theta+\Delta\theta}$. At the beginning of each cycle, the values in the two storages are identical. During the heat balance, the temperatures in the "T at θ" block are used to compute new values which go into the "T at $(\theta + \Delta\theta)$" block, the old temperature values remaining unchanged. At the end of the heat balance, the temperatures in the "θ" block are set equal to those in the "$(\theta + \Delta\theta)$" block and the process repeated.

METHOD USED TO DETERMINE COMPUTING INTERVAL

In the section on lump size, it was stated that smaller lumps result in a smaller $\Delta\theta$. At each time point θ, the machine computes the time constant, or RC product, of each node for which a non-zero capacitor is specified. This time constant is defined as the product of the capacity of the node times the equivalent resistance to that node. This equivalent resistance is defined as the parallel combination of all resistors connected to the node in question. Therefore,

$$(RC)_k = \frac{C_k}{\sum\limits_j \frac{1}{R_j}} \qquad (3\text{-}6)$$

where $(RC)_k$ = time constant of node k ~ seconds (if R in sec-$^{\circ}$F/Btu).
Since, for conduction, $R = \delta/kA$, and for capacity, $C = \rho c A \delta$, the RC product of a node is a function of the square of its thickness (δ) in the direction of the heat transfer, viz.

$$RC = \frac{C}{\sum_j \frac{1}{R_j}} = f\left[\frac{\delta}{kA} \cdot \rho c \delta A\right] = f\left[\frac{\rho c}{k} \cdot \delta^2\right] \qquad (3\text{-}7)$$

To obtain the computing interval, $\Delta\theta$, the computer searches the network to find the minimum RC product and compares this with the printing interval (the real-time increments for which the output is desired). The computer then takes whichever is smaller and multiplies it by a certain fraction to obtain $\Delta\theta$. Ordinarily this factor is 0.25, and unless changed by the user in the FUNCT subroutine, it remains so. However, there are times when a factor other than 0.25 is convenient, for example:

1. Increased $\Delta\theta$/RC (possibly as large as 0.9) can be used when only a small percentage of the nodes have small RC products compared with the others, and when a major reduction in machine time can be obtained at the cost of a small reduction in accuracy.

2. Reduced $\Delta\theta$/RC (as small as 0.1 or even smaller) should be used when it is desired to have the machine compute more frequently during certain portions of the program. In conjunction with the MMF function to be described subsequently, this procedure enables the computer to pick up the maximum and minimum temperatures of a particular node with a greater degree of accuracy. Since it is possible to make $\Delta\theta$/RC a variable, a reduced $\Delta\theta$/RC during one portion of the program can be offset, as far as machine time is concerned, by an increased $\Delta\theta$/RC during the other portions of the program.

CALCULATION OF CIRCUIT PARAMETERS

This section presents the general procedures employed to calculate capacitors and resistors, and the methods used to accommodate the various modes of heat transfer (conduction, convection, and radiation). The procedures required to transcribe the network parameters to data input sheets are described in Section V.

In the discussions which follow, it is assumed that ordinary engineering units will be used for program input. Specifically, the following units are assumed:

a. Heat Flow, Btu/sec

b. Temperature, °F

c. Time, sec

d. Resistance, sec °F/Btu

e. Capacitors, Btu/°F

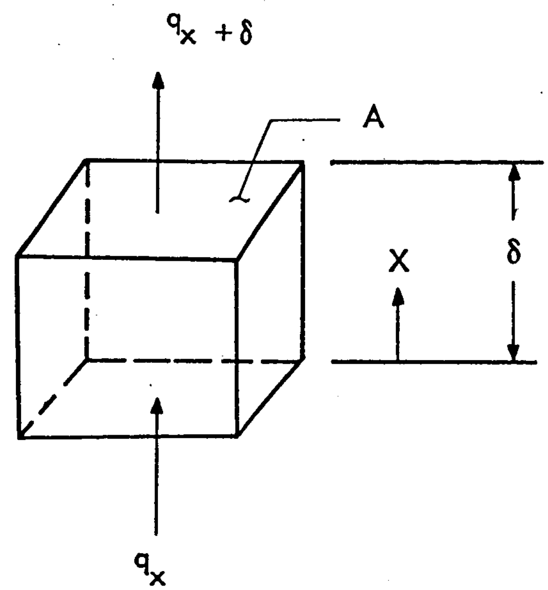
In cases not involving radiation, any consistent set of units may be employed for program input. Also in such cases, the time scale can be changed and all temperatures input as absolute (°R or °K) quantities. However, for radiation resistors the machine adds 460°F to the appropriate temperatures, and since a dimensional quantity, the Stefan-Boltzmann constant, is incorporated into the program as discussed later, it is not recommended that units other than those tabulated above be used for cases involving radiation.

Resistor Values

Resistors must be of the dimension time °F/Btu. For most practical engineering problems, particularly those involving materials with low RC products, the most convenient unit of time is seconds. Consequently, resistance is given in sec °F/Btu, and the heat flow is in Btu/sec. The form of the resistor depends on the particular mode of heat transfer involved.

Conduction Resistors - In all cases, conduction resistors are computed by the formula

$$R = 3600 \int_0^{\delta} \frac{dx}{kA}, \quad \frac{\text{sec } ^\circ\text{F}}{\text{Btu}} \qquad (3\text{-}8)$$



where:

k = thermal conductivity, Btu/hr ft °F

A = cross-sectional conductive heat transfer area, ft^2

x = distance along conductive path, ft

R = resistance, sec- °F/Btu

For a rectangular parallelepiped or a cylinder with vertical sides, or for any configuration with constant cross-section, vertical sides, and parallel faces,

$$R = \frac{3600\,\delta}{kA}, \; \frac{\text{sec °F}}{\text{Btu}} \tag{3-9}$$

Radiation Resistors - Since the radiation interchange from surface 1 to surface 2 is

$$q_{1-2} = \frac{T_1 - T_2}{R} = \frac{\epsilon_{12} A_1 F_{12} \sigma \left(\tau_1^{\,4} - \tau_2^{\,4}\right)}{3600} \text{ Btu/sec,} \tag{3-10}$$

(where τ_i = absolute temperature of node i ~ °R)
R will be of the form

$$R = \frac{3600}{\epsilon_{12} A_1 F_{12} \sigma \left(\tau_1^2 + \tau_2^2\right) (\tau_1 + \tau_2)} \qquad (3\text{-}11)$$

where:

ϵ_{12} = emissivity factor

A_1 = area of radiating surface, ft^2

F_{12} = shape factor from surface 1 to surface 2

σ = Stefan-Boltzmann Constant = 0.1713×10^{-8} Btu/hr ft^2 $°R^4$

The thermal analyzer program computes the radiation resistor

$$R = \frac{1.0}{\sigma K_{rad} \left[(T_1 + 460)^2 + (T_2 + 460)^2 \right] \left[(T_1 + 460) + (T_2 + 460) \right]} \qquad (3\text{-}12)$$

given the value

$$K_{rad} = \frac{\epsilon_{12} A_1 F_{12}}{3600} \quad \frac{ft^2 \ hr}{sec} \qquad (3\text{-}13)$$

The value for σ must not be included since it is built into the program. Note that since the program adds 460°F to each temperature, all temperatures must be in °F.

In addition to the above, it is possible to have the computer evaluate its own value for K_{rad} as will be discussed in Section IV.

Convection Resistors, No Change in Fluid Temperature -

$$R = \frac{3600}{\int_0^A h \, d A_h} \quad \frac{sec \ °F}{Btu} \qquad (3\text{-}14)$$

where:

R = resistance, sec-°F/Btu

h = heat transfer coefficient, Btu/hr-ft^2-°F

A_h = convective heat transfer area, ft^2

For common sections when an average value of h can be used for the given area A_h,

$$R = \frac{3600}{h\ A_h} \quad \frac{\text{sec }°\text{F}}{\text{Btu}} \tag{3-15}$$

Convection Resistors, With Change in Fluid Temperature (Duct Temperature Drop) - The problem may be approached from two directions. The first assumes that the time for the fluid to pass a lump portion of the duct wall is very small compared with the time constant of the wall lump, i.e.,

$$\frac{\theta_L}{T_w} = \frac{L\ h\ P}{3600\ \rho_w\ c_w\ A_w\ V} << 1$$

and that the thermal capacity of the fluid element is very small compared with the thermal capacity of the wall element, i.e.,

$$\frac{C_a}{C_w} = \frac{\rho_a\ c_v\ A_c}{\rho_w\ c_w\ A_w} << 1$$

where:

L = total passage length, ft

θ_L = time for fluid to flow through the passage, sec

T_w = wall time constant, sec

h = heat transfer coefficient, Btu/hr-ft^2-°F

P = perimeter of passage, ft

ρ_w = density of wall material, lb/ft^3

c_w = specific heat of wall material, Btu/lb °F

A_w = wall cross-section, ft^2

C_a = thermal capacity of fluid, Btu/°F

C_w = thermal capacity of wall, Btu/°F

ρ_a = density of fluid, lb/ft^3

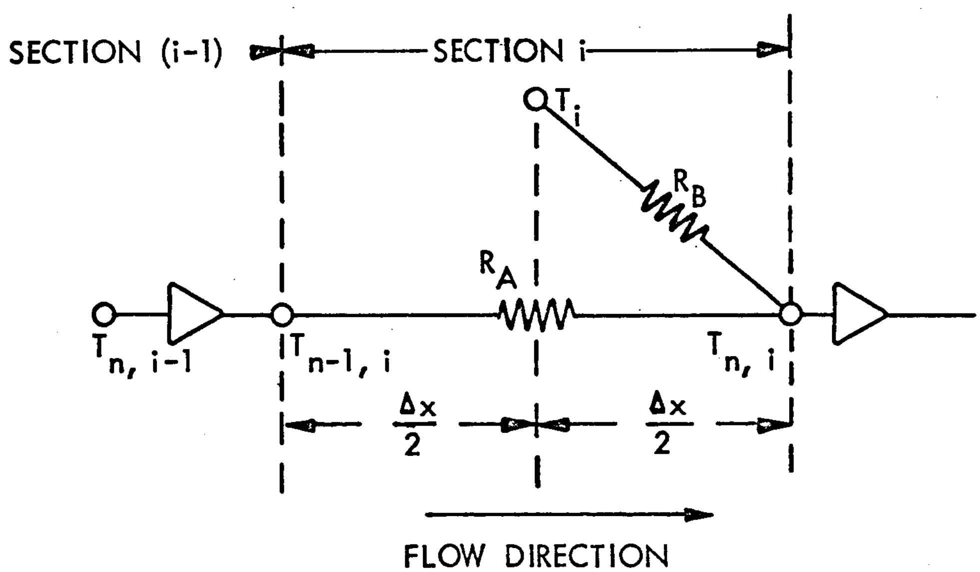
c_v = fluid specific heat at constant volume, Btu/lb °F

A_c = passage cross-section area, ft^2

V = velocity, ft/sec

This problem is treated as in methods 1 and 2, in the following paragraphs. If these ratios are not very small, the problem must be treated as in method 3:

1. With uniform temperature in the passage walls in planes normal to the airflow direction,



$$R_A = \frac{3600}{W\ c_p}$$

$$R_B = \frac{3600}{W\ c_p\ (e^{\beta} - 1)}$$

where:

$\beta = h A_h / W c_p$

h = heat transfer coefficient, Btu/hr-ft^2-°F

A_h = convective heat transfer area, ft^2

W = weight flow of fluid, lb/hr

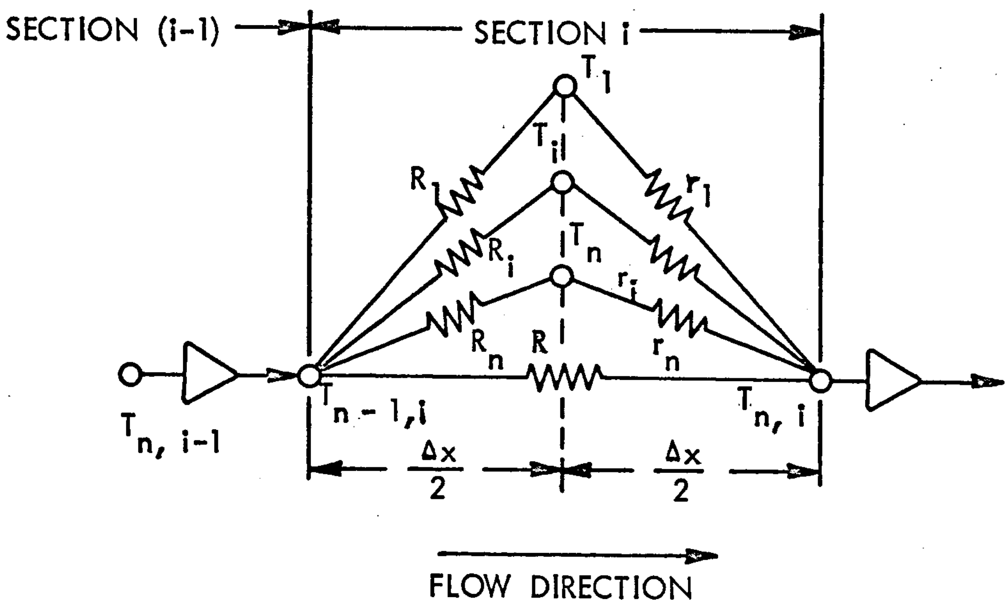
c_p = fluid specific heat at constant pressure, Btu/lb-°F

T_i = wall temperature, °F

T_{n-1} = air temperature at inlet of the passage, °F

T_n = air temperature at passage outlet, °F

2. With temperature gradients in the passage walls in planes normal to the flow direction as well as in the flow direction, the setup becomes somewhat more complicated.



at the inlet end

$$R_i = \frac{3600}{(h\ A)_i \left| \frac{1}{\beta} - e^{-\beta} (1 - e^{-\beta})^{-1} \right|}$$

LOCKHEED CALIFORNIA COMPANY

and at the outlet end

$$r_i = \frac{3600}{(h\ A)_i \left| (1 - e^{-\beta})^{-1} - \frac{1}{\beta} \right|}$$

$$R = \frac{3600\ (e^{\beta} - 1)}{a \left| (1 - e^{-\beta})^{-1} - \frac{1}{\beta} \right|}$$

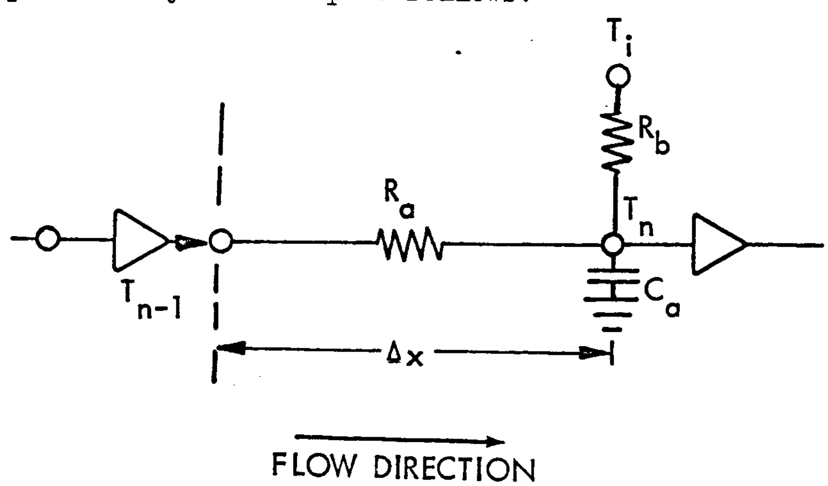
where

$$a = \sum_{i=1}^{n} (h\ A_h)_i$$

$$\beta = \sum_{i=1}^{n} \frac{(h\ A_h)_i}{W\ c_p}$$

and T_i, T_{n-1} and T_n are defined as before.

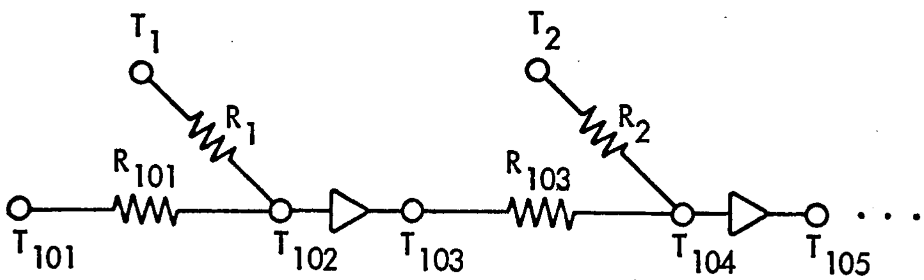
3. If the conditions $\theta_L/T_w \ll 1$ and $C_a/C_w \ll 1$ are not met, the problem may be set up as follows:



$C_a = \rho c_v\ A_c \Delta x$ $\qquad R_a = \frac{3600}{W\ c_p}$ $\qquad R_b = \frac{3600}{h\ A_h}$

This is a more general case; however, the conditions necessary for using Methods 1 or 2 are met in ordinary passage flow.

Derivations and plots of the equations presented above are given in Appendix C. The symbols ——▷—— indicate that the inlet air temperature at section (i) is to be set equal to the outlet air temperature at section (i-1); i.e., to cause the temperatures to be influenced by upstream heat transfer but independent of downstream heat transfer. Failure to do this will cause the equations describing the network response to contain terms not present in the lumped parameter equations of the thermal system. For those cases where method 1 or 2 can be used, the computer is equipped to transfer the new temperature at the exit of one lump to the "new" temperature block of the inlet temperature of the next lump downstream. This transfer is made in the Function routine by the statement T(i) = T(i-1). The statement TN(i) = TN(i-1) should also appear if it is desired to print out the temperatures since T(i) will be destroyed in the heat balance before printing occurs. As an example, consider the network:



In the FUNCT subroutine, the instructions

T(103) = T(102)

T(105) = T(104)

etc.

should appear. The capacitance of nodes 102, 104, etc. should be specified as zero. Nodes 103, 105, etc. should not be mentioned in the capacitor block.

In the event that method 3 is required, the same instructions must appear in the FUNCT subroutine, but the true capacitance must be specified for the fluid nodes. The dummy nodes (T_{n-1} in the figure on page 3-18) should not be mentioned in the capacitor block.

Capacitor Values

The thermal capacity of a lump is calculated in all cases through the formula

$$C = A\delta \int_{T_1}^{T_2} \frac{\rho c}{T} \, dT + \int_0^{\delta} \rho c \, A \, dx \qquad (3\text{-}16)$$

where:

C = thermal capacity, Btu/°F

ρ = density, lb/ft^3

c = specific heat, Btu/lb°F

In most cases of practical interest, ρ and c are constant over a large temperature range, and the above formula reduces to

$$C = \rho c \int_0^{\delta} A \, dx \qquad (3\text{-}17)$$

where:

$$\int_0^{\delta} A \, dx = \text{volume of lump, ft}^3$$

Variable Resistors and Capacitors

In many cases of practical interest, material properties, and hence resistor and capacitor values, are temperature dependent. To handle this problem the user can input the value as a function of temperature in curve form, or the machine can be given tables of the material properties and directed to compute its own values of resistance and capacitance as described in Section V. To accomplish the latter, the Thermal Analyzer Program has a special library tape containing thermal properties of several commonly used structural materials, insulations, propellants, and pressurant gases. The data contained on this tape, and the functions required to call for it, are discussed in Section IV.

Heat Inputs

Heat inputs to a particular node can be called out directly in the function subroutine as a constant or variable quantity, as described in Section IV. Many times, however, it is convenient to specify the heat input to a particular node in the form of a temperature through a resistor, where

$$R_k = \frac{T_i - T_k}{q_k}$$

If $T_i \gg T_k$, then q_k is essentially independent of T_k, and R_k can be computed by the relationship

$$R_k = \frac{T_i}{q_k} .$$

As an example, to input 15 Btu/sec into a particular node set $T_i = 15 \times 10^{10}$ (for 15 Btu/sec), and therefore

$$R_k = \frac{15 \times 10^{10}}{15} = 10^{10} \quad \frac{\text{sec } ^\circ\text{F}}{\text{Btu}}$$

IV - FUNCTIONS AND SPECIAL SUBROUTINES

The FUNCT subroutine is the most important and the most powerful section of the Thermal Analyzer Program. This subroutine contains a listing of all arithmetic operations, curve interpolations, and special functions including radiation, which are to be performed during program execution. The functions are executed at the start of each cycle (before the heat balance) in the order specified, a characteristic which becomes important when the execution of one function involves the result of another. This section presents a brief description of the functions and special subroutines available, what they are capable of, and where input errors are most likely to occur. The FUNCT subroutine input format is described in Section V.

DEFINITION OF TERMS

In the following discussion, several terms are used repeatedly to refer to the various elements of a given function callout. These terms are defined below:

1. Floating-Point Number - All numerical values are input in floating point, and as such must contain a decimal point. The number 30.6 is an example. For very large or very small numbers, such as 10^{10} or 10^{-10}, the capital letter E is used to signify a power of 10. For example, 1.30E6 means $1.30x10^6$ and 6.91E-4 means $6.91x10^{-4}$.

2. Fixed-Point Number - Fixed point numbers are integers, either positive or negative, and as such are characterized by the absence of a decimal point. They are used to designate data sub-blocks and in numerous instructions to indicate the number of steps or items involved in the execution of that instruction.

3. Literal Numbers - These elements are characterized by a letter followed by a number in parentheses, such as T(6), R(14), etc. These are the addresses of the various items of numerical data, i.e., the location in the computer core where the data is stored. They tell the computer where to look for the numerical data required to perform a given instruction. As a result of this, they are referred to as "addressable elements."

In discussing the various functions the following notations are used:

1. When the variables of a function are arbitrary circuit elements, the independent variable is denoted by X and the dependent variable by Y.

2. N denotes the designation number of a data sub-block.

3. Unless otherwise specified, the letters A, B, C, D, E, F, G denote floating point constants whose values may appear explicitly in the function callout, or may be stored in an addressable element.

4. When the variables of a function are temperatures, Y refers to the temperature at time $\theta + \Delta\theta$ and X refers to the temperature at time θ.

LIST OF ADDRESSABLE ELEMENTS

Each item of numerical data of interest to the user is assigned an address which can then be used to refer to that item. A list of these addresses follows:

T(i)	"Old" temperature of the i-th node (see below)
TN(i)	"New" temperature of the i-th node
R(i)	Resistor number i
C(i)	Capacitor number i
RC(i)	The RC product at C(i)
Q(i)	The arbitrary heat input into C(i)
P(LN + i)	The i-th number in the data block whose designation number is N, and where LN = LOC(N) (see Section V). In determining the i-th number the designation number itself is not counted. Also, when specified in this manner, the PER in a periodic curve is not counted.
M(i)	The i-th miscellaneous element, where

M(1)	Current time, θ
M(2)	Initial time

M(3)	Final time
M(4)	Print interval
M(5)	Computing interval, $\Delta\theta$
M(6)	Minimum RC product of the network
M(7)	The factor by which M(6) is multiplied to produce M(5). Unless changed by the variables, a factor of 0.25 is used.
M(8)	Print trigger, normally zero. If a variable puts any non-zero into this storage, a print-out of the output block will result at the end of that cycle.
M(9)	The number of that capacitor which produces the minimum RC product.
M(11)	The time of the next regular print
M(12)	The recovery temperature in the Eckert Aerodynamic Heating Subroutine (see Appendix A).
M(14)	The computing interval used during the previous cycle, ie., $\Delta\theta_{(\theta - \Delta\theta)}$.
M(15)	The product of M(6) and M(7). Ordinarily, M(15) = M(5). However, M(5) may be reduced at a print time.
M(16)	The total number of computing cycles since the problem was begun.

The missing elements M(10) and M(13) were used in a former version of the Thermal Analyzer Program, written in machine language.

The temperature designated as T(i) is the "old" temperature of node i and is usually used as the independent variable when calculations involving temperature are performed, eg., heat balances. TN(i) is the "new" temperature of node i and is used in the functions as the dependent variable, or the new calculated temperature. At the end of each heat balance T(i) is set equal to TN(i).

MATHEMATICAL OPERATIONS

Any of the built-in subroutines (or functions) of the FORTRAN IV system may be used as an arithmetic statement in the FUNCT routine. Because of the many operations available, only a few will be mentioned here. For a more detailed listing, the user is referred to the FORTRAN reference manual.

The arithmetic operation symbols +, -, *, /, ** denote addition, subtraction, multiplication, division, and exponentiation, respectively. Unless changed by the use of parentheses, the order of computation is understood to be as follows:

(a)	exponentiation	(**)
(b)	multiplication and division	(* and /)
(c)	addition and subtraction	(+ and -)

For example, the expression

Y = A + B/C - D**E*F - G

will be taken to mean

$$Y = A + \frac{B}{C} - D^{E}F - G$$

Parentheses may be used to override the order in which the operations are to be computed. If parentheses appear, the expression within the innermost parentheses is computed first, following the order of computation given above. The computation then proceeds outward to the next parentheses, and so forth. As examples, the expression

Y = (A + B)/C - D**(E*F) - G

will be taken to mean

$$Y = \frac{A + B}{C} - D^{EF} - G$$

and the expression

$$Y = A + (B/(C - D))**(E*F - G)$$

will be taken to mean

$$Y = A + \left(\frac{B}{C-D}\right)^{EF-G}$$

In addition to these arithmetic operations, the common mathematical functions such as logarithm, exponential, sine, cosine, square root, arctangent, and absolute value are available to the program user. Since their use is self-explanatory, they are listed below with only a brief explanation.

Function	Description
LOG	Logarithm to the base e
EXP	Powers of e or exponential
SIN	Sine of an angle whose measure is given in radians
COS	Cosine of an angle whose measure is given in radians
SQRT	Square root
ATAN	Arctangent or angle in radians of a given tangent value
ABS	Absolute value

As an example,

$$Y = A + SQRT(B**C)$$

means

$$Y = A + \sqrt{B^C}$$

In the FORTRAN language it is also possible to have functions of functions. As an example

```
Y =  EXP (SQRT((COS(2.*A)) **2. + B))
```

means

$$Y = e^{(\cos^2 2A + B)^{1/2}}$$

CURVE INTERPOLATION

One of the most commonly used functions involves some sort of curve interpolation, of which three types are available, viz., linear, parabolic, and linear bivariate. With curve interpolation, any addressable element listed above can be made a function of any other, or others, including itself. In addition to specifying simple interpolation, the instructions may be modified to provide for multiplying the curve value by some other number which can be either a fixed factor or some other variable.

A description of the various interpolation routines follows. The use of these routines will be further illustrated by the example problems of Section VI.

Linear Interpolation

This is the simplest and most commonly used interpolation routine. The callout is

```
Y = LIN(X, N)
```

where

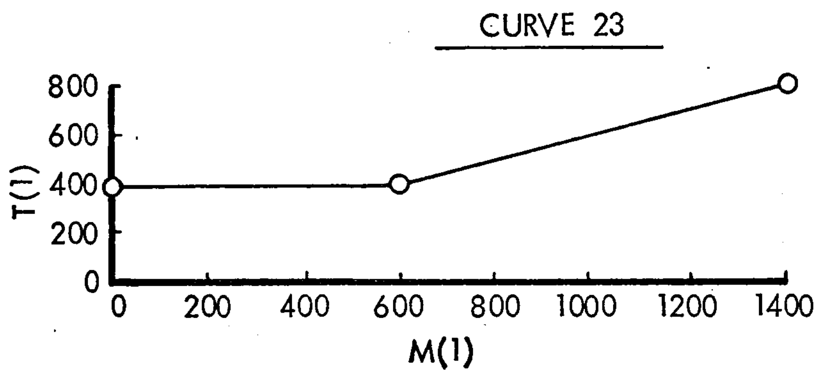
Y = linear function of X given by curve N

As an example, consider

$$T(1) = LIN\ (M(1),\ 23)$$

The temperature of node 1 is a linear function of time given by curve 23.

Curve 23 might appear as shown below:



The curve would be described on input data sheets as follows:

COLUMN

→ 1	5	10	15	25	35	45	55	65	69 72
DEC		23							4001
DEC06			0.	400.	600.	400.	1400.	800.	4002
DEC01			0.						4003
DEC		-23							4004

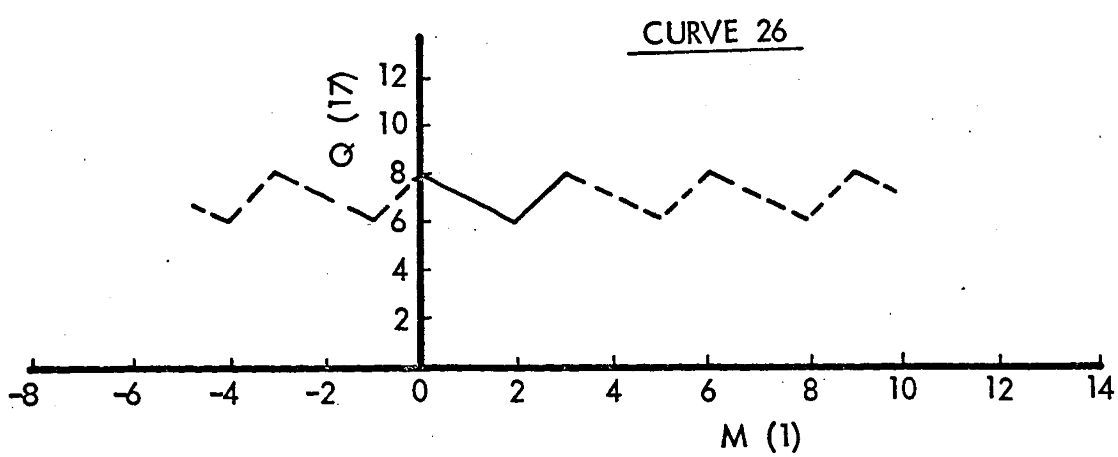
The input format is described in detail in Section V. Briefly, a card containing DEC in columns 1, 2, and 3 and the curve designation number in columns 6 through 10 must precede the data, and a card containing DEC in columns 1, 2, and 3 and the negative of the curve designation number in columns 6 through 10 must terminate the curve. The curve designation numbers are arbitrary. The actual curve is described by listing the coordinates of the points circled on the plot. Each independent variable must immediately precede the corresponding dependent variable, and the independent variables are listed in increasing order. The integers in column 5 specify the number of floating point values on that card. The zero on the third card is a flag indicating the end of the data. Either a four- or five-digit card sequence number, at the user's discretion, is placed in columns 68 through 72.

In the above example, if current time, M(1), lies between 0. and 600., interpolation for T(1) takes place along the straight line connecting these two points. If M(1) lies between 600. and 1400., interpolation takes place along the straight line connecting these two points. If M(1) is less than 0., or greater than 1400., the case is terminated.

As mentioned previously, it is also possible to have periodic functions. An example of a periodic curve is the external heat input to a shell node of an orbiting satellite. The fact that a curve is periodic does not alter the function specification; only the curve itself need be modified. Consider the example

$$Q(17) = LIN(M(1), 26)$$

where curve 26 is to be described by linear interpolation and is periodic. This curve is described to the program by the period (3 sec) followed by the curve itself as shown below.

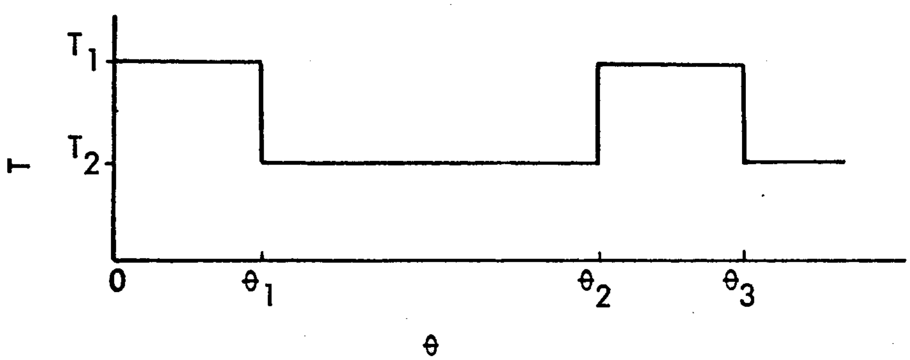


COLUMN

→ 1	5	10	15	25	35	45	55	65
DEC		26						
PER			3.					
DEC06			0.	8.	2.	6.	3.	8.
DEC01			0.					
DEC		-26						

The interpolation itself takes place exactly as in the linear interpolation described above, with the exception that the independent variable cannot lie outside the curve since the curve is understood to be indefinitely extended in both directions as indicated in the diagram.

When fitting a linear curve to a sharp discontinuity, such as a heating cycle,



the input specification for the curve should list the following coordinates:

$0.\ ,\ T_1$

$\theta_1\ ,\ T_1$

$\theta_1\ ,\ T_2$

$\theta_2\ ,\ T_2$

$\theta_2\ ,\ T_1$

$\theta_3\ ,\ T_1$

$\theta_3\ ,\ T_2$

etc.

If the program finds itself computing at a time at which one of the discontinuities occurs, it will recognize the former value of the dependent variable. For example, if a computing point should occur at θ_1 or θ_3, T_1 will be chosen as the proper temperature; if a computing point occurs at θ_2, T_2 will be chosen. This problem will occur only if a computing time point occurs exactly at a point of discontinuity.

Parabolic Interpolation

Where additional accuracy is desired, parabolic interpolation may be employed. However, it requires greater care in the selection of input points as discussed below. The callout is

Y = PAR(X,N)

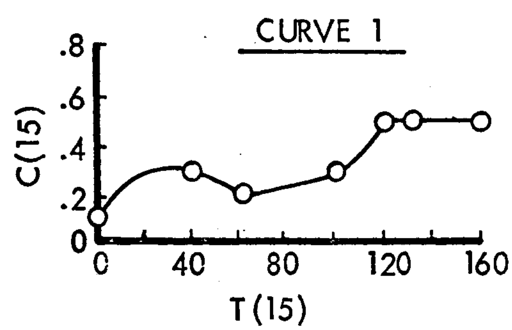
where

Y = parabolic function of X given by curve N

As an example, consider

C(15) = PAR (T(15),1)

The capacitance of node 15 is a function of the temperature of node 15 as given by curve 1. Curve 1 is to be described by parabolic interpolation. The curve and the data points input to the program are shown below:



COLUMN

1 5	10	15	25	35	45	55	65
DEC	1						
DEC06		0.	0.	40.	.3	60.	.2
DEC06		100.	.3	120.	.5	125.	.5
DEC03		160.	.5	0.			
DEC	-1						

If the independent variable, T(15), lies between 0. and 60., interpolation takes place by using that interpolating parabola which passes through the first 3 points and whose axis is parallel to the Y-axis. If T(15) lies between 60. and 120., the interpolating parabola is that parabola passing through the 3rd, 4th, and 5th points and parallel to the Y-axis. If T(15) lies between 120. and 160., the interpolating parabola is that parabola passing through the 5th, 6th, and 7th points and parallel to the Y-axis. If T(15) is off the curve (less than 0. or greater than 160.) the case is terminated.

These consecutive interpolating parabolas having only one point in common make it possible to describe discontinuous curves as well as curves which are made up of parabolic segments and linear segments (as curve 1). However, this manner of choosing parabolas always requires that an odd number of points be used in the curve description.

Since the parabolic curve-interpolation subroutine fits a curve to each successive series of three points, the choice of curve points is very critical. It must be emphasized that any sharp discontinuity should provide the end of one and the beginning of another set of three points. For example, Figure 4-1 shows the different interpretations of the same curve which would be given by different divisions. Note the extreme example of faulty interpolation which can occur in the case of a very steep curve.

When fitting a parabolic curve to a sharp discontinuity, such as the heating cycle described under Linear Interpretation in this section,

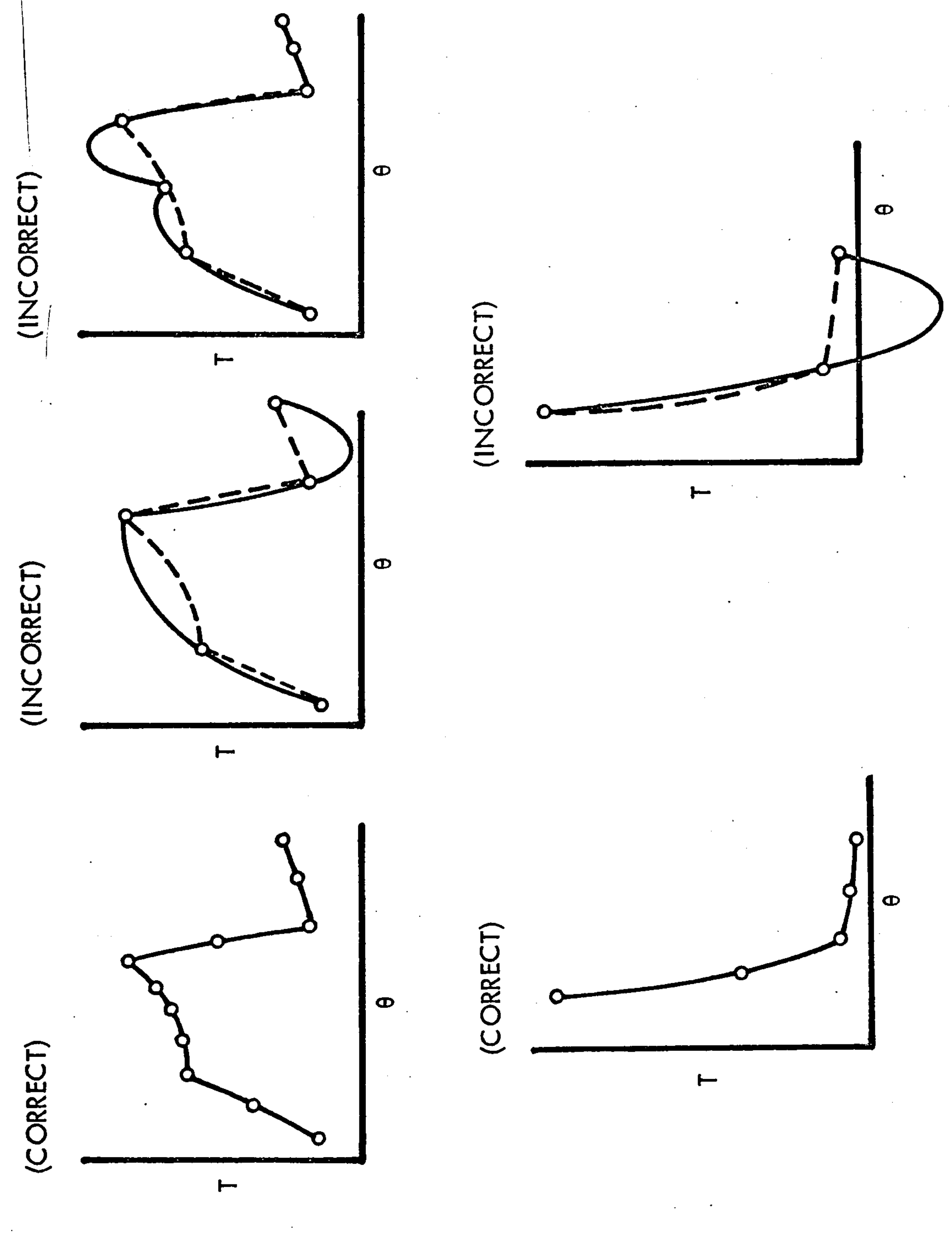
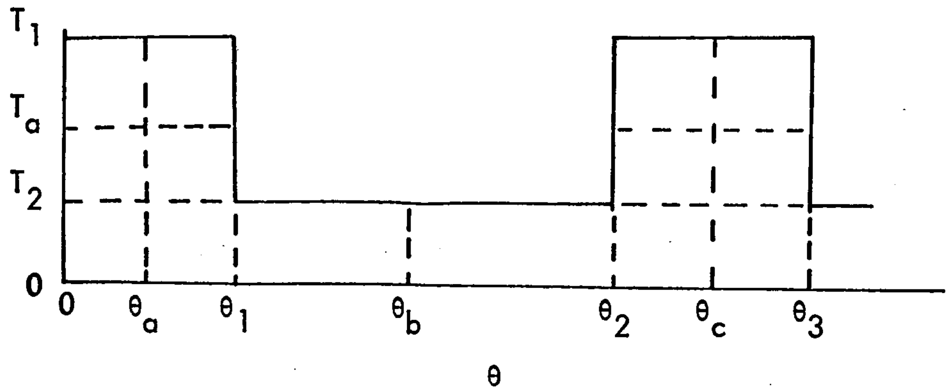


Figure 4-1. Correct and Incorrect Inputs for Curves Described by Parabolic Interpolation



$T_2 < T_a < T_1$

$0 < \theta_a < \theta_1 < \theta_b < \theta_2$, etc.

the input specification for the curve should contain the following coordinates:

0. ,	T_1
θ_a ,	T_1
θ_1 ,	T_1
θ_1 ,	T_a
θ_1 ,	T_2
θ_b ,	T_2
θ_2 ,	T_2
θ_2 ,	T_a
θ_2 ,	T_1
θ_c ,	T_1
θ_3 ,	T_1

etc.

If the program should find itself computing at a time which coincides with one of the discontinuities, it will recognize the previous value of the dependent variable. For example, for the above cyclic temperature, if the computing time point should occur at θ_1 or θ_3, T_1 will be chosen as the proper temperature; if the program should be computing at time θ_2, T_2 will be chosen.

Bivariate Interpolation

Frequently, there are situations where a variable is a function of more than one independent variable. Examples of this are:

(a) Internal losses in electronic components, which are commonly functions of current consumption and temperature.

(b) Thermodynamic and transport property data, which are usually temperature and pressure dependent.

The Thermal Analyzer Program is equipped to handle such problems through use of the linear bivariate interpolation routine. The callout is

Y = BIV (X1, X2, N)

where

Y is a bivariate function of X1 and X2 given by Table N.

In setting up a table to be described by bivariate interpolation, the first word on the first line following the table designation number must be uuuvvv. where uuu is a three-digit number equal to the number of rows in the table and vvv is a three-digit number equal to the number of columns in the table. Do not count the row and column that uuuvvv appears in. Following the first word of the first line the values of the independent variable X2 are listed. For each of these values there is an entry below it corresponding to the value of the independent variable X1 appearing at the left. In other words, the data appear on the input sheet just as one might tabulate it on a sheet of paper, with the addition of the number uuuvvv. The table need not be terminated with a flag indicating the end of the data since the quantity of data is specified by the number uuuvvv.

As an example of the use of bivariate interpolation, consider the problem of the internal heat dissipation of a battery, where the losses are a function of both the battery temperature and the current consumption. In this example, the battery will be designated as node 3. It will be assumed that the current is a known function of time and is to be linearly interpolated from curve 1. The battery losses in watts are tabulated below:

LOSSES (WATTS) VS. LOAD (AMPS)

Current ~ amps → ↓ Temperature ~ °F	0.	1.0	2.0	4.0	6.0
10.	0.	1.5	3.3	8.1	14.5
40.	0.	1.3	2.8	7.0	12.6
70.	0.	1.1	2.4	5.9	10.7
100.	0.	0.9	1.9	4.8	8.8
130.	0.	0.7	1.5	3.7	6.9

The bivariate curve will be designated as Table 8 and is described in the data block as follows:

COLUMN

```
→    1   5        10   15         25         35         45         55         65
     DEC           8
     DEC06     005005.             0.        1.0        2.0        4.0        6.0
     DEC06          10.            0.        1.5        3.3        8.1       14.5
     DEC06          40.            0.        1.3        2.8        7.0       12.6
     DEC06          70.            0.        1.1        2.4        5.9       10.7
     DEC06         100.            0.        0.9        1.9        4.8        8.8
     DEC06         130.            0.        0.7        1.5        3.7        6.9
     DEC          -8
```

The function specification for this example is

Q(3) = BIV(T(3),LIN(M(1),1),8) * .000946 where the constant is the conversion factor from watts to Btu/sec. In the function callout the independent variables, current and temperature, must appear in the order listed because Table 8 is set-up with temperature values listed in the first column. Otherwise, the program would attempt to horizontally interpolate using T(3) as independent variable and would run off the curve since the values listed range only from 0 to 6.

If there are more than five values of the independent variable X2, the data cannot be listed on input sheets in the tabular arrangement shown above because each card is restricted to a maximum of six floating point entries. In that event, the elements of the table are listed on input sheets in consecutive order, reading from left to right on each row and down each column of the table. To illustrate, consider the addition of a sixth and seventh column to the tabulation of "Losses vs. Load" as shown below:

Current ~ amps → / ↓ Temperature ~ °F	0.	1.0	2.0	4.0	6.0	8.0	10.0
10.	0.	1.5	3.3	8.1	14.5	21.8	30.3
40.	0.	1.3	2.8	7.0	12.6	19.7	28.1
70.	0.	1.1	2.4	5.9	10.7	16.5	25.0
100.	0.	0.9	1.9	4.8	8.8	14.2	22.2
130.	0.	0.7	1.5	3.7	6.9	12.1	20.0

Table 8 might appear on the input sheet as follows:

COLUMN

→ 1 5	10 15	25	35	45	55	65
DEC	8					
DEC06	005007.	0.	1.0	2.0	4.0	6.0
DEC02	8.0	10.0				
DEC06	10.0	0.	1.5	3.3	8.1	14.5
DEC02	21.8	30.3				
DEC06	40.0	0.	1.3	2.8	7.0	12.6

COLUMN →

1 5	10	15	25	35	45	55	65
DEC02		19.7	28.1				
DEC06		70.0	0.	1.1	2.4	5.9	10.7
DEC02		16.5	25.0				
DEC06		100.	0.	0.9	1.9	4.8	8.8
DEC02		14.2	22.2				
DEC06		130.	0.	0.7	1.5	3.7	6.9
DEC02		12.1	20.0				
DEC	-8						

Although six quantities may be listed on each card, the method shown above is preferred since the input is easier to read. The only requirement is that the table elements be listed in proper order.

RADIATION FUNCTIONS

As mentioned in Section III, "Calculation of Circuit Parameters," the machine is equipped to compute radiation resistors by the formula

$$R = \frac{1}{\sigma K_{rad} \left[(T_1 + 460)^2 + (T_2 + 460)^2 \right] \left[(T_1 + 460) + (T_2 + 460) \right]} \qquad (4\text{-}1)$$

where the user computes and inputs to the program in the FUNCT subroutine the value

$$K_{rad} = \frac{\epsilon_{12} A_1 F_{12}}{3600} \qquad \text{for } R \sim \frac{\text{sec } ^{\circ}\text{F}}{\text{Btu}} \qquad (4\text{-}2)$$

Three radiation resistor functions are available as discussed in the following sections.

Radiation With Constant Factor

This is the simplest and most commonly used radiation function. It is used in those situations where a fixed K_{rad} is applicable. The basic callout is:

$$R(A) = RAD\ (B,\ C,\ K_{rad})$$

where A is the designation number of the resistor connecting nodes B and C.

As an example, consider the case of a one square foot plate with a surface emissivity of 0.8 radiating to outer space (shape factor F = 1.0). For this case,

$$K_{rad} = \frac{(0.8)(1.0)(1.0)}{3600} = 2.22 \times 10^{-4}$$

If the resistor designation number is 106 and the plate and space are nodes 7 and 100, the function callout is

R(106) = RAD(7, 100, 2.22E - 4)

The order in which the two nodes are specified is immaterial since the program merely uses this information to solve for the resistor value using equation 4-1.

<u>Radiation With Variable Factor</u>

This is the same as the previous function except the value of K_{rad} is a variable. Examples of this are cases where the system geometry varies with time, or, more commonly, where the surface emissivity is temperature dependent. The callout is:

R(A) = RAD(B, C, X)

where the value of K_{rad} is stored in X, which may be either an addressable circuit element or a curve interpolation callout.

Two examples of the radiation with variable factor instruction are:

R(106) = RAD(7, 100, R(200))

R(106) = RAD(7, 100, (LIN(T(7), 13)))

In the first example, the value of K_{rad} has been temporarily stored in an unused resistor location. This procedure is often used when the value of K_{rad} is to be changed through a restart, as described in Section V. In the second example, the value of K_{rad} is linearly interpolated from curve 13 using the temperature of node 7 as independent variable. This instruction is convenient when the surface emissivity is temperature dependent.

Radiation With Matrix

In using the radiation functions described above, it is necessary to know K_{rad} in advance, and this in turn requires a knowledge of the geometric view factor F_{12} and the emissivity factor. Calculation of shape factors is beyond the scope of the program and is not covered here (see, however, references 1 and 2).

In most cases, the emissivity factor is computed by one of the following formulas:

infinite parallel plate

$$\epsilon_{12} = \frac{1}{\frac{1}{\epsilon_1} + \frac{1}{\epsilon_2} - 1}$$

concentric cylinder or sphere with $A_2 > A_1$

$$\epsilon_{12} = \frac{1}{\frac{1}{\epsilon_1} + \frac{A_1}{A_2}\left(\frac{1}{\epsilon_2} - 1\right)}$$

two surfaces whose size is small compared with their distance apart

$$\epsilon_{12} = \epsilon_1 \epsilon_2$$

surface A_1 much smaller than, and completely enclosed by, surface A_2

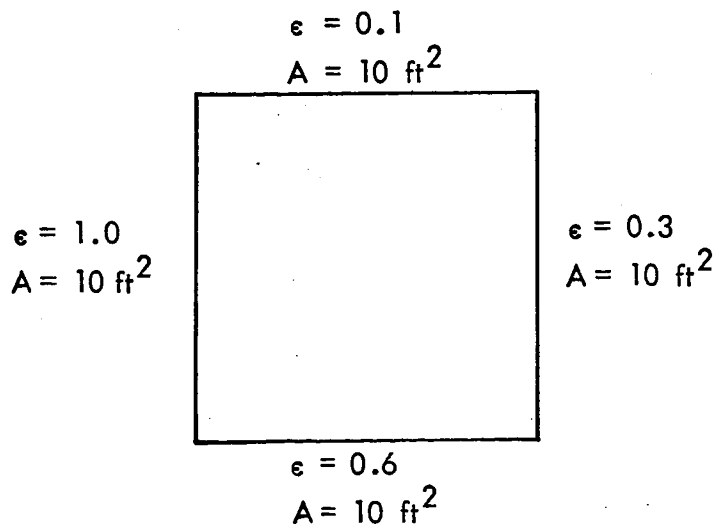
$$\epsilon_{12} = \epsilon_1$$

However, for those problems where several nonparallel surfaces are involved and the above formulas are not considered satisfactory, the program is equipped to compute the overall shape factor $\mathcal{F}_{12}$ in contrast to the geometrical shape factor F_{12}. The method used is discussed in detail in several heat transfer texts, such as McAdams (Reference 3), and will not be covered here. The method basically is to set up the radiation energy interchange equations and solve them through use of determinants. It is to be noted that the machine performs the calculation of K_{rad} only once, and hence the various quantities such as emissivity must all be assumed to be constants.

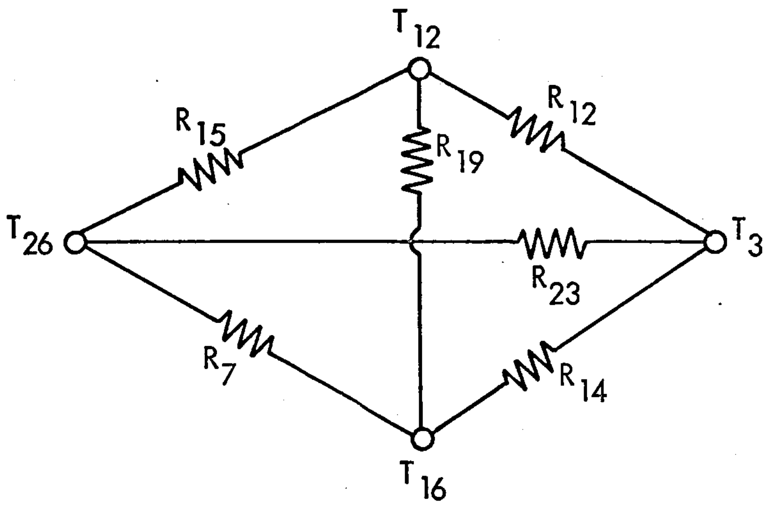
The Radiation with Matrix callout is

R(A) = RRM(B, C, N)

where A is the designation number of the resistor connecting nodes B and C, and N is the designation number of the table containing the data used to compute K_{rad}. Since this routine is used where several surfaces are involved, table N will appear in several resistor callouts. Also, it is possible to have more than one such system in a given network, and to have the same surface appear in more than one matrix. Consider the case of four walls arranged in a square pattern:



The shape factors have been computed to be 0.4 between opposite walls, and 0.3 between adjacent walls. The corresponding circuit is shown below. The node and resistor number assignments are completely arbitrary.



The RRM callouts for the above network are:

R(15) = RRM(12, 26, 6)
R(19) = RRM(12, 16, 6)
R(12) = RRM(12, 3, 6)
R(23) = RRM(26, 3, 6)
R(7) = RRM(16, 26, 6)
R(14) = RRM(16, 3, 6)

Table 6 contains the following information:

- the order of matrix of configuration factors, which in this example is 4.
- the matrix of configuration factors (row order)
- each node number and the area, emissivity, and reflectivity of each of these nodes.

For a 4th order matrix, the input would be as shown below. The subscripts refer to the order in which the nodes are listed, and are not necessarily the node or resistor designation numbers.

$(A_1F_{11} - A_1/\rho_1)$	A_1F_{12}	A_1F_{13}	A_1F_{14}
A_2F_{21}	$(A_2F_{22} - A_2/\rho_2)$	A_2F_{23}	A_2F_{24}
A_3F_{31}	A_3F_{32}	$(A_3F_{33} - A_3/\rho_3)$	A_3F_{34}
A_4F_{41}	A_4F_{42}	A_4F_{43}	$(A_4F_{44} - A_4/\rho_4)$

$N_1, A_1, \epsilon_1, \rho_1$
$N_2, A_2, \epsilon_2, \rho_2$
$N_3, A_3, \epsilon_3, \rho_3$
$N_4, A_4, \epsilon_4, \rho_4$

} the order in which the nodes are listed must correspond to the order of matrix rows.

where N_i = node number i
A_i = area of node i
ϵ_i = emissivity of node i
ρ_i = reflectivity of node i, usually equal to $(1 - \epsilon_i)$

The input for this particular example is shown below. (The reflectivity of node 26 has been taken as 0.01 to avoid division by zero).

COLUMN → 1 5 10 15 25 35 45

DEC	6		TABLE DESIGNATION NUMBER		
DEC01		4.	ORDER OF MATRIX		
DEC04		-11.1	3.0	4.0	3.0
DEC04		3.0	-14.3	3.0	4.0
DEC04		4.0	3.0	-25.0	3.0
DEC04		3.0	4.0	3.0	-1000.

COLUMN

1 5	10	15	25	35	45
DEC04		12.	10.	.10	.90
DEC04		3.	10.	.30	.70
DEC04		16.	10.	.60	.40
DEC04		26.	10.	.99	.01
DEC	-6				

Note that, with the exception of the curve designation numbers, all values in the table are input as floating point numbers.

The machine computes the radiation K factor by the formula

$$(K_{rad})_{ij} = \frac{1}{3600} \frac{\epsilon_i \epsilon_j}{\rho_i \rho_j} A_i A_j \left| \frac{D_{ij}}{D} \right|$$

where:

D is the determinant of the matrix

Di,j is the minor of the element (i,j), i.e., the determinant of the matrix formed by removing the i-th row and the j-th column.

The absolute value sign around the ratio D_{ij}/D comes about because as the equations were written, the machine could compute a negative K-factor, whereas the direct solution of the simultaneous equations involved results in positive K-factors.

CONVERGENCE AND STEADY-STATE SUBROUTINES

Since most problems to be solved by the Thermal Analyzer involve transient analysis, the initial values chosen for resistance, capacity, and temperature assume some importance. If it is desired to begin a transient problem at some steady-state condition and then apply variable boundary conditions (as would often occur in aerodynamic heating problems), it is very important that the initial conditions approximate the true steady-state conditions as nearly as possible before the transient conditions are applied. In many cases, the starting temperatures are uniform throughout the network and thus present no problem. However, many problems, such as the transient heating of a reentry vehicle, involve nonuniform starting temperatures. To handle this problem, the converge (CVG) function is used, the callout being

CALL CVG(A, N, M)

where

$|T_{\theta+\Delta\theta} - T_{\theta}| \leq$ A = degree of convergence (defined below).

N = maximum number of iterations to be executed in the convergence attempt.

M = number of iterations between prints of the output block.

When this function is encountered in the FUNCT routine, the heat balance and list of functions are evaluated repeatedly with time held constant (although with a finite computing interval) until the temperatures reach the required convergence, i.e., until the largest temperature difference of any node on consecutive passes is less than A (usually input as 0.005). When the temperatures have converged, the convergence function is removed from the function list, the regular output block is printed with "time" set to the number of iterations required for convergence and with "$\Delta\theta$" set to the designation number of the last node to converge. The case is then executed in the normal matter. If the temperatures fail to converge after N iterations, the case is discontinued, with a print of the regular output block with "time" set to N.

When using the CVG function, it should be remembered that the degree of convergence, A, is not necessarily the maximum deviation from the true converged temperatures, but merely the maximum temperature change of any node between two successive cycles. Consequently, it is quite possible that the deviation from the converged temperature could be several orders of magnitude greater than A and could amount to several degrees. For this reason, a small value of A, such as 0.005, should be used.

As input above, the CVG uses the capacitor values as input. To speed the convergence process, an optional form of the input can be used, namely

CALL CVG(A, N, - M)

where the negative sign on M is a flag which causes the convergence iteration to be executed with all capacitors set to zero. However, some care must be exercised when using zero capacitors in those cases which involve radiation resistors. Because radiation resistors vary inversely as the cube of the corresponding temperatures, they are very sensitive to small temperature changes. If the converged temperatures are very much different from the initial guesses, oscillations could be set up which will prevent convergence. In most cases, however, a reasonable set of initial temperatures should converge even if the zero capacitor option is used in cases involving radiation.

When the CVG function is used to solve purely steady-state cases, it is necessary to specify some non-zero final time in the time block to guarantee that the CVG function is executed properly. The value used should be of the order of the minimum RC product of the network to guarantee that a few "normal" cycles, i.e., with progressive time, are computed.

In addition to the CVG routine, a similar function (STS) is available which operates on the heat balance only and does not execute the function list during each iteration. Therefore, if there are radiation resistors or heat inputs called out in the functions, it is generally preferable to use the CVG subroutine. The steady-state function callout

CALL STS (A)

assigns zeros to all capacitors, and causes iteration through the circuit to proceed until the largest temperature difference of any node on consecutive passes is less than A. The original values of the capacitors are then restored, a flag is set to ignore the steady-state function in ensuing passes, and the case continued.

MAXIMUM-MINIMUM SUBROUTINE

In addition to the temperature-time history of the various nodes, it is often desired to know also the maximum and minimum temperatures of these nodes, and to be able to obtain them without having to search through the normal output data. This is especially true in large programs where several hundred temperatures may be involved.

This problem is handled by use of the maximum-minimum function (MMF), the callout being

$$\text{CALL MMF}(\theta_i, \theta_f, N, S)$$

where θ_i and θ_f are explicit (floating point) numbers, or addressable elements. During the time interval $\theta_i < \theta < \theta_f$, the maximum-minimum function records all local maximum and minimum temperatures of all nodes whose designation numbers are listed in data table N. The value S indicates the frequency with which the program is to search the output to determine the maximum and minimum temperatures, e.g., a 1 requires a search every cycle, a 2 requires a search every second cycle, etc. The node designation numbers appear in table N as floating point numbers. This table is terminated by a zero flag. To illustrate the use of the MMF function, assume that over the time interval 0-100 sec, the program is to conduct a search every other computing cycle to determine the maximum and minimum temperatures of nodes 7, 15, 83, and 94, which will be listed in Table 3. The table appears as follows:

```
DEC     3
DEC05          7.     15.     83.     94.     0.
DEC    -3
```

The instruction in the FUNCT subroutine is

```
CALL MMF(0., 100., 3, 2)
```

Generally, the search is required for the entire time span of the case and the following instruction would be used:

```
CALL MMF (M(2), M(3), 3, 2)
```

M(2) and M(3) are the initial and final times for the case, respectively.

The maximum and minimum temperatures are automatically printed following the regular time history print. This maximum-minimum output contains the range of the function (since there can be more than one MMF in a case), followed by a table with five non-zero columns. The first column is the designation number of the nodes (in the same order as given in the input table N), the second column is the list of the local maximums, the third column is the list of local minimums, the fourth column contains the times of the local maximums, and the fifth column contains the times of the local minimums. The MMF subroutine is illustrated in Example #2 of Section VI.

Only distinct local maximums and minimums are recorded. Therefore, if the rate of temperature change with time for a node does not change sign during the range of the MMF function, no maximum or minimum is recorded for that node. On the other hand, if the temperature of a node, after increasing for a while, remains constant for a time, and then decreases, a maximum would be recorded at the last time point for which the temperature was constant.

As a result of the manner in which the computer handles MMF data, there is a limitation on the amount of such data which can be obtained from a single case. Specifically, there must be less than 6000 maximums and minimums (representing 3000 cycles) in order for all of them to be printed. (A case with 200 nodes undergoing 20 temperature cycles would total 2x200x20=8000 and this exceeds this limit). If the total exceeds 6000, only the first (chronologically speaking) 6000 will be printed and all subsequent max-min data will be lost. This limitation can be overcome by dividing the case into two or more segments, with the additional cases run as restarts using the SAV function described below.

SAVE CURRENT DATA SUBROUTINE

This function makes it possible to use the output of one case as input to another in conjunction with the restart feature. An example of its use would be a full-life analysis of a recoverable satellite, with separate cases being run for ascent, orbit and recovery.

Ordinarily a restart case uses as its basis the data which existed at the start of the previous case. The Save Current Data (SAV) function provides the means whereby the answers of one case may be used as input into the next. The callout is

CALL SAV (A, B)

At the first time point for which A is greater than or equal to B, the data currently in core, i.e., the current values of temperature, resistance, and capacitance, are stored on tape to be used in the subsequent case. In nearly all cases A will be current time, M(1). The value of B may appear explicitly (floating point) or it may be an addressable element, e.g., final time, M(3). As an example,

CALL SAV (M(1), M(3))

will save the data in core at the end of each case, to be used as the initial conditions for the subsequent case.

TEMPERATURE CARD PUNCH SUBROUTINE

This subroutine is frequently used in complex cases as an alternate restart feature. It provides for the automatic recording of all temperatures on punched cards in proper format for the initial temperature block. The function specification is

CALL PUNCHT(θ, A, B)

where θ is the time at which the temperatures are to be saved, and cards are required for nodes A through B. θ may be an explicit (floating point) number or an addressable element. A and B are listed in fixed-point notation. If the function is specified as CALL PUNCHT(θ, 0, 0) the program will interpret the zeros as a special code requiring the punching of temperature cards for all nodes.

The advantage of this subroutine is that once the temperatures are recorded, the case may be removed from the computer and resumed later when machine time becomes available. Also, in complex cases requiring several hours of computer time, it is customary to call for punched cards at various times throughout the run. Then, if an error occurs, the case may be re-run starting at the last time for which valid temperature cards were obtained.

To help identify the punched cards, the information in columns 6 through 14 of the case identification card (Section V) and the time θ are automatically punched on each card. Thus, if this subroutine is used, it is desirable that the information on the case identification card begin with some code, for example, CASE B-2.

CURVE PLOTTING SUBROUTINE

In many problems it is desirable to obtain the results in curve form in addition to the tabulated output. To accomplish this, the Thermal Analyzer Program has a subroutine to write a plot tape which controls a machine plotter. The function specification for the plotting routine is

CALL TPLOT(N)

where N is the designation number of the table containing the information to write the tape. Table N must contain the following items, in the order listed:

(a) the initial time of the plot.

(b) the final time of the plot.

(c) the time interval between successive plots (discussed below).

(d) all nodes whose temperatures are to be plotted.

(e) a zero flag terminating the table.

All of these entries must be floating point. The plotting routine does not automatically plot the temperatures at each computing cycle, the frequency of the plot being specified by the user. The program subtracts the final time from the initial, and divides that quantity by the specified frequency (item C above) to determine how often the temperatures are to be plotted. If these times do not coincide with computing times, the machine will plot the temperatures for each subsequent computing time. For example, if the initial and final times are 0 and 100 sec, the time interval between plots specified at 10 sec, and the computing interval set at 3 sec, the temperatures would be plotted at 0 sec, 12 sec, 21 sec, 30 seconds, etc. If the user desires a plot every computing cycle, the third storage location of table N must be set equal to the computing interval M(5) by the following function specifications:

```
LN = LOC(N)
P(LN + 3) = M(5)
CALL TPLOT(N)
```

Although this routine will only plot temperatures, in practice any addressable quantity can be plotted simply by storing that quantity in an unused temperature location. For example, to plot R(4), one could set TN(200) = R(4) and plot T(200). Note that at the end of the heat balance, T(200) is set equal to TN(200), so it is necessary to store R(4) in the "new" temperature location.

This routine is currently set up to plot a maximum of 5 temperatures per curve. If, for example, 8 nodes are specified in table N, the first 5 will be plotted on one curve and the remaining 3 on a separate curve. A maximum of 20 nodes may be listed per table and a maximum of 20 tables may be used.

As an example of the input for the plotting routine, suppose that it is desired to plot the first and last 100 sec of a case that runs continuously from 0 to 500 sec. The temperatures of nodes 24, 68, 72, and 93 are to be plotted at 10-sec intervals. For this case two tables could be set up as follows:

COLUMN →

1 5	10	15	25	35	45	55	65
DEC	20						
DEC06		0.	100.	10.	24.	68.	72.
DEC02		93	0.				
DEC	-20						
DEC	21						
DEC06		400.	500.	10.	24.	68.	72.
DEC02		93.	0.				
DEC	-21						

The function specifications are

CALL TPLOT (20)
CALL TPLOT (21)

In Section VI, plots have been obtained for 5 nodes in Example Problem #2 to illustrate the use of the function and the quality of the curves. Note that the information on the case identification card (see Section V) is automatically printed at the top of each curve.

AERODYNAMIC HEATING AND ABLATION SUBROUTINES

The Eckert Aerodynamic Heating subroutine computes the surface convective heat transfer by the reference temperature method. The primary advantage of this method is that it allows incompressible skin friction relations to be employed in evaluating high-velocity-flow heat transfer. This is accomplished by evaluating the air properties at a suitably defined reference temperature, which depends only on the wall, local inviscid flow, and recovery temperatures.

The Ablation subroutine computes the amount of material that is ablated (if the temperature is above the ablation temperature) as a function of time as part of the heat balance. This solution is applicable to one-dimensional heat transfer to a pure subliming (non-charring) surface.

These specialized subroutines are discussed in detail in Appendices A and B. Example problems are set up and written on data input sheets to demonstrate the use of each subroutine.

MISCELLANEOUS FORTRAN OPERATIONS

Although a knowledge of FORTRAN is not required to use the Thermal Analyzer Program, an understanding of a few basic FORTRAN operations is highly desirable. The following is a brief discussion of some of the most commonly used operations, along with some general information useful in writing the FUNCT subroutine. The topics include the use of fixed- and floating-point variables, statement numbers, comment cards, and control statements. The treatment of each topic is brief, and the reader is referred to the FORTRAN reference manual for more detailed information.

Variables

A variable is a symbol or name which refers to a place in memory where the number or value represented by that name is stored. The name of any variable consists of one or more alphabetic or numerical characters, the first of which must be alphabetic (A through Z). The maximum number of characters permitted in the name is 6 for the 7094 processor. Names beginning with the letters I, J, K, L, M, or N are variable names for fixed-point numbers. All other names represent floating-point values. For example, K15 and NEXT are valid integer variables and A50 and VALUE are valid floating-point variables. In general, fixed- and floating-point variables cannot be mixed. As examples:

B = A * I	is not permitted since B and A are floating-point numbers and I is a fixed-point number.
B = A + 2	is not permitted since B and A are floating-point numbers and 2 (without decimal) is fixed-point.
B = A + 2.	is permitted since A, B, and 2. (with decimal) are all fixed-point numbers.

Statement Numbers

It is frequently desirable to identify a statement by name or number. This is accomplished by a numerical statement number (any integer from 1 to 32767) placed in columns 1 through 5 of the data input sheet.

The only other restriction is that a statement number must not be repeated in any subroutine. The use of statement numbers will become more apparent in the following sections.

Comment Cards

Occasionally the user may wish to have comments printed out in the FUNCT and PRINT subroutines. This is accomplished by placing the letter C in column 1 of the data input sheet and the desired comments in columns 7 through 72. The comment card is then ignored by the FORTRAN processor, and serves merely to furnish commentary information in the program listing.

GO TO Statements

Unconditional transfer statements which begin with the words GO TO permit the user to alter arbitrarily the sequence in which the program statements are to be executed. A frequent use of the GO TO statement is to skip an inapplicable instruction as shown by the first example in the following section.

To illustrate an unconditional transfer statement, the instruction

```
GO TO 17
```

means that the next statement to be executed is the one labeled 17. Statement 17 may either precede or follow the GO TO statement. All statements in between will be skipped and, once statement 17 is executed, the computer will execute those which follow, in the order in which they appear.

Computed GO TO statements are used as a switch for branching to one of several places in the program, depending on the integer value of a test location. To illustrate, the statement

```
GO TO (7, 4, 28, 9), K
```

transfers control to the statement labeled 7, 4, 28, or 9 when the current value of K is 1, 2, 3, or 4, respectively.

Conditional Control and Logical Expressions

Frequently it is desirable to execute a particular function, or set of functions, only under certain specified conditions. This is accomplished by the IF statement which directs control to one of three different statements, depending on the value of an arithmetic expression. The statement

```
IF(X) A, B, C
```

will direct the computer to immediately execute statement A if the value of (X) is negative, to execute statement B if (X) is zero, and to execute statement C if (X) is positive. Any statements between the IF statement and the next executed statement (A, B, or C) will be skipped. The expression (X) may be any legal FORTRAN arithmetic expression discussed in the preceding sections.

To illustrate the use of the conditional control statement, consider a lunar spacecraft which is slowly rotating about one axis. The solar heating rate to shell node 10 is to be linearly interpolated from curve 10. At 45,000 sec, however, the vehicle stops rolling and the heating rate to node 10 is a constant value, say 0.085 Btu/sec. The appropriate functions might appear as follows:

```
     IF (M(1) - 45000.) 2, 3, 3
  2  Q(10) = LIN (M(1), 10)
     GO TO 4
  3  Q(10) = 0.085
```

If current time is less than 45,000 sec, statement 2 is executed. Statement 3 is then skipped by the insertion of a GO TO statement which directs control to statement 4. The latter is not shown but would presumably follow statement 3. If current time is equal to or greater than 45,000 sec, the IF statement directs control to statement 3, after which statement 4 and succeeding functions are executed in a normal manner.

A logical IF statement has been added to the FORTRAN IV language so that decisions can be based directly on the true or false value of a quantity which is logical rather than arithmetic in nature. The statement

IF(Y)F

means that if the logical expression (Y) is true, execute the statement F and then proceed to the statement following F; otherwise (if Y is false), do not execute F, but simply proceed to the following statement. F may be any single executable statement except another logical IF statement or a DO statement.

Comparisons can be made by means of the logical operators .OR., .AND., or .NOT., or by any of the following relational operators:

Symbol	Meaning
.EQ.	Equal to
.NE.	Not equal to
.LT.	Less than
.LE.	Less than or equal to
.GT.	Greater than
.GE.	Greater than or equal to

Two examples of the use of logical IF statements are given below:

```
IF (M(6).EQ.80..OR.T(26).GE.500.) GO TO 5
```

means that if the minimum RC product equals 80 or if the temperature of node 26 is greater than or equal to 500, proceed immediately to statement 5.

```
IF (M(1).GE.35..AND.M(1).LE.70.) T(30) = 1100.
```

means that if current time is between 35 and 70, set the temperature of node 30 equal to 1100.

With the logical IF statement it is possible to simplify the functions required for the lunar spacecraft example given above. To illustrate, the two statements

```
Q(10) = 0.085
IF (M(1).LT. 45000.)  Q(10) = LIN (M(1),10)
```

are equivalent to the four statements listed previously. The second function merely overrides the first if current time is less than 45,000 sec.

DO Statements

Most computer programs involve a group of steps which are to be executed in a repetitive fashion. The iteration or DO statement greatly facilitates the definition and control of these repetitive steps. The statement

```
DO A I = B, C, N
```

controls the repetition of all succeeding statements down through and including the statement labeled A. Repetition of these statements is controlled by varying the index called I, from an initial value of B to a terminal value of C in increments of N. If the integer N does not appear the increment is understood to be 1. For example, the statements

```
   DO 13 I = 1, 3
   Q(I) = 0.25
   R(I + 100) = RAD (I, 100, .3E - 5)
13 CONTINUE
```

will cause the following operations to be performed:

```
Q(1)   =  0.25
Q(2)   =  0.25
Q(3)   =  0.25
R(101) =  RAD(1, 100, 0.3E - 5)
R(102) =  RAD(2, 100, 0.3E - 5)
R(103) =  RAD(3, 100, 0.3E - 5)
```

The terminating statement need not always be a CONTINUE card as shown in the above example. However, the last statement in an iteration loop cannot be a transfer statement, conditional or unconditional, or another iteration statement. The termination of a DO loop with a CONTINUE card is always acceptable and is recommended as a means to avoid the above difficulties. Another restriction on the use of DO loops is that control must never be transferred from outside to any card within the iteration loop with the exception of the first, or DO statement, card.

THERMAL PROPERTIES LIBRARY

The thermal properties library contains tables of physical properties of a number of propellants, pressurants, simulated propellants, structural materials, and insulations. Special data search and interpolation routines are incorporated into the Thermal Analyzer Program to utilize the library data.

Library Tape

The thermal properties and library tape are discussed in detail and completely listed in Reference 2. The property tables contain data on density, specific heat, and thermal conductivity vs. temperature for liquids and solids; vapor pressure, viscosity, heat of vaporization, and surface tension vs. temperature for liquids; and specific heat, thermal conductivity, and compressibility factor vs. temperature and pressure for gases. The properties data are given in the following units:

Density, lb/ft^3

Specific heat, Btu/lb °F

Thermal conductivity, Btu/hr ft °F

Vapor pressure, psia

Viscosity, lb/ft sec

Heat of vaporization, Btu/lb

Surface tension, lb/ft

The library itself is contained on a reserved tape. Each curve is uniquely identified by a six-character code consisting of both a material and a property identification. Table 4-1 is a complete listing of the identification codes and titles of every library table. The use of the library tape is described below.

A flag in the data-block of the Thermal Analyzer will cause the compiler to search the data tape for the specific table called for by the flag. That table will then be stored in place of the flag. There must be an exact correspondence between the identification of the tables stored on tape and the flags used to call the tables to be used in the program. The flag consists of six alphameric characters in columns 6-11, preceded by the mnemonic code "TAP" in columns 1-3. (See Table 4-1.) The data entry and flag are as follows:

COLUMN →

1	10	
DEC	401	Table No.
TAP	NTO-13	I.D. of table stored on tape
DEC	-401	End-of-table flag

This entry means that the material properties data table identified as Table "NTO-13" (Thermal Conductivity of Liquid Nitrogen Tetroxide) is to be stored as Table 401 within the Thermal Analyzer input data block.

In all cases where data are available, the tables are accurate for linear interpolation within ±5% over the appropriate temperature range. To construct a useful data library, it was necessary to extend

TABLE 4-1

THERMAL PROPERTIES LIBRARY IDENTIFICATION CODES

TAP	NTO-11	DENSITY OF LIQUID NITROGEN TETROXIDE	101012266
TAP	NTO-12	SPECIFIC HEAT OF LIQUID NITROGEN TETROXIDE	102012266
TAP	NTO-13	THERMAL CONDUCTIVITY OF LIQUID NITROGEN TETROXIDE	103012266
TAP	NTO-14	VAPOR PRESSURE OF LIQUID NITROGEN TETROXIDE	104012266
TAP	NTO-15	VISCOSITY OF LIQUID NITROGEN TETROXIDE	105012266
TAP	NTO-16	HEAT OF VAPORIZATION OF LIQUID NITROGEN TETROXIDE	106012266
TAP	NTO-17	SURFACE TENSION OF LIQUID NITROGEN TETROXIDE	107012266
TAP	NTO-22	SPECIFIC HEAT OF GASEOUS NITROGEN TETROXIDE	108012266
TAP	NTO-23	THERMAL CONDUCTIVITY OF GASEOUS NITROGEN TETROXIDE	109012266
TAP	NTO-28	COMPRESSIBILITY FACTOR OF GASEOUS NITROGEN TETROXIDE	110012266
TAP	OXY-11	DENSITY OF LIQUID OXYGEN	111012266
TAP	OXY-12	SPECIFIC HEAT OF LIQUID OXYGEN	112012266
TAP	OXY-13	THERMAL CONDUCTIVITY OF LIQUID OXYGEN	113012266
TAP	OXY-14	VAPOR PRESSURE OF LIQUID OXYGEN	114012266
TAP	OXY-15	VISCOSITY OF LIQUID OXYGEN	115012266
TAP	OXY-16	HEAT OF VAPORIZATION OF LIQUID OXYGEN	116012266
TAP	OXY-17	SURFACE TENSION OF LIQUID OXYGEN	117012266
TAP	OXY-22	SPECIFIC HEAT OF GASEOUS OXYGEN	118012266
TAP	OXY-23	THERMAL CONDUCTIVITY OF GASEOUS OXYGEN	119012266
TAP	OXY-28	COMPRESSIBILITY FACTOR OF GASEOUS OXYGEN	120012266
TAP	FLU-11	DENSITY OF LIQUID FLUORINE	121012266
TAP	FLU-12	SPECIFIC HEAT OF LIQUID FLUORINE	122012266
TAP	FLU-13	THERMAL CONDUCTIVITY OF LIQUID FLUORINE	123012266
TAP	FLU-14	VAPOR PRESSURE OF LIQUID FLUORINE	124012266
TAP	FLU-15	VISCOSITY OF LIQUID FLUORINE	125012266
TAP	FLU-16	HEAT OF VAPORIZATION OF LIQUID FLUORINE	126012266
TAP	FLU-17	SURFACE TENSION OF LIQUID FLUORINE	127012266
TAP	FLU-22	SPECIFIC HEAT OF GASEOUS FLUORINE	128012266
TAP	FLU-23	THERMAL CONDUCTIVITY OF GASEOUS FLUORINE	129012266
TAP	FLU-28	COMPRESSIBILITY FACTOR OF GASEOUS FLUORINE	130012266
TAP	ODF-11	DENSITY OF LIQUID OXYGEN DIFLUORIDE	131012266
TAP	ODF-12	SPECIFIC HEAT OF LIQUID OXYGEN DIFLUORIDE	132012266
TAP	ODF-13	THERMAL CONDUCTIVITY OF LIQUID OXYGEN DIFLUORIDE	133012266
TAP	ODF-14	VAPOR PRESSURE OF LIQUID OXYGEN DIFLUORIDE	134012266
TAP	ODF-15	VISCOSITY OF LIQUID OXYGEN DIFLUORIDE	135012266
TAP	ODF-16	HEAT OF VAPORIZATION OF LIQUID OXYGEN DIFLUORIDE	136012266
TAP	ODF-22	SPECIFIC HEAT OF GASEOUS OXYGEN DIFLUORIDE	138012266
TAP	ODF-28	COMPRESSIBILITY FACTOR OF LIQUID OXYGEN DIFLUORIDE	140012266
TAP	CTF-11	DENSITY OF LIQUID CHLORINE TRIFLUORIDE	141012266
TAP	CTF-12	SPECIFIC HEAT OF LIQUID CHLORINE TRIFLUORIDE	142012266
TAP	CTF-13	THERMAL CONDUCTIVITY OF LIQUID CHLORINE TRIFLUORIDE	143012266
TAP	CTF-14	VAPOR PRESSURE OF LIQUID CHLORINE TRIFLUORIDE	144012266
TAP	CTF-15	VISCOSITY OF LIQUID CHLORINE TRIFLUORIDE	145012266
TAP	CTF-16	HEAT OF VAPORIZATION OF LIQUID CHLORINE TRIFLUORIDE	146012266
TAP	CTF-17	SURFACE TENSION OF LIQUID CHLORINE TRIFLUORIDE	147012266
TAP	CTF-22	SPECIFIC HEAT OF GASEOUS CHLORINE TRIFLUORIDE	148012266
TAP	CTF-28	COMPRESSIBILITY FACTOR, GASEOUS CHLORINE TRIFLUORIDE	150012266
TAP	AER-11	DENSITY OF LIQUID AEROZINE 50	151012266
TAP	AER-12	SPECIFIC HEAT OF LIQUID AEROZINE 50	152012266
TAP	AER-13	THERMAL CONDUCTIVITY OF LIQUID AEROZINE 50	153012266
TAP	AER-14	VAPOR PRESSURE OF LIQUID AEROZINE 50	154012266
TAP	AER-15	VISCOSITY OF LIQUID AEROZINE 50	155012266
TAP	AER-16	HEAT OF VAPORIZATION OF LIQUID AEROZINE 50	156012266
TAP	AER-17	SURFACE TENSION OF LIQUID AEROZINE 50	157012266
TAP	AER-22	SPECIFIC HEAT OF GASEOUS AEROZINE 50	158012266
TAP	AER-23	THERMAL CONDUCTIVITY OF GASEOUS AEROZINE 50	159012266
TAP	AER-28	COMPRESSIBILITY FACTOR OF GASEOUS AEROZINE 50	160012266
TAP	MMH-11	DENSITY OF LIQUID MONOMETHYL HYDRAZINE	161012266
TAP	MMH-12	SPECIFIC HEAT OF LIQUID MONOMETHYL HYDRAZINE	162012266
TAP	MMH-13	THERMAL CONDUCTIVITY OF LIQUID MONOMETHYL HYDRAZINE	163012266
TAP	MMH-14	VAPOR PRESSURE OF LIQUID MONOMETHYL HYDRAZINE	164012266

TABLE 4-1

(Continued)

TAP	MMH-15	VISCOSITY OF LIQUID MONOMETHYL HYDRAZINE	165012266
TAP	MMH-16	HEAT OF VAPORIZATION OF LIQUID MONOMETHYL HYDRAZINE	166012266
TAP	MMH-17	SURFACE TENSION OF LIQUID MONOMETHYL HYDRAZINE	167012266
TAP	MMH-22	SPECIFIC HEAT OF GASEOUS MONOMETHYL HYDRAZINE	168012266
TAP	MMH-23	THERMAL CONDUCTIVITY , GASEOUS MONOMETHYL HYDRAZINE	169012266
TAP	MMH-28	COMPRESSIBILITY FACTOR,GASEOUS MONOMETHYL HYDRAZINE	170012266
TAP	DIB-11	DENSITY OF LIQUID DIBORANE	171012266
TAP	DIB-12	SPECIFIC HEAT OF LIQUID DIBORANE	172012266
TAP	DIB-13	THERMAL CONDUCTIVITY OF LIQUID DIBORANE	173012266
TAP	DIB-14	VAPOR PRESSURE OF LIQUID DIBORANE	174012266
TAP	DIB-15	VISCOSITY OF LIQUID DIBORANE	175012266
TAP	DIB-16	HEAT OF VAPORIZATION OF LIQUID DIBORANE	176012266
TAP	DIB-22	SPECIFIC HEAT OF GASEOUS DIBORANE	178012266
TAP	DIB-28	COMPRESSIBILITY FACTOR OF GASEOUS DIBORANE	180012266
TAP	HYD-11	DENSITY OF LIQUID HYDROGEN	181012266
TAP	HYD-12	SPECIFIC HEAT OF LIQUID HYDROGEN	182012266
TAP	HYD-13	THERMAL CONDUCTIVITY OF LIQUID HYDROGEN	183012266
TAP	HYD-14	VAPOR PRESSURE OF LIQUID HYDROGEN	184012266
TAP	HYD-15	VISCOSITY OF LIQUID HYDROGEN	185012266
TAP	HYD-16	HEAT OF VAPORIZATION OF LIQUID HYDROGEN	186012266
TAP	HYD-17	SURFACE TENSION OF LIQUID HYDROGEN	187012266
TAP	HYD-22	SPECIFIC HEAT OF GASEOUS NORMAL HYDROGEN	188012266
TAP	HYD-23	THERMAL CONDUCTIVITY OF GASEOUS PARA HYDROGEN	189012266
TAP	HYD-28	COMPRESSIBILITY FACTOR OF GASEOUS PARA HYDROGEN	190012266
TAP	HA5-11	DENSITY OF LIQUID HYBALINE A5	191012266
TAP	HA5-12	SPECIFIC HEAT OF LIQUID HYBALINE A5	192012266
TAP	HA5-13	THERMAL CONDUCTIVITY OF LIQUID HYBALINE A5	193012266
TAP	HA5-14	VAPOR PRESSURE OF LIQUID HYBALINE A5	194012266
TAP	HA5-15	VISCOSITY OF LIQUID HYBALINE A5	195012266
TAP	HA5-22	SPECIFIC HEAT OF GASEOUS HYBALINE A5	198012266
TAP	HA5-23	THERMAL CONDUCTIVITY OF GASEOUS HYBALINE A5	199012266
TAP	HA5-28	COMPRESSIBILITY FACTOR OF GASEOUS HYBALINE A5	200012266
TAP	NIT-11	DENSITY OF LIQUID NITROGEN	201012266
TAP	NIT-12	SPECIFIC HEAT OF LIQUID NITROGEN	202012266
TAP	NIT-13	THERMAL CONDUCTIVITY OF LIQUID NITROGEN	203012266
TAP	NIT-14	VAPOR PRESSURE OF LIQUID NITROGEN	204012266
TAP	NIT-15	VISCOSITY OF LIQUID NITROGEN	205012266
TAP	NIT-16	HEAT OF VAPORIZATION OF LIQUID NITROGEN	206012266
TAP	NIT-17	SURFACE TENSION OF LIQUID NITROGEN	207012266
TAP	NIT-22	SPECIFIC HEAT OF GASEOUS NITROGEN	208012266
TAP	NIT-23	THERMAL CONDUCTIVITY OF GASEOUS NITROGEN	209012266
TAP	NIT-28	COMPRESSIBILITY FACTOR OF GASEOUS NITROGEN	210012266
TAP	HEL-22	SPECIFIC HEAT OF GASEOUS HELIUM	218012266
TAP	HEL-23	THERMAL CONDUCTIVITY OF GASEOUS HELIUM	219012266
TAP	HEL-28	COMPRESSIBILITY FACTOR OF GASEOUS HELIUM	220012266
TAP	G60-11	DENSITY OF LIQUID .6 ETHYLENE GLYCOL	221012266
TAP	G60-12	SPECIFIC HEAT OF LIQUID .6 ETHYLENE GLYCOL	222012266
TAP	G60-13	THERMAL CONDUCTIVITY OF LIQUID .6 ETHYLENE GLYCOL	223012266
TAP	G60-14	VAPOR PRESSURE OF LIQUID .6 ETHYLENE GLYCOL	224012266
TAP	G60-15	VISCOSITY OF LIQUID .6 ETHYLENE GLYCOL	225012266
TAP	G60-22	SPECIFIC HEAT OF GASEOUS .6 ETHYLENE GLYCOL	228012266
TAP	G60-23	THERMAL CONDUCTIVITY OF GASEOUS .6 ETHYLENE GLYCOL	229012266
TAP	G60-28	COMPRESSIBILITY FACTOR , GASEOUS .6 ETHYLENE GLYCOL	230012266
TAP	F11-11	DENSITY OF LIQUID FREON 11	231012266
TAP	F11-12	SPECIFIC HEAT OF LIQUID FREON 11	232012266
TAP	F11-13	THERMAL CONDUCTIVITY OF LIQUID FREON 11	233012266
TAP	F11-14	VAPOR PRESSURE OF LIQUID FREON 11	234012266
TAP	F11-15	VISCOSITY OF LIQUID FREON 11	235012266
TAP	F11-16	HEAT OF VAPORIZATION OF LIQUID FREON 11	236012266
TAP	F11-17	SURFACE TENSION OF LIQUID FREON 11	237012266
TAP	F11-22	SPECIFIC HEAT OF GASEOUS FREON 11	238012266

TABLE 4-1

(Continued)

TAP	L92A-3	THERMAL CONDUCTIVITY OF LINDE SI-92 (80 LAYERS/INCH)	483012266
TAP	L92B-1	DENSITY OF LINDE SI-92 (100 LAYERS/INCH)	491012266
TAP	L92B-2	SPECIFIC HEAT OF LINDE SI-92	492012266
TAP	L92B-3	THERMAL CONDUCTIVITY OF LINDE SI-92 (100 LAYERS/INCH)	493012266
TAP	L92C-1	DENSITY OF LINDE SI-92 (120 LAYERS/INCH)	501012266
TAP	L92C-2	SPECIFIC HEAT OF LINDE SI-92	502012266
TAP	L92C-3	THERMAL CONDUCTIVITY OF LINDE SI-92 (120 LAYERS/INCH)	503012266
TAP	L92D-1	DENSITY OF LINDE SI-92 (160 LAYERS/INCH)	511012266
TAP	L92D-2	SPECIFIC HEAT OF LINDE SI-92	512012266
TAP	L92D-3	THERMAL CONDUCTIVITY OF LINDE SI-92 (160 LAYERS/INCH)	513012266
TAP	LFWA-1	DENSITY OF LINDE FT WT (40 LAYERS/INCH)	521012266
TAP	LFWA-2	SPECIFIC HEAT OF LINDE FLT WT	522012266
TAP	LFWA-3	THERMAL CONDUCTIVITY OF LINDE FLT WT (40 LAYERS/INCH)	523012266
TAP	LFWB-1	DENSITY OF LINDE FT WT (50 LAYERS(INCH)	531012266
TAP	LFWB-2	SPECIFIC HEAT OF LINDE FLT WT	532012266
TAP	LFWB-3	THERMAL CONDUCTIVITY OF LINDE FLT WT (50 LAYERS/INCH)	533012266
TAP	LFWC-1	DENSITY OF LINDE FT WT (60 LAYERS/INCH)	541012266
TAP	LFWC-2	SPECIFIC HEAT OF LINDE FLT WT	542012266
TAP	LFWC-3	THERMAL CONDUCTIVITY OF LINDE FLT WT (60 LAYERS/INCH)	543012266
TAP	LFWD-1	DENSITY OF LINDE FT WT (80 LAYERS/INCH)	551012266
TAP	LFWD-2	SPECIFIC HEAT OF LINDE FLT WT	552012266
TAP	LFWD-3	THERMAL CONDUCTIVITY OF LINDE FLT WT (80 LAYERS/INCH)	553012266
TAP	LHTA-1	DENSITY OF LINDE HIGH TEMP (60 LAYERS/INCH)	561012266
TAP	LHTA-2	SPECIFIC HEAT OF LINDE HIGH TEMP	562012266
TAP	LHTA-3	THERMAL CONDUCTIVITY OF LINDE HI TEMP(60 LAYERS/INCH)	563012266
TAP	LHTB-1	DENSITY OF LINDE HIGH TEMP (120 LAYERS/INCH)	571012266
TAP	LHTB-2	SPECIFIC HEAT OF LINDE HIGH TEMP	572012266
TAP	LHTB-3	THERMAL CONDUCTIVITY OF LINDE HI TEMP(120 LAYER/INCH)	573012266
TAP	LHTC-1	DENSITY OF LINDE HIGH TEMP (180 LAYERS/INCH)	581012266
TAP	LHTC-2	SPECIFIC HEAT OF LINDE HIGH TEMP	582012266
TAP	LHTC-3	THERMAL CONDUCTIVITY OF LINDE HI TEMP(180 LAYER/INCH)	583012266
TAP	NRCA-1	DENSITY OF NRC-2 (40 LAYERS/INCH)	601012266
TAP	NRCA-2	SPECIFIC HEAT OF NRC-2	602012266
TAP	NRCA-3	THERMAL CONDUCTIVITY OF NRC-2 (40 LAYERS/INCH)	603012266
TAP	NRCB-1	DENSITY OF NRC-2 (100 LAYERS/INCH)	611012266
TAP	NRCB-2	SPECIFIC HEAT OF NRC-2	612012266
TAP	NRCB-3	THERMAL CONDUCTIVITY OF NRC-2 (100 LAYERS/INCH)	613012266
TAP	NRCC-1	DENSITY OF NRC-2 (160 LAYERS/INCH)	621012266
TAP	NRCC-2	SPECIFIC HEAT OF NRC-2	622012266
TAP	NRCC-3	THERMAL CONDUCTIVITY OF NRC-2 (160 LAYERS/INCH)	623012266
TAP	FGLS-1	DENSITY OF FIBERGLASS	701012266
TAP	FGLS-2	SPECIFIC HEAT OF FIBERGLASS	702012266
TAP	FGLS-3	THERMAL CONDUCTIVITY OF FIBERGLASS	703012266
TAP	MQTZ-1	DENSITY OF MICRO QUARTZ	751012266
TAP	MQTZ-2	SPECIFIC HEAT OF MICRO QUARTZ	752012266
TAP	MQTZ-3	THERMAL CONDUCTIVITY OF MICRO QUARTZ	753012266

TABLE 4-1

(Continued)

TAP	F11-23	THERMAL CONDUCTIVITY OF GASEOUS FREON 11	239012266
TAP	F11-28	COMPRESSIBILITY FACTOR OF GASEOUS FREON 11	240012266
TAP	BERL-1	DENSITY OF BERYLLIUM	291012266
TAP	BERL-2	SPECIFIC HEAT OF BERYLLIUM	292012266
TAP	BERL-3	THERMAL CONDUCTIVITY OF BERYLLIUM	293012266
TAP	AL22-1	DENSITY OF ALUMINUM 2219-T87	301012266
TAP	AL22-2	SPECIFIC HEAT OF ALUMINUM 2219-T87	302012266
TAP	AL22-3	THERMAL CONDUCTIVITY OF ALUMINUM 2219-T87	303012266
TAP	AL70-1	DENSITY OF ALUMINUM 7075-T6	311012266
TAP	AL70-2	SPECIFIC HEAT OF ALUMINUM 7075-T6	312012266
TAP	AL70-3	THERMAL CONDUCTIVITY OF ALUMINUM 7075-T6,AS RECEIVED	313012266
TAP	AL70-4	THERMAL CONDUCTIVITY OF ALUMINUM 7075-T6,ANNEALED	314012266
TAP	MGA3-1	DENSITY OF MAGNESIUM AZ31B-H24	321012266
TAP	MGA3-2	SPECIFIC HEAT OF MAGNESIUM AZ31B-H24	322012266
TAP	MGA3-3	THERMAL CONDUCTIVITY OF MAGNESIUM AZ31B-H24	323012266
TAP	T6AL-1	DENSITY OF TITANIUM 6AL4V	331012266
TAP	T6AL-2	SPECIFIC HEAT OF TITANIUM 6AL4V	332012266
TAP	T6AL-3	THERMAL CONDUCTIVITY OF TITANIUM 6AL4V	333012266
TAP	T110-1	DENSITY OF TITANIUM A110AT	341012266
TAP	T110-2	SPECIFIC HEAT OF TITANIUM A110AT	342012266
TAP	T110-3	THERMAL CONDUCTIVITY OF TITANIUM A110AT	343012266
TAP	C103-1	DENSITY OF COLUMBIUM C-103	351012266
TAP	C103-2	SPECIFIC HEAT OF COLUMBIUM C-103	352012266
TAP	C103-3	THERMAL CONDUCTIVITY OF COLUMBIUM C-103	353012266
TAP	S321-1	DENSITY OF STAINLESS STEEL 321	361012266
TAP	S321-2	SPECIFIC HEAT OF STAINLESS STEEL 321	362012266
TAP	S321-3	THERMAL CONDUCTIVITY OF STAINLESS STEEL 321	363012266
TAP	INCX-1	DENSITY OF INCONEL X	371012266
TAP	INCX-2	SPECIFIC HEAT OF INCONEL X	372012266
TAP	INCX-3	THERMAL CONDUCTIVITY OF INCONEL X	373012266
TAP	INCX-4	THERMAL CONDUCTIVITY OF INCONEL X, SOLUTION TREATED	374012266
TAP	RE41-1	DENSITY OF RENE 41	381012266
TAP	RE41-2	SPECIFIC HEAT OF RENE 41	382012266
TAP	RE41-3	THERMAL CONDUCTIVITY OF RENE 41, 2HR SOLN TREATED	383012266
TAP	RE41-4	THERMAL CONDUCTIVITY OF RENE 41, 4HR SOLN TREATED	384012266
TAP	L12A-1	DENSITY OF LINDE SI-12 (8 LAYERS/INCH)	401012266
TAP	L12A-2	SPECIFIC HEAT OF LINDE SI-12	402012266
TAP	L12A-3	THERMAL CONDUCTIVITY OF LINDE SI-12 (8 LAYERS/INCH)	403012266
TAP	L12B-1	DENSITY OF LINDE SI-12 (10 LAYERS/INCH)	411012266
TAP	L12B-2	SPECIFIC HEAT OF LINDE SI-12	412012266
TAP	L12B-3	THERMAL CONDUCTIVITY OF LINDE SI-12 (10 LAYERS/INCH)	413012266
TAP	L12C-1	DENSITY OF LINDE SI-12 (12 LAYERS/INCH)	421012266
TAP	L12C-2	SPECIFIC HEAT OF LINDE SI-12	422012266
TAP	L12C-3	THERMAL CONDUCTIVITY OF LINDE SI-12 (12 LAYERS/INCH)	423012266
TAP	L12D-1	DENSITY OF LINDE SI-12 (14 LAYERS/INCH)	431012266
TAP	L12D-2	SPECIFIC HEAT OF LINDE SI-12	432012266
TAP	L12D-3	THERMAL CONDUCTIVITY OF LINDE SI-12 (14 LAYERS/INCH)	433012266
TAP	L62A-1	DENSITY OF LINDE SI-62 (40 LAYERS/INCH)	441012266
TAP	L62A-2	SPECIFIC HEAT OF LINDE SI-62	442012266
TAP	L62A-3	THERMAL CONDUCTIVITY OF LINDE SI-62 (40 LAYERS/INCH)	443012266
TAP	L62B-1	DENSITY OF LINDE SI-62 (60 LAYERS/INCH)	451012266
TAP	L62B-2	SPECIFIC HEAT OF LINDE SI-62	452012266
TAP	L62B-3	THERMAL CONDUCTIVITY OF LINDE SI-62 (60 LAYERS/INCH)	453012266
TAP	L62C-1	DENSITY OF LINDE SI-62 (80 LAYERS/INCH)	461012266
TAP	L62C-2	SPECIFIC HEAT OF LINDE SI-62	462012266
TAP	L62C-3	THERMAL CONDUCTIVITY OF LINDE SI-62 (80 LAYERS/INCH)	463012266
TAP	L62D-1	DENSITY OF LINDE SI-62 (100 LAYERS/INCH)	471012266
TAP	L62D-2	SPECIFIC HEAT OF LINDE SI-62	472012266
TAP	L62D-3	THERMAL CONDUCTIVITY OF LINDE SI-62 (100 LAYERS/INCH)	473012266
TAP	L92A-1	DENSITY OF LINDE SI-92 (80 LAYERS/INCH)	481012266
TAP	L92A-2	SPECIFIC HEAT OF LINDE SI-92	482012266

some properties data by means of extrapolation and/or estimation into regions where data were unavailable. All such "extended" data are flagged and interpolations based on these data are automatically noted by the interpolation routines, and suitable explanations are printed out. These "extended" data are not necessarily accurate to within ±5% but are included as a convenience.

Interpolation Routines

Once the required tables are stored in the data block, the properties may be called for in the FUNCT subroutine, using either the linear or bivariate interpolation routines discussed previously, or the linear interpolation with integration routine discussed below. Linear interpolation should be used for the solid and liquid properties data, except for the conductivity of insulations which requires the special integration feature. For gases, the bivariate interpolation routine is required since the properties are temperature and pressure dependent. The tables are interpolated horizontally with temperature as independent variable and vertically with pressure as independent variable. As a result, the function specification must be of the form

Y = BIV (pressure, temperature, table number)

An integer code has been added at the beginning of each table to indicate the special nature of the table and the method of acquiring the data, if applicable. The absence of the code or a code "0" indicates that no special procedures (such as extrapolation) were required to obtain the data, and that interpolation is to take place in a normal fashion. The data classification codes and their explanations are given in Table 4-2.

Since the actual property values in such tables are always positive, the absolute magnitude of the value is used during interpolation. "Extended" data are entered as negative numbers, thus allowing the interpolation routine to recognize when such values have been used, and to print a comment that this has occurred. Note, however, that if the code is zero or does not appear, negative values are treated as negative.

TABLE 4-2

DATA CLASSIFICATION CODES

Code	Explanation
0	Normal curve
1	Extrapolated data
2	Fitted parabola
3	Estimated values
4	"Dummy" values
5	Values for 1 atmosphere pressure
6	Values for saturation line
7	Special bivariate interpolation routine, used in vicinity of saturation line
15	1 plus 5
16	1 plus 6
71	7 plus 1
73	7 plus 3

A special situation exists for interpolation on incomplete bivariate curves indicated by code "7." The properties of gases involve the saturation line, beyond which data are meaningless. In the neighborhood of the saturation line, interpolation will be attempted with only one, two, or three meaningful points. This difficulty is surmounted by entering all points beyond the saturation line as zeros. The routine then recognizes the situation, and shifts its base points until it has four meaningful values. It then extrapolates to the required point. Note, again, that this will occur only if a non-zero code is entered in the table. (See Reference 2.) If the code is zero or does not appear, normal linear bivariate interpolation will occur, and any zero or negative value is treated as zero or negative.

A special linear interpolation and integration subroutine, XLIN, has been included which determines the area under a specified curve, N, from X1 to X2 divided by X2 - X1:

$$XLIN(X1, X2, N) = \frac{1}{X2 - X1} \int_{X1}^{X2} F_N(X)dX \qquad (4-3)$$

This routine is intended specifically to compute temperature-dependent conductivity of insulation. The routine integrates by the trapezoidal rule from the left end of the curve to X1, from the left end of the curve to X2, and then divides the difference by X2 - X1. Equation (4-3) above then becomes:

$$XLIN(X1, X2, N) = \frac{1}{X2 - X1} * \left(\int_{-460}^{X_2} F_N(X)dX - \int_{-460}^{X_1} F_N(X)dX \right) \qquad (4-4)$$

XLIN is a variation of function LIN, and uses identical coding in evaluating the curve at first X1, then X2. It integrates the preceeding part of the curve as it locates the interval in which it interpolates. Having integrated and interpolated for X1, the answers are saved, and a second pass is made to evaluate the same functions for X2. Then the difference between the integrals is taken, divided by (X2 - X1), and that value is returned as the value of function XLIN.

The curve, data block N, is required to be monotonically increasing in the independent variable, but following the last value of the dependent variable there must be a flag less than the last value of the independent variable. This is the same rule that applies to function LIN.

An example of the function specification is

K = XLIN (T(14), T(15), 28)

where the conductivity for the resistor connecting nodes 14 and 15 is to be obtained by integrating curve 28.

V - PROGRAM INPUT

This section describes how to transcribe the sketch of the electrical analog network to data cards. It also describes how to prepare the subroutines which contain the functions and the print format.

The problem description for each case is divided into five distinct blocks and two subroutines. The first three blocks define the network and give the initial values of the temperatures, resistors, and capacitors. The fourth block is a list of sub-blocks of data required by the functions and subroutines. The fifth block lists the print interval, and the initial and final times of the case. The functions and subroutines used in the case are listed in FORTRAN statements in the FUNCT subroutine. The PRINT subroutine contains the list of quantities to be printed at each specified printing interval, plus the desired output format.

A scheme whereby all circuit elements may be referenced between blocks is completed by the user as he describes the circuit. In block 1, each node is assigned an integer which is the designation number of that node throughout the blocks following. Likewise, in block 2, each resistor is assigned a designation number. Each set of data in block 4 is also assigned a designation number. Moreover, the program automatically assigns designation numbers to certain computed and input quantities which are thereby available to the user. A complete list of these addressable quantities was given in Section IV.

The problem description is written on input data sheets or FORTRAN coding sheets of a standard form. Each line of data (blocks 1 through 5) will become a card with columns 1, 2, and 3 containing a three-lettered mnemonic code, columns 4 and 5 either containing a numerical value or left blank, and columns 6 through N containing the data. Columns N + 1 through 72 are then available for user's comments and a card sequence number, if desired.

SPECIAL DATA CARDS

The three-lettered mnemonic code in columns 1, 2, and 3 will take one of the following seven forms:

1. Case Identification Card (CID). This is a card placed at the beginning of each case to identify the output. The letters CID appear in columns 1, 2, and 3, followed by user's comments in columns 6 through 72. If the user desires, these comments may be printed out at the top of each page of output belonging to that case (see "FORMAT Declarations" discussed later in this section).

2. DEC Card. The letters DEC in columns 1, 2, and 3, and a numerical value greater than zero in columns 4 and 5 indicate that the data on that card are not to be handled in any special way, but merely to be entered into consecutive storage. The integers in columns 4 and 5 represent the number of values or sets of values on that card, as discussed in the following sections.

3. Increment Card (INC). This card is used to abbreviate a repetitious list of data. It can only be used in the initial temperature, resistor, and capacitor blocks, and in the data block. It is best explained by an example:

DEC01	a	b.	
INC	2	-3.0	4

means that card DEC01 a b is to be repeated 4 times, each time incrementing the first number on the card by 2 and the second number by -3. Thus the 2 cards above are equivalent to the following 5 cards:

DEC01	a	b.
DEC01	a+2	b-3.0
DEC01	a+4	b-6.0
DEC01	a+6	b-9.0
DEC01	a+8	b-12.0

Note that fixed- and floating-point numbers must be incremented by fixed- and floating-point numbers, respectively. The number of times the card is to be incremented (4 in the above example) is given in fixed point as shown. As a further example, suppose that the following cards are to be incremented:

DEC01	4	5	50	0.
DEC01	5	6	50	0.
DEC01	6	7	50	0.
DEC01	7	8	50	0.
DEC01	8	9	50	0.

The proper instructions are as follows:

DEC01	4	5	50	0.	
INC	1	1	0	0.	4

4. Periodic Data Card (PER). The letters PER in columns 1, 2, and 3, with columns 4 and 5 left blank indicate a periodic curve. The period of the curve is given in floating-point notation in columns 6 through 15. When a periodic curve is specified, the PER card follows the DEC card which contains the curve designation number.

5. End of Block Card (NBK). Each of the first five blocks is terminated by a card having NBK in columns 1, 2, and 3.

6. Geometric Resistor Card (RES). This card is used to specify a resistor whose value is automatically computed by the Thermal Analyzer, given the resistor geometry and the designation number of the curve describing the material conductivity vs temperature. See page 5-6 for details.

7. Geometric Capacitor Card (CAP). This card is used to specify a capacitor whose value is automatically computed by the Thermal Analyzer, given the capacitor geometry and the designation numbers of the curves describing the material specific heat and density vs temperature. See page 5-8 for details.

DATA CARD FORMATS

All CID, INC, PER, and NBK cards have columns 4 and 5 blank. RES and CAP cards always contain a zero (or a blank) in column 4 and a 1 in column 5. DEC cards have columns 4 and 5 blank only when the information contained on the card is the designation number of a curve or table or when the card is a flag in the restart block. In other words, a DEC card will have columns 4 and 5 blank if an only if the card does not contain a floating-point number. Otherwise, a zero (or blank) is placed in column 4 and an integer equal to the number of floating-point numbers contained on the card is placed in column 5.

The numerical data are contained in columns 6 through 65. The first field always begins in column 6. Fixed-point numbers have a field width of 5 columns and must end in the last column of the field in which they appear. Floating-point numbers have a field width of 10 columns and do not have to end in the last column of the field except when the E format (indicating a power of 10) is used to input the value. It is recommended, however, that all values end in the last column of the field to facilitate checking of the input data.

Any column to the right of the last field containing numerical data can be used for comments. However, certain of these comment columns should be reserved for a card sequence number. Ordinarily a 4-digit number starting in column 69, or a 5-digit number starting in column 68 is sufficient. The choice and use of sequence numbers is purely arbitrary, and is at the discretion of the user. As a general rule, however, a systematic scheme, such as that shown in the examples which follow, should be employed to facilitate subsequent changes in the deck and to allow for machine sorting of the cards. Five-digit sequence numbers are useful in cases where a large number of curves are required. A commonly used and convenient procedure is to let the first three digits of the sequence number represent the curve number, and the last two digits represent the card number of that curve. This system is used by the Orbital Radiation Program (Ref. 4) which has an output option to supply the external heating rate history in the form of curves punched in proper format for the Thermal Analyzer Program.

BLOCK 1, INITIAL TEMPERATURES

In block 1, each node is listed by a designation number (fixed-point) by which the node will be referenced in later blocks, followed by the initial value of the temperature at that node (floating-point number). Up to 4 nodes and their initial temperatures may be listed on each card. An example of a temperature input listing two nodes on a card is shown below.

```
COLUMN
----> 1  5    10         20   25      35                                  69 72
      DEC02    5         70.   6      105.                                1005
```

- initial temperature of node 5 in °F in a ten column field ending in column 20.
- node designation number in a five column field ending in column 10
- 2 initial temperatures listed on this card
- 4-digit sequence number consisting of a "1" (to signify temperature block) followed by the designation number of the first node listed on the card.

Note that fixed-point numbers are listed in 5-column fields, and floating-point numbers are listed in 10-column fields. No columns may be skipped between fields.

Since in many practical cases a uniform initial temperature applies to a large portion of the network, the computer is programmed automatically to increment initial temperatures. For example:

DEC01	1	70.
DEC01	50	100.
DEC01	80	20.

would be interpreted as

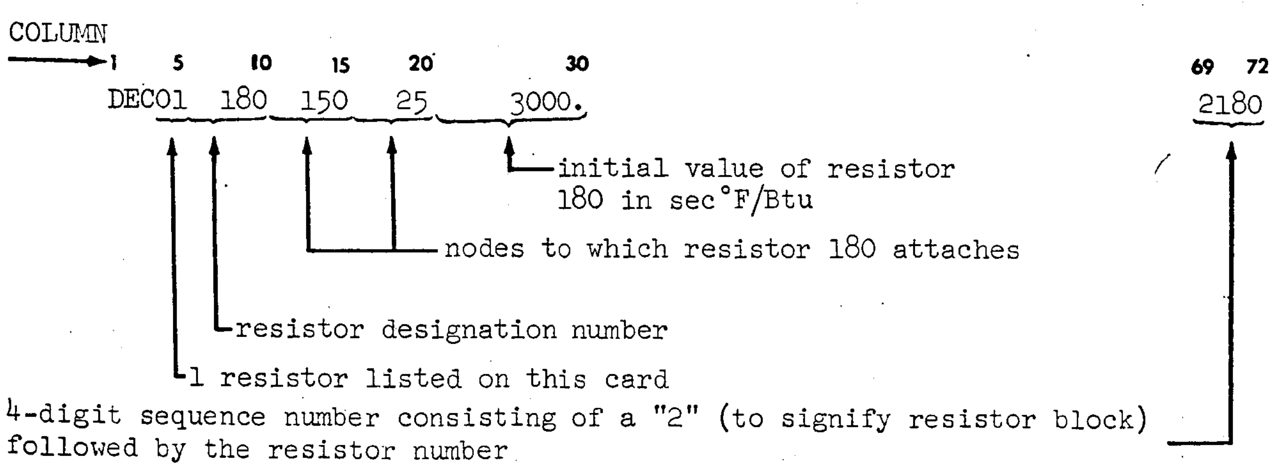
T(1) through T(49) have an initial temperature of 70°F.
T(50) through T(79) have an initial temperature of 100°F.
T(80) has an initial temperature of 20°F.

Note that storage is reserved for nodes 1 through 80 whether they are used or not. Thus, these locations may be used for storing miscellaneous floating-point data.

The initial temperature block, as well as the resistor, capacitor, data, and time blocks, must be terminated with an NBK card.

BLOCK 2, RESISTORS

Each resistor is listed by the designation number of the resistor, followed by the designation numbers of the two nodes which the resistor connects, followed by the initial value of the resistor. In those cases where all values of a resistor are to be computed by a function, a dummy initial value (as zero) must be given. All resistors must be given in this block even though they may be described later (for instance, radiation resistors). Up to 2 resistors may be listed on a card. An example of a resistor input is shown below.



An input option provides for the program to compute its own resistor values, given the resistor geometry and a curve of the material conductivity as a function of temperature. The resistor is computed by the formula

$$R = \frac{\delta}{kwd} \sim \frac{sec\,°F}{Btu}$$

where

δ = the distance along conductive path

k = the thermal conductivity

w,d = the width and depth of the cross-sectional conductive heat transfer area

The resistor is automatically computed if described in block 2 as follows:

COLUMN → 1	5	10	15	20	30	40	50	55
RES01		A	B	C	δ	w	d	N

where

A = the resistor designation number

B,C = the nodes which resistor A connects

δ,w,d = the length, width, and depth of the resistor

N = the designation number of the curve containing the material conductivity as a function of temperature

Any consistent set of units may be employed. A, B, C, and N are listed in fixed-point notation in 5-column fields. δ,w, and d are given in floating-point in 10-column fields. The program updates the resistor value each computing cycle, after interpolating for the conductivity as a function of the temperature of node B.

This option is applicable only when the resistor cross-section is uniform. The conductivity curve can be described in the data block, or may be called in from the thermal properties library.

BLOCK 3, CAPACITORS

Each node at which there is a capacitor is listed by that node's designation number (already assigned in block 1), followed by the value of the capacitor. If the capacity is to be computed by a function, a dummy initial value must be given.

At each node for which a capacitor is specified, i.e., appears explicitly in the capacitor block, a new temperature is computed for that node. If, however, no capacitor is specified for a given node, no heat balance is computed, and the temperature of that node remains unchanged, unless changed by a function statement.

Up to four capacitor values can be listed on a card. An example of a capacitor input is shown below:

COLUMN → 1 5 10 20 ... 69 72

DEC01 25 .117 ... 3025

- .117 — value of capacitor 25 in Btu/°F
- 25 — capacitor (node) designation number
- 01 — number of capacitors listed on this card
- 3025 — 4-digit sequence number consisting of "3" (to signify capacitor block) followed by the capacitor designation number.

An input option provides for the program to compute its own capacitor values, given the node geometry and curves of density and specific heat as a function of temperature. The capacitor is computed by the formula

$$C = \delta w d \rho c \qquad \frac{\text{Btu}}{{}^\circ\text{F}}$$

where

δ, w, d = the length, width, and depth of the node

ρ = the material density

c = the material specific heat

The capacitor is automatically computed if described in block 3 as follows:

COLUMN → 1	5	10	20	30	40	45	50
CAP01		A	δ	w	d	N_1	N_2

where

A = the capacitor designation number

N_1 = the designation number of the curve containing the material specific heat vs temperature

N_2 = the designation number of the curve containing the material density vs temperature

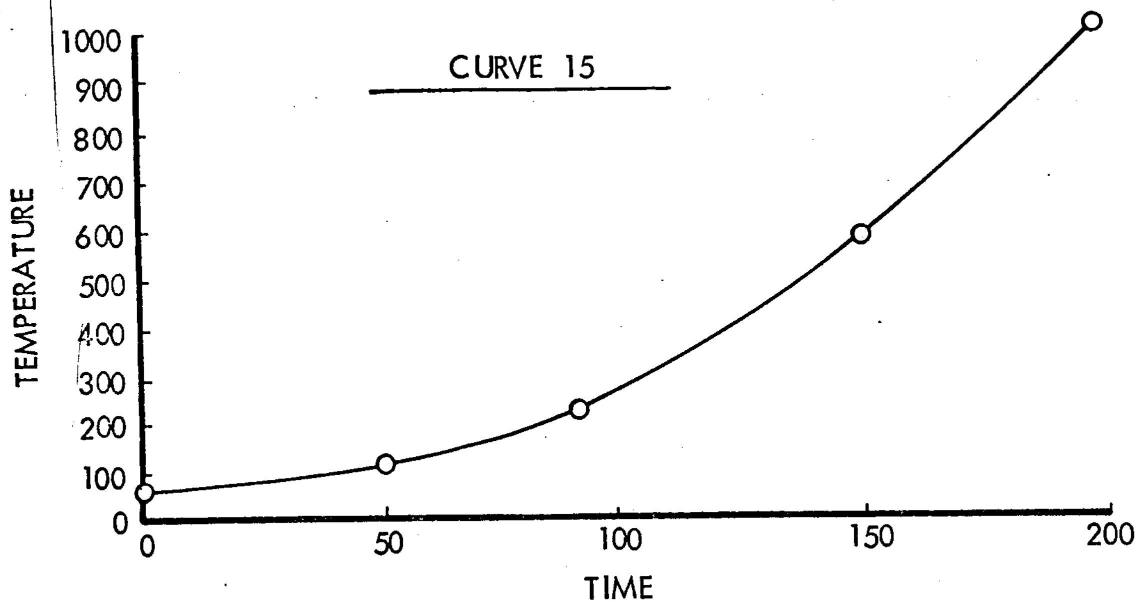
Any consistent set of units may be employed. A, N_1 and N_2 are given in fixed-point notation in 5-column fields, and δ, w, and d are given in floating-point notation in 10-column fields. The capacitance of node A is updated each computing cycle, after the program interpolates curves N_1 and N_2 for the specific heat and density.

BLOCK 4, INPUT DATA

Block 4 consists of individual sub-blocks of input data which are used by the functions. The function which uses the data determines the nature of the numbers in the sub-block. For instance, the data might be the point by point description of a curve which the function will use for interpolation. Also, the data block is very convenient as a storage for data constants used for special calculations, or as temporary storage to be used as required during various portions of the program.

Each sub-block of data is identified by a fixed-point number called the designation number of the sub-block. The designation numbers are arbitrary, but can be used only once. A DEC card with the designation number listed in columns 6 through 10 must precede the sub-block, and a DEC card with the negative of the designation number listed in columns 6 through 10 must terminate the sub-block.

To illustrate, suppose that the curve plotted below is to be described in the data block.



Curve 15 would appear on the input data sheet as follows:

COLUMN →

1	5	10	15	25	35	45	55	65	69 72
DEC		15							4001
DEC06			0.	60.	50.	110.	90.	220.	4002
DEC05			150.	570.	200.	1000.	0.		4003
DEC		-15							4004

In block 4, the sequence numbers are often assigned as consecutive integers starting with 4001. In the above example, card 4001 contains the curve designating number, and card 4004 contains the negative of the curve designation number. The actual curve is described on cards 4002 and 4003 by listing the coordinates of the points circled on the plotted curve. Each independent variable must immediately precede the corresponding dependent variable, and the independent variables must be listed in increasing order. The number of points needed to describe a curve depends on the accuracy required, and the type of interpolation routine employed. The machine as yet does not know whether the variables are time, temperature, or something else;

this information being given in the FUNCT subroutine. The zero on card 4003 is a flag indicating the end of the curve. This flag can be any number as long as it is less than the preceding value of the independent variable, which in this example is 200. This point is important, for if all values of the independent variable were negative, then the zero would not be correct. If, for example, the last value of the independent variable were -50., the end-of-curve flag would have to be something like -51.

The designation number of a sub-block may be used in the FUNCT and PRINT subroutines to refer to a particular element in the sub-block. Before such a reference can be made, however, a locating statement must be made in the subroutine in which the reference is made. For example, to refer to a value in curve 15 above, the statement

L15 = LOC(15)

is first necessary. Then the sixth number in curve 15 (in this case the value 220.) would be referred to as

P(L15 + 6)

In determining the sixth number note that the table designation number itself is not counted. Also, if this had been a periodic curve, the value given for the period would not be counted.

As stated previously, the data block must be terminated with an NBK card. Occasionally, a thermal analyzer problem will be run in which block 4 is not required. In this case two consecutive NBK cards must appear at the end of the capacitor block.

BLOCK 5, TIME CONSTANTS

Block 5 contains the printing interval, the final time of the case, and the initial time of the case. These are the only three quantities to appear in block 5, and they must be given in the order listed above. The unit of time must be consistent with that used throughout the problem, normally seconds. A sample time block is shown on the following page.

```
COLUMN
——→ 1   5         15        25        35                                 69  72
    DEC03       5.       150.       0.                                     5001
                ↑          ↑         ↑— initial time                        ↑
                |          └—final time                                     |
                └— print interval                                           |
    4-digit sequence number starting with a "5" (to signify time block) ————┘
```

These items can, of course, be listed on separate cards, thus allowing room for comments following the numbers.

RESTART BLOCKS

Many times it is desired to run several cases which are basically similar, differing only in the values of certain parameters. An example of this is a parametric study of the insulation thickness required for a given application where only resistor and capacitor values are changed between cases. Another example is an Earth-orbiting vehicle, where solar inputs can vary depending upon the angle between the Earth-sun line and the orbit plane. The restart can also be used to string several transient cases together, with the final temperatures of one case being used as the initial temperatures for the case which follows.

To use the restart feature, one or more restart blocks (each one constituting a separate case) can be added, with each restart block causing the preceding case to be re-run as a new case with certain changes as indicated in the restart block. A restart block can change any or all of the data sub-blocks, but if a data sub-block, e.g., a curve, is changed, it must be restarted in its entirety together with the same designation number used in the original case and it must contain the same or fewer numbers than the data sub-block being replaced. If the curve is periodic, the PER card must be included in the restart. Also, a restart block can change the time block, but if the time block is changed, it must be restarted in its entirety. If more than one restart block follows a case, each succeeding restart block is interpreted to be changes to the immediately preceding case, and not changes to the original case.

Since each restart is a separate case, a CID card is inserted ahead of each restart to identify the output. This card is not mandatory, but its use is recommended. The next card is a flag indicating to which block the cards following the flag refer. The flags are as follows (note absence of decimal point):

COLUMN

→ 1	10	
DEC	1	New initial temperatures
DEC	2	New resistors
DEC	3	New capacitances
DEC	4	New time block
DEC	5	New data sub-blocks

Immediately following each of the initial value change flags (1, 2, or 3) are the change cards. Each of these change cards contains two items, the first being the designation number of the item to be changed and the second being the new initial value of that item. A DEC 0 card ends each set of change cards pertaining to a particular class of quantities, e.g., resistors. If the time block is to be changed, a DEC 4 card followed by the complete time block, i.e., print interval, final time and initial time, plus a DEC 0 at the end is required. A DEC 5 card precedes the new data sub-blocks, with a DEC 0 card after the last new sub-block. Each restart block, i.e., all the changes required for a given case, is terminated with an NBK card.

As an example of a restart, the following changes are to be made in a case:

(a) T(17) = 50.
(b) R(15) = 6000.
(c) C(11) = 0.031
(d) New linear curve 40, where,

y = 17. at x = 0.
y = 53. at x = 100.

(e) New time block, where,

M(4) = 10.
M(3) = 500.
M(2) = 150.

The input is shown on the following page. The order in which the various blocks are changed is arbitrary.

```
COLUMN
--->  1        10   15   20    25
     CID  RESTART EXAMPLE
     DEC       2                       Change resistors
     DEC01    15       6000.
     DEC       0
     DEC       1                       Change temperatures
     DEC01    17          50.
     DEC       0
     DEC       4                       Change time block
     DEC01          10.                New print interval
     DEC01         500.                New final time
     DEC01         150.                New initial time
     DEC       0
     DEC       3                       Change capacitors
     DEC01    11         .031
     DEC       0
     DEC       5                       Change data block
     DEC      40                       New curve 40
     DEC02           0.        17.
     DEC02         100.        53.
     DEC01           0.                End curve 40
     DEC     -40
     DEC       0                       End curve changes
     NBK                               End restart block
```

One very useful application of the restart feature is in the analysis of an Earth-orbiting vehicle for which several sets of solar inputs may be applicable depending upon the orientation of the orbit plane with respect to the Earth-sun line. For this case the basic deck incorporates only the storage required to accommodate the solar data. To prevent the basic deck from running, the initial and final times should be input as zero. The correct time block can then be included in the various restart decks which load the particular solar inputs for each orbit orientation.

The following points should be kept in mind when using restarts:

- A restart block can only change values which had been given in the original case.
- A restart cannot change the structure of the network, i.e., the way the various resistors and capacitors are connected.
- A restart cannot add or delete data sub-blocks, but merely change the data contained in these sub-blocks.
- A restart block cannot change the FUNCT or the PRINT subroutines.
- The data within a restart block can be incremented, but those increment cards used in the original case are no longer applicable and must be repeated if the restart data is to be incremented.

FUNCT SUBROUTINE

The FUNCT subroutine is an ordinary FORTRAN subroutine in which the user lists all curve interpolations, arithmetic operations, and special functions to be performed during program execution. These functions are executed at the beginning of each computing cycle, before the heat balance is performed. A standard input format is described below so that the user need not have a knowledge of FORTRAN to prepare this routine. If the user is familiar with FORTRAN he should still learn to fill out the forms as described, but then should not hesitate to employ his knowledge to supplement the capabilities of the prepared program.

The input form for the FUNCT subroutine is illustrated in Table 5-1. All of the information shown must appear in the subroutine and, with the exception of the functions, these are generally the only items which must appear. With the exception of the first card, the information begins in column 7. A four-digit sequence number is placed in columns 77 through 80 of each card as a convenience in sorting or making changes to the deck.

The statement $IBFTC FUNCT on the first card identifies the function subroutine and must be input as shown. The statement NODECK,NOREF will delete from the output the internal reference listing for the subroutine and will eliminate the punching of a binary deck. If either of these items is desired, replace NODECK (or NOREF) with DECK (or REF), leaving no blank spaces; e.g., DECK,REF.

The statement SUBROUTINE FUNCT sets up the linkage between the FUNCT subroutine and the calling program, and must be input as shown.

The variables listed in the COMMON declaration are assigned locations in a reserved section of core so they will not be destroyed by overlay operations. Card 6003 must also appear as shown.

The DIMENSION declaration tells the processor how much space in memory must be allocated or reserved for each collection of elements. Each variable which appears in the program in subscripted form must appear in a DIMENSION statement, and the statement must precede the first appearance of the variable. For example, on card 6004, 16 storages are reserved for the collection of elements called M, which are defined in Section IV. The DIMENSION declaration should be filled out exactly as shown. If the user wishes to add another

TABLE 5-1

INPUT FORMAT FOR FUNCT SUBROUTINE

GENERAL PURPOSE DATA SHEET

LOCKHEED-CALIFORNIA COMPANY
A DIVISION OF LOCKHEED AIRCRAFT CORPORATION

PREPARED	NAME	DATE	TITLE	PAGE OF
CHECKED			WO EWA	JOB NO.
				GROUP

FORM DX 7924.1

SEQ. 77-80	1–72	ID 73–76
6001	`$IBFTC FUNCT    NØDECK,NØREF`	
6002	`      SUBRØUTINE FUNCT`	
6003	`      CØMMØN P,M`	
6004	`      DIMENSIØN P(16000),M(16),TN(1),T(1),R(1),C(1),Q(1),RC(1)`	
6005	`      EQUIVALENCE (P(1),TN(1)),`	
6006	`     1 (P(     ),T(1)),`	
6007	`     2 (P(     ),R(1)),`	
6008	`     3 (P(     ),C(1)),`	
6009	`     4 (P(     ),Q(1)),`	
6010	`     5 (P(     ),RC(1))`	
6011	`      REAL M,LIN`	
6998	`      RETURN`	
6999	`      END`	

subscripted variable to the subroutine, storage may be reserved by including on card 6004 the name of the variable followed in parentheses by the number of quantities requiring storage space. Alternately, a second DIMENSION card could be added to accomplish the same purpose.

From the user's standpoint the EQUIVALENCE declaration is required so that he may refer to the temperatures, resistors, capacitors and heating rates in terms which are meaningful to him. As an example, it allows the user to refer to a particular temperature by a T followed in parentheses by the node designation number. The user must fill in the blank spaces within the parentheses on cards 6006 through 6010 with the five integers defined below:

CARD	INTEGER
6006	j=1 + greatest temperature designation number
6007	k=j + greatest temperature designation number
6008	m=k + greatest resistor designation number
6009	n=m + greatest capacitor designation number
6010	p=n + greatest capacitor designation number

Since columns 7 through 72 are all available for data input, the EQUIVALENCE declaration could actually be written on two cards, instead of six as shown. The integers in column 6 are codes indicating that the information on that card is a continuation of the information on the previous card.

The REAL statement defines the variables M and LIN as floating-point values, since ordinarily these variable names are reserved for fixed-point values.

Next, all functions which the program is required to execute are listed, in the order of their execution. The order of execution is generally unimportant except when the execution of one function involves the result of another. For the user's convenience, the functions discussed in Section IV are summarized in Table 5-2.

The FUNCT subroutine is always terminated with the RETURN statement and the END declaration. The RETURN statement marks the completion of the intended task, and returns control to the calling program. The END declaration merely signals the processor that there are no more cards to be translated for that program.

TABLE 5-2.

SUMMARY OF FUNCTIONS AND SPECIAL SUBROUTINES

FUNCTION	EXAMPLE	MEANING
LINEAR INTERPOLATION	Y=LIN(X,N)	Y is a linear function of X given by curve N
PARABOLIC INTERPOLATION	Y=PAR(X,N)	Y is a parabolic function of X given by curve N
BIVARIATE INTERPOLATION	Y=BIV(XI,X2,N)	Y is a bivariate function of X1 and X2 given by table N
LINEAR INTERPOLATION WITH INTEGRATION	K=XLIN(X1,X2,N)	K (thermal conductivity of insulating material) is $\frac{1}{X2-X1}\int_{X1}^{X2} F_N(X)\,dX$, where X1 and X2 are the temperatures of the nodes on each side of the insulation
RADIATION WITH CONSTANT FACTOR	R(A)=RAD(B,C,K)	Resistor A connects nodes B and C and the K factor is a fixed constant.
RADIATION WITH VARIABLE FACTOR	R(A)=RAD(B,C,X)	Resistor A connects nodes B and C and the K factor is stored in the circuit element X.
RADIATION WITH MATRIX	R(A)=RRM(B,C,N)	Resistor A connects nodes B and C and table N contains the data used to compute the K factors.
CONVERGENCE	CALL CVG(A,N,M)	The heat balance and list of functions are evaluated repeatedly until the temperatures reach the required convergence, A. A maximum of N iterations are to be performed, and a print is required every M-th iteration. If M is negative, convergence will be attempted with all capacitors set to zero.

TABLE 5-2 (Cont)

FUNCTION	EXAMPLE	MEANING
STEADY STATE	CALL STS(A)	All capacitors are set to zero, the function list is ignored, and the heat balance evaluated repeatedly until the required convergence, A, is obtained.
MAXIMUM-MINIMUM	CALL MMF(θ_i, θ_f, N,S)	The program searches the output every S-th computing cycle between times θ_i and θ_f to determine the maximum and the minimum temperatures of the nodes listed in table N.
SAVE CURRENT DATA	CALL SAV(A,B)	If A (usually current time) is greater than B, the data currently in core are stored on tape and used as initial conditions in the subsequent case.
INITIAL TEMPERATURE CARD PUNCH	CALL PUNCHT (θ,A,B)	The temperatures A through B at time θ are recorded on punched cards in proper format for the initial temperature block.
CURVE PLOTTING	CALL TPLOT(N)	The temperatures of all nodes listed in table N are to be machine plotted.
AERODYNAMIC HEATING	CALL EAH i(j,K, H,α_o,F)	The aerodynamic heating to node j is to be computed by the reference temperature method. K is the Eckert K factor, H is the location where the calculated heat transfer coefficient is to be stored, α_o is the surface angle of attack, and F is a code indicating whether node j is located on an upper or lower surface. H, α_o, and F are optional inputs.
ABLATION	CALL ABL(i,N)	The surface (ablating) node number is i and table N contains the ablator properties used to calculate the network parameters.

PRINT SUBROUTINE

The PRINT subroutine is an ordinary FORTRAN subroutine in which the user specifies the quantities that are to be printed out at each printing interval, and the desired output format. Like the FUNCT subroutine, no knowledge of FORTRAN is necessary to prepare this routine. However, to take advantage of the ability to specify the output format, the user is encouraged to learn the basic rules in writing format statements. In the discussion which follows, emphasis is placed on the input for a "standard" print format which will be acceptable to the user in most cases. This format lists the data in six full columns, with the corresponding time appearing by itself to the left of the first row of output. None of the quantities are labeled, but this is of little consequence unless a very large number of quantities are being printed. The advantages of this format are its simplicity and the fact that it requires no knowledge of format statements by the user. Following the explanation of this "standard" format, some of the general rules pertaining to the writing of format statements are reviewed for those who wish to label their output.

Standard Print Format

The input form for the PRINT subroutine is illustrated in Table 5-3. The first eleven cards contain essentially the same information as the corresponding cards in the FUNCT subroutine (Table 5-1). The blank spaces in the EQUIVALENCE declaration must be filled in with the integers j, k, m, n, and p defined in the preceding section. The REAL statement in the PRINT subroutine does not declare LIN to be a real, or floating-point, value since the variable LIN will not ordinarily appear in this subroutine.

The WRITE statement specifies the quantities to be printed and references the statement number of the FORMAT declaration which specifies the arrangement of the output data. The general form of the WRITE statement is

```
WRITE (6,N) A,B,C,....
```

where 6 is the designation number of the system output tape, N is the number of the corresponding FORMAT declaration, and the quantities A, B, C, etc.,

TABLE 5-3

INPUT FORMAT FOR PRINT SUBROUTINE

FORM DX 7924.1 — GENERAL PURPOSE DATA SHEET — LOCKHEED-CALIFORNIA COMPANY, A DIVISION OF LOCKHEED AIRCRAFT CORPORATION

PREPARED | NAME | DATE | TITLE | PAGE OF | JOB NO.
CHECKED | WO | EWA | GROUP

SEQ.	Columns 1–72
7001	`$IBFTC PRINT     NØDECK,NØREF`
7002	`      SUBRØUTINE PRINT`
7003	`      CØMMØN P,M`
7004	`      DIMENSIØN P(16000),M(16),TN(1),T(1),R(1),C(1),Q(1),RC(1)`
7005	`      EQUIVALENCE (P(1),TN(1)),`
7006	`     1 (P(     ),T(1)),`
7007	`     2 (P(     ),R(1)),`
7008	`     3 (P(     ),C(1)),`
7009	`     4 (P(     ),Q(1)),`
7010	`     5 (P(     ),RC(1))`
7011	`      REAL M`
7012	`      WRITE (6,10) M(1),`
7013	`     1`
7014	`     2`
7015	`     3`
7016	`     4`
7017	`     5`
7018	`     6`
7019	`     7`
7020	`     8`
7021	`     9`
7022	`     A`
7023	`  10  FØRMAT (1H0,7E18.8/(19X,6E18.8))`
7024	`      RETURN`
7025	`      END`

LOCKHEED CALIFORNIA COMPANY

are to be printed in that order. On card 7012, for example, the statement

WRITE (6,10) M(1),

will cause the value of current time to be printed first. The user should then list the remaining items of interest, separated by commas, in the order in which they are to be printed. Any addressable element may be printed out. Although not required by the program, if the standard six-column format is used, these elements should be listed six to a card for convenience in identifying the output. There is no limit to the number of items that may be printed. Any character other than 0 (zero) or a blank may be used in column 6 as a continuation flag. However, the maximum number of continuation cards in any statement is 19. If more than 20 lines are required, the remaining items are listed on additional WRITE statements, all of which begin

WRITE (6,10)

When the designation numbers to be listed in a WRITE statement are consecutive, they can be abbreviated. To illustrate, if the temperatures of nodes 20 through 85 are to be printed consecutively, an acceptable WRITE statement is

WRITE (6,10) (T(I), I=20,85)

The FORMAT statement shown in Table 5-3 calls for the values to be printed out using the E format (for powers of 10), with seven quantities listed on the first line, and six on each remaining line.

The PRINT subroutine must be terminated with the RETURN and END cards.

The output format described above is illustrated in Example #1 of Section VI. The following section describes the FORMAT declaration in more detail for those who wish to take advantage of the opportunity to specify the output format.

FORMAT Declarations

When the results are to be printed on one or more lines of paper, the computer must know how the information is to be distributed among the columns on each printed line. The information to be printed on a particular line may be thought of as a unit output record. Some line printers for the IBM 1401 and 1410 computers can accept unit output records containing 100 characters while other printers, such as the 1401 with special features, provide 132 printer columns. The following discussion pertaining to output will assume the use of a line printer which can accept records containing 132 characters. Regardless of the printer used, the first character of each record is actually not printed. Instead, it is interpreted by the printer as a special code for control of paper movement just prior to printing the record. As a result, only 131 characters are available to the user to become printed information on 131 columns of the printed line.

Each unit output record is made up of one or more fields, a field being a group of one or more columns whose contents can or must be described separately. Each field is described by a format code which specifies the form, size, and location of each field from left to right within each of one or more records. The purpose of the FORMAT declaration is to make this code available to the computer during execution of the WRITE statement. The remainder of this section is concerned with the definition, description, and use of format codes.

Field Specification Codes - Numerical output will take one of the following three forms:

1. Integers (I-fields) - Integers are printed out without decimal point; for example,

$$\pm XXX$$

2. Floating Point With Exponent (E-fields) - When using this form, the decimal point is printed immediately to the left of the leading significant digit; for example,

$$\pm 0.X_1X_2X_3X_4E \pm n_1n_2$$

applies to a request for 4 decimal places.

3. Floating Point Without Exponentiation (F-fields) - An example of this form applicable to a request for 5 decimal places is

$$\pm X_1X_2X_3.X_4X_5X_6X_7X_8$$

In all three examples, the negative sign will appear if the quantity is negative, but the + sign will never appear.

A field specification for an I-, E-, or F-field consists of one of the letters I, E, or F followed by an integer which specifies the size of the field, i.e., the number of available columns to be used. If the E or F field is specified, an additional code is required to denote the location of the decimal point. For example, I6 specifies a six-column integer field, and F10.5 specifies a ten column F-field with five numbers following the decimal point. Caution should be exercised in specifying the field width of E-fields since accuracy to one significant digit only requires an E-field of eight columns, i.e.,

$$\pm 0.X_1E \pm m_1n_1$$

As examples of the general output appearance of E and F fields, consider the number -13.175492. The output appearance for several different field specification codes are shown below:

Code	Output Appearance
F10.6	-13.175492
F10.3	-13.175
F10.0	-13.
E10.3	-0.132E 02
E10.4	0.1318E 02 (invalid)

Note that E-field specifications must provide at least 7 columns in excess of the number of decimal places required. Regardless of the format code, the last digit always appears in the last column of the field.

Format Declarations for Records Containing Numerical Data Only - The general form of the FORMAT declaration is simply a statement number followed by the word FORMAT followed by the format code in parentheses; i.e.,

NN FORMAT (format code)

Its use is best explained by examples. Suppose that we wish to print the variables M(5), T(6), R(18), C(26), J1, and KLOD in that order. The first four are floating point variables, and the latter two are fixed point. If the WRITE statement were

```
WRITE(6,35) M(5), T(6), R(18), C(26), J1, KLOD
```

the FORMAT declaration might appear as

```
35 FORMAT (F10.4, F10.4, F10.4, F10.4, I5, I5)
```

or

```
35 FORMAT (4F10.4, 2I5)
```

Both formats would result in identical outputs consisting of a 10-column F-field for the floating point numbers, and a 5 column integer field for the fixed point numbers. The printed line might appear as follows (note that the first column is always skipped):

```
COLUMN
———→     2      10       20        30        40   45   50

          50.0000 -31.7183 8267.9526   1.8913    6  -83
```

Parentheses may be used within the format code to indicate repetition of the format description within the parentheses for all succeeding lines. For example, the WRITE statement

```
WRITE (6,10)(T(I), I=1,50)
```

and corresponding FORMAT declaration

```
10 FORMAT (6E15.4/(7F15.2))
```

will cause the first six temperatures to be printed on the first line using the E-field format. Then, the remaining 44 temperatures will be listed, seven per line using the F-field format. The slash (/) indicates the end of a record, i.e., the end of a printed line, as explained in the next section.

Printer Carriage Control - In order to print an output record, the line printer must first be told on which line that record should be printed. With this information, the printer can then move the carriage which holds the paper the desired amount just prior to the printing of a line. The printer receives the desired carriage control information in the form of a one character code placed at the first of the output record being readied for printing. The first position of the record, therefore, is never printed and is always assumed to be a carriage control code. It is therefore imperative that each output record be format coded to provide a carriage control code in the first column.

Single line spacing means that the paper is advanced one line before printing. The code for single line spacing is a blank (□). With some risk of error, a blank is ensured if the first field is an I-, E-, or F-field which has been provided with more than enough columns to print the given number. A much safer approach is to force a blank in the first character of the output record by making the first field a skip field. This is accomplished by incorporating in the format code the specification mnX, where mn is an integer representing the number of columns to be skipped. Some examples are given below.

In the last example of the preceding section

FORMAT (6E15.4/(7F15.2))

was assumed to be a suitable FORMAT declaration. A much preferred form, in which we are certain to have a blank as the leading character of each record, is

FORMAT (1X,6E15.4/(1X,7F15.2))

When this form is used, each new record described begins with a 1X field. The first line has the format 1X,6E15.4 and all succeeding lines are described as 1X,7F15.2.

Line spacing is accomplished by use of a slash (/) or multiple slashes ahead of or following any record description. The above example is now repeated several times to illustrate this technique:

Example 1

FORMAT (1X,6E15.4//(1X,7F15.2))

The double slash causes an extra blank line to appear after the first line is printed. The first slash marks the end of the first line and the second slash marks the end of the second line.

Example 2

FORMAT (1X,6E15.4////(1X,7F15.2))

By the same reasoning quadruple slashes cause three extra blank lines. In general, N slashes appearing at the end of a record description will produce N-1 extra lines after that record.

Example 3

FORMAT (1X,6E15.4/(1X,7F15.2)/)

The slash before the final right parentheses does not cause an extra line. The combination "/)" has the same effect as the final parentheses ")" by itself.

Example 4

FORMAT (1X,6E15.4/(1X,7F15.2)//)

The double slash causes one extra blank line after each record coded (1X,7F15.2).

Example 5

FORMAT (//1X,6E15.4/(1X,7F15.2))

The double slash causes two blank lines before the first record is printed, so the printing begins on the third line. In general, if N slashes appear at the very beginning of the format code, N extra blank lines will result.

Hollerith Fields - The use of a Hollerith or H-field enables the user to provide for the printing of alphameric words or phrases in the form of comments, titles, headings, etc., to explain the numerical results being printed. Use of H-fields, while frequently desirable, is optional. An H-field is of the form sssHccc...ccc

where

sss is a space count which may be any integer up to 132.

H is the symbol identifying the Hollerith field.

ccc...ccc denotes the contents of the sss spaces following immediately after the letter H; ccc...ccc includes all blanks and special characters, i.e., all blank spaces indicated by the symbol □ must be included in the space count.

Hollerith fields are especially useful to print messages, titles, and headings without calling for the value of variables. Also, they can be used for line-printer carriage control. Since the use of a line printer requires that the first position of any output record be a carriage control character it is convenient to introduce a one space Hollerith field as the first field of every output record, as shown below.

Desired Paper Movement Upward	Required Carriage Control Code	Simplest Way of Coding
one line	□ (a blank)	1 H □
two lines	0 (zero)	1 H 0
skip to top of next page	1	1 H 1

To illustrate, the format used as an example above

FORMAT (1X,6E15.4/(1X,7F15.2))

could be written

FORMAT (1H□6E15.4/(1H□7F15.2))

and in each case the output would appear the same, i.e., single-spaced. If we wished to start printing at the top of the page, the appropriate format code is

FORMAT (1H16E15.4/(1H□7F15.2))

To further illustrate the use of Hollerith fields assume that it is desired to list the current time M(1), print interval M(15), minimum RC product M(6), and the node at which the minimum RC product appears M(9).

Furthermore, this information is to be printed on one line and labeled as follows:

TIME=XXXXXXX.^SECONDS^^^^^COMP.^INTERVAL=XXXX.XXX^SECONDS^^^^^(RC)MIN=XXXX.XXX^SECONDS^AT^NODEXXXXX.

The carets (^) indicate that a blank space is to be provided. If a paper movement of two lines (double space) is desired before writing this line, the appropriate WRITE statement and FORMAT declarations appear as follows:

```
                 WRITE (6,837) M(1), M(15), M(6), M(9)

837 FORMAT (6H0TIME=,F8.0,8H□SECONDS,5X,15HCOMP.□INTERVAL=F8.3,8H□SECONDS,
    1  5X,8H(RC)MIN=,F8.3,16H□SECONDS□AT□NODE,F6.0)
```

The following remarks pertaining to the above FORMAT declaration are applicable:

1. The zero in the first Hollerith field serves as carriage control, moving the paper upward two lines.

2. Any character lying within the Hollerith field, i.e., any character found within the sss spaces following the H, will be printed exactly as it appears in the format code. These spaces within the H-field must be counted carefully to ensure that the correct space count is used. An error in the space count will usually render the format code invalid.

3. It is not necessary to separate by a comma an H-field from any characters which follow it on the right. Thus, in

   8H(RC)MIN=,F8.3....

   the comma could be optionally omitted after the equal sign. The comma in

   F8.3,16H□SECONDS.....

   is required. The reason is that commas are required to mark the end of one field code and the beginning of another when the computer has no other means of determining this demarcation. Due to space count, an H-field, unlike other field codes, provides sufficient information for the computer to determine where it ends.

4. The integer 1 on the second line of the FORMAT declaration is a continuation code, and appears in column 6.

The reader should now be in a position to interpret the "standard" Format code given in the Section "Standard Print Format", i.e.,

FORMAT (1H0,7E18.8/(19X,6E18.8))

After the paper is advanced 2 lines, the first 7 numerical quantities are listed, using an 18 column E-field format. On each succeeding line the first 19 spaces are skipped, followed by 6 numerical quantities again using the E-field format.

Alphameric Fields - Alphameric or "BCD" information may be stored internally in the computer and printed out when desired using the A-field code. This code has the following form:

Axx

where xx is the field width or the number of columns allocated for the field, and A is the code letter for this type of field.

The use of A-fields will not be discussed in detail except for explaining the procedure to have the information on the CID card printed in the output. To accomplish this, the COMMON card in the PRINT subroutine must appear as

COMMON P,M,MISCEL(23),CID(12)

Then, assuming the FORMAT declaration is statement number 101 and it is desired to skip to the top of the next page, the WRITE statement and FORMAT declarations appear as follows:

```
WRITE (6,101)(CID(I),I=1,12)

101  FORMAT (1H1///33X,12A6)
```

The triple slash will cause the top two lines to be skipped. Thirty-three spaces are skipped to approximately center the CID information on the top of the page. The line skipping and centering are, of course, optional.

VI - EXAMPLE PROBLEMS

Two examples demonstrating the various program features are presented in this section. These problems are worked in detail to show the complete process involved in converting the physical system into an equivalent RC network and the transcription of this network onto the data sheets from which the data cards are punched. The computer output is shown for both problems.

EXAMPLE #1 - Temperature response of an equipment-mounting plate during sub-orbital test flight of a lifting re-entry vehicle.

Problem Description

A number of heat-dissipating electronic components are mounted on a 0.10-in. aluminum plate in the equipment bay of an unmanned lifting re-entry test vehicle. The plate and equipment radiate on both sides to quartz fiber insulation attached to the internal surfaces of the upper and lower skins. The skin temperatures are known functions of time from a separate ascent and re-entry heating analysis. Assuming a mean heat dissipation of 100-w/sq ft (0.095 Btu/sec-ft^2) and an equipment density of 25 psf, the problem is to determine the temperature rise of the aluminum plate and equipment during the suborbital flight. The physical picture and corresponding thermal network are shown in Figure 6-1.

The insulation is divided into lumps either 1.0 in. or 0.5 in. thick, with nodes appearing on the interior boundaries where radiation to the equipment plate occurs. Although node and resistor designation numbers are completely arbitrary a systematic scheme is employed for convenience. The following material properties are assumed.

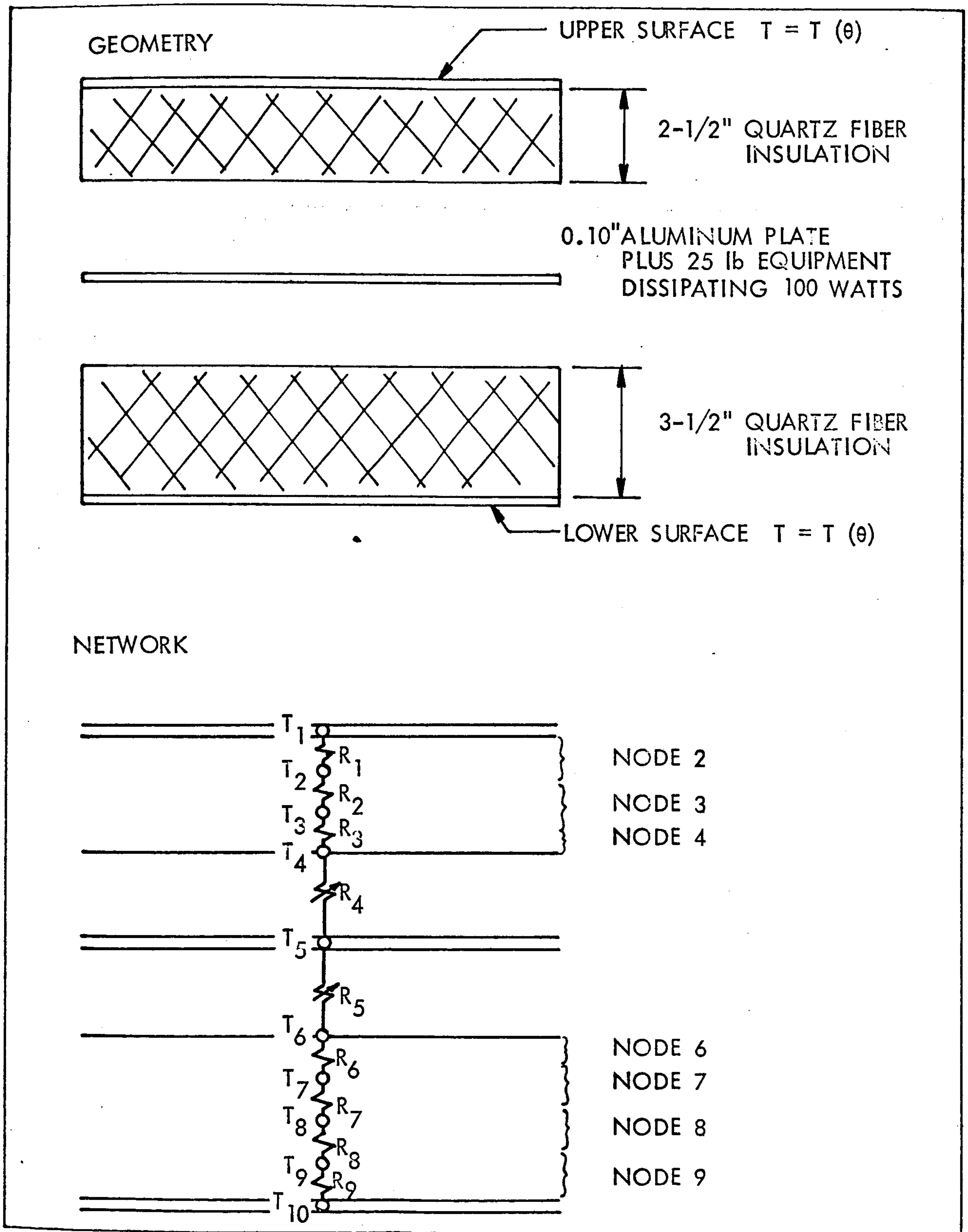


Figure 6-1. Physical Geometry and Corresponding Thermal Network for Example Problem 1

Property	Insulation	Aluminum
k = thermal conductivity (Btu/hr ft°F)	0.06	NR
c = specific heat (Btu/lb°F)	0.25	0.20
ρ = density (lb/ft^3)	3.0	175
ε = surface emissivity	0.8	0.1

Conduction Resistors

$$R2 = R3 = R6 = R7 = R8 = \frac{3600\,\delta}{kA} = \frac{(3600)\left(\frac{1}{12}\right)}{(0.06)(1)} = 5000\ \frac{\sec{}^\circ F}{Btu}$$

$$R1 = R9 = \frac{3600\,\delta}{kA} = \frac{3600\ \frac{0.5}{12}}{(0.06)(1)} = 2500\ \frac{\sec{}^\circ F}{Btu}$$

The resistance through the external skins is neglected, since it is negligible compared with that through the insulation.

Radiation Resistors

The radiation K factors are computed using the effective emissivity factor given by the infinite parallel plate formula, and a view factor of unity.

$$\epsilon_{12} = \frac{1}{\frac{1}{\epsilon_1} + \frac{1}{\epsilon_2} - 1} = \frac{1}{\frac{1}{0.1} + \frac{1}{0.8} - 1} = 0.0975$$

$$K_4 = K_5 = \frac{\epsilon_{12}\,FA}{3600} = \frac{(0.0975)(1.0)(1.0)}{3600} = 2.71 \times 10^{-5}$$

Capacitors

For the insulation nodes:

$$C2 = C3 = C7 = C8 = C9 = \rho c \delta A = (3.0)(0.25)\left(\frac{1}{12}\right)(1) = 0.0625\ \frac{Btu}{{}^\circ F}$$

$$C4 = C6 = \rho c \delta A = (3.0)(0.25)\left(\frac{0.5}{12}\right)(1) = 0.0313\ \frac{Btu}{{}^\circ F}$$

For the equipment and mounting plate, assuming an average specific heat of 0.15 for the electronic components:

$$C5 = 25(0.15) + \rho c \delta A = 3.75 + 175(0.2)\left(\frac{0.1}{12}\right)(1) = 4.04 \ \frac{\text{Btu}}{°\text{F}}$$

No capacitors are required for nodes 1 and 10 since their temperatures are specified by a curve.

RC Products

Although not required by the computer, the user should be familiar with the computation of RC products. To illustrate the procedure, the RC products are computed for this example.

Nodes 2 and 9:

$$(RC)_2 = (RC)_9 = \frac{C2}{\frac{1}{R1} + \frac{1}{R2}} = \frac{0.0625}{\frac{1}{2500} + \frac{1}{5000}} = 104 \text{ sec}$$

Nodes 3, 7, and 8:

$$(RC)_3 = (RC)_7 = (RC)_8 = \frac{C3}{\frac{1}{R2} + \frac{1}{R3}} = \frac{0.0625}{\frac{1}{5000}} = 313 \text{ sec}$$

The RC products of nodes 4, 5, and 6 depend on the value of the radiation resistors 4 and 5, which are temperature dependent. Assuming all three nodes are at 150°F:

$$R4 = R5 = \frac{1.0}{\sigma K_4 \left[(T_4 + 460)^2 + (T_5 + 460)^2\right]\left[(T_4 + 460) + (T_5 + 460)\right]}$$

$$= \frac{1}{(0.1713 \times 10^{-8})(2.71 \times 10^{-5})(4)(610)^3} = 23700 \ \frac{\text{sec } °\text{F}}{\text{Btu}}$$

Nodes 4 and 6:

$$(RC)_4 = (RC)_6 = \frac{C4}{\frac{1}{R3} + \frac{1}{R4}} = \frac{0.0313}{\frac{1}{5000} + \frac{1}{23700}} = 129 \text{ sec}$$

Node 5:

$$(RC)_5 = \frac{C5}{\frac{1}{R4} + \frac{1}{R5}} = \frac{4.04}{\frac{1}{23700} + \frac{1}{23700}} = 47900 \text{ sec}$$

Initially, nodes 2 and 9 have the lowest RC products, but if the temperatures of nodes 4, 5, or 6 become quite large, resistors 4 and 5 decrease and it is possible that later in the problem $(RC)_{min}$ would appear at either node 4 or 6. If several nodes all have the same RC product, the one with the lowest designation number is used to determine the computing interval. In this example, however, the initial $(RC)_{min}$ is greater than the specified 100-sec printing interval and therefore

$$\Delta\theta = (0.25)(100) = 25 \text{ sec}$$

Later in the problem, if $(RC)_{min}$ becomes less than 100 sec, the computing interval will be 0.25 $(RC)_{min}$.

Curves

Two curves are required, namely the variation of T(1) and T(10) with time. These temperature histories are assumed to be known and are input as curves 1 and 10, respectively. These curves are shown in Figure 6-2.

Time Block

The following times are assumed:

Print interval = 100 sec

Final time = 2300 sec

Initial time = 0 sec

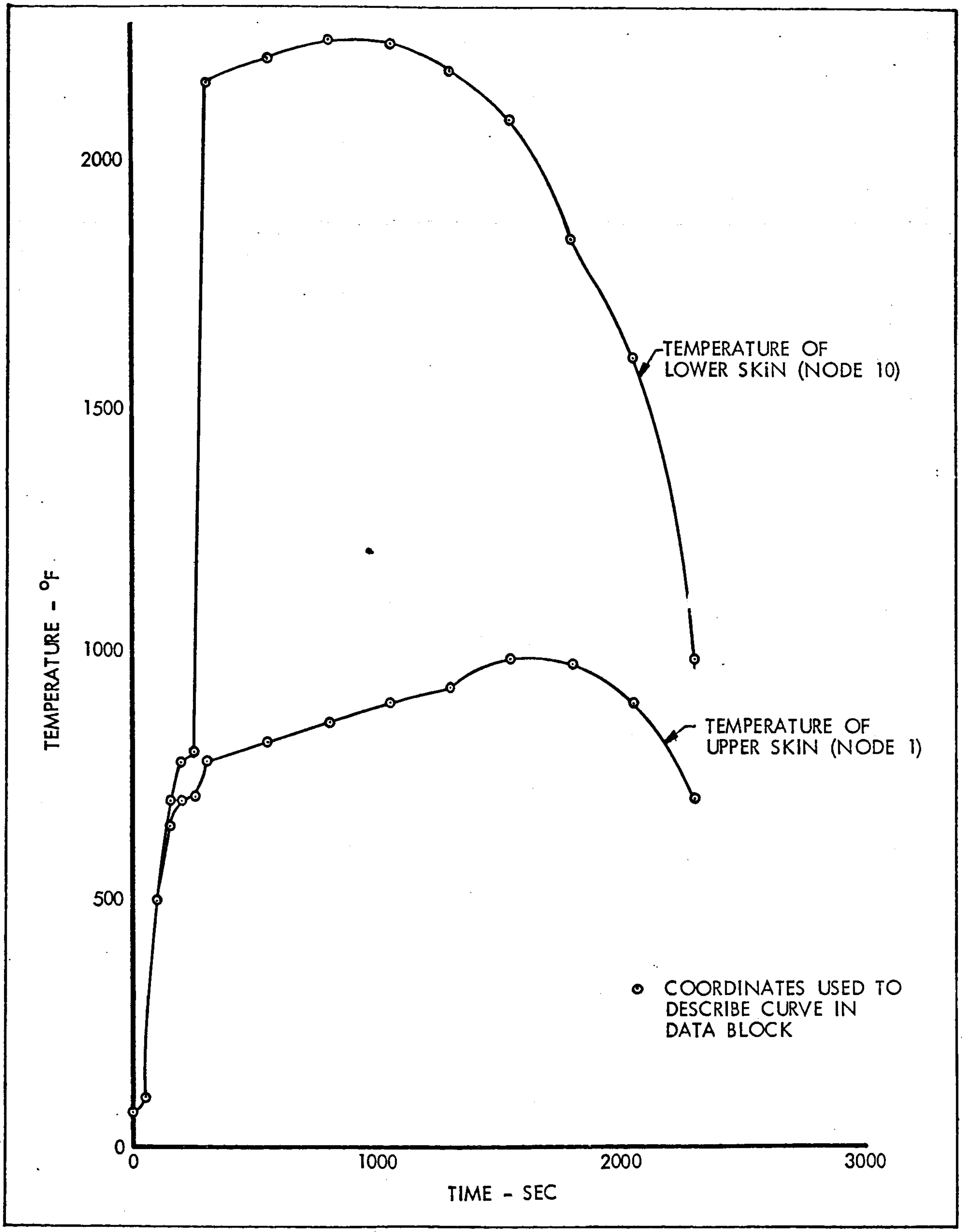


Figure 6-2. Temperature History of Upper and Lower Surfaces

FUNCT Subroutine

The five constants required for the EQUIVALENCE declaration are as follows:

j = 1 + greatest temperature designation number = 1 + 10 = 11
k = j + greatest temperature designation number = 11 + 10 = 21
m = k + greatest resistor designation number = 21 + 9 = 30
n = m + greatest capacitor designation number = 30 + 9 = 39
p = n + greatest capacitor designation number = 39 + 9 = 48

The following function callouts are required:

(a) Variation of T(1) and T(10) with time

T(1) = LIN (M(1),1)
T(10) = LIN (M(1),10)

(b) Radiation with constant K factor

R(4) = RAD (4, 5, 2.71E-5)
R(5) = RAD (5, 6, 2.71E-5)

(c) Heat input to node 5

Q(5) = 0.095

Two additional instructions TN(1) = T(1) and TN(10) = T(10) are necessary only if it is desired to print the temperatures of nodes 1 and 10 in the output. The reason is that, unless otherwise specified, the "new" temperatures of these nodes never change from their initial value of 70°F (since no capacitors are specified and hence no heat balance is performed). Then, at the end of each computing cycle, T(i) is set equal to TN(i) and the temperatures interpolated from curves 1 and 10 are replaced by the value 70°F, which is then printed in the output. Note that the heat balance on the nodes with capacitors is properly executed whether these additional instructions are added or not since the independent variable used in the heat balance is T(i), not TN(i).

PRINT Subroutine

The print format is set up to list the data in six full columns, with the corresponding time appearing by itself to the left of the first row of output. This is the standard output format described in Section V. The output specification includes all temperatures, radiation resistors 4 and 5, the minimum RC product, and the node at which $(RC)_{min}$ appears.

Computer Input for Example #1

The input for Example #1 is shown on data input sheets in Table 6-1. In addition to the previous instructions the following remarks are applicable:

1. The order in which nodes, resistors, capacitors, and curves are input is arbitrary.
2. The user's name and phone number should appear on the first temperature card to identify the output.
3. A liberal use of the space available for comments will help identify various information for future reference.

Computer Output

The computer output for Example #1 is shown in Table 6-2. The first two pages list the FUNCT subroutine and the PRINT subroutine. The third page lists the initial temperature, resistor, capacitor, data, and time blocks just as they appear on the input sheets (Table 6-1). The number 2266 added to the sequence number during keypunch is the Lockheed-California Company designation number for the Thermal Analyzer Program.

The answers are printed on the fourth and fifth pages of Table 6-2. As an example of how the output is read, the temperatures, etc., existing at 1500 sec are tabulated below.

Current time = 1500 sec

T(1) = 979.0°F
T(2) = 844.7°F
T(3) = 631.8°F
T(4) = 469.0°F
T(5) = 117.2°F
T(6) = 630.1°F
T(7) = 958.9°F
T(8) = 1383.4°F

TABLE 6-1

INPUT FOR EXAMPLE 1 (1 of 3)

FORM DX 7924.1

GENERAL PURPOSE DATA SHEET

LOCKHEED-CALIFORNIA COMPANY
A DIVISION OF LOCKHEED AIRCRAFT CORPORATION

PREPARED | NAME | DATE | TITLE | PAGE OF | JOB NO.
CHECKED | WO | EWA | GROUP

1–6	10	15	20	25	30	35	40	45	50	55	60	65	70–72 (ID)
CID		THERMAL ANALYZER EXAMPLE PRØBLEM ØNE											1000
DEC01	1		70.			NAME, PHØNE NØ.							1001
DEC01	10		70.			DATE							1002
NBK													1999
DEC01	1	1	2		2500.	RESISTØR BLØCK							2001
DEC02	2	2	3		5000.	3	3	4		5000.			2002
DEC02	4	4	5		0.	5	5	6		0.			2003
DEC02	6	6	7		5000.	7	7	8		5000.			2004
DEC02	8	8	9		5000.	9	9	10		2500.			2005
NBK													2999
DEC01	2		0.0625			CAPACITØR BLØCK							3001
DEC04	3		0.0625	4		0.0313	6		0.0313	7		0.0625	3002
DEC02	8		0.0625	9		0.0625							3003
NBK													3999
DEC	1	TEMPERATURE ØF UPPER SURFACE VS TIME											4001
DEC06		0.		70.		50.		100.		100.		500.	4002
		150.		650.		200.		700.		250.		710.	4003
		300.		780.		550.		820.		800.		860.	4004
		1050.		900.		1300.		930.		1550.		990.	4005
DEC06		1800.		980.		2050.		900.		2300.		710.	4006
DEC01		0.											4007
DEC	-1												4008
DEC	10	TEMPERATURE ØF LØWER SURFACE VS TIME											4009
DEC06		0.		70.		50.		100.		100.		500.	4010
DEC06		150.		700.		200.		770.		250.		800.	4011

LOCKHEED
CALIFORNIA COMPANY

TABLE 6-1

INPUT FOR EXAMPLE 1 (2 of 3)

GENERAL PURPOSE DATA SHEET — LOCKHEED-CALIFORNIA COMPANY, A DIVISION OF LOCKHEED AIRCRAFT CORPORATION

PAGE ___ OF ___ | JOB NO. | PREPARED | NAME | DATE | TITLE | CHECKED | WO | EWA | GROUP

FORM DX 7924.1

```
SEQ.  1                                                                    ID
DEC06        300.      2150.      550.      2200.      800.      2240.    4012
DEC06       1050.      2230.     1300.      2180.     1550.      2080.    4013
DEC06       1800.      1840.     2050.      1600.     2300.       990.    4014
DEC01          0.                                                        4015
DEC     -10                                                              4016
                                                                         4999
NBK                                                                      5001
DEC01        100.                PRINT INTERVAL                          5002
DEC01       2300.                FINAL TIME                              5003
DEC01          0.                INITIAL TIME                            5999
NBK
0001 $IBFTC FUNCT    NØDECK,NØREF
0002       SUBRØUTINE FUNCT
0003       CØMMØN P,M
0004       DIMENSIØN P(16000),M(16),TN(1),T(1),R(1),C(1),Q(1),RC(1)
0005       EQUIVALENCE (P(1),TN(1)),(P(11),T(1)),(P(21),R(1)),(P(30),C(1)),
0006      1 (P(39),Q(1)),(P(48),RC(1))
0007       REAL M,LIN
0008       T(1)=LIN(M(1),1)
0009       T(10)=LIN(M(1),10)
0010       R(4)=RAD(4,5,2.71E-5)
0011       R(5)=RAD(5,6,2.71E-5)
0012       Q(5)=.095
0013       TN(1)=T(1)
0014       TN(10)=T(10)
0015       RETURN
```


LOCKHEED CALIFORNIA COMPANY

TABLE 6-1

INPUT FOR EXAMPLE 1 (3 of 3)

GENERAL PURPOSE DATA SHEET — LOCKHEED-CALIFORNIA COMPANY, A DIVISION OF LOCKHEED AIRCRAFT CORPORATION

PREPARED | NAME | DATE | TITLE | PAGE OF
CHECKED | | | WO EWA | JOB NO. | GROUP

FORM DX 7924.1

```
SEQ.
77 80
0016        END

0101 $IBFTC PRINT    NØDECK,NØREF
0102        SUBRØUTINE PRINT
0103        CØMMØN P,M
0104        DIMENSIØN P(16000),M(16),TN(1),T(1),R(1),C(1),Q(1),RC(1)
0105        EQUIVALENCE (P(1),TN(1)),(P(11),T(1)),(P(21),R(1)),(P(30),C(1)),
0106       1 (P(39),Q(1)),(P(48),RC(1))
0107        REAL M
0108        WRITE (6,10) M(1),T(1),T(2),T(3),T(4),T(5),T(6),
0109       1 T(7),T(8),T(9),T(10),R(4),R(5),
0110       2 M(6),M(9)
0111  10    FØRMAT (1H0,7E18.8/(19X,6E18.8))
0112        RETURN
0113        END
```


LOCKHEED CALIFORNIA COMPANY

TABLE 6-2. COMPUTER OUTPUT FOR EXAMPLE 1

```
            FUNCT                                                   02/11/65
     EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

 SUBROUTINE FUNCT                                                22660002
 COMMON P,M                                                      22660003
 DIMENSION P(16000),M(16),TN(1),T(1),R(1),C(1),Q(1),RC(1)        22660004
 EQUIVALENCE (P(1),TN(1)),(P(11),T(1)),(P(21),R(1)),(P(30),C(1)), 22660005
1 (P(39),Q(1)),(P(48),RC(1))                                     22660006
 REAL M,LIN                                                      22660007
 T(1)=LIN(M(1),1)                                                22660008    ,1
 T(10)=LIN(M(1),10)                                              22660009    ,2
 R(4)=RAD(4,5,2.71E-5)                                           22660010    ,3
 R(5)=RAD(5,6,2.71E-5)                                           22660011    ,4
 Q(5)=.095                                                       22660012    ,5
 TN(1)=T(1)                                                      22660013    ,6
 TN(10)=T(10)                                                    22660014    ,7
 RETURN                                                          22660015    ,8
 END                                                             22660016    ,9
```

TABLE 6-2. (CONTINUED)

```
                 PRINT                                                 02/11/65
          EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

      SUBROUTINE PRINT                                                 22660102
      COMMON P,M                                                       22660103
      DIMENSION P(16000),M(16),TN(1),T(1),R(1),C(1),Q(1),RC(1)         22660104
      EQUIVALENCE (P(1),TN(1)),(P(11),T(1)),(P(21),R(1)),(P(30),C(1)), 22660105
     1 (P(39),Q(1)),(P(48),RC(1))                                      22660106
      REAL M                                                           22660107
      WRITE (6,10) M(1),T(1),T(2),T(3),T(4),T(5),T(6),                 22660108
     1 T(7),T(8),T(9),T(10),R(4),R(5),                                 22660109
     2 M(6),M(9)                                                       22660110   ,1   ,2   ,3
 10   FORMAT (1HO,7E18.8/(19X,6E18.8))                                 22660111
      RETURN                                                           22660112   ,4
      END                                                              22660113   ,5
```

TABLE 6-2. (CONTINUED)

```
                     THERMAL ANALYZER EXAMPLE PROBLEM ONE                          10002266
DEC 1    1        70.                   NAME,PHONE NO.                             10012266
DEC 1   10        70.                   DATE                                       10022266
NBK 0                                                                              19992266
DEC 1    1    1    2      2500.         RESISTOR BLOCK                             20012266
DEC 2    2    2    3      5000.         3    3    4      5000.                     20022266
DEC 2    4    4    5         0.         5    5    6         0.                     20032266
DEC 2    6    6    7      5000.         7    7    8      5000.                     20042266
DEC 2    8    8    9      5000.         9    9   10      2500.                     20052266
NBK 0                                                                              29992266
DEC 1    2       0.0625                 CAPACITOR BLOCK                            30012266
DEC 4    3       0.0625    4     0.0313     6     0.0313     7     0.0625          30022266
DEC 3    8       0.0625    9     0.0625     5       4.04                           30032266
NBK 0                                                                              39992266
DEC 0    1       TEMPERATURE OF UPPER SURFACE VS TIME                              40012266
DEC 6         0.        70.        50.        100.        100.        500.         40022266
DEC 6       150.       650.       200.        700.        250.        710.         40032266
DEC 6       300.       780.       550.        820.        800.        860.         40042266
DEC 6      1050.       900.      1300.        930.       1550.        990.         40052266
DEC 6      1800.       980.      2050.        900.       2300.        710.         40062266
DEC 1         0.                                                                   40072266
DEC 0   -1                                                                         40082266
DEC 0   10       TEMPERATURE OF LOWER SURFACE VS TIME                              40092266
DEC 6         0.        70.        50.        100.        100.        500.         40102266
DEC 6       150.       700.       200.        770.        250.        800.         40112266
DEC 6       300.      2150.       550.       2200.        800.       2240.         40122266
DEC 6      1050.      2230.      1300.       2180.       1550.       2080.         40132266
DEC 6      1800.      1840.      2050.       1600.       2300.        990.         40142266
DEC 1         0.                                                                   40152266
DEC 0  -10                                                                         40162266
NBK 0                                                                              49992266
DEC 1       100.                        PRINT INTERVAL                             50012266
DEC 1      2300.                        FINAL TIME                                 50022266
DEC 1         0.                        INITIAL TIME                               50032266
NBK 0                                                                              59992266
```

TABLE 6-2. (CONTINUED)

0.	0.70000000E 02	0.70000000E 02	0.70000000E 02	0.70000000E 02	0.70000000E 02	0.70000000E 02
	0.70000000E 02	0.70000000E 02	0.70000000E 02	0.70000000E 02	0.36255127E 05	0.36255127E 05
	0.09999999E 03	0.				
0.09999999E 03	0.50000000E 03	0.17066107E 03	0.73934828E 02	0.70204655E 02	0.72350296E 02	0.70071597E 02
	0.70071997E 02	0.73929887E 02	0.17066099E 03	0.50000000E 03	0.36067738E 05	0.36070864E 05
	0.09999999E 03	0.				
0.20000000E 03	0.70000000E 03	0.37191358E 03	0.12153303E 03	0.80528519E 02	0.74699096E 02	0.70925584E 02
	0.75395026E 02	0.12358608E 03	0.39688206E 03	0.77000000E 03	0.35262220E 05	0.35789173E 05
	0.09999999E 03	0.				
0.30000000E 03	0.78000000E 03	0.48854344E 03	0.19034791E 03	0.11440361E 03	0.77060451E 02	0.76948616E 02
	0.94762371E 02	0.20600600E 03	0.81977893E 03	0.21500000E 04	0.32368295E 05	0.35134542E 05
	0.09999999E 03	0.				
0.40000000E 03	0.79600000E 03	0.56403683E 03	0.26024413E 03	0.16290020E 03	0.79459974E 02	0.93392884E 02
	0.13894192E 03	0.39682660E 03	0.12989896E 04	0.21700000E 04	0.28362109E 05	0.33649234E 05
	0.09999999E 03	0.				
0.50000000E 03	0.81200000E 03	0.61122482E 03	0.32316525E 03	0.21530426E 03	0.81929317E 02	0.12988888E 03
	0.21740769E 03	0.59247985E 03	0.15123849E 04	0.21900000E 04	0.24508080E 05	0.30707234E 05
	0.09999999E 03	0.				
0.60000000E 03	0.82800000E 03	0.64719986E 03	0.37781873E 03	0.26415143E 03	0.84507910E 02	0.18766356E 03
	0.31353240E 03	0.75350880E 03	0.16320350E 04	0.22080000E 04	0.21381519E 05	0.26432671E 05
	0.09999999E 03	0.				
0.70000000E 03	0.84400000E 03	0.67766861E 03	0.42498742E 03	0.30645786E 03	0.87238984E 02	0.25820921E 03
	0.41440805E 03	0.88359604E 03	0.17115814E 04	0.22239999E 04	0.19012549E 05	0.21955337E 05
	0.09999999E 03	0.				
0.80000000E 03	0.86000000E 03	0.70473283E 03	0.46559777E 03	0.34186970E 03	0.90165205E 02	0.33156329E 03
	0.51290637E 03	0.99115331E 03	0.17716763E 04	0.22400000E 04	0.17245023E 05	0.18124875E 05
	0.09999999E 03	0.				
0.90000000E 03	0.87600000E 03	0.72929558E 03	0.50053615E 03	0.37113343E 03	0.93321960E 02	0.40028416E 03
	0.60512380E 03	0.10821043E 04	0.18146640E 04	0.22360000E 04	0.15913963E 05	0.15190412E 05
	0.09999999E 03	0.				
0.09999999E 04	0.89200000E 03	0.75185104E 03	0.53064865E 03	0.39530033E 03	0.96730053E 02	0.46034737E 03
	0.68867914E 03	0.11590212E 04	0.18459523E 04	0.22319999E 04	0.14891052E 05	0.13055789E 05
	0.09999999E 03	0.				
0.11000000E 04	0.90600000E 03	0.77230520E 03	0.55671351E 03	0.41538855E 03	0.10039252E 03	0.51050630E 03
	0.76230648E 03	0.12242496E 04	0.18687038E 04	0.22200000E 04	0.14085317E 05	0.11528993E 05
	0.09999999E 03	0.				
0.12000000E 04	0.91800000E 03	0.78994734E 03	0.57918097E 03	0.43222253E 03	0.10429703E 03	0.55128353E 03
	0.82572646E 03	0.12786131E 04	0.18810684E 04	0.22000000E 04	0.13434789E 05	0.10432640E 05
	0.09999999E 03	0.				

TABLE 6-2. (CONTINUED)

0.13000000E 04	0.93000000E 03	0.80582423E 03	0.59851721E 03	0.44638080E 03	0.10842127E 03	0.58393256E 03
	0.87917111E 03	0.13229030E 04	0.18873320E 04	0.21799999E 04	0.12899929E 05	0.96338193E 04
	0.09999999E 03	0.				
0.13999999E 04	0.95400000E 03	0.82426919E 03	0.61561422E 03	0.45840325E 03	0.11273802E 03	0.60982078E 03
	0.92334776E 03	0.13579357E 04	0.18831363E 04	0.21399999E 04	0.12452172E 05	0.90417093E 04
	0.09999999E 03	0.				
0.15000000E 04	0.97800000E 03	0.84476697E 03	0.63188629E 03	0.46906038E 03	0.11721960E 03	0.63016707E 03
	0.95898052E 03	0.13834853E 04	0.18707935E 04	0.20999999E 04	0.11973185E 05	0.85025568E 04
	0.98548014E 02	0.59999999E 01				
0.16000000E 04	0.98800000E 03	0.86272401E 03	0.64769918E 03	0.47901399E 03	0.12184091E 03	0.64589774E 03
	0.98677257E 03	0.14004283E 04	0.18477961E 04	0.20320000E 04	0.11630581E 05	0.81892129E 04
	0.97171213E 02	0.59999999E 01				
0.16999999E 04	0.98400000E 03	0.87081108E 03	0.66133881E 03	0.48824015E 03	0.12657857E 03	0.65775389E 03
	0.10072728E 04	0.14064947E 04	0.18015420E 04	0.19360000E 04	0.11315288E 05	0.79487673E 04
	0.96069537E 02	0.59999999E 01				
0.18000000E 04	0.98000000E 03	0.87445297E 03	0.67177222E 03	0.49600982E 03	0.13140851E 03	0.66609942E 03
	0.10202923E 04	0.14008209E 04	0.17438259E 04	0.18400000E 04	0.11042460E 05	0.77686273E 04
	0.95216984E 02	0.59999999E 01				
0.19000000E 04	0.94800000E 03	0.86732227E 03	0.67864677E 03	0.50202907E 03	0.13630372E 03	0.67109285E 03
	0.10259638E 04	0.13852209E 04	0.16797774E 04	0.17439999E 04	0.10816120E 05	0.76421797E 04
	0.94604029E 02	0.59999999E 01				
0.20000000E 04	0.91600000E 03	0.85182516E 03	0.68050308E 03	0.50578301E 03	0.14123559E 03	0.67298458E 03
	0.10248831E 04	0.13615756E 04	0.16115524E 04	0.16480000E 04	0.10644735E 05	0.75631223E 04
	0.94214528E 02	0.59999999E 01				
0.20999999E 04	0.86200000E 03	0.82781392E 03	0.67755535E 03	0.50694314E 03	0.14617427E 03	0.67211336E 03
	0.10178459E 04	0.13308184E 04	0.15248423E 04	0.14780000E 04	0.10533698E 05	0.75251579E 04
	0.94025739E 02	0.59999999E 01				
0.22000000E 04	0.78600000E 03	0.78578499E 03	0.66791797E 03	0.50523114E 03	0.15109108E 03	0.66872783E 03
	0.10047156E 04	0.12847551E 04	0.13827737E 04	0.12340000E 04	0.10486240E 05	0.75238769E 04
	0.94019347E 02	0.59999999E 01				
0.23000000E 04	0.71000000E 03	0.73511326E 03	0.65068914E 03	0.49975958E 03	0.15595522E 03	0.66234028E 03
	0.98326895E 03	0.12188146E 04	0.12148283E 04	0.99000000E 03	0.10521859E 05	0.75660551E 04
	0.94229066E 02	0.59999999E 01				

$T(9) = 1870.7°F$

$T(10) = 2100.0°F$

$R(4) = 11973. sec°F/Btu$

$R(5) = 8502. sec°F/Btu$

$(RC)_{min} = 98.5$ sec

$(RC)_{min}$ occurs at node 6

Prior to 1500 sec $(RC)_{min}$ is greater than 100 sec so the computing interval is based on the specified 100-sec print interval. As a result, the program prints 100. for $(RC)_{min}$ and 0. for the node number.

EXAMPLE #2 - Temperature response of a satellite equipment bay from launch through the first orbit.

Problem Description

The electronic equipment rack of an earth-orbiting satellite is shown in Figure 6-3. A thermal analysis of this bay is to be performed for the time from launch through the first orbit, assuming an adiabatic interface with the rest of the vehicle. The bay consists of two intersecting aluminum webs on which three electronic components are mounted, and an aluminum outer shell. The heat dissipation of two of these components is constant, while that of the third is periodic with time.

Two sets of external heating curves are required: (1) a curve showing the ascent heating up to the time of orbit insertion, and (2) six periodic curves showing the incident orbital radiation. The ascent heating pulse is estimated in this example. The orbital heating curves were obtained from the Orbital Radiation Program (Ref. 4) for an earth-oriented horizontal cylinder in a 115-mile circular orbit. A noon launch at an inclination angle of 32.5 degrees and a zenith angle of 180 degrees at the center of the equipment bay are assumed.

The initial values of surface emissivity and solar absorptivity will be changed in a restart. In addition, certain items will be plotted to demonstrate the plotting routine and a max-min search will be conducted for some of the more important nodes.

The physical geometry and equivalent thermal network are shown in Figure 6-3.

Since component temperatures are of primary interest here, the various capacities are assumed to be concentrated at points corresponding to equipment locations. The capacitance of the flush-mounted equipment is lumped with that of the adjacent web. The bracket-mounted equipment is represented by a separate node connected by a conduction resistor to the adjacent web. At the juncture between the two webs, a string of zero capacitance nodes (designated by ⊗ and often referred to as "dummy" nodes) are used to effect a connection between webs. Perfect thermal contact at the web juncture, and at the web-shell intersection is assumed in this example. Also, the placement of shell nodes at the web-shell intersection implies that the external heating (or cooling) rates are much greater than heat losses by conduction to the vehicle interior.

The following material properties are assumed:

		Property	Aluminum
k	=	thermal conductivity, Btu/hr ft °F	70.
c	=	specific heat, Btu/lb°F	0.20
ρ	=	density, lb/ft^3	175.

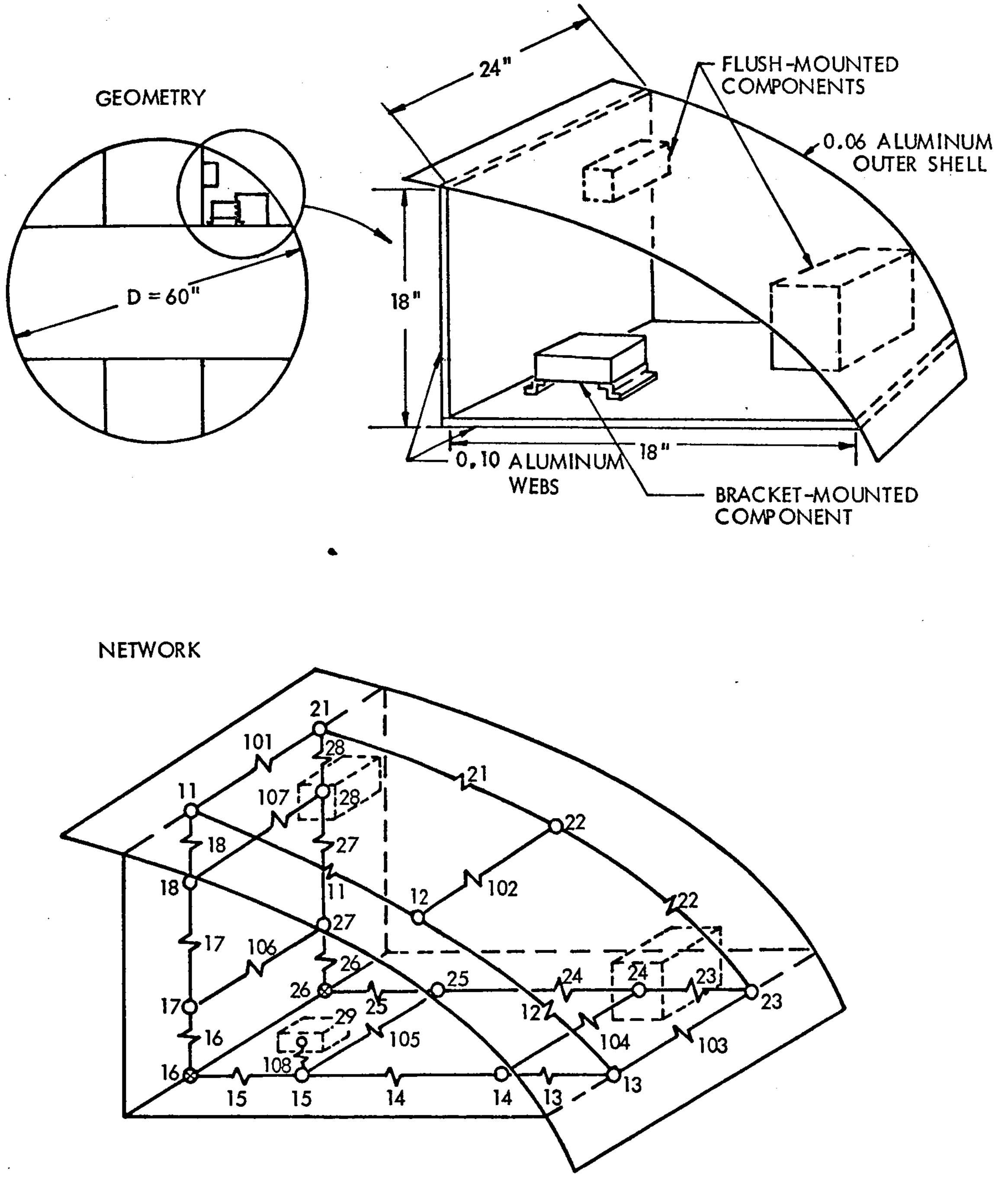
Conduction Resistors

On the webs

$$R14 = R17 = R24 = R27 = \frac{3600\,\delta}{kA} = \frac{3600\left(\frac{9}{12}\right)}{(70)\left(\frac{0.10}{12}\right)\left(\frac{12}{12}\right)} = 4620\ \frac{\sec{}^\circ F}{Btu}$$

$$R13 = R15 = R16 = R18 = R23 = R25 = R26 = R28 = \frac{R14}{2} = 2310\ \frac{\sec{}^\circ F}{Btu}$$

$$R104 = R105 = R106 = R107 = \frac{3600\,\delta}{kA} = \frac{3600\left(\frac{12}{12}\right)}{(70)\left(\frac{0.10}{12}\right)\left(\frac{9}{12}\right)} = 8240\ \frac{\sec{}^\circ F}{Btu}$$



NOTE: THE DESIGNATION NUMBER OF EACH SHELL TO SPACE RADIATION RESISTOR (NOT SHOWN) IS 1000 + THE SHELL NODE NUMBER. SPACE IS NODE 1.

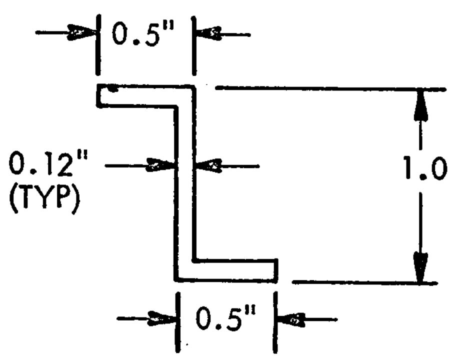
Figure 6-3. Physical Geometry and Corresponding Thermal Network for Example Problem 2

On the shell

$$R11 = R12 = R21 = R22 = \frac{3600\,\delta}{kA} = \frac{3600\left(\frac{13}{12}\right)}{(70)\left(\frac{0.06}{12}\right)\left(\frac{12}{12}\right)} = 11100\ \frac{\sec{}^\circ F}{Btu}$$

$$R101 = R102 = R103 = \frac{3600\delta}{kA} = \frac{3600\ \frac{12}{12}}{(70)\left(\frac{0.06}{12}\right)\left(\frac{13}{12}\right)} = 9500\ \frac{\sec{}^\circ F}{Btu}$$

To compute R108, assume that each aluminum Z section is 5 in. in length with a cross-section as follows:



Ignore the resistance through each flange but assume a contact conductance of 500 Btu/hr ft^2 °F between the lower flange and the web, and between the upper flange and the equipment.

$$R_{CONTACT} = \frac{3600}{kA} = \frac{3600}{(500)\left(\frac{0.5}{12}\right)\left(\frac{5}{12}\right)} = 415\ \frac{\sec{}^\circ F}{Btu}$$

The resistance through the web of the Z section is

$$R_{WEB} = \frac{3600\delta}{kA} = \frac{3600\left(\frac{1.0}{12}\right)}{70\left(\frac{0.12}{12}\right)\left(\frac{5}{12}\right)} = 1030\ \frac{\sec{}^\circ F}{Btu}$$

The total resistance through one Z section is

$$R = 2R_{CONTACT} + R_{WEB} = 2\ (415) + 1030 = 1860\ \frac{\sec{}^\circ F}{Btu}$$

Since there are two of these in parallel

$$R108 = \frac{R}{2} = 930\ \frac{\sec{}^\circ F}{Btu}$$

Radiation Resistors

Internal radiation is ignored in this example; however, each shell node must have a radiation resistor to space. In the basic deck assume $\alpha_s = 0.8$ and $\epsilon = 0.8$. In the restart assume $\alpha_s = 0.4$ and $\epsilon = 0.7$.

Basic deck

$$K11 = K12 = K13 = K21 = K22 = K23 = \frac{\epsilon FA}{3600} = \frac{(0.8)\ (1)\left(\frac{13 \times 12}{144}\right)}{3600} = 2.40 \times 10^{-4}$$

Restart

$$K11 = K12 = K13 = K21 = K22 = K23 = \frac{(0.7)\ (1)\left(\frac{13 \times 12}{144}\right)}{3600} = 2.10 \times 10^{-4}$$

Since both α_s and ϵ are required in the shell heating functions (see below) these values, as well as the value of the radiation k factor, are stored in Table 103. The appropriate changes are made to Table 103 in the restart.

Capacitors

For the webs, excluding the capacitance of the electronic components

$$C14 = C15 = C17 = C18 = C24 = C25 = C27 = C28$$

$$= \rho c \delta A = (175)\ (0.2) \left(\frac{0.06 \times 12 \times 9}{1728}\right) = 0.131\ \frac{Btu}{{}^\circ F}$$

For the shell

$$C11 = C12 = C13 = C21 = C22 = C23$$

$$= \rho c \delta A = (175)\ (0.2)\left(\frac{0.10 \times 12 \times 13}{1728}\right) = 0.316\ \frac{\text{Btu}}{°\text{F}}$$

$$C16 = C26 = 0.$$

Assume that each of the three electronic components has a capacitance of 1.5 Btu/°F (including the mounting bracket for node 29). Therefore,

$$C29 = 1.5 \text{ Btu/°F}$$

$$C24 = 1.5 + 0.131 = 1.63 \text{ Btu/°F}$$

$$C28 = 1.5 + 0.131 = 1.63 \text{ Btu/°F}$$

<u>RC Products</u>

The RC products for web node 15 and shell node 21 are computed to estimate the computing interval for this problem. On the web

$$(RC)_{15} = \frac{C15}{\frac{1}{R105} + \frac{1}{R14} + \frac{1}{R15} + \frac{1}{R108}} = \frac{0.131}{\frac{1}{8240} + \frac{1}{4620} + \frac{1}{2310} + \frac{1}{930}} = 71 \text{ sec}$$

Since the network for node 15 is similar to that of the other web nodes with the addition of the small resistor, R108, it appears that the minimum RC product for the webs occurs at node 15.

Because shell node 21 has a radiation resistor attached to it, its RC product is temperature dependent. If, for example, T (21) = 250°F,

$$R\ (1021) = \frac{1}{\sigma K_{21} 4 [T\ (21) + 460.]^3}$$

$$= \frac{1}{(0.1713 \times 10^{-8})\ (2.4 \times 10^{-4})(4)(710)^3} = 1700\ \frac{\text{sec °F}}{\text{Btu}}$$

and

$$(RC)_{21} = \frac{C(21)}{\frac{1}{R21} + \frac{1}{R101} + \frac{1}{R28} + \frac{1}{R1021}} = \frac{0.316}{\frac{1}{11100} + \frac{1}{9500} + \frac{1}{2310} + \frac{1}{1700}} = 260 \text{ sec}$$

If T (21) = -100°F

$$R\ (1021) = \frac{1}{(0.1713 \times 10^{-8})\ (2.4 \times 10^{-4})(4)(360)^3} = 13000\ \frac{\text{sec°F}}{\text{Btu}}$$

and

$$(RC)_{21} = \frac{0.316}{\frac{1}{11100} + \frac{1}{9500} + \frac{1}{2310} + \frac{1}{13000}} = 448 \text{ sec}$$

It appears, therefore, that the minimum RC product will occur at node 15, and will have a value of about 71 sec. In this example problem M(7) is set at 0.6, causing a computing interval of about 43 sec. During ascent, however, the specified print interval is less than 43 seconds. Hence, during this time the computing interval is determined by the print interval rather than the (RC) min.

Curves

The following curves are required:

CURVE NO.	DESCRIPTION
101	Shell ascent heating (including radiation) from liftoff to the time of orbit insertion (700 sec). This one curve will be assumed applicable to all six shell nodes. The value of the heating rate for times equal to or greater than 700 sec is set at zero so this curve has no effect on the orbital analysis.
102	Print interval M(4) vs time.

CURVE NO.	DESCRIPTION
103	A table storing the constants α_s, ϵ, and the shell radiation K factor.
104	A table listing the nodes (11, 12, 24, 28, and 29) for the maximum-minimum search.
105	A table listing the nodes (11, 12, 24, 28 and 29) and time intervals for the plotting function.
106	A periodic curve showing the heat dissipation of the component at node 24.
111	A periodic curve showing the solar spectrum radiation to shell nodes 11 and 21.
211	A periodic curve showing the infra-red radiation to shell nodes 11 and 21.
112	A periodic curve showing the solar spectrum radiation to shell nodes 12 and 22.
212	A periodic curve showing the infra-red radiation to shell nodes 12 and 22.
113	A periodic curve showing the solar spectrum radiation to shell nodes 13 and 23.
213	A periodic curve showing the infra-red radiation to shell nodes 13 and 23.

Time Block

The following times are assumed:

Print interval = 5 sec for $0. \leq M(1) \leq 150.$,
10 sec for $150. < M(1) \leq 200.$, and
300 sec for $M(1) > 200.$

Final time = 6200 sec

Initial time = 0 sec

Restart Block

The only changes required are the values of α_s, ϵ, and K_{RAD} which are stored in Table 3. Note that this table must be changed in its entirety, including the table designation number.

FUNCT Subroutine

The following functions are required:

(a) An instruction to set M(7) = 0.6.

(b) An instruction to change the temperature of the shell radiation sink, node 1, from its value of 50°F applicable to ascent, to a value of -460°F, appropriate for the orbital phase. Only a slight inaccuracy results if this change is made instantaneously at 150 sec since the shell temperature is sufficiently large that the choice of sink temperature is relatively insignificant.

(c) Callouts for the ascent heating and radiation functions for the external shell. These instructions are combined in a DO loop. Since the radiation K factor is obtained from storage Table 103 a locating statement L103 = LOC(103) precedes the radiation callout.

(d) Callouts for the shell orbital heat inputs. A logical IF statement will precede these instructions so that they are skipped if time is less than 700 sec, the assumed orbit insertion time. The instructions will appear as

```
Q(11) = Q(11) + Area * Absorptivity * LIN (M(1), 111)
              + Area * Emissivity * LIN (M(1), 211)
etc.
```

The term Q(11) on the right side of the equation is required when more than one heat input is called out for a particular node, and it is desired that they be summed. Otherwise, each successive callout would merely replace the previous value. In this example problem, the term is actually not required since the first callout (item (C) above) specifies that Q(11) = 0. for M(1) > 700 sec.

(e) Callouts for the heat dissipation of nodes 24, 28, and 29.

(f) An instruction to obtain the print interval, M(4), from curve 102.

(g) The plotting function callout for nodes 11, 12, 24, 28, and 29.

(h) The max-min function callout for nodes 11, 12, 24, 28, and 29.

The five constants required for the EQUIVALENCE statement are:

$$j = 1 + 29 = 30$$

$$k = 30 + 29 = 59$$

$$m = 59 + 1023 = 1082$$

$$n = 1082 + 29 = 1111$$

$$p = 1111 + 29 = 1140$$

PRINT Subroutine

The output specification for this case consists of all temperatures, and the heat rejection at nodes 24, 28, and 29. Each of these items is given an appropriate label to illustrate the flexibility of the FORTRAN language in writing Format statements. The first line of output lists the current time, computing interval, minimum RC product, and the node at which the (RC) min appears.

Four prints are specified on each page of output. To accomplish this, the variable BLOCK is used to sum the number of prints and when BLOCK = 5., the program is instructed to eject, print the information contained on the CID card, and then set BLOCK = 1. to indicate the first print on that page. The remaining WRITE statements are then executed in a normal manner.

Computer Input

The input data for Example #2 is shown in Table 6-3. Note that curves 111, 211, 112, 212, 113, and 213 are missing since they are obtained directly as punched cards from the Orbital Radiation Program.

Computer Output

The computer output for Example #2 is shown in Table 6-4. The first two pages list the FUNCT and PRINT subroutines. The next five pages list the initial temperature, resistor, capacitor, and data blocks exactly as they appear on the input sheets (Table 6-3), with the addition of the heating rate curves obtained from the Orbital Radiation Program.

The answers for the basic case are printed on the next 14 pages. During the time that a 5 or 10 sec print interval is requested, the computing interval is six-tenths of the print interval. When a 300-sec print interval is requested, the computing interval is six-tenths of the $(RC)_{min}$ which, as anticipated, appears at node 15 and has a value of about 71 sec. The Maximum-Minimum output appears immediately following the answers. Zeros are printed for the maximum and minimum temperatures of nodes 28 and 29 since the temperatures increase monotonically. Four distinct maximums and minimums were recorded for shell nodes 11 and 12.

The new data required for the restart are printed following the Max-Min output, exactly as it appears on the input sheet. The succeeding pages show the answers for the restart and the Maximum-Minimum output. It appears that the changes in surface radiation properties has little effect on the temperatures of the electronic components.

The machine plotted temperatures are shown in Figures 6-4 and 6-5 for the basic case and restart, respectively. The information on the CID card is automatically printed on the top of each figure. The format used to identify the node numbers and symbols is also standard. The scales are selected by the program according to the overall range of temperatures and time.

TABLE 6-3

INPUT FOR EXAMPLE 2 (1 of 6)

FORM DX 7924.1

GENERAL PURPOSE DATA SHEET — LOCKHEED-CALIFORNIA COMPANY, A DIVISION OF LOCKHEED AIRCRAFT CORPORATION

PREPARED	NAME	DATE	TITLE	PAGE OF
CHECKED			WO / EWA	JOB NO. / GROUP

Cols 1–6	7–12	13–18	19–24	25–30	31–36	37–42	43–48	49–54	Text	69–72
CID									THERMAL ANALYZER EXAMPLE PROBLEM TWO	1000
DEC01	1		50.						NAME, PHONE NUMBER	1001
DEC02	11		70.	29	70.				DATE	1002
NBK										1999
DEC01	11	11	12	11100.					RESISTOR BLOCK	2001
DEC02	12	12	13	11100.	13	13	14	2310.		2002
	14	14	15	4620.	15	15	16	2310.		2003
	16	16	17	2310.	17	17	18	4620.		2004
	18	18	11	2310.	21	21	22	11100.		2005
	22	22	23	11100.	23	23	24	2310.		2006
	24	24	25	4620.	25	25	26	2310.		2007
DEC02	26	26	27	2310.	27	27	28	4620.		2008
DEC01	28	28	21	2310.						2009
DEC01	101	11	21	9500.						2010
INC	1	1	1	0.	2					2011
DEC01	104	14	24	8240.						2012
INC	1	1	1	0.	3					2013
DEC01	1011	11	1	0.						2014
INC	1	1	0	0.	2					2015
DEC01	1021	21	1	0.						2016
INC	1	1	0	0.	2					2017
DEC01	108	15	29	930.						2018
NBK										2999
DEC01	11	0.316							CAPACITOR BLOCK	3001
INC	1		0.	2						3002

LOCKHEED CALIFORNIA COMPANY

TABLE 6-3

INPUT FOR EXAMPLE 2 (2 of 6)

GENERAL PURPOSE DATA SHEET — LOCKHEED-CALIFORNIA COMPANY, A DIVISION OF LOCKHEED AIRCRAFT CORPORATION

PREPARED | NAME | DATE | TITLE | PAGE OF | CHECKED | WO | EWA | JOB NO. | GROUP

FORM DX 7924.1

Card	No.									ID
DEC04		14	0.131	15	0.131	17	0.131	18	0.131	3003
DEC03		21	0.316	22	0.316	23	0.316			3004
DEC04		24	1.63	25	0.131	27	0.131	28	1.63	3005
DEC03		29	1.5	16	0.	26	0.			3006
NBK										3999
DEC	101	SHELL HEATING DURING ASCENT								10101
DEC06		0.	0.	35.	0.	40.	.05			2
DEC06		60.	1.4	75.	1.5	85.	1.4			3
DEC06		115.	.27	130.	.023	700.	.023			4
DEC05		700.	0.	7000.	0.	0.				5
DEC	-101									10106
DEC	102	VARIABLE PRINT INTERVAL								10201
DEC06		0.	5.	145.	5.	145.	10.			2
DEC06		190.	10.	190.	300.	7000.	300.			3
DEC01		0.								4
DEC	-102									10205
DEC	103	SHELL SØLAR ABSØRPTIVITY, EMISSIVITY, AND RAD. K-FACTØR								10301
DEC03		.8	.8	2.4E-4						10302
DEC	-103									10303
DEC	104	NØDES FØR MAX-MIN FUNCTIØN								10401
DEC06		24.	28.	29.	11	12.	0.			10402
DEC	-104									10403
DEC	105	NØDES TØ BE PLØTTED								10501
DEC06		0.	6200.	150.	24.	28.	29.			10502
DEC03		11.	12.	0.						10503

LOCKHEED CALIFORNIA COMPANY

TABLE 6-3

INPUT FOR EXAMPLE 2 (3 of 6)

FORM DX 7924.1

GENERAL PURPOSE DATA SHEET — LOCKHEED-CALIFORNIA COMPANY, A DIVISION OF LOCKHEED AIRCRAFT CORPORATION

PREPARED | NAME | DATE | TITLE | PAGE OF | CHECKED | WO | EWA | JOB NO. | GROUP

```
DEC   -105                                                         10504
DEC    106    NØDE 24 HEAT DISSIPATIØN                             10601
PER       5290.                                                        2
DEC06        0.        0.     1700.        0.    1700.      .085       3
DEC06     3700.      .085     3700.        0.    5290.        0.       4
DEC01        0.                                                        5
DEC   -106                                                         10606
NBK                                                                 4999
DEC03        5.     6200.        0.    TIME BLØCK                   5001
NBK                                                                 5999
CID      RESTART FØR EXAMPLE PRØBLEM TWØ                            6001
DEC        5    NEW DATA                                               2
DEC      103    CHANGE TABLE                                           3
DEC03        .4        .7   2.1E-4                                     4
DEC   -103                                                             5
DEC        0                                                           6
NBK             END RESTART                                         6007
```


LOCKHEED CALIFORNIA COMPANY

TABLE 6-3

INPUT FOR EXAMPLE 2 (4 of 6)

FORM DX 7924.1

GENERAL PURPOSE DATA SHEET — LOCKHEED-CALIFORNIA COMPANY, A DIVISION OF LOCKHEED AIRCRAFT CORPORATION

PREPARED | NAME | DATE | TITLE | PAGE OF
CHECKED | WO | EWA | JOB NO. | GROUP

```
SEQ.  1    6
0001 #IBFTC FUNCT    NØDECK,NØREF
0002       SUBRØUTINE FUNCT
0003       CØMMØN P,M
0004       DIMENSIØN P(16000),M(16),TN(1),T(1),R(1),C(1),Q(1),RC(1)
0005       EQUIVALENCE (P(1),TN(1)),(P(30),T(1)),(P(59),R(1)),(P(1082),C(1)),
0006      1 (P(1111),Q(1)),(P(1140),RC(1))
0007       REAL M,LIN
0008       M(7)=0.6
0009       IF (M(1).EQ.150.) T(1)=-460.
0010       L103=LØC(103)
0011       DØ 1 I=1,3
0012       Q(I+10)=LIN(M(1),101)
0013       Q(I+20)=Q(I+10)
0014       R(I+1010)=RAD(10+I,1,P(L103+3))
0015       R(I+1020)=RAD(20+I,1,P(L103+3))
0016  1    CØNTINUE
0017       IF (M(1).LT.700.) GØ TØ 3
0018       DØ 2 I=1,3
0019       Q(I+10)=Q(I+10)+1.08*P(L103+1)*LIN(M(1),110+I)+1.08*P(L103+2)*
0020      1 LIN(M(1),210+I)
0021       Q(I+20)=Q(I+10)
0022  2    CØNTINUE
0023  3    Q(24)=LIN(M(1),106)
0024       Q(28)=0.03
0025       Q(29)=0.04
```


LOCKHEED CALIFORNIA COMPANY

TABLE 6-3

INPUT FOR EXAMPLE 2 (5 of 6)

FORM DX 7924.1

GENERAL PURPOSE DATA SHEET — LOCKHEED-CALIFORNIA COMPANY, A DIVISION OF LOCKHEED AIRCRAFT CORPORATION

PREPARED | NAME | DATE | TITLE | PAGE OF | JOB NO. | CHECKED | WO | EWA | GROUP

```
SEQ.
0026        M(4)=LIN(M(1),102)
0027        CALL TPLØT(105)
0028        CALL MMF(M(2),M(3),104,1)
0029        TN(1)=T(1)
0030        RETURN
0031        END

0101$IBFTC PRINT    NØDECK,NØREF
0102        SUBRØUTINE PRINT
0103        CØMMØN P,M,MISCEL(23),CID(12)
0104        DIMENSIØN P(16000),M(16),TN(1),T(1),R(1),C(1),Q(1),RC(1)
0105        EQUIVALENCE (P(1),TN(1)),(P(30),T(1)),(P(59),R(1)),(P(1082),C(1)),
0106       1 (P(1111),Q(1)),(P(1140),RC(1))
0107        REAL M
0108        IF (M(1).EQ.0.) BLØCK=4.
0109        BLØCK=BLØCK+1.
0110        IF (BLØCK.EQ.5.) WRITE (6,100) (CID(I),I=1,12)
0111        IF (BLØCK.EQ.5.) BLØCK=1.
0112        WRITE (6,101) M(1),M(15),M(6),M(9)
0113        WRITE (6,102)
0114       1 T(11),T(12),T(13),T(21),T(22),T(23),
0115       2 T(14),T(15),T(16),T(17),T(18),T(24),T(25),T(26),T(27),T(28)
```

TABLE 6-3

INPUT FOR EXAMPLE 2 (6 of 6)

FORM DX 7924.1

GENERAL PURPOSE DATA SHEET — LOCKHEED-CALIFORNIA COMPANY, A DIVISION OF LOCKHEED AIRCRAFT CORPORATION

PREPARED | NAME | DATE | TITLE | PAGE OF | CHECKED | JOB NO. | WO | EWA | GROUP

```
SEQ.
0116        WRITE (6,103) T(24),T(28),T(29),Q(24),Q(28),Q(29)
0117   100  FORMAT (1H1////1H ,12A6)
0118   101  FORMAT (///6H TIME=,F8.0,8H SECONDS,5X,20H COMPUTING INTERVAL=,
0119       1 F8.3,8H SECONDS,5X,8H(RC)MIN=,F8.3,16H SECONDS AT NODE,F6.0)
0120   102  FORMAT (53H0TEMPERATURE OF EXTERNAL SHELL NODES                ,
0121       1     3X,53H TEMPERATURE OF INTERNAL WEB NODES                  /
0122       2   127H NODE         11         12         13        21        22        23
0123       3   14        15        16        17        18        24        25        26        27        2
0124       48/8H TEMP.   ,6F7.0,7X,10F7.0)
0125   103  FORMAT (53H0EQUIPMENT TEMPERATURES AND HEATING RATES          /
0126       1          53H NODE                  24         28         29    /
0127       2          11H TEMP.      ,3F10.0/
0128       3          11H Q(BTU/SEC),3F10.4)
0129        RETURN
0130        END
```


LOCKHEED CALIFORNIA COMPANY

TABLE 6-4

COMPUTER OUTPUT FOR EXAMPLE 2

```
         FUNCT                                                              06/15/65
      EXTERNAL FORMULA NUMBER    -    SOURCE STATEMENT    -   INTERNAL FORMULA NUMBER(S)

      SUBROUTINE FUNCT                                                   22660002
      COMMON P,M                                                         22660003
      DIMENSION P(16000),M(16),TN(1),T(1),R(1),C(1),Q(1),RC(1)           22660004
      EQUIVALENCE (P(1),TN(1)),(P(30),T(1)),(P(59),R(1)),(P(1082),C(1)),22660005
     1 (P(1111),Q(1)),(P(1140),RC(1))                                    22660006
      REAL M,LIN                                                         22660007
      M(7)=0.6                                                           22660008   ,1
      IF (M(1).EQ.150.) T(1)=-460.                                       22660009   ,2    ,3    ,4
      L103=LOC(103)                                                      22660010   ,5
      DO 1 I=1,3                                                         22660011   ,6
      Q(I+10)=LIN(M(1),101)                                              22660012   ,7
      Q(I+20)=Q(I+10)                                                    22660013   ,8
      R(I+1010)=RAD(10+I,1,P(L103+3))                                    22660014   ,9
      R(I+1020)=RAD(20+I,1,P(L103+3))                                    22660015   ,10
1     CONTINUE                                                           22660016   ,11   ,12
      IF (M(1).LT.700.) GO TO 3                                          22660017   ,13   ,14   ,15
      DO 2 I=1,3                                                         22660018   ,16
      Q(I+10)=Q(I+10)+1.08*P(L103+1)*LIN(M(1),110+I)+1.08*P(L103+2)*     22660019
     1 LIN(M(1),210+I)                                                   22660020   ,17
      Q(I+20)=Q(I+10)                                                    22660021   ,18
2     CONTINUE                                                           22660022   ,19   ,20
3     Q(24)=LIN(M(1),106)                                                22660023   ,21
      Q(28)=0.03                                                         22660024   ,22
      Q(29)=0.04                                                         22660025   ,23
      M(4)=LIN(M(1),102)                                                 22660026   ,24
      CALL TPLOT(105)                                                    22660027   ,25
      CALL MMF(M(2),M(3),104,1)                                          22660028   ,26
      TN(1)=T(1)                                                         22660029   ,27
      RETURN                                                             22660030   ,28
      END                                                                22660031   ,29
```

TABLE 6-4 (Continued)

```
          PRINT                                                           06/15/65
     EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

       SUBROUTINE PRINT                                                  22660102
       COMMON P,M,MISCEL(23),CID(12)                                     22660103
       DIMENSION P(16000),M(16),TN(1),T(1),R(1),C(1),Q(1),RC(1)          22660104
       EQUIVALENCE (P(1),TN(1)),(P(30),T(1)),(P(59),R(1)),(P(1082),C(1)),22660105
      1 (P(1111),Q(1)),(P(1140),RC(1))                                   22660106
       REAL M                                                            22660107
       IF (M(1).EQ.0.) BLOCK=4.                                          22660108   ,1   ,2   ,3
       BLOCK=BLOCK+1.                                                    22660109   ,4
       IF (BLOCK.EQ.5.) WRITE (6,100) (CID(I),I=1,12)                    22660110   ,5   ,6   ,7   ,8   ,9   ,10
                                                                                    ,11
       IF (BLOCK.EQ.5.) BLOCK=1.                                         22660111   ,12  ,13  ,14
       WRITE (6,101) M(1),M(15),M(6),M(9)                                22660112   ,15  ,16  ,17
       WRITE (6,102)                                                     22660113
      1 T(11),T(12),T(13),T(21),T(22),T(23),                             22660114
      2 T(14),T(15),T(16),T(17),T(18),T(24),T(25),T(26),T(27),T(28)      22660115   ,18  ,19  ,20
       WRITE (6,103) T(24),T(28),T(29),Q(24),Q(28),Q(29)                 22660116   ,21  ,22  ,23
 100   FORMAT (1H1///1H ,12A6)                                           22660117
 101   FORMAT (///6H TIME=,F8.0,9H SECONDS,5X,20H COMPUTING INTERVAL=,   22660118
      1 F8.3,8H SECONDS,5X,8H(RC)MIN=,F8.3,16H SECONDS AT NODE,F6.0)     22660119
 102   FORMAT (53H0TEMPERATURE OF EXTERNAL SHELL NODES                  , 22660120
      1      3X,53H TEMPERATURE OF INTERNAL WEB NODES                   / 22660121
      2  127H NODE          11      12      13      21      22      23     22660122
      3  14      15      16      17      18      24      25      26      27 222660123
      48/9H TEMP.   ,6F7.0,7X,10F7.0)                                    22660124
 103   FORMAT (53H0EQUIPMENT TEMPERATURES AND HEATING RATES             / 22660125
      1          53H NODE              24           28          29      / 22660126
      2          11H TEMP.     ,3F10.0/                                  22660127
      3          11H Q(BTU/SEC),3F10.4)                                  22660128
       RETURN                                                            22660129   ,24
       END                                                               22660130   ,25
```

TABLE 6-4 (Continued)

```
                  THERMAL ANALYZER EXAMPLE PROBLEM TWO                                10002266
DEC 1    1        50.                       NAME,PHONE NUMBER                         10012266
DEC 2   11        70.    29         70.     DATE                                      10022266
NBK 0                                                                                 19992266
DEC 1   11   11   12     11100.             RESISTOR BLOCK                            20012266
DEC 2   12   12   13     11100.    13   13   14      2310.                            20022266
DEC 2   14   14   15      4620.    15   15   16      2310.                            20032266
DEC 2   16   16   17      2310.    17   17   18      4620.                            20042266
DEC 2   18   18   11      2310.    21   21   22     11100.                            20052266
DEC 2   22   22   23     11100.    23   23   24      2310.                            20062266
DEC 2   24   24   25      4620.    25   25   26      2310.                            20072266
DEC 2   26   26   27      2310.    27   27   28      4620.                            20082266
DEC 1   28   28   21      2310.                                                       20092266
DEC 1  101   11   21      9500.                                                       20102266
INC 0    1    1    1        0.     2                                                  20112266
DEC 1  104   14   24      8240.                                                       20122266
INC 0    1    1    1        0.     3                                                  20132266
DEC 1 1011   11    1        0.                                                        20142266
INC 0    1    1    0        0.     2                                                  20152266
DEC 1 1021   21    1        0.                                                        20162266
INC 0    1    1    0        0.     2                                                  20172266
DEC 1  108   15   29       930.                                                       20182266
NBK 0                                                                                 29992266
DEC 1   11       0.316                          CAPACITOR BLOCK                       30012266
INC 0    1         0.    2                                                            30022266
DEC 4   14       0.131  15        0.131   17       0.131    18        0.131           30032266
DEC 3   21       0.316  22        0.316   23       0.316                              30042266
DEC 4   24        1.63  25        0.131   27       0.131    28         1.63           30052266
DEC 3   29         1.5  16           0.   26          0.                              30062266
NBK 0                                                                                 39992266
DEC 0  101      SHELL HEATING DURING ASCENT                                          101012266
DEC 6          0.          0.        35.         0.         40.        .05           101022266
DEC 6         60.         1.4        75.        1.5         85.        1.4           101032266
DEC 6        115.         .27       130.       .023        700.       .023           101042266
DEC 5        700.          0.      7000.         0.          0.                      101052266
DEC 0 -101                                                                           101062266
DEC 0  102      VARIABLE PRINT INTERVAL                                              102012266
DEC 6          0.          5.       145.         5.        145.        10.           102022266
DEC 6        190.         10.       190.       300.       7000.       300.           102032266
DEC 1          0.                                                                    102042266
DEC 0 -102                                                                           102052266
DEC 0  103      SHELL SOLAR ABSORPTIVITY,EMISSIVITY,AND RAD.K-FACTOR                 103012266
DEC 3          .8          .8     2.4E-4                                             103022266
DEC 0 -103                                                                           103032266
DEC 0  104      NODES FOR MAX-MIN FUNCTION                                           104012266
DEC 6         24.         28.        29.        11.         12.         0.           104022266
DEC 0 -104                                                                           104032266
DEC 0  105      NODES TO BE PLOTTED                                                  105012266
DEC 6          0.       6200.       150.        24.         28.        29.           105022266
DEC 3         11.         12.         0.                                             105032266
DEC 0 -105                                                                           105042266
DEC 0  106      NODE 24 HEAT DISSIPATION                                             106012266
PER 0      5290.                                                                     106022266
DEC 6          0.          0.      1700.         0.       1700.       .085           106032266
DEC 6       3700.        .085      3700.         0.       5290.         0.           106042266
DEC 1          0.                                                                    106052266
DEC 0 -106                                                                           106062266
DEC 0  111                                                   SOLAR SPEC              11101
PER 1 5289.7793                                                                      11102
DEC 2     0.         0.0202                                                          11103
```

TABLE 6-4 (Continued)

		THERMAL ANALYZER EXAMPLE PROBLEM TWO				10002266
DEC	2	220.4074	0.0139			11104
DEC	2	440.8149	0.0065			11105
DEC	2	661.2223	0.0024			11106
DEC	2	837.5482	0.0270			11107
DEC	2	852.2421	0.			11108
DEC	2	881.6299	0.			11109
DEC	2	1102.0372	0.			11110
DEC	2	1322.4447	0.			11111
DEC	2	1542.8521	0.			11112
DEC	2	1763.2595	0.			11113
DEC	2	1983.6670	0.			11114
DEC	2	2204.0744	0.			11115
DEC	2	2424.4819	0.			11116
DEC	2	2644.8895	0.			11117
DEC	2	2865.2968	0.			11118
DEC	2	3085.7042	0.			11119
DEC	2	3085.7042	0.			11120
DEC	2	3100.3978	0.0267			11121
DEC	2	3306.1117	0.0009			11122
DEC	2	3526.5191	0.0097			11123
DEC	2	3746.9265	0.0186			11124
DEC	2	3967.3338	0.0272			11125
DEC	2	4187.7412	0.0322			11126
DEC	2	4408.1487	0.0305			11127
DEC	2	4628.5561	0.0273			11128
DEC	2	4848.9635	0.0262			11129
DEC	2	5069.3709	0.0246			11130
DEC	2	5289.7783	0.0202			11131
DEC	2	0.	0.			11132
DEC	0-	111				11133
DEC	0	211			INFRA-RED	21101
PER	1	5289.7783				21102
DEC	2	0.	0.0202			21103
DEC	2	220.4074	0.0203			21104
DEC	2	440.8149	0.0205			21105
DEC	2	661.2223	0.0205			21106
DEC	2	837.5482	0.0203			21107
DEC	2	852.2421	0.0203			21108
DEC	2	881.6299	0.0203			21109
DEC	2	1102.0372	0.0196			21110
DEC	2	1322.4447	0.0200			21111
DEC	2	1542.8521	0.0201			21112
DEC	2	1763.2595	0.0201			21113
DEC	2	1983.6670	0.0197			21114
DEC	2	2204.0744	0.0196			21115
DEC	2	2424.4819	0.0191			21116
DEC	2	2644.8895	0.0190			21117
DEC	2	2865.2969	0.0191			21118
DEC	2	3085.7042	0.0197			21119
DEC	2	3085.7042	0.0197			21120
DEC	2	3100.3978	0.0197			21121
DEC	2	3306.1117	0.0197			21122
DEC	2	3526.5191	0.0201			21123
DEC	2	3746.9265	0.0201			21124
DEC	2	3967.3338	0.0197			21125
DEC	2	4187.7412	0.0198			21126
DEC	2	4408.1487	0.0203			21127
DEC	2	4628.5561	0.0205			21128
DEC	2	4848.9635	0.0205			21129

TABLE 6-4 (Continued)

```
               THERMAL ANALYZER EXAMPLE PROBLEM TWO                              10002266
DEC 2 5069.3709    0.0203                                                        21130
DEC 2 5289.7783    0.0202                                                        21131
DEC 2        0.        0.                                                        21132
DEC 0- 211                                                                       21133
DEC 0  112                                  SOLAR SPEC                           11201
PER 1 5289.7783                                                                  11202
DEC 2    0.        0.0197                                                        11203
DEC 2  220.4074    0.0136                                                        11204
DEC 2  440.8149    0.0064                                                        11205
DEC 2  661.2223    0.0001                                                        11206
DEC 2  837.5492    0.0232                                                        11207
DEC 2  852.2421    0.                                                            11208
DEC 2  881.6298    0.                                                            11209
DEC 2 1102.0372    0.                                                            11210
DEC 2 1322.4447    0.                                                            11211
DEC 2 1542.8521    0.                                                            11212
DEC 2 1763.2595    0.                                                            11213
DEC 2 1983.6670    0.                                                            11214
DEC 2 2204.0744    0.                                                            11215
DEC 2 2424.4819    0.                                                            11216
DEC 2 2644.8895    0.                                                            11217
DEC 2 2865.2968    0.                                                            11218
DEC 2 3085.7042    0.                                                            11219
DEC 2 3085.7042    0.                                                            11220
DEC 2 3100.3978    0.0229                                                        11221
DEC 2 3306.1117    0.0008                                                        11222
DEC 2 3526.5191    0.0094                                                        11223
DEC 2 3746.9265    0.0181                                                        11224
DEC 2 3967.3338    0.0266                                                        11225
DEC 2 4187.7412    0.0310                                                        11226
DEC 2 4408.1487    0.0296                                                        11227
DEC 2 4628.5561    0.0264                                                        11228
DEC 2 4848.9635    0.0256                                                        11229
DEC 2 5069.3709    0.0241                                                        11230
DEC 2 5289.7783    0.0197                                                        11231
DEC 2        0.        0.                                                        11232
DEC 0- 112                                                                       11233
DEC 0  212                                  INFRA-RED                            21201
PER 1 5289.7783                                                                  21202
DEC 2    0.        0.0197                                                        21203
DEC 2  220.4074    0.0197                                                        21204
DEC 2  440.8149    0.0200                                                        21205
DEC 2  661.2223    0.0200                                                        21206
DEC 2  837.5492    0.0198                                                        21207
DEC 2  852.2421    0.0198                                                        21208
DEC 2  881.6298    0.0198                                                        21209
DEC 2 1102.0372    0.0192                                                        21210
DEC 2 1322.4447    0.0194                                                        21211
DEC 2 1542.8521    0.0196                                                        21212
DEC 2 1763.2595    0.0196                                                        21213
DEC 2 1983.6670    0.0192                                                        21214
DEC 2 2204.0744    0.0192                                                        21215
DEC 2 2424.4819    0.0187                                                        21216
DEC 2 2644.8895    0.0186                                                        21217
DEC 2 2865.2968    0.0187                                                        21218
DEC 2 3085.7042    0.0192                                                        21219
DEC 2 3085.7042    0.0192                                                        21220
DEC 2 3100.3978    0.0192                                                        21221
DEC 2 3306.1117    0.0193                                                        21222
```

TABLE 6-4 (Continued)

```
               THERMAL ANALYZER EXAMPLE PROBLEM TWO                              10002266
DEC 2 3526.5191     0.0196                                                      21223
DEC 2 3746.9265     0.0195                                                      21224
DEC 2 3967.3338     0.0192                                                      21225
DEC 2 4187.7412     0.0193                                                      21226
DEC 2 4408.1487     0.0198                                                      21227
DEC 2 4628.5561     0.0200                                                      21228
DEC 2 4848.9635     0.0200                                                      21229
DEC 2 5069.3709     0.0197                                                      21230
DEC 2 5289.7783     0.0197                                                      21231
DEC 2          0.          0.                                                   21232
DEC 0- 212                                                                      21233
DEC 0  113                                     SOLAR SPEC                       11301
PER 1 5289.7783                                                                 11302
DEC 2    0.         0.0184                                                      11303
DEC 2  220.4074     0.0127                                                      11304
DEC 2  440.8149     0.0059                                                      11305
DEC 2  661.2223     0.0001                                                      11306
DEC 2  837.5482     0.0182                                                      11307
DEC 2  852.2421     0.                                                          11308
DEC 2  881.6298     0.                                                          11309
DEC 2 1102.0372     0.                                                          11310
DEC 2 1322.4447     0.                                                          11311
DEC 2 1542.8521     0.                                                          11312
DEC 2 1763.2595     0.                                                          11313
DEC 2 1983.6670     0.                                                          11314
DEC 2 2204.0744     0.                                                          11315
DEC 2 2424.4819     0.                                                          11316
DEC 2 2644.8895     0.                                                          11317
DEC 2 2865.2968     0.                                                          11318
DEC 2 3085.7042     0.                                                          11319
DEC 2 3085.7042     0.                                                          11320
DEC 2 3100.3978     0.0180                                                      11321
DEC 2 3306.1117     0.0007                                                      11322
DEC 2 3526.5191     0.0087                                                      11323
DEC 2 3746.9265     0.0168                                                      11324
DEC 2 3967.3338     0.0248                                                      11325
DEC 2 4187.7412     0.0285                                                      11326
DEC 2 4408.1487     0.0275                                                      11327
DEC 2 4628.5561     0.0245                                                      11328
DEC 2 4848.9635     0.0239                                                      11329
DEC 2 5069.3709     0.0226                                                      11330
DEC 2 5289.7783     0.0184                                                      11331
DEC 2          0.          0.                                                   11332
DEC 0- 113                                                                      11333
DEC 0  213                                     INFRA-RED                        21301
PER 1 5289.7783                                                                 21302
DEC 2    0.         0.0183                                                      21303
DEC 2  220.4074     0.0184                                                      21304
DEC 2  440.8149     0.0186                                                      21305
DEC 2  661.2223     0.0186                                                      21306
DEC 2  837.5482     0.0185                                                      21307
DEC 2  852.2421     0.0185                                                      21308
DEC 2  881.6298     0.0185                                                      21309
DEC 2 1102.0372     0.0179                                                      21310
DEC 2 1322.4447     0.0181                                                      21311
DEC 2 1542.8521     0.0182                                                      21312
DEC 2 1763.2595     0.0183                                                      21313
DEC 2 1983.6670     0.0179                                                      21314
DEC 2 2204.0744     0.0179                                                      21315
```

TABLE 6-4 (Continued)

```
                 THERMAL ANALYZER EXAMPLE PROBLEM TWO                     10002266
DEC 2 2424.4919     0.0174                                               21316
DEC 2 2644.8895     0.0173                                               21317
DEC 2 2865.2968     0.0174                                               21318
DEC 2 3085.7042     0.0179                                               21319
DEC 2 3085.7042     0.0179                                               21320
DEC 2 3100.3978     0.0179                                               21321
DEC 2 3306.1117     0.0179                                               21322
DEC 2 3526.5191     0.0183                                               21323
DEC 2 3746.9265     0.0182                                               21324
DEC 2 3967.3338     0.0178                                               21325
DEC 2 4187.7412     0.0180                                               21326
DEC 2 4408.1487     0.0185                                               21327
DEC 2 4628.5561     0.0186                                               21328
DEC 2 4848.9635     0.0186                                               21329
DEC 2 5069.3709     0.0184                                               21330
DEC 2 5289.7783     0.0183                                               21331
DEC 2          0.        0.                                              21332
DEC 0- 213                                                               21333
NBK 0                                                                     49992266
DEC 3          5.     6200.          0.     TIME BLOCK                    50012266
NBK 0                                                                     59992266
```


LOCKHEED
CALIFORNIA COMPANY

TABLE 6-4. (CONTINUED)

THERMAL ANALYZER EXAMPLE PROBLEM TWO 10002266

TIME= 0. SECONDS COMPUTING INTERVAL= 0. SECONDS (RC)MIN= 5.000 SECONDS AT NODE 0.

TEMPERATURE OF EXTERNAL SHELL NODES

NODE	11	12	13	21	22	23
TEMP.	70.	70.	70.	70.	70.	70.

TEMPERATURE OF INTERNAL WEB NODES

NODE	14	15	16	17	18	24	25	26	27	28
TEMP.	70.	70.	70.	70.	70.	70.	70.	70.	70.	70.

EQUIPMENT TEMPERATURES AND HEATING RATES

NODE	24	28	29
TEMP.	70.	70.	70.
Q(BTU/SEC)	0.	0.0300	0.0400

TIME= 5. SECONDS COMPUTING INTERVAL= 3.000 SECONDS (RC)MIN= 5.000 SECONDS AT NODE 0.

TEMPERATURE OF EXTERNAL SHELL NODES

NODE	11	12	13	21	22	23
TEMP.	70.	70.	70.	70.	70.	70.

TEMPERATURE OF INTERNAL WEB NODES

NODE	14	15	16	17	18	24	25	26	27	28
TEMP.	70.	70.	70.	70.	70.	70.	70.	70.	70.	70.

EQUIPMENT TEMPERATURES AND HEATING RATES

NODE	24	28	29
TEMP.	70.	70.	70.
Q(BTU/SEC)	0.	0.0300	0.0400

TIME= 10. SECONDS COMPUTING INTERVAL= 3.000 SECONDS (RC)MIN= 5.000 SECONDS AT NODE 0.

TEMPERATURE OF EXTERNAL SHELL NODES

NODE	11	12	13	21	22	23
TEMP.	70.	70.	70.	70.	70.	70.

TEMPERATURE OF INTERNAL WEB NODES

NODE	14	15	16	17	18	24	25	26	27	28
TEMP.	70.	70.	70.	70.	70.	70.	70.	70.	70.	70.

EQUIPMENT TEMPERATURES AND HEATING RATES

NODE	24	28	29
TEMP.	70.	70.	70.
Q(BTU/SEC)	0.	0.0300	0.0400

TIME= 15. SECONDS COMPUTING INTERVAL= 3.000 SECONDS (RC)MIN= 5.000 SECONDS AT NODE 0.

TEMPERATURE OF EXTERNAL SHELL NODES

NODE	11	12	13	21	22	23
TEMP.	70.	70.	70.	70.	70.	70.

TEMPERATURE OF INTERNAL WEB NODES

NODE	14	15	16	17	18	24	25	26	27	28
TEMP.	70.	70.	70.	70.	70.	70.	70.	70.	70.	70.

EQUIPMENT TEMPERATURES AND HEATING RATES

NODE	24	28	29
TEMP.	70.	70.	70.
Q(BTU/SEC)	0.	0.0300	0.0400

TABLE 6-4. (CONTINUED)

```
THERMAL ANALYZER EXAMPLE PROBLEM TWO                    10002266

TIME=     20. SECONDS      COMPUTING INTERVAL=   3.000 SECONDS     (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                         TEMPERATURE OF INTERNAL WEB NODES
NODE           11      12      13      21      22      23        14      15      16      17      18      24      25      26      27      28
TEMP.         70.     70.     70.     70.     70.     70.       70.     70.     70.     70.     70.     70.     70.     70.     70.     70.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                   24          28          29
TEMP.                 70.         70.         71.
Q(BTU/SEC)     0.          0.0300      0.0400


TIME=     25. SECONDS      COMPUTING INTERVAL=   3.000 SECONDS     (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                         TEMPERATURE OF INTERNAL WEB NODES
NODE           11      12      13      21      22      23        14      15      16      17      18      24      25      26      27      28
TEMP.         70.     70.     70.     70.     70.     70.       70.     70.     70.     70.     70.     70.     70.     70.     70.     70.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                   24          28          29
TEMP.                 70.         70.         71.
Q(BTU/SEC)     0.          0.0300      0.0400


TIME=     30. SECONDS      COMPUTING INTERVAL=   3.000 SECONDS     (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                         TEMPERATURE OF INTERNAL WEB NODES
NODE           11      12      13      21      22      23        14      15      16      17      18      24      25      26      27      28
TEMP.         70.     70.     70.     70.     70.     70.       70.     70.     70.     70.     70.     70.     70.     70.     70.     71.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                   24          28          29
TEMP.                 70.         71.         71.
Q(BTU/SEC)     0.          0.0300      0.0400


TIME=     35. SECONDS      COMPUTING INTERVAL=   3.000 SECONDS     (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                         TEMPERATURE OF INTERNAL WEB NODES
NODE           11      12      13      21      22      23        14      15      16      17      18      24      25      26      27      28
TEMP.         70.     69.     70.     70.     69.     70.       70.     70.     70.     70.     70.     70.     70.     70.     70.     71.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                   24          28          29
TEMP.                 70.         71.         71.
Q(BTU/SEC)     0.          0.0300      0.0400
```

TABLE 6-4. (CONTINUED)

```
THERMAL ANALYZER EXAMPLE PROBLEM TWO                  10002266

TIME=     40. SECONDS      COMPUTING INTERVAL=   3.000 SECONDS      (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                          TEMPERATURE OF INTERNAL WEB NODES
NODE        11     12     13     21     22     23               14     15     16     17     18     24     25     26     27     28
TEMP.      70.    70.    70.    70.    70.    70.              70.    70.    70.    70.    70.    70.    70.    70.    70.    71.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                24          28          29
TEMP.              70.         71.         71.
Q(BTU/SEC)    0.           0.0300      0.0400

TIME=     45. SECONDS      COMPUTING INTERVAL=   3.000 SECONDS      (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                          TEMPERATURE OF INTERNAL WEB NODES
NODE        11     12     13     21     22     23               14     15     16     17     18     24     25     26     27     28
TEMP.      75.    75.    75.    75.    75.    75.              70.    70.    70.    70.    70.    70.    70.    70.    70.    71.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                24          28          29
TEMP.              70.         71.         71.
Q(BTU/SEC)    0.           0.0300      0.0400

TIME=     50. SECONDS      COMPUTING INTERVAL=   3.000 SECONDS      (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                          TEMPERATURE OF INTERNAL WEB NODES
NODE        11     12     13     21     22     23               14     15     16     17     18     24     25     26     27     28
TEMP.      85.    85.    85.    85.    85.    85.              70.    70.    70.    70.    70.    70.    70.    70.    70.    71.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                24          28          29
TEMP.              70.         71.         71.
Q(BTU/SEC)    0.           0.0300      0.0400

TIME=     55. SECONDS      COMPUTING INTERVAL=   3.000 SECONDS      (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                          TEMPERATURE OF INTERNAL WEB NODES
NODE        11     12     13     21     22     23               14     15     16     17     18     24     25     26     27     28
TEMP.     100.   100.   100.   100.   100.   100.              70.    70.    70.    70.    70.    70.    70.    70.    70.    71.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                24          28          29
TEMP.              70.         71.         71.
Q(BTU/SEC)    0.           0.0300      0.0400
```

TABLE 6-4. (CONTINUED)

THERMAL ANALYZER EXAMPLE PROBLEM TWO 10002266

TIME= 60. SECONDS COMPUTING INTERVAL= 3.000 SECONDS (RC)MIN= 5.000 SECONDS AT NODE 0.

TEMPERATURE OF EXTERNAL SHELL NODES

NODE	11	12	13	21	22	23
TEMP.	121.	121.	121.	121.	121.	121.

TEMPERATURE OF INTERNAL WEB NODES

NODE	14	15	16	17	18	24	25	26	27	28
TEMP.	71.	70.	70.	70.	71.	70.	70.	70.	70.	71.

EQUIPMENT TEMPERATURES AND HEATING RATES

NODE	24	28	29
TEMP.	70.	71.	72.
Q(BTU/SEC)	0.	0.0300	0.0400

TIME= 65. SECONDS COMPUTING INTERVAL= 3.000 SECONDS (RC)MIN= 5.000 SECONDS AT NODE 0.

TEMPERATURE OF EXTERNAL SHELL NODES

NODE	11	12	13	21	22	23
TEMP.	142.	143.	142.	142.	143.	142.

TEMPERATURE OF INTERNAL WEB NODES

NODE	14	15	16	17	18	24	25	26	27	28
TEMP.	72.	70.	70.	70.	72.	70.	70.	70.	70.	71.

EQUIPMENT TEMPERATURES AND HEATING RATES

NODE	24	28	29
TEMP.	70.	71.	72.
Q(BTU/SEC)	0.	0.0300	0.0400

TIME= 70. SECONDS COMPUTING INTERVAL= 3.000 SECONDS (RC)MIN= 5.000 SECONDS AT NODE 0.

TEMPERATURE OF EXTERNAL SHELL NODES

NODE	11	12	13	21	22	23
TEMP.	164.	166.	164.	165.	166.	164.

TEMPERATURE OF INTERNAL WEB NODES

NODE	14	15	16	17	18	24	25	26	27	28
TEMP.	73.	70.	70.	70.	73.	70.	70.	70.	70.	72.

EQUIPMENT TEMPERATURES AND HEATING RATES

NODE	24	28	29
TEMP.	70.	72.	72.
Q(BTU/SEC)	0.	0.0300	0.0400

TIME= 75. SECONDS COMPUTING INTERVAL= 3.000 SECONDS (RC)MIN= 5.000 SECONDS AT NODE 0.

TEMPERATURE OF EXTERNAL SHELL NODES

NODE	11	12	13	21	22	23
TEMP.	187.	189.	187.	187.	189.	187.

TEMPERATURE OF INTERNAL WEB NODES

NODE	14	15	16	17	18	24	25	26	27	28
TEMP.	75.	71.	70.	70.	75.	70.	70.	70.	70.	72.

EQUIPMENT TEMPERATURES AND HEATING RATES

NODE	24	28	29
TEMP.	70.	72.	72.
Q(BTU/SEC)	0.	0.0300	0.0400

TABLE 6-4. (CONTINUED)

```
THERMAL ANALYZER EXAMPLE PROBLEM TWO                    10002266

TIME=     80. SECONDS      COMPUTING INTERVAL=   3.000 SECONDS     (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                          TEMPERATURE OF INTERNAL WEB NODES
NODE          11      12      13      21      22      23             14      15      16      17      18      24      25      26      27      28
TEMP.       208.    211.    208.    208.    211.    208.            77.     71.     70.     70.     77.     71.     70.     70.     70.     72.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                24          28          29
TEMP.              71.         72.         72.
Q(BTU/SEC)     0.          0.0300      0.0400


TIME=     85. SECONDS      COMPUTING INTERVAL=   3.000 SECONDS     (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                          TEMPERATURE OF INTERNAL WEB NODES
NODE          11      12      13      21      22      23             14      15      16      17      18      24      25      26      27      28
TEMP.       229.    233.    229.    229.    233.    229.            79.     71.     70.     70.     79.     71.     70.     70.     70.     72.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                24          28          29
TEMP.              71.         72.         72.
Q(BTU/SEC)     0.          0.0300      0.0400


TIME=     90. SECONDS      COMPUTING INTERVAL=   3.000 SECONDS     (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                          TEMPERATURE OF INTERNAL WEB NODES
NODE          11      12      13      21      22      23             14      15      16      17      18      24      25      26      27      28
TEMP.       247.    251.    247.    247.    251.    247.            81.     71.     70.     70.     81.     71.     70.     70.     70.     73.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                24          28          29
TEMP.              71.         73.         72.
Q(BTU/SEC)     0.          0.0300      0.0400


TIME=     95. SECONDS      COMPUTING INTERVAL=   3.000 SECONDS     (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                          TEMPERATURE OF INTERNAL WEB NODES
NODE          11      12      13      21      22      23             14      15      16      17      18      24      25      26      27      28
TEMP.       261.    267.    261.    261.    267.    261.            84.     71.     71.     70.     84.     71.     70.     70.     70.     73.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                24          28          29
TEMP.              71.         73.         72.
Q(BTU/SEC)     0.          0.0300      0.0400
```

TABLE 6-4. (CONTINUED)

```
THERMAL ANALYZER EXAMPLE PROBLEM TWO                    10002266

TIME=   100. SECONDS     COMPUTING INTERVAL=   3.000 SECONDS     (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                       TEMPERATURE OF INTERNAL WEB NODES
NODE       11     12     13     21     22     23            14     15     16     17     18     24     25     26     27     28
TEMP.    273.   279.   273.   272.   279.   272.           87.    71.    71.    70.    87.    71.    70.    70.    70.    73.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE               24         28         29
TEMP.             71.        73.        73.
Q(BTU/SEC)    0.          0.0300     0.0400

TIME=   105. SECONDS     COMPUTING INTERVAL=   3.000 SECONDS     (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                       TEMPERATURE OF INTERNAL WEB NODES
NODE       11     12     13     21     22     23            14     15     16     17     18     24     25     26     27     28
TEMP.    281.   289.   281.   281.   289.   280.           89.    71.    71.    71.    90.    72.    70.    70.    70.    74.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE               24         28         29
TEMP.             72.        74.        73.
Q(BTU/SEC)    0.          0.0300     0.0400

TIME=   110. SECONDS     COMPUTING INTERVAL=   3.000 SECONDS     (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                       TEMPERATURE OF INTERNAL WEB NODES
NODE       11     12     13     21     22     23            14     15     16     17     18     24     25     26     27     28
TEMP.    286.   295.   286.   286.   295.   285.           92.    72.    71.    71.    93.    72.    70.    70.    70.    74.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE               24         28         29
TEMP.             72.        74.        73.
Q(BTU/SEC)    0.          0.0300     0.0400

TIME=   115. SECONDS     COMPUTING INTERVAL=   3.000 SECONDS     (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                       TEMPERATURE OF INTERNAL WEB NODES
NODE       11     12     13     21     22     23            14     15     16     17     18     24     25     26     27     28
TEMP.    288.   298.   288.   288.   298.   287.           95.    72.    71.    71.    96.    72.    70.    70.    70.    74.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE               24         28         29
TEMP.             72.        74.        73.
Q(BTU/SEC)    0.          0.0300     0.0400
```

TABLE 6-4. (CONTINUED)

```
THERMAL ANALYZER EXAMPLE PROBLEM TWO                 10002266

TIME=    120. SECONDS     COMPUTING INTERVAL=   3.000 SECONDS     (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                        TEMPERATURE OF INTERNAL WEB NODES
NODE          11      12      13      21      22      23           14      15      16      17      18      24      25      26      27      28
TEMP.       288.    300.    288.    288.    300.    288.          98.     72.     71.     71.     99.     73.     70.     70.     70.     75.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                24          28          29
TEMP.              73.         75.         73.
Q(BTU/SEC)    0.          0.0300      0.0400


TIME=    125. SECONDS     COMPUTING INTERVAL=   3.000 SECONDS     (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                        TEMPERATURE OF INTERNAL WEB NODES
NODE          11      12      13      21      22      23           14      15      16      17      18      24      25      26      27      28
TEMP.       288.    300.    288.    287.    300.    287.         101.     72.     72.     71.    102.     73.     70.     70.     70.     75.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                24          28          29
TEMP.              73.         75.         73.
Q(BTU/SEC)    0.          0.0300      0.0400


TIME=    130. SECONDS     COMPUTING INTERVAL=   3.000 SECONDS     (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                        TEMPERATURE OF INTERNAL WEB NODES
NODE          11      12      13      21      22      23           14      15      16      17      18      24      25      26      27      28
TEMP.       285.    299.    285.    284.    299.    284.         104.     72.     72.     72.    104.     73.     70.     70.     70.     75.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                24          28          29
TEMP.              73.         75.         73.
Q(BTU/SEC)    0.          0.0300      0.0400


TIME=    135. SECONDS     COMPUTING INTERVAL=   3.000 SECONDS     (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                        TEMPERATURE OF INTERNAL WEB NODES
NODE          11      12      13      21      22      23           14      15      16      17      18      24      25      26      27      28
TEMP.       283.    298.    283.    282.    298.    282.         106.     73.     72.     72.    107.     73.     70.     71.     70.     76.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                24          28          29
TEMP.              73.         76.         73.
Q(BTU/SEC)    0.          0.0300      0.0400
```

TABLE 6-4. (CONTINUED)

```
THERMAL ANALYZER EXAMPLE PROBLEM TWO                      10002266

TIME=    140. SECONDS      COMPUTING INTERVAL=   3.000 SECONDS     (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                        TEMPERATURE OF INTERNAL WEB NODES
NODE          11      12      13      21      22      23          14      15      16      17      18      24      25      26      27      28
TEMP.       281.    296.    281.    279.    296.    279.        109.     73.     72.     72.    110.     74.     70.     71.     70.     76.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                  24          28          29
TEMP.                74.         76.         74.
Q(BTU/SEC)     0.           0.0300      0.0400


TIME=    145. SECONDS      COMPUTING INTERVAL=   3.000 SECONDS     (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                        TEMPERATURE OF INTERNAL WEB NODES
NODE          11      12      13      21      22      23          14      15      16      17      18      24      25      26      27      28
TEMP.       278.    295.    278.    277.    295.    277.        111.     73.     73.     73.    112.     74.     70.     71.     71.     77.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                  24          28          29
TEMP.                74.         77.         74.
Q(BTU/SEC)     0.           0.0300      0.0400


TIME=    150. SECONDS      COMPUTING INTERVAL=   6.000 SECONDS     (RC)MIN=  10.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                        TEMPERATURE OF INTERNAL WEB NODES
NODE          11      12      13      21      22      23          14      15      16      17      18      24      25      26      27      28
TEMP.       276.    293.    276.    274.    293.    274.        113.     74.     73.     73.    115.     74.     70.     71.     71.     77.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                  24          28          29
TEMP.                74.         77.         74.
Q(BTU/SEC)     0.           0.0300      0.0400


TIME=    160. SECONDS      COMPUTING INTERVAL=   6.000 SECONDS     (RC)MIN=  10.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                        TEMPERATURE OF INTERNAL WEB NODES
NODE          11      12      13      21      22      23          14      15      16      17      18      24      25      26      27      28
TEMP.       271.    290.    271.    269.    290.    268.        118.     74.     73.     74.    119.     75.     71.     71.     71.     78.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                  24          28          29
TEMP.                75.         78.         74.
Q(BTU/SEC)     0.           0.0300      0.0400
```

TABLE 6-4. (CONTINUED)

THERMAL ANALYZER EXAMPLE PROBLEM TWO 10002266

TIME= 170. SECONDS COMPUTING INTERVAL= 6.000 SECONDS (RC)MIN= 10.000 SECONDS AT NODE 0.

TEMPERATURE OF EXTERNAL SHELL NODES						
NODE	11	12	13	21	22	23
TEMP.	266.	286.	266.	263.	286.	263.

TEMPERATURE OF INTERNAL WEB NODES										
NODE	14	15	16	17	18	24	25	26	27	28
TEMP.	121.	75.	74.	74.	123.	75.	71.	71.	71.	78.

EQUIPMENT TEMPERATURES AND HEATING RATES			
NODE	24	28	29
TEMP.	75.	78.	74.
Q(BTU/SEC)	0.	0.0300	0.0400

TIME= 180. SECONDS COMPUTING INTERVAL= 6.000 SECONDS (RC)MIN= 10.000 SECONDS AT NODE 0.

TEMPERATURE OF EXTERNAL SHELL NODES						
NODE	11	12	13	21	22	23
TEMP.	261.	283.	261.	258.	283.	258.

TEMPERATURE OF INTERNAL WEB NODES										
NODE	14	15	16	17	18	24	25	26	27	28
TEMP.	125.	76.	75.	75.	127.	76.	71.	71.	71.	79.

EQUIPMENT TEMPERATURES AND HEATING RATES			
NODE	24	28	29
TEMP.	76.	79.	75.
Q(BTU/SEC)	0.	0.0300	0.0400

TIME= 190. SECONDS COMPUTING INTERVAL= 6.000 SECONDS (RC)MIN= 10.000 SECONDS AT NODE 0.

TEMPERATURE OF EXTERNAL SHELL NODES						
NODE	11	12	13	21	22	23
TEMP.	257.	280.	256.	253.	280.	253.

TEMPERATURE OF INTERNAL WEB NODES										
NODE	14	15	16	17	18	24	25	26	27	28
TEMP.	128.	76.	75.	76.	131.	76.	71.	71.	71.	80.

EQUIPMENT TEMPERATURES AND HEATING RATES			
NODE	24	28	29
TEMP.	76.	80.	75.
Q(BTU/SEC)	0.	0.0300	0.0400

TIME= 200. SECONDS COMPUTING INTERVAL= 42.579 SECONDS (RC)MIN= 70.965 SECONDS AT NODE 15.

TEMPERATURE OF EXTERNAL SHELL NODES						
NODE	11	12	13	21	22	23
TEMP.	252.	276.	252.	248.	276.	248.

TEMPERATURE OF INTERNAL WEB NODES										
NODE	14	15	16	17	18	24	25	26	27	28
TEMP.	131.	77.	76.	77.	134.	77.	71.	72.	71.	80.

EQUIPMENT TEMPERATURES AND HEATING RATES			
NODE	24	28	29
TEMP.	77.	80.	75.
Q(BTU/SEC)	0.	0.0300	0.0400

TABLE 6-4. (CONTINUED)

```
THERMAL ANALYZER EXAMPLE PROBLEM TWO                    10002266


TIME=    500. SECONDS      COMPUTING INTERVAL=  42.579 SECONDS     (RC)MIN=  70.965 SECONDS AT NODE   15.

TEMPERATURE OF EXTERNAL SHELL NODES                        TEMPERATURE OF INTERNAL WEB NODES
NODE         11     12     13     21     22     23             14     15     16     17     18     24     25     26     27     28
TEMP.      172.   199.   170.   153.   198.   152.           146.    91.    94.   100.   158.    87.    78.    81.    80.    94.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                 24          28          29
TEMP.               87.         94.         84.
Q(BTU/SEC)     0.          0.0300      0.0400


TIME=    800. SECONDS      COMPUTING INTERVAL=  42.579 SECONDS     (RC)MIN=  70.965 SECONDS AT NODE   15.

TEMPERATURE OF EXTERNAL SHELL NODES                        TEMPERATURE OF INTERNAL WEB NODES
NODE         11     12     13     21     22     23             14     15     16     17     18     24     25     26     27     28
TEMP.      139.   154.   132.   120.   151.   114.           126.    97.   101.   109.   140.    92.    87.    90.    91.   102.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                 24          28          29
TEMP.               92.        102.         94.
Q(BTU/SEC)     0.          0.0300      0.0400


TIME=   1100. SECONDS      COMPUTING INTERVAL=  42.579 SECONDS     (RC)MIN=  70.965 SECONDS AT NODE   15.

TEMPERATURE OF EXTERNAL SHELL NODES                        TEMPERATURE OF INTERNAL WEB NODES
NODE         11     12     13     21     22     23             14     15     16     17     18     24     25     26     27     28
TEMP.      113.   119.   104.    99.   115.    91.           110.   102.   105.   110.   123.    93.    93.    97.    99.   108.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                 24          28          29
TEMP.               93.        108.        102.
Q(BTU/SEC)     0.          0.0300      0.0400


TIME=   1400. SECONDS      COMPUTING INTERVAL=  42.579 SECONDS     (RC)MIN=  70.965 SECONDS AT NODE   15.

TEMPERATURE OF EXTERNAL SHELL NODES                        TEMPERATURE OF INTERNAL WEB NODES
NODE         11     12     13     21     22     23             14     15     16     17     18     24     25     26     27     28
TEMP.       94.    93.    85.    87.    90.    78.            98.   106.   106.   107.   108.    93.    97.   101.   103.   112.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                 24          28          29
TEMP.               93.        112.        110.
Q(BTU/SEC)     0.          0.0300      0.0400
```

TABLE 6-4. (CONTINUED)

THERMAL ANALYZER EXAMPLE PROBLEM TWO 10002266

TIME= 1700. SECONDS COMPUTING INTERVAL= 42.579 SECONDS (RC)MIN= 70.965 SECONDS AT NODE 15.

TEMPERATURE OF EXTERNAL SHELL NODES							TEMPERATURE OF INTERNAL WEB NODES								
NODE	11	12	13	21	22	23	14	15	16	17	18	24	25	26	27	28
TEMP.	81.	74.	73.	81.	72.	70.	90.	109.	106.	105.	95.	91.	100.	103.	105.	115.

EQUIPMENT TEMPERATURES AND HEATING RATES

NODE	24	28	29
TEMP.	91.	115.	116.
Q(BTU/SEC)	0.	0.0300	0.0400

TIME= 2000. SECONDS COMPUTING INTERVAL= 42.579 SECONDS (RC)MIN= 70.965 SECONDS AT NODE 15.

TEMPERATURE OF EXTERNAL SHELL NODES							TEMPERATURE OF INTERNAL WEB NODES								
NODE	11	12	13	21	22	23	14	15	16	17	18	24	25	26	27	28
TEMP.	71.	61.	65.	78.	60.	66.	86.	112.	107.	102.	86.	105.	103.	105.	107.	117.

EQUIPMENT TEMPERATURES AND HEATING RATES

NODE	24	28	29
TEMP.	105.	117.	123.
Q(BTU/SEC)	0.0850	0.0300	0.0400

TIME= 2300. SECONDS COMPUTING INTERVAL= 42.579 SECONDS (RC)MIN= 70.965 SECONDS AT NODE 15.

TEMPERATURE OF EXTERNAL SHELL NODES							TEMPERATURE OF INTERNAL WEB NODES								
NODE	11	12	13	21	22	23	14	15	16	17	18	24	25	26	27	28
TEMP.	63.	50.	60.	76.	51.	67.	84.	116.	108.	101.	80.	116.	109.	109.	109.	119.

EQUIPMENT TEMPERATURES AND HEATING RATES

NODE	24	28	29
TEMP.	116.	119.	128.
Q(BTU/SEC)	0.0850	0.0300	0.0400

TIME= 2600. SECONDS COMPUTING INTERVAL= 42.579 SECONDS (RC)MIN= 70.965 SECONDS AT NODE 15.

TEMPERATURE OF EXTERNAL SHELL NODES							TEMPERATURE OF INTERNAL WEB NODES								
NODE	11	12	13	21	22	23	14	15	16	17	18	24	25	26	27	28
TEMP.	57.	42.	57.	74.	44.	70.	85.	120.	110.	100.	75.	126.	114.	112.	111.	120.

EQUIPMENT TEMPERATURES AND HEATING RATES

NODE	24	28	29
TEMP.	126.	120.	134.
Q(BTU/SEC)	0.0850	0.0300	0.0400

TABLE 6-4. (CONTINUED)

THERMAL ANALYZER EXAMPLE PROBLEM TWO 10002266

TIME= 2900. SECONDS COMPUTING INTERVAL= 42.579 SECONDS (RC)MIN= 70.965 SECONDS AT NODE 15.

TEMPERATURE OF EXTERNAL SHELL NODES

NODE	11	12	13	21	22	23
TEMP.	53.	36.	56.	73.	40.	73.

TEMPERATURE OF INTERNAL WEB NODES

14	15	16	17	18	24	25	26	27	28
86.	124.	113.	101.	71.	136.	120.	116.	113.	122.

EQUIPMENT TEMPERATURES AND HEATING RATES

NODE	24	28	29
TEMP.	136.	122.	139.
Q(BTU/SEC)	0.0850	0.0300	0.0400

TIME= 3200. SECONDS COMPUTING INTERVAL= 42.579 SECONDS (RC)MIN= 70.965 SECONDS AT NODE 15.

TEMPERATURE OF EXTERNAL SHELL NODES

NODE	11	12	13	21	22	23
TEMP.	57.	38.	60.	79.	43.	82.

TEMPERATURE OF INTERNAL WEB NODES

14	15	16	17	18	24	25	26	27	28
89.	128.	115.	101.	70.	145.	125.	120.	116.	123.

EQUIPMENT TEMPERATURES AND HEATING RATES

NODE	24	28	29
TEMP.	145.	123.	143.
Q(BTU/SEC)	0.0850	0.0300	0.0400

TIME= 3500. SECONDS COMPUTING INTERVAL= 42.579 SECONDS (RC)MIN= 70.965 SECONDS AT NODE 15.

TEMPERATURE OF EXTERNAL SHELL NODES

NODE	11	12	13	21	22	23
TEMP.	56.	38.	63.	80.	44.	87.

TEMPERATURE OF INTERNAL WEB NODES

14	15	16	17	18	24	25	26	27	28
94.	132.	119.	104.	71.	153.	130.	124.	119.	125.

EQUIPMENT TEMPERATURES AND HEATING RATES

NODE	24	28	29
TEMP.	153.	125.	148.
Q(BTU/SEC)	0.0850	0.0300	0.0400

TIME= 3800. SECONDS COMPUTING INTERVAL= 42.579 SECONDS (RC)MIN= 70.965 SECONDS AT NODE 15.

TEMPERATURE OF EXTERNAL SHELL NODES

NODE	11	12	13	21	22	23
TEMP.	64.	47.	72.	88.	53.	98.

TEMPERATURE OF INTERNAL WEB NODES

14	15	16	17	18	24	25	26	27	28
100.	137.	122.	107.	74.	154.	135.	128.	122.	127.

EQUIPMENT TEMPERATURES AND HEATING RATES

NODE	24	28	29
TEMP.	154.	127.	153.
Q(BTU/SEC)	0.	0.0300	0.0400

TABLE 6-4. (CONTINUED)

```
THERMAL ANALYZER EXAMPLE PROBLEM TWO                    10002266

TIME=  4100. SECONDS     COMPUTING INTERVAL=  42.579 SECONDS     (RC)MIN=  70.965 SECONDS AT NODE   15.

TEMPERATURE OF EXTERNAL SHELL NODES                     TEMPERATURE OF INTERNAL WEB NODES
NODE        11     12     13     21     22     23          14     15     16     17     18     24     25     26     27     28
TEMP.      76.    61.    85.    99.    68.   108.        107.   142.   127.   111.    81.   149.   137.   130.   125.   129.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE               24          28          29
TEMP.            149.        129.        157.
Q(BTU/SEC)     0.          0.0300      0.0400


TIME=  4400. SECONDS     COMPUTING INTERVAL=  42.579 SECONDS     (RC)MIN=  70.965 SECONDS AT NODE   15.

TEMPERATURE OF EXTERNAL SHELL NODES                     TEMPERATURE OF INTERNAL WEB NODES
NODE        11     12     13     21     22     23          14     15     16     17     18     24     25     26     27     28
TEMP.      87.    75.    95.   108.    82.   114.        114.   146.   131.   116.    91.   145.   137.   132.   128.   132.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE               24          28          29
TEMP.            145.        132.        162.
Q(BTU/SEC)     0.          0.0300      0.0400


TIME=  4700. SECONDS     COMPUTING INTERVAL=  42.579 SECONDS     (RC)MIN=  70.965 SECONDS AT NODE   15.

TEMPERATURE OF EXTERNAL SHELL NODES                     TEMPERATURE OF INTERNAL WEB NODES
NODE        11     12     13     21     22     23          14     15     16     17     18     24     25     26     27     28
TEMP.      94.    84.   101.   113.    90.   115.        119.   151.   136.   121.    99.   142.   138.   135.   131.   136.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE               24          28          29
TEMP.            142.        136.        167.
Q(BTU/SEC)     0.          0.0300      0.0400


TIME=  5000. SECONDS     COMPUTING INTERVAL=  42.579 SECONDS     (RC)MIN=  70.965 SECONDS AT NODE   15.

TEMPERATURE OF EXTERNAL SHELL NODES                     TEMPERATURE OF INTERNAL WEB NODES
NODE        11     12     13     21     22     23          14     15     16     17     18     24     25     26     27     28
TEMP.      98.    89.   104.   115.    94.   114.        123.   155.   140.   126.   105.   139.   140.   137.   134.   139.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE               24          28          29
TEMP.            139.        139.        171.
Q(BTU/SEC)     0.          0.0300      0.0400
```

TABLE 6-4. (CONTINUED)

```
THERMAL ANALYZER EXAMPLE PROBLEM TWO                    10002266

TIME=   5300. SECONDS      COMPUTING INTERVAL=  42.579 SECONDS     (RC)MIN=  70.965 SECONDS AT NODE   15.

TEMPERATURE OF EXTERNAL SHELL NODES                        TEMPERATURE OF INTERNAL WEB NODES
NODE         11      12      13      21      22      23          14      15      16      17      18      24      25      26      27      28
TEMP.        99.     90.    104.    115.     94.    112.        124.    159.    144.    131.    108.    137.    141.    140.    137.    143.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE               24          28          29
TEMP.             137.        143.        176.
Q(BTU/SEC)    0.          0.0300      0.0400


TIME=   5600. SECONDS      COMPUTING INTERVAL=  42.579 SECONDS     (RC)MIN=  70.965 SECONDS AT NODE   15.

TEMPERATURE OF EXTERNAL SHELL NODES                        TEMPERATURE OF INTERNAL WEB NODES
NODE         11      12      13      21      22      23          14      15      16      17      18      24      25      26      27      28
TEMP.        95.     86.    100.    111.     89.    106.        124.    162.    147.    134.    109.    135.    142.    142.    140.    145.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE               24          28          29
TEMP.             135.        145.        180.
Q(BTU/SEC)    0.          0.0300      0.0400


TIME=   5900. SECONDS      COMPUTING INTERVAL=  42.579 SECONDS     (RC)MIN=  70.965 SECONDS AT NODE   15.

TEMPERATURE OF EXTERNAL SHELL NODES                        TEMPERATURE OF INTERNAL WEB NODES
NODE         11      12      13      21      22      23          14      15      16      17      18      24      25      26      27      28
TEMP.        88.     75.     92.    104.     79.     97.        121.    165.    150.    136.    107.    132.    143.    144.    142.    148.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE               24          28          29
TEMP.             132.        148.        184.
Q(BTU/SEC)    0.          0.0300      0.0400


TIME=   6200. SECONDS      COMPUTING INTERVAL=  42.579 SECONDS     (RC)MIN=  70.965 SECONDS AT NODE   15.

TEMPERATURE OF EXTERNAL SHELL NODES                        TEMPERATURE OF INTERNAL WEB NODES
NODE         11      12      13      21      22      23          14      15      16      17      18      24      25      26      27      28
TEMP.        85.     69.     87.    103.     73.     93.        118.    168.    151.    136.    104.    130.    144.    145.    144.    149.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE               24          28          29
TEMP.             130.        149.        188.
Q(BTU/SEC)    0.          0.0300      0.0400
```

LOCKHEED
CALIFORNIA COMPANY

TABLE 6-4. (CONTINUED)

RANGE OF MINIMUM-MAXIMUM

0.   0.62000000E 04

NODE NO.	MAX T	MIN T	TIME OF MAX	TIME OF MIN
24	0.92973553E 02	0.69997304E 02	0.11851580E 04	0.40000000E 02
	0.15754692E 03	0.91462558E 02	0.36703160E 04	0.16999999E 04
28	0.	0.	0.	0.
29	0.	0.	0.	0.
11	0.28848826E 03	0.69505806E 02	0.12000000E 03	0.34999999E 02
	0.56979638E 02	0.51479299E 02	0.32425790E 04	0.30703159E 04
	0.99294759E 02	0.55820698E 02	0.52128948E 04	0.34128950E 04
	0.88702224E 02	0.86379557E 02	0.61128948E 04	0.59851579E 04
12	0.30027721E 03	0.69495180E 02	0.12300000E 03	0.34999999E 02
	0.38503425E 02	0.33908306E 02	0.32425790E 04	0.30703159E 04
	0.90621306E 02	0.37677485E 02	0.52128948E 04	0.33703160E 04
	0.72856805E 02	0.71894490E 02	0.61128948E 04	0.60277369E 04

LOCKHEED CALIFORNIA COMPANY

TABLE 6-4. (CONTINUED)

```
           RESTART FOR EXAMPLE PROBLEM TWO                                  60012266
DEC 0.   5     NEW DATA                                                     60022266
DEC 0  103     CHANGE TABLE 3                                               60032266
DEC 3          .4          .7     2.1E-4                                    60042266
DEC 0 -103                                                                  60052266
DEC 0    0                                                                  60062266
NBK 0          END RESTART                                                  60072266
```

TABLE 6-4. (CONTINUED)

```
RESTART FOR EXAMPLE PROBLEM TWO                                        60012266


TIME=      0. SECONDS      COMPUTING INTERVAL=   0.      SECONDS      (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                             TEMPERATURE OF INTERNAL WEB NODES
NODE          11       12       13       21       22       23          14       15       16       17       18       24       25       26       27       28
TEMP.        70.      70.      70.      70.      70.      70.         70.      70.      70.      70.      70.      70.      70.      70.      70.      70.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                  24           28           29
TEMP.                70.          70.          70.
Q(BTU/SEC)     0.          0.0300       0.0400


TIME=      5. SECONDS      COMPUTING INTERVAL=   3.000 SECONDS      (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                             TEMPERATURE OF INTERNAL WEB NODES
NODE          11       12       13       21       22       23          14       15       16       17       18       24       25       26       27       28
TEMP.        70.      70.      70.      70.      70.      70.         70.      70.      70.      70.      70.      70.      70.      70.      70.      70.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                  24           28           29
TEMP.                70.          70.          70.
Q(BTU/SEC)     0.          0.0300       0.0400


TIME=     10. SECONDS      COMPUTING INTERVAL=   3.000 SECONDS      (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                             TEMPERATURE OF INTERNAL WEB NODES
NODE          11       12       13       21       22       23          14       15       16       17       18       24       25       26       27       28
TEMP.        70.      70.      70.      70.      70.      70.         70.      70.      70.      70.      70.      70.      70.      70.      70.      70.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                  24           28           29
TEMP.                70.          70.          70.
Q(BTU/SEC)     0.          0.0300       0.0400


TIME=     15. SECONDS      COMPUTING INTERVAL=   3.000 SECONDS      (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                             TEMPERATURE OF INTERNAL WEB NODES
NODE          11       12       13       21       22       23          14       15       16       17       18       24       25       26       27       28
TEMP.        70.      70.      70.      70.      70.      70.         70.      70.      70.      70.      70.      70.      70.      70.      70.      70.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                  24           28           29
TEMP.                70.          70.          70.
Q(BTU/SEC)     0.          0.0300       0.0400
```

TABLE 6-4. (CONTINUED)

```
RESTART FOR EXAMPLE PROBLEM TWO                                60012266

TIME=     20. SECONDS      COMPUTING INTERVAL=   3.000 SECONDS      (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                          TEMPERATURE OF INTERNAL WEB NODES
NODE          11      12      13      21      22      23            14      15      16      17      18      24      25      26      27      28
TEMP.        70.     70.     70.     70.     70.     70.           70.     70.     70.     70.     70.     70.     70.     70.     70.     70.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                24           28          29
TEMP.              70.          70.         71.
Q(BTU/SEC)    0.          0.0300      0.0400


TIME=     25. SECONDS      COMPUTING INTERVAL=   3.000 SECONDS      (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                          TEMPERATURE OF INTERNAL WEB NODES
NODE          11      12      13      21      22      23            14      15      16      17      18      24      25      26      27      28
TEMP.        70.     70.     70.     70.     70.     70.           70.     70.     70.     70.     70.     70.     70.     70.     70.     70.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                24           28          29
TEMP.              70.          70.         71.
Q(BTU/SEC)    0.          0.0300      0.0400


TIME=     30. SECONDS      COMPUTING INTERVAL=   3.000 SECONDS      (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                          TEMPERATURE OF INTERNAL WEB NODES
NODE          11      12      13      21      22      23            14      15      16      17      18      24      25      26      27      28
TEMP.        70.     70.     70.     70.     70.     70.           70.     70.     70.     70.     70.     70.     70.     70.     70.     71.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                24           28          29
TEMP.              70.          71.         71.
Q(BTU/SEC)    0.          0.0300      0.0400


TIME=     35. SECONDS      COMPUTING INTERVAL=   3.000 SECONDS      (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                          TEMPERATURE OF INTERNAL WEB NODES
NODE          11      12      13      21      22      23            14      15      16      17      18      24      25      26      27      28
TEMP.        70.     70.     70.     70.     70.     70.           70.     70.     70.     70.     70.     70.     70.     70.     70.     71.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                24           28          29
TEMP.              70.          71.         71.
Q(BTU/SEC)    0.          0.0300      0.0400
```

TABLE 6-4. (CONTINUED)

```
RESTART FOR EXAMPLE PROBLEM TWO                                    60012266


TIME=    40. SECONDS      COMPUTING INTERVAL=   3.000 SECONDS     (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                       TEMPERATURE OF INTERNAL WEB NODES
NODE          11      12      13      21      22      23          14      15      16      17      18      24      25      26      27      28
TEMP.        70.     70.     70.     70.     70.     70.         70.     70.     70.     70.     70.     70.     70.     70.     70.     71.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                 24          28          29
TEMP.               70.         71.         71.
Q(BTU/SEC)     0.            0.0300      0.0400


TIME=    45. SECONDS      COMPUTING INTERVAL=   3.000 SECONDS     (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                       TEMPERATURE OF INTERNAL WEB NODES
NODE          11      12      13      21      22      23          14      15      16      17      18      24      25      26      27      28
TEMP.        75.     75.     75.     75.     75.     75.         70.     70.     70.     70.     70.     70.     70.     70.     70.     71.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                 24          28          29
TEMP.               70.         71.         71.
Q(BTU/SEC)     0.            0.0300      0.0400


TIME=    50. SECONDS      COMPUTING INTERVAL=   3.000 SECONDS     (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                       TEMPERATURE OF INTERNAL WEB NODES
NODE          11      12      13      21      22      23          14      15      16      17      18      24      25      26      27      28
TEMP.        85.     85.     85.     85.     85.     85.         70.     70.     70.     70.     70.     70.     70.     70.     70.     71.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                 24          28          29
TEMP.               70.         71.         71.
Q(BTU/SEC)     0.            0.0300      0.0400


TIME=    55. SECONDS      COMPUTING INTERVAL=   3.000 SECONDS     (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                       TEMPERATURE OF INTERNAL WEB NODES
NODE          11      12      13      21      22      23          14      15      16      17      18      24      25      26      27      28
TEMP.       100.    100.    100.    100.    100.    100.         70.     70.     70.     70.     70.     70.     70.     70.     70.     71.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                 24          28          29
TEMP.               70.         71.         71.
Q(BTU/SEC)     0.            0.0300      0.0400
```

TABLE 6-4. (CONTINUED)

```
RESTART FOR EXAMPLE PROBLEM TWO                          60012266

TIME=     60. SECONDS      COMPUTING INTERVAL=   3.000 SECONDS     (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                           TEMPERATURE OF INTERNAL WEB NODES
NODE          11      12      13      21      22      23             14      15      16      17      18      24      25      26      27      28
TEMP.       121.    121.    121.    121.    121.    121.             71.     70.     70.     70.     71.     70.     70.     70.     70.     71.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                24          28          29
TEMP.               70.         71.         72.
Q(BTU/SEC)    0.            0.0300      0.0400


TIME=     65. SECONDS      COMPUTING INTERVAL=   3.000 SECONDS     (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                           TEMPERATURE OF INTERNAL WEB NODES
NODE          11      12      13      21      22      23             14      15      16      17      18      24      25      26      27      28
TEMP.       143.    143.    143.    143.    143.    143.             72.     70.     70.     70.     72.     70.     70.     70.     70.     71.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                24          28          29
TEMP.               70.         71.         72.
Q(BTU/SEC)    0.            0.0300      0.0400


TIME=     70. SECONDS      COMPUTING INTERVAL=   3.000 SECONDS     (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                           TEMPERATURE OF INTERNAL WEB NODES
NODE          11      12      13      21      22      23             14      15      16      17      18      24      25      26      27      28
TEMP.       165.    166.    165.    165.    166.    165.             73.     70.     70.     70.     73.     70.     70.     70.     70.     72.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                24          28          29
TEMP.               70.         72.         72.
Q(BTU/SEC)    0.            0.0300      0.0400


TIME=     75. SECONDS      COMPUTING INTERVAL=   3.000 SECONDS     (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                           TEMPERATURE OF INTERNAL WEB NODES
NODE          11      12      13      21      22      23             14      15      16      17      18      24      25      26      27      28
TEMP.       187.    189.    187.    187.    189.    187.             75.     71.     70.     70.     75.     70.     70.     70.     70.     72.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                24          28          29
TEMP.               70.         72.         72.
Q(BTU/SEC)    0.            0.0300      0.0400
```

TABLE 6-4. (CONTINUED)

```
RESTART FOR EXAMPLE PROBLEM TWO                              60012266


TIME=     80. SECONDS      COMPUTING INTERVAL=   3.000 SECONDS      (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                        TEMPERATURE OF INTERNAL WEB NODES
NODE          11      12      13      21      22      23          14      15      16      17      18      24      25      26      27      28
TEMP.       209.    212.    209.    209.    212.    209.         77.     71.     70.     70.     77.     71.     70.     70.     70.     72.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                 24          28          29
TEMP.               71.         72.         72.
Q(BTU/SEC)    0.            0.0300      0.0400


TIME=     85. SECONDS      COMPUTING INTERVAL=   3.000 SECONDS      (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                        TEMPERATURE OF INTERNAL WEB NODES
NODE          11      12      13      21      22      23          14      15      16      17      18      24      25      26      27      28
TEMP.       229.    233.    229.    229.    233.    229.         79.     71.     70.     70.     79.     71.     70.     70.     70.     72.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                 24          28          29
TEMP.               71.         72.         72.
Q(BTU/SEC)    0.            0.0300      0.0400


TIME=     90. SECONDS      COMPUTING INTERVAL=   3.000 SECONDS      (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                        TEMPERATURE OF INTERNAL WEB NODES
NODE          11      12      13      21      22      23          14      15      16      17      18      24      25      26      27      28
TEMP.       247.    252.    247.    247.    252.    247.         81.     71.     70.     70.     81.     71.     70.     70.     70.     73.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                 24          28          29
TEMP.               71.         73.         72.
Q(BTU/SEC)    0.            0.0300      0.0400


TIME=     95. SECONDS      COMPUTING INTERVAL=   3.000 SECONDS      (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                        TEMPERATURE OF INTERNAL WEB NODES
NODE          11      12      13      21      22      23          14      15      16      17      18      24      25      26      27      28
TEMP.       262.    268.    262.    262.    268.    262.         84.     71.     71.     70.     84.     71.     70.     70.     70.     73.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                 24          28          29
TEMP.               71.         73.         72.
Q(BTU/SEC)    0.            0.0300      0.0400
```

TABLE 6-4. (CONTINUED)

```
RESTART FOR EXAMPLE PROBLEM TWO                        60012266

TIME=    100. SECONDS     COMPUTING INTERVAL=   3.000 SECONDS     (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                       TEMPERATURE OF INTERNAL WEB NODES
NODE        11     12     13     21     22     23             14     15     16     17     18     24     25     26     27     28
TEMP.     274.   280.   274.   273.   280.   273.            87.    71.    71.    70.    87.    71.    70.    70.    70.    73.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE               24          28          29
TEMP.              71.         73.         73.
Q(BTU/SEC)    0.          0.0300      0.0400


TIME=    105. SECONDS     COMPUTING INTERVAL=   3.000 SECONDS     (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                       TEMPERATURE OF INTERNAL WEB NODES
NODE        11     12     13     21     22     23             14     15     16     17     18     24     25     26     27     28
TEMP.     282.   290.   282.   282.   290.   282.            90.    71.    71.    71.    90.    72.    70.    70.    70.    74.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE               24          28          29
TEMP.              72.         74.         73.
Q(BTU/SEC)    0.          0.0300      0.0400


TIME=    110. SECONDS     COMPUTING INTERVAL=   3.000 SECONDS     (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                       TEMPERATURE OF INTERNAL WEB NODES
NODE        11     12     13     21     22     23             14     15     16     17     18     24     25     26     27     28
TEMP.     287.   297.   287.   287.   297.   287.            92.    72.    71.    71.    93.    72.    70.    70.    70.    74.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE               24          28          29
TEMP.              72.         74.         73.
Q(BTU/SEC)    0.          0.0300      0.0400


TIME=    115. SECONDS     COMPUTING INTERVAL=   3.000 SECONDS     (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                       TEMPERATURE OF INTERNAL WEB NODES
NODE        11     12     13     21     22     23             14     15     16     17     18     24     25     26     27     28
TEMP.     290.   300.   290.   289.   300.   289.            95.    72.    71.    71.    96.    72.    70.    70.    70.    74.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE               24          28          29
TEMP.              72.         74.         73.
Q(BTU/SEC)    0.          0.0300      0.0400
```

TABLE 6-4. (CONTINUED)

```
RESTART FOR EXAMPLE PROBLEM TWO                                      60012266
```

TIME= 120. SECONDS COMPUTING INTERVAL= 3.000 SECONDS (RC)MIN= 5.000 SECONDS AT NODE 0.

TEMPERATURE OF EXTERNAL SHELL NODES

NODE	11	12	13	21	22	23
TEMP.	290.	302.	290.	289.	302.	289.

TEMPERATURE OF INTERNAL WEB NODES

NODE	14	15	16	17	18	24	25	26	27	28
TEMP.	98.	72.	71.	71.	99.	73.	70.	70.	70.	75.

EQUIPMENT TEMPERATURES AND HEATING RATES

NODE	24	28	29
TEMP.	73.	75.	73.
Q(BTU/SEC)	0.	0.0300	0.0400

TIME= 125. SECONDS COMPUTING INTERVAL= 3.000 SECONDS (RC)MIN= 5.000 SECONDS AT NODE 0.

TEMPERATURE OF EXTERNAL SHELL NODES

NODE	11	12	13	21	22	23
TEMP.	289.	302.	289.	289.	302.	288.

TEMPERATURE OF INTERNAL WEB NODES

NODE	14	15	16	17	18	24	25	26	27	28
TEMP.	101.	72.	72.	71.	102.	73.	70.	70.	70.	75.

EQUIPMENT TEMPERATURES AND HEATING RATES

NODE	24	28	29
TEMP.	73.	75.	73.
Q(BTU/SEC)	0.	0.0300	0.0400

TIME= 130. SECONDS COMPUTING INTERVAL= 3.000 SECONDS (RC)MIN= 5.000 SECONDS AT NODE 0.

TEMPERATURE OF EXTERNAL SHELL NODES

NODE	11	12	13	21	22	23
TEMP.	287.	301.	287.	286.	301.	286.

TEMPERATURE OF INTERNAL WEB NODES

NODE	14	15	16	17	18	24	25	26	27	28
TEMP.	104.	72.	72.	72.	105.	73.	70.	70.	70.	75.

EQUIPMENT TEMPERATURES AND HEATING RATES

NODE	24	28	29
TEMP.	73.	75.	73.
Q(BTU/SEC)	0.	0.0300	0.0400

TIME= 135. SECONDS COMPUTING INTERVAL= 3.000 SECONDS (RC)MIN= 5.000 SECONDS AT NODE 0.

TEMPERATURE OF EXTERNAL SHELL NODES

NODE	11	12	13	21	22	23
TEMP.	285.	300.	285.	284.	300.	284.

TEMPERATURE OF INTERNAL WEB NODES

NODE	14	15	16	17	18	24	25	26	27	28
TEMP.	106.	73.	72.	72.	107.	73.	70.	71.	70.	76.

EQUIPMENT TEMPERATURES AND HEATING RATES

NODE	24	28	29
TEMP.	73.	76.	74.
Q(BTU/SEC)	0.	0.0300	0.0400

TABLE 6-4. (CONTINUED)

```
RESTART FOR EXAMPLE PROBLEM TWO                              60012266

TIME=    140. SECONDS      COMPUTING INTERVAL=   3.000 SECONDS     (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                          TEMPERATURE OF INTERNAL WEB NODES
NODE          11      12      13      21      22      23          14      15      16      17      18      24      25      26      27      28
TEMP.       283.    299.    283.    282.    299.    281.        109.     73.     72.     72.    110.     74.     70.     71.     70.     76.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                24          28          29
TEMP.              74.         76.         74.
Q(BTU/SEC)    0.          0.0300      0.0400

TIME=    145. SECONDS      COMPUTING INTERVAL=   3.000 SECONDS     (RC)MIN=   5.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                          TEMPERATURE OF INTERNAL WEB NODES
NODE          11      12      13      21      22      23          14      15      16      17      18      24      25      26      27      28
TEMP.       281.    298.    281.    279.    298.    279.        111.     73.     73.     73.    112.     74.     70.     71.     71.     77.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                24          28          29
TEMP.              74.         77.         74.
Q(BTU/SEC)    0.          0.0300      0.0400

TIME=    150. SECONDS      COMPUTING INTERVAL=   6.000 SECONDS     (RC)MIN=  10.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                          TEMPERATURE OF INTERNAL WEB NODES
NODE          11      12      13      21      22      23          14      15      16      17      18      24      25      26      27      28
TEMP.       279.    296.    279.    277.    296.    277.        114.     74.     73.     73.    115.     74.     70.     71.     71.     77.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                24          28          29
TEMP.              74.         77.         74.
Q(BTU/SEC)    0.          0.0300      0.0400

TIME=    160. SECONDS      COMPUTING INTERVAL=   6.000 SECONDS     (RC)MIN=  10.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                          TEMPERATURE OF INTERNAL WEB NODES
NODE          11      12      13      21      22      23          14      15      16      17      18      24      25      26      27      28
TEMP.       274.    293.    274.    272.    293.    271.        118.     74.     73.     74.    119.     75.     71.     71.     71.     78.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                24          28          29
TEMP.              75.         78.         74.
Q(BTU/SEC)    0.          0.0300      0.0400
```

TABLE 6-4. (CONTINUED)

```
RESTART FOR EXAMPLE PROBLEM TWO                                60012266

TIME=    170. SECONDS      COMPUTING INTERVAL=   6.000 SECONDS     (RC)MIN=  10.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                          TEMPERATURE OF INTERNAL WEB NODES
NODE         11      12      13      21      22      23          14      15      16      17      18      24      25      26      27      28
TEMP.      269.    290.    269.    266.    290.    266.        122.     75.     74.     74.    124.     75.     71.     71.     71.     78.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                 24          28          29
TEMP.               75.         78.         74.
Q(BTU/SEC)     0.            0.0300      0.0400

TIME=    180. SECONDS      COMPUTING INTERVAL=   6.000 SECONDS     (RC)MIN=  10.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                          TEMPERATURE OF INTERNAL WEB NODES
NODE         11      12      13      21      22      23          14      15      16      17      18      24      25      26      27      28
TEMP.      265.    287.    265.    262.    287.    261.        125.     76.     75.     75.    128.     76.     71.     71.     71.     79.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                 24          28          29
TEMP.               76.         79.         75.
Q(BTU/SEC)     0.            0.0300      0.0400

TIME=    190. SECONDS      COMPUTING INTERVAL=   6.000 SECONDS     (RC)MIN=  10.000 SECONDS AT NODE    0.

TEMPERATURE OF EXTERNAL SHELL NODES                          TEMPERATURE OF INTERNAL WEB NODES
NODE         11      12      13      21      22      23          14      15      16      17      18      24      25      26      27      28
TEMP.      261.    284.    261.    257.    284.    256.        129.     76.     75.     76.    131.     77.     71.     71.     71.     80.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                 24          28          29
TEMP.               77.         80.         75.
Q(BTU/SEC)     0.            0.0300      0.0400

TIME=    200. SECONDS      COMPUTING INTERVAL=  42.579 SECONDS     (RC)MIN=  70.965 SECONDS AT NODE   15.

TEMPERATURE OF EXTERNAL SHELL NODES                          TEMPERATURE OF INTERNAL WEB NODES
NODE         11      12      13      21      22      23          14      15      16      17      18      24      25      26      27      28
TEMP.      257.    281.    256.    252.    281.    252.        132.     77.     76.     77.    135.     77.     71.     72.     71.     80.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                 24          28          29
TEMP.               77.         80.         75.
Q(BTU/SEC)     0.            0.0300      0.0400
```

TABLE 6-4. (CONTINUED)

```
RESTART FOR EXAMPLE PROBLEM TWO                                    60012266

TIME=   500. SECONDS     COMPUTING INTERVAL=  42.579 SECONDS     (RC)MIN=  70.965 SECONDS AT NODE   15.

TEMPERATURE OF EXTERNAL SHELL NODES                       TEMPERATURE OF INTERNAL WEB NODES
NODE         11      12      13      21      22      23          14      15      16      17      18      24      25      26      27      28
TEMP.       182.    212.    179.    162.    210.    160.        150.     91.     94.    101.    163.     88.     79.     81.     81.     95.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                 24          28          29
TEMP.                88.         95.         84.
Q(BTU/SEC)     0.            0.0300      0.0400


TIME=   800. SECONDS     COMPUTING INTERVAL=  42.579 SECONDS     (RC)MIN=  70.965 SECONDS AT NODE   15.

TEMPERATURE OF EXTERNAL SHELL NODES                       TEMPERATURE OF INTERNAL WEB NODES
NODE         11      12      13      21      22      23          14      15      16      17      18      24      25      26      27      28
TEMP.       147.    166.    141.    126.    163.    121.        131.     98.    103.    111.    147.     93.     88.     91.     92.    103.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                 24          28          29
TEMP.                93.        103.         94.
Q(BTU/SEC)     0.            0.0300      0.0400


TIME=  1100. SECONDS     COMPUTING INTERVAL=  42.579 SECONDS     (RC)MIN=  70.965 SECONDS AT NODE   15.

TEMPERATURE OF EXTERNAL SHELL NODES                       TEMPERATURE OF INTERNAL WEB NODES
NODE         11      12      13      21      22      23          14      15      16      17      18      24      25      26      27      28
TEMP.       121.    130.    112.    104.    126.     97.        115.    103.    107.    112.    129.     95.     95.     98.    100.    109.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                 24          28          29
TEMP.                95.        109.        102.
Q(BTU/SEC)     0.            0.0300      0.0400


TIME=  1400. SECONDS     COMPUTING INTERVAL=  42.579 SECONDS     (RC)MIN=  70.965 SECONDS AT NODE   15.

TEMPERATURE OF EXTERNAL SHELL NODES                       TEMPERATURE OF INTERNAL WEB NODES
NODE         11      12      13      21      22      23          14      15      16      17      18      24      25      26      27      28
TEMP.       102.    104.     93.     93.    100.     84.        103.    107.    108.    110.    114.     95.     99.    103.    105.    113.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                 24          28          29
TEMP.                95.        113.        110.
Q(BTU/SEC)     0.            0.0300      0.0400
```

TABLE 6-4. (CONTINUED)

```
     RESTART FOR EXAMPLE PROBLEM TWO                              60012266


TIME=   1700. SECONDS      COMPUTING INTERVAL=  42.579 SECONDS     (RC)MIN=  70.965 SECONDS AT NODE   15.

TEMPERATURE OF EXTERNAL SHELL NODES                          TEMPERATURE OF INTERNAL WEB NODES
NODE         11      12      13      21      22      23             14      15      16      17      18      24      25      26      27      28
TEMP.        88.     85.     81.     87.     82.     76.            95.    111.    109.    108.    102.     94.    102.    105.    107.    117.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                 24        28        29
TEMP.                94.      117.      117.
Q(BTU/SEC)     0.           0.0300    0.0400


TIME=   2000. SECONDS      COMPUTING INTERVAL=  42.579 SECONDS     (RC)MIN=  70.965 SECONDS AT NODE   15.

TEMPERATURE OF EXTERNAL SHELL NODES                          TEMPERATURE OF INTERNAL WEB NODES
NODE         11      12      13      21      22      23             14      15      16      17      18      24      25      26      27      28
TEMP.        78.     70.     72.     84.     69.     72.            91.    114.    110.    106.     93.    107.    105.    108.    109.    119.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                 24        28        29
TEMP.               107.      119.      124.
Q(BTU/SEC)     0.0850      0.0300    0.0400


TIME=   2300. SECONDS      COMPUTING INTERVAL=  42.579 SECONDS     (RC)MIN=  70.965 SECONDS AT NODE   15.

TEMPERATURE OF EXTERNAL SHELL NODES                          TEMPERATURE OF INTERNAL WEB NODES
NODE         11      12      13      21      22      23             14      15      16      17      18      24      25      26      27      28
TEMP.        70.     59.     66.     81.     59.     73.            89.    118.    111.    104.     86.    119.    111.    111.    111.    121.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                 24        28        29
TEMP.               119.      121.      129.
Q(BTU/SEC)     0.0850      0.0300    0.0400


TIME=   2600. SECONDS      COMPUTING INTERVAL=  42.579 SECONDS     (RC)MIN=  70.965 SECONDS AT NODE   15.

TEMPERATURE OF EXTERNAL SHELL NODES                          TEMPERATURE OF INTERNAL WEB NODES
NODE         11      12      13      21      22      23             14      15      16      17      18      24      25      26      27      28
TEMP.        64.     50.     63.     80.     52.     76.            90.    122.    113.    104.     81.    130.    117.    115.    114.    123.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                 24        28        29
TEMP.               130.      123.      135.
Q(BTU/SEC)     0.0850      0.0300    0.0400
```

TABLE 6-4. (CONTINUED)

```
    RESTART FOR EXAMPLE PROBLEM TWO                              60012266

TIME=   2900. SECONDS      COMPUTING INTERVAL=  42.579 SECONDS     (RC)MIN=  70.965 SECONDS AT NODE   15.

TEMPERATURE OF EXTERNAL SHELL NODES                          TEMPERATURE OF INTERNAL WEB NODES
NODE          11      12      13      21      22      23            14      15      16      17      18      24      25      26      27      28
TEMP.        60.     44.     62.     79.     47.     79.           91.    126.    116.    104.     77.    139.    122.    119.    116.    125.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                 24          28          29
TEMP.              139.        125.        140.
Q(BTU/SEC)       0.0850      0.0300      0.0400


TIME=   3200. SECONDS      COMPUTING INTERVAL=  42.579 SECONDS     (RC)MIN=  70.965 SECONDS AT NODE   15.

TEMPERATURE OF EXTERNAL SHELL NODES                          TEMPERATURE OF INTERNAL WEB NODES
NODE          11      12      13      21      22      23            14      15      16      17      18      24      25      26      27      28
TEMP.        60.     43.     64.     82.     47.     86.           94.    131.    118.    105.     75.    148.    128.    123.    119.    126.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                 24          28          29
TEMP.              148.        126.        145.
Q(BTU/SEC)       0.0850      0.0300      0.0400


TIME=   3500. SECONDS      COMPUTING INTERVAL=  42.579 SECONDS     (RC)MIN=  70.965 SECONDS AT NODE   15.

TEMPERATURE OF EXTERNAL SHELL NODES                          TEMPERATURE OF INTERNAL WEB NODES
NODE          11      12      13      21      22      23            14      15      16      17      18      24      25      26      27      28
TEMP.        59.     41.     66.     82.     47.     90.           97.    135.    122.    107.     75.    157.    133.    127.    122.    128.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                 24          28          29
TEMP.              157.        128.        150.
Q(BTU/SEC)       0.0850      0.0300      0.0400


TIME=   3800. SECONDS      COMPUTING INTERVAL=  42.579 SECONDS     (RC)MIN=  70.965 SECONDS AT NODE   15.

TEMPERATURE OF EXTERNAL SHELL NODES                          TEMPERATURE OF INTERNAL WEB NODES
NODE          11      12      13      21      22      23            14      15      16      17      18      24      25      26      27      28
TEMP.        62.     44.     71.     86.     50.     98.          102.    139.    125.    110.     76.    158.    138.    131.    125.    130.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                 24          28          29
TEMP.              158.        130.        155.
Q(BTU/SEC)       0.          0.0300      0.0400
```

TABLE 6-4. (CONTINUED)

```
RESTART FOR EXAMPLE PROBLEM TWO                                60012266

TIME=   4100. SECONDS      COMPUTING INTERVAL=  42.579 SECONDS     (RC)MIN=  70.965 SECONDS AT NODE   15.

TEMPERATURE OF EXTERNAL SHELL NODES                        TEMPERATURE OF INTERNAL WEB NODES
NODE         11     12     13     21     22     23                14     15     16     17     18     24     25     26     27     28
TEMP.        68.    51.    79.    92.    58.   103.              106.   143.   129.   113.    80.   152.   139.   133.   128.   132.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                 24          28         29
TEMP.               152.        132.       159.
Q(BTU/SEC)     0.          0.0300     0.0400


TIME=   4400. SECONDS      COMPUTING INTERVAL=  42.579 SECONDS     (RC)MIN=  70.965 SECONDS AT NODE   15.

TEMPERATURE OF EXTERNAL SHELL NODES                        TEMPERATURE OF INTERNAL WEB NODES
NODE         11     12     13     21     22     23                14     15     16     17     18     24     25     26     27     28
TEMP.        74.    58.    85.    97.    65.   106.              110.   147.   132.   116.    84.   147.   140.   135.   130.   134.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                 24          28         29
TEMP.               147.        134.       164.
Q(BTU/SEC)     0.          0.0300     0.0400


TIME=   4700. SECONDS      COMPUTING INTERVAL=  42.579 SECONDS     (RC)MIN=  70.965 SECONDS AT NODE   15.

TEMPERATURE OF EXTERNAL SHELL NODES                        TEMPERATURE OF INTERNAL WEB NODES
NODE         11     12     13     21     22     23                14     15     16     17     18     24     25     26     27     28
TEMP.        77.    63.    88.   100.    69.   105.              113.   151.   136.   120.    89.   143.   140.   136.   132.   137.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                 24          28         29
TEMP.               143.        137.       168.
Q(BTU/SEC)     0.          0.0300     0.0400


TIME=   5000. SECONDS      COMPUTING INTERVAL=  42.579 SECONDS     (RC)MIN=  70.965 SECONDS AT NODE   15.

TEMPERATURE OF EXTERNAL SHELL NODES                        TEMPERATURE OF INTERNAL WEB NODES
NODE         11     12     13     21     22     23                14     15     16     17     18     24     25     26     27     28
TEMP.        80.    66.    90.   102.    72.   104.              115.   155.   139.   123.    92.   139.   140.   138.   134.   139.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                 24          28         29
TEMP.               139.        139.       173.
Q(BTU/SEC)     0.          0.0300     0.0400
```

TABLE 6-4. (CONTINUED)

```
RESTART FOR EXAMPLE PROBLEM TWO                                        60012266

TIME=   5300. SECONDS     COMPUTING INTERVAL=  42.579 SECONDS     (RC)MIN=  70.965 SECONDS AT NODE   15.

TEMPERATURE OF EXTERNAL SHELL NODES                          TEMPERATURE OF INTERNAL WEB NODES
NODE         11      12      13      21      22      23            14      15      16      17      18      24      25      26      27      28
TEMP.        81.     67.     90.    102.     72.    102.          116.    158.    142.    126.     95.    136.    141.    139.    136.    141.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                 24          28          29
TEMP.               136.        141.        177.
Q(BTU/SEC)     0.          0.0300      0.0400


TIME=   5600. SECONDS     COMPUTING INTERVAL=  42.579 SECONDS     (RC)MIN=  70.965 SECONDS AT NODE   15.

TEMPERATURE OF EXTERNAL SHELL NODES                          TEMPERATURE OF INTERNAL WEB NODES
NODE         11      12      13      21      22      23            14      15      16      17      18      24      25      26      27      28
TEMP.        80.     65.     88.    101.     70.     98.          116.    161.    144.    128.     96.    133.    141.    140.    138.    143.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                 24          28          29
TEMP.               133.        143.        181.
Q(BTU/SEC)     0.          0.0300      0.0400


TIME=   5900. SECONDS     COMPUTING INTERVAL=  42.579 SECONDS     (RC)MIN=  70.965 SECONDS AT NODE   15.

TEMPERATURE OF EXTERNAL SHELL NODES                          TEMPERATURE OF INTERNAL WEB NODES
NODE         11      12      13      21      22      23            14      15      16      17      18      24      25      26      27      28
TEMP.        76.     60.     84.     98.     65.     92.          115.    164.    146.    130.     95.    130.    142.    141.    140.    145.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                 24          28          29
TEMP.               130.        145.        184.
Q(BTU/SEC)     0.          0.0300      0.0400


TIME=   6200. SECONDS     COMPUTING INTERVAL=  42.579 SECONDS     (RC)MIN=  70.965 SECONDS AT NODE   15.

TEMPERATURE OF EXTERNAL SHELL NODES                          TEMPERATURE OF INTERNAL WEB NODES
NODE         11      12      13      21      22      23            14      15      16      17      18      24      25      26      27      28
TEMP.        75.     56.     81.     98.     61.     89.          113.    166.    148.    131.     94.    127.    142.    142.    141.    147.

EQUIPMENT TEMPERATURES AND HEATING RATES
NODE                 24          28          29
TEMP.               127.        147.        188.
Q(BTU/SEC)     0.          0.0300      0.0400
```

TABLE 6-4. (CONTINUED)

RANGE OF MINIMUM-MAXIMUM

0. 0.62000000E 04

NODE NO.	MAX T	MIN T	TIME OF MAX	TIME OF MIN
24	0.94871831E 02	0.69997656E 02	0.12703160E 04	0.40000000E 02
	0.16111676E 03	0.93935698E 02	0.36703160E 04	0.16999999E 04
28	0.	0.	0.	0.
29	0.	0.	0.	0.
11	0.29009773E 03	0.69566916E 02	0.12000000E 03	0.34999999E 02
	0.60110962E 02	0.58092370E 02	0.31999999E 04	0.30703159E 04
	0.80980001E 02	0.58823948E 02	0.52999999E 04	0.34554740E 04
	0.76623477E 02	0.75392339E 02	0.61128948E 04	0.59851579E 04
12	0.30216596E 03	0.69557594E 02	0.12500000E 03	0.34999999E 02
	0.42506868E 02	0.41300423E 02	0.31999999E 04	0.30703159E 04
	0.66655913E 02	0.40919013E 02	0.52554738E 04	0.34554740E 04
	0.58164551E 02	0.57733718E 02	0.61128948E 04	0.60277369E 04

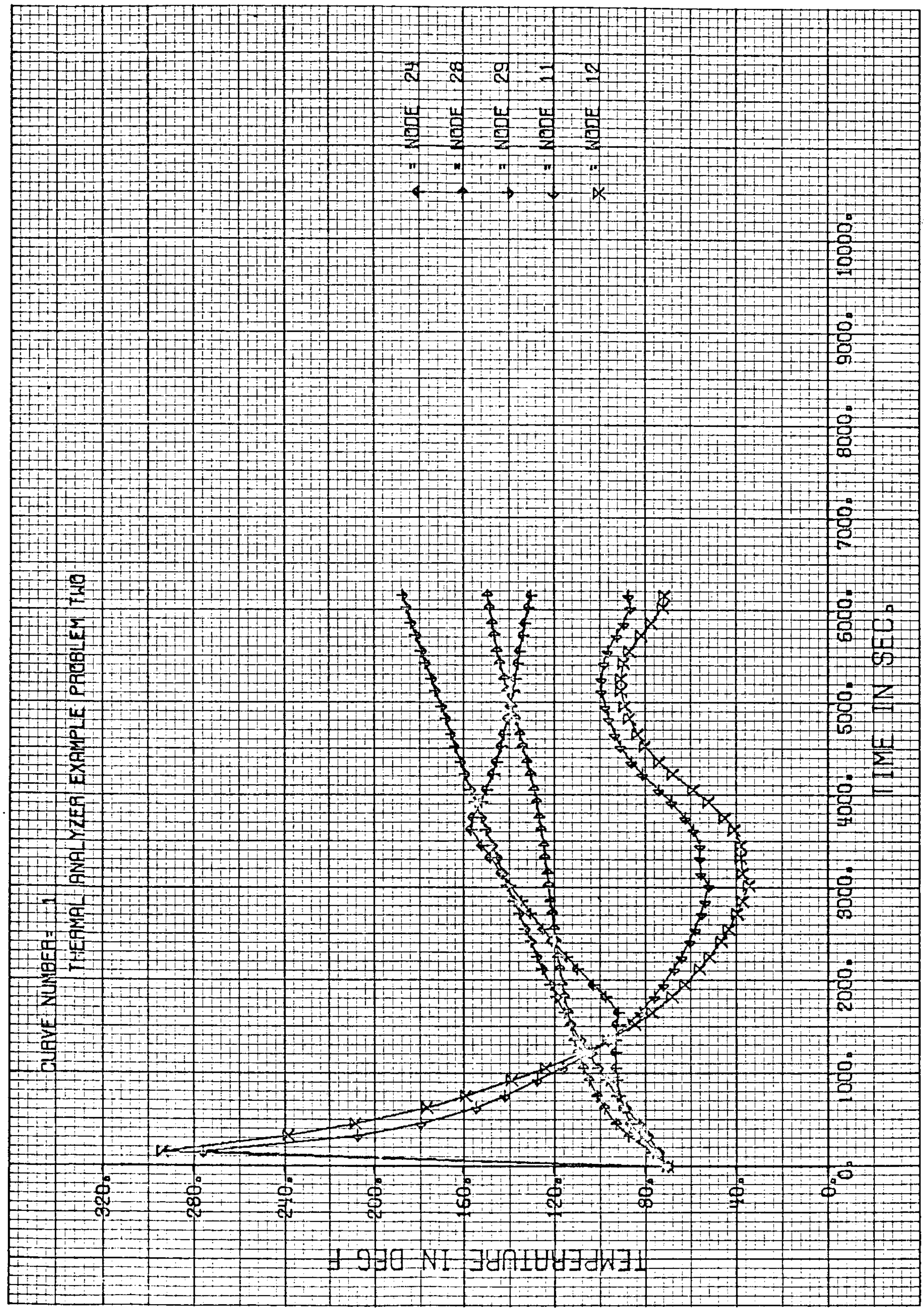


Figure 6-4. Temperature Plots for Example Problem 2

LOCKHEED
CALIFORNIA COMPANY

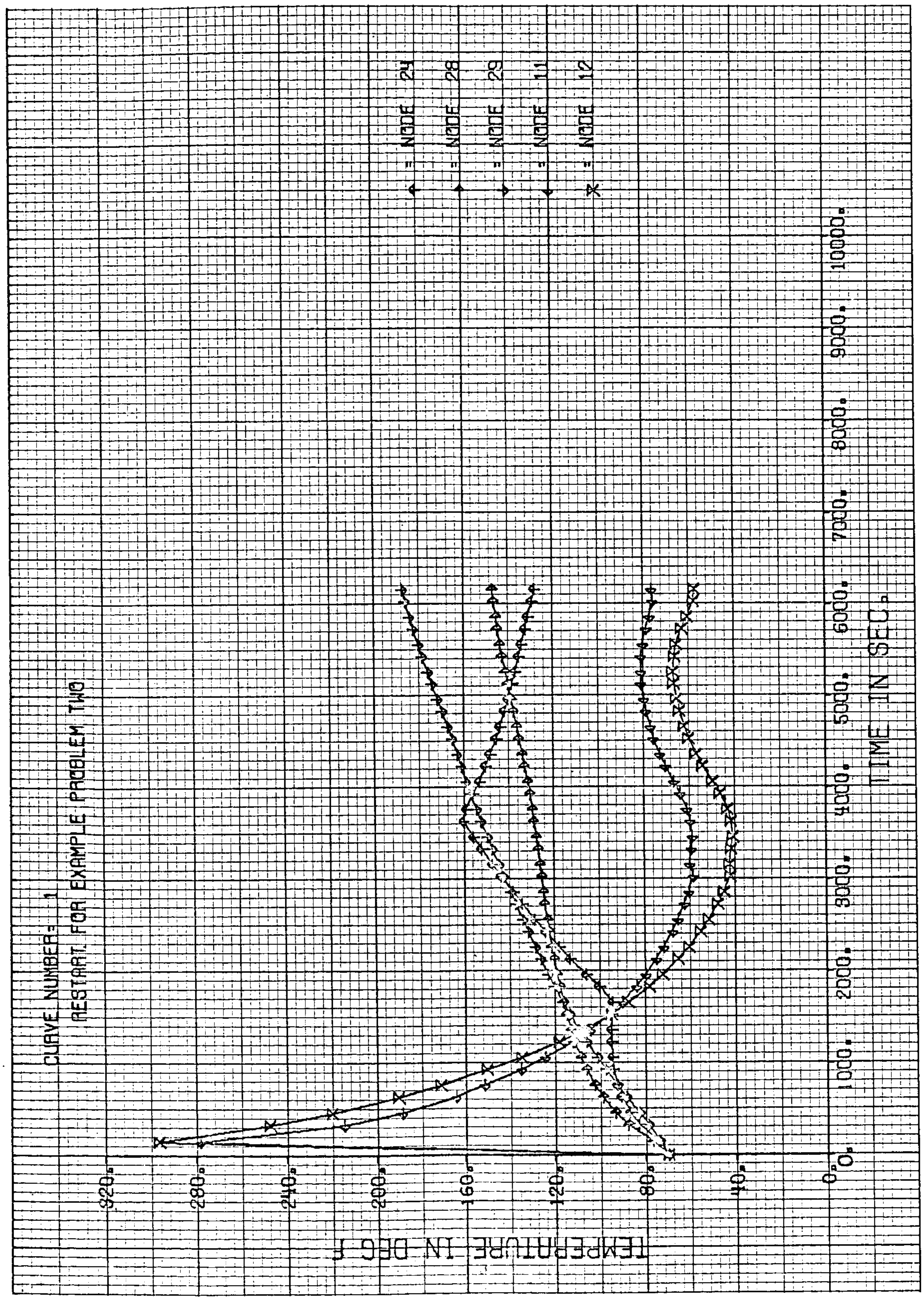


Figure 6-5. Temperature Plots for Example Problem 2 Restart

VII - GENERAL PROGRAM INFORMATION

INCORRECT DATA INPUT

Common Input Errors

In a program as complex as the Thermal Analyzer, a wide variety of input errors can occur. Some of the most frequent are:

1. Number of items on a DEC card does not match the number in columns 4 & 5. This usually occurs in the data block and results in the input data on that card being truncated after the number of items specified in columns 4 & 5 have been stored. However, if more items are asked for than exist, blank fields are read, and zeros are inserted into the data at that point.

2. Floating-point number is punched without a decimal point. Depending on the position of the number in the field, this results in a multiplication by a power of ten.

3. Integer number is not right-adjusted (blanks left at the right end of the field). This error results in a multiplication by a power of ten. Usually, the integers are either node or resistor numbers, and a multiplication by ten changes the entire circuit, and often results in values being stored into the wrong block or even completely out of the data area.

4. A capacitor is specified for a node whose temperature is to remain constant or is to be supplied by the FUNCT subroutine. Any capacitor listed in the capacitor block causes a heat balance to be performed at that node and the computed temperature to be stored, thus destroying the value assigned. This is true for zero-valued capacitors as well as positive capacitors.

Most other errors in input are checked by the program. However, the foregoing errors are impossible to detect as the current program makes use of the FORTRAN system library routines to read and convert the data.

Function Subroutine Errors

Certain errors occur fairly often in the FUNCT subroutine. These too, cannot be checked in the current program, because the Function subroutine (FUNCT) is compiled by FORTRAN.

One of the more common Function errors is incorrect use of the "old" and the "new" temperature blocks. For each listed capacitor, a new temperature is computed as a function of the capacitance, time step, its old temperature, the old temperatures of neighboring nodes, and any arbitrary heat inputs. In the special case of a zero-valued capacitor, the new temperature is a function only of the old temperatures of its neighboring nodes, and any other heat inputs. The new temperatures are computed during the heat balance. Then all temperatures in the new temperature block are moved into the old temperature block in preparation for the next cycle. Therefore, any value stored into the "old" temperature block will have an effect on the heat balances performed on itself and on its neighbors, but it will be replaced by its "new" value immediately thereafter. If it has no capacity, its new value will be its initial input value.

Another frequent Function subroutine error is misuse of the time step, $\Delta\theta$. Three values of time step are provided in the miscellaneous block, M(I). M(5) contains the actual time step used to arrive at the current time. Usually, at the time of a print, a short time step is needed to arrive at the print time, and M(5), if printed, gives a false indication of step size. Therefore the time step that would have been used, had this not been a print cycle, is provided in M(15). For cycles other than print cycles, M(5) = M(15). For certain purposes it is necessary to know the time step used in the previous heat balance. This quantity is provided in M(14).

Control of the printing interval by means of a value interpolated from a time-dependent curve is often a convenient device, but care must be taken in the choice of points for the curve. If, at time = θ, it is desired to change the printing interval from PI_1 to PI_2, one may use a step-function curve of the form:

$$\theta_{initial}, PI_1$$

$$\theta - PI_1, PI_1$$

$$\theta - PI_1, PI_2$$

$$\theta_{final}, PI_2$$

The print interval must be changed at or before one print interval before the time at which the new interval is to take effect, since the next time to print is decided at the time of printing, but before the function subroutine is entered.

Data Diagnostic

A large amount of data diagnostic is included in the program. The diagnostic routines always print a comment describing the type of error found. For example, if one of the node numbers mentioned in the resistor block is greater than the highest numbered node mentioned in the temperature block, the program sets an error flag to prevent execution of the program, and prints the comment:

RESISTOR∧∧NNN∧CONNECTS∧NODES∧∧MMM,∧∧LLL∧GREATER∧THAN∧MAX∧TEMP.NO.=∧JJJ

If the input data overflows the data storage region, P, the program sets an error flag to prevent execution, and prints the comment:

MMM∧DATA∧STORAGE∧EXCEEDED.∧TOTAL∧OF∧∧∧NNN

In these examples, the three-letter symbols, MMM, NNN, LLL, and JJJ represent numbers that appear in the diagnostic comment.

Unfortunately, not all errors can be checked by the program. For example, if a resistor is connected to a wrong node, but the node number is legitimate, the problem is still a legal problem, but not the problem that the user wants to solve. Note, again, that some errors cause the FORTRAN system to reject the problem, and this situation can mean that not all of the data has been examined. For example, an error in the Function or Print routines that deletes compilation means that no data has been read at all. A decimal point or other non-integer character in an integer field means that the data inspection is terminated at that point and the data following has not been examined. Any error that is illegal to FORTRAN will delete the problem at that point.

Some diagnostic has also been provided in the execution phase. This includes such errors as a zero or negative time step, attempted division by zero in certain situations, interpolation requested outside the range of the curve, and a number of other more specific errors which are checked in specific subroutines. Undoubtedly there are errors in this phase that could be but have not been diagnosed. But the diagnostic continually becomes more complete as new errors are encountered.

Data Debugging Routine

A short version of the compiler portion of the program has been devised, for the purpose of checking data only. This program, described in Appendix E, includes the same diagnostic included in the main program, excepting the error checks during execution.

PROGRAM CAPACITY

Currently there are three versions of the program. Two versions have 16000 storage locations available for data, but do not include the fluid storage and pressurization subroutines. One of these, Version A, has been set up to run short problems with minimum overlay. The other, Version B, has been set up with maximum overlay to allow inclusion of the largest possible function and print routines. Version C, including fluid storage and pressurization, has 13000 storage locations available for data.

Version A (Short Problems)

There are 16000 storage locations available for data. This is sufficient to handle the largest problem encountered to date, although some of the largest have had to be revised somewhat to fit into the available storage. This allows approximately 1000 temperatures and capacitors, 2500 resistors, and approximately 3500 words in the data blocks. Approximately 2700 storage locations are available for the FUNCT and PRINT subroutines. This means that the degree of sophistication allowable is strictly limited.

Version B (Maximum Problems)

The data storage available is the same as for version A, above. The linkage has been changed to allow maximum storage for the function and print routines. There are about 4800 storage locations available for these routines in version B. Additional space can be gained in two ways:

1. By eliminating unused subroutines, such as linear, parabolic and/or bivariate interpolation, or by eliminating the larger unused routines in dependent links, such as the radiation resistor matrix. Note that the largest routine actually used limits the possible saving.

2. By breaking the Function and/or Print routines into several functions, with unique names for each, and adding $INCLUDE cards at the proper places.

Version C (Pressurization)

13000 storage locations are available for standard input data in version C. (The dimension size is 14000, but approximately 1000 locations are used for special fluid storage and pressurization program storage.) For further details of version C, see Reference 9.

MACHINE EXECUTION TIME

The execution time for the Thermal Analyzer is almost wholly dependent on the particular problem to be solved. Small problems (depending also on the particular functions used) will often run faster than 0.01 min/cycle, where a cycle is defined as one full pass through the program. On the other end of the scale, the largest problems to date take about 0.25 min/cycle. The time required is a function of the number and type of arithmetic operations, the size of the network, the subroutine functions used, and the amount of overlaying required. The number of cycles required to complete execution is the time range, M(3) - M(2), divided by the computing interval, $\Delta\theta$. Generally, $\Delta\theta$ is a variable and is computed as the product of the network $(RC)_{min}$ and the factor stored in M(7). Unless changed by the user in the FUNCT subroutine,

LOCKHEED CALIFORNIA COMPANY

M(7) = 0.25. If the printing interval is less than the $(RC)_{min}$, the computing interval is determined as the product of the printing interval and M(7).

A quick estimate of $\Delta \theta$ may be obtained by multiplying M(7) by the smallest of the following three quantities:

1. The print interval.

2. The smallest capacitor times its smallest connecting resistor.

3. The smallest resistor times its smallest connecting capacitor.

This method is far from fool-proof, since in particular, radiation resistors are variable inversely with the cube of the temperatures of connected nodes. Also, computations 2 and 3 above, may not give the minimum RC value. In anticipation of a long run, perhaps the best method of getting a time estimate is to run the program for a very short time history and print out the time step or the number of cycles used.

APPENDIX A

ECKERT AERODYNAMIC HEATING

The Eckert Aerodynamic Heating subroutine is a valuable tool in performing convective heat transfer calculations for high-velocity flows. The program computes the aerodynamic heating rate when given trajectory, flow field, and air property data in a prescribed manner. Eckert's calculation procedure, or the reference temperature method, as it is commonly called, eliminates the dependence of skin friction, and hence wall heating rate, on variable fluid properties associated with high-velocity flow. This allows heat transfer calculations to be accomplished as with incompressible flow where property variations across the boundary layer are negligible.

ECKERT'S RECOMMENDED PROCEDURE

The recommended heating equations follow. Additional explanation can be found in Reference 5. The symbols used are defined in the table of nomenclature given at the end of this section.

For two-dimensional laminar flow over an isothermal and isobaric surface

$$St = 0.332\ (Re^*)^{-0.5}\ (Pr^*)^{-0.667} \qquad \text{(A-1)}$$

$$q_w = h\ (T_R - T_w) \qquad \text{(A-2)}$$

where the asterisk denotes property values to be evaluated at a reference temperature given by

$$T^* = 0.28\ T_e + 0.22\ T_R + 0.50\ T_w \qquad \text{(A-3)}$$

For two-dimensional turbulent flow over an isothermal and isobaric surface

$$St^{*} = 0.0296 \, (Re^{*})^{-0.2} \, (Pr^{*})^{-0.667} \qquad \text{(A-4)}$$

The wall heat transfer is calculated from equation A-2. The reference temperature given by equation A-3 is assumed valid for both laminar and turbulent flows.

MODIFICATION OF ECKERT'S RELATIONS FOR DIGITAL COMPUTING USE

In the form given in the previous section the equations do not lend themselves to computer calculation; therefore, they are modified as explained here:

For turbulent flow, from equation A-4:

$$h_T = 0.0296 \, \frac{k^{*}}{X} \, (Re^{*})^{0.8} \, (Pr^{*})^{0.333} \qquad \text{(A-5)}$$

Combining equation A-5 with equation A-2 results in

$$q_{wT} = \frac{0.0296}{X^{0.2}} \left[\frac{Re^{*}}{X} \; \frac{M_\infty}{M_\infty} \right]^{0.8} (Pr^{*})^{0.333} \; k^{*} \, (T_R - T_w) \qquad \text{(A-6)}$$

Using the Mach number definition

$$M = \frac{u}{c}$$

and the perfect gas relation

$$\frac{c}{c_\infty} = \left(\frac{T}{T_\infty} \right)^{0.5}$$

the Reynolds number term in equation A-6 is modified, resulting in the following expression:

$$\frac{Re^*}{X} = \left[\frac{Re_\infty}{X}\left(\frac{M_e}{M_\infty}\right)\left(\frac{T_e}{T_\infty}\right)^{0.5}\left(\frac{\rho_e}{\rho_\infty}\right)\right]\left(\frac{\rho^*}{\rho_e}\right)\left(\frac{\mu_\infty}{\mu_e}\right)\left(\frac{\mu_e}{\mu^*}\right) \tag{A-7}$$

Next, the viscosity relation

$$\frac{\mu_\infty}{\mu_e} = \left(\frac{T_\infty}{T_e}\right)^{0.69}$$

and perfect gas law

$$\frac{\rho^*}{\rho_e} = \frac{T_e}{T^*}$$

are used to modify equation A-7 to the following form:

$$\frac{Re^*}{X} = \left[\frac{Re_\infty}{X}\left(\frac{M_e}{M_\infty}\right)\left(\frac{\rho_e}{\rho_\infty}\right)\left(\frac{T_e}{T_\infty}\right)^{-0.2}\right]\left(\frac{Te}{T^*}\right)^{1.69} \tag{A-8}$$

Defining the Reynolds number ratio as

$$\frac{Re_e}{Re_\infty} = \left(\frac{M_e}{M_\infty}\right)\left(\frac{\rho_e}{\rho_\infty}\right)\left(\frac{T_e}{T_\infty}\right)^{-0.2}$$

the expression for q_w becomes

$$q_{wT} = \frac{0.0296}{X^{0.2}}\left[\left(\frac{Re_\infty}{XM_\infty}\right) M_\infty\left(\frac{Re_e}{Re_\infty}\right)\left(\frac{Te}{T^*}\right)^{1.69}\right]^{0.8}(Pr^*)^{0.333}$$

$$k^*\left(T_R - T_w\right) \tag{A-9}$$

By a similar process the expression for laminar flow becomes

$$q_{wL} = \frac{0.332}{X^{0.5}} \left[\left(\frac{Re_\infty}{XM_\infty}\right) M_\infty \left(\frac{Re_e}{Re_\infty}\right) \left(\frac{T_e}{T^*}\right)^{1.69} \right]^{0.5} \left(Pr^*\right)^{0.333} k^* \left(T_R - T_w\right) \tag{A-10}$$

Equations A-9 and A-10 are combined to give the programmed form

$$q_w = K\beta \left[\left(\frac{Re_\infty}{XM_\infty}\right) M_\infty \left(\frac{Re_e}{Re_\infty}\right) \left(\frac{Te}{T^*}\right)^{1.69} \right]^{a} \left(Pr^*\right)^{0.333} k^* M_\infty{}^{b} \left(T_R - T_w\right) \tag{A-11}$$

where the factor β has been included as an angle of attack modifier. Its use is optional and β will have a value of unity unless changed by the user as described below. The term $M_\infty{}^b$ is included to give the Eckert equation the same form as that used by another aerodynamic heating method, which is no longer being used. The recovery temperature is computed from the following:

$$T_R = T_e \left[1 + Pr^e \left(\frac{\gamma - 1}{2}\right) M_e{}^2\right] \tag{A-12}$$

where Pr^e is the temperature recovery factor. A constant value of Pr = 0.71 is used in the program.

PROGRAM INPUT

Several tables must always be provided in the data block when the Eckert heating routine is used. The designation numbers of these tables are permanently reserved and linear interpolation of the data is understood. The user must always provide tables 1, 3, 4, 5, 6, 8, 11, 12, and 13. If the angle of attack multiplier β is not understood to be unity, tables 2 and 7 must also be provided. The following list shows the composition of each table.

TABLE DESIGNATION NUMBER	INDEPENDENT VARIABLE	DEPENDENT VARIABLE
1	θ	H
3	H	$\log_{10}\left(\frac{Re_\infty}{XM_\infty}\right)$
4	T^*	$\log_{10}\left(Pr^{0.333}\ k\right)$
5	M_∞	$\frac{Re_e}{Re_\infty}$
6	M_∞	$\frac{T_e}{T_\infty}$
8	Constants a, b, e, γ in that order	
11	θ	M_∞
12	H	T_∞
13	M_∞	$\frac{M_e}{M_\infty}$
2	θ	α_R
7	α	β or constants K_W and K_L

Note that tables 3 and 12 contain only atmospheric data and as such are standard, reusable tables. This is also true of table 4, which contains thermodynamic and transport properties data. The trajectory is defined in tables 1 and 11, and the local flow field in tables 5, 6, and 13. Table 8 lists the constants a, b, e, and γ from equations 11 and 12. Using the Blasius expressions for skin friction (equations 1 and 4) and assuming the fluid is air with $\gamma = 1.4$, these constants are:

BOUNDARY LAYER	a	b	e	γ
laminar	0.5	0	0.5	1.4
turbulent	0.8	0	0.333	1.4

The angle of attack multiplier β is optional. Two possible methods exist for its use.

Method 1

In this method the angle of attack multiplier is input in table 7 as

$$\beta = f(\alpha)$$

where the angle of attack is obtained from

For the top surface, F = 0 $\qquad \alpha = -(\alpha_R - \alpha_o)$

For the bottom surface, F = 1 $\qquad \alpha = (\alpha_R - \alpha_o)$

Both F and α_o are input as part of the function callout as described below. The vehicle angle of attack α_R is given in table 2 as a function of time.

Method 2

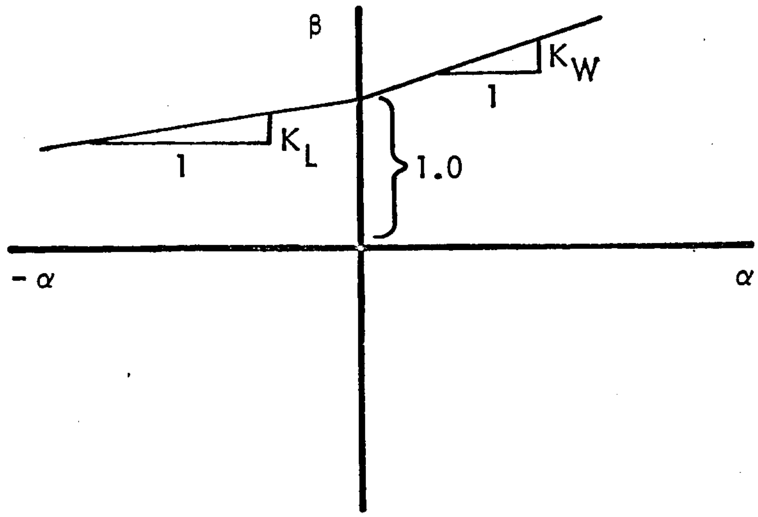
This method consists of inputing two slopes, K_W and K_L, in table 7, and having the machine compute the angle-of-attack multiplier by one of the following equations:

$\alpha > 0$ (a windward surface) $\qquad \beta = 1 + K_W \alpha$

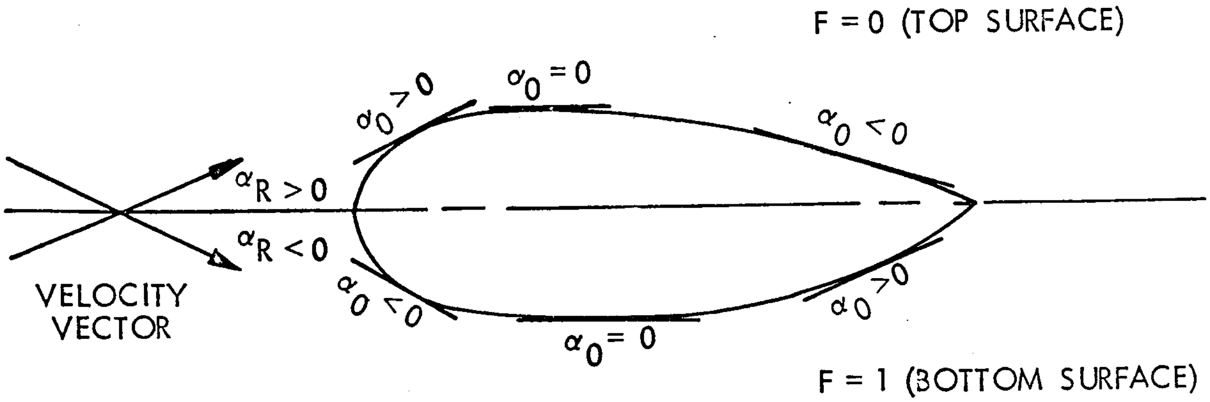
$\alpha < 0$ (a leeward surface) $\qquad \beta = 1 + K_L \alpha$

The vehicle angle of attack α_R is again given in table 2 as a function of time.

Shown below is a sketch of the β - α relation used in Method 2.



The sign convention used to determine α_R and α_o is indicated below.



One additional quantity is required as data input to the computer when using the Eckert Aerodynamic heating routine. This is the K factor in equation A-11 which, as can be seen by comparing equations A-9, A-10, and A-11, is given by the following:

Laminar flow $K = \frac{0.332A}{X^{0.5}}$ (A-13)

Turbulent flow $K = \frac{0.0296A}{X^{0.2}}$ (A-14)

The subroutine callout takes one of the following forms:

CALL EAH4 (j, K, H, α_o, F)

CALL EAH3 (j, K, α_o, F)

CALL EAH2 (j, K, H)

CALL EAH1 (j, K)

where

j is the node number for which q is computed.

K is the floating point constant K, or the location where K is stored.

H is the location where the value of the heat transfer coefficient is to be stored, if desired.

α_o is the vehicle surface angle in radians. If not specified, the angle-of-attack multiplier β is understood to be unity.

F is a flag which is 0 for an upper surface and 1 for a lower surface. F is vacant if α_o is vacant.

Several example callouts are:

1. CALL EAH1 (16,0.02)

   Means compute the aerodynamic heating to node 16 with K = 0.02 and β is understood to be 1.

2. CALL EAH1 (16,P(L + 3))

   Means compute the aerodynamic heating to node 16, and find the value of the K factor in the third storage location of the table whose designation number corresponds to L.

This arrangement allows K to be a variable expressed by some other function.

3. CALL EAH2 $\left(16,0.02,\ P(L20 + 2)\right)$

   Means the same as (1) except that the value of the heat transfer coefficient is to be stored in the second storage location of the table whose designation number corresponds to L20. This arrangement allows h to be called out in the print block.

4. CALL EAH3 (16,0.02, 0.2, 0)

   Means the same as (1) except that the angle of attack multiplier β is computed by Method I above with α_o = 0.2 radians, and the node is located on the upper surface.

5. CALL EAH4 $\left(16,\ P(L + 3),\ P(L20 + 2),\ 0.02,\ 1\right)$

   Means compute the aerodynamic heating to node 16, find the K factor in the third storage location of the table whose designation number corresponds to L, and store the value of the heat transfer coefficient in the second storage location of the table whose designation number corresponds to L20. The angle of attack multiplier β is computed by Method I with α_o = 0.2 radians, and the node is located on a bottom surface.

The recovery temperature in ^oF is automatically stored in the addressable element M(12).

LIMITATIONS OF ECKERT'S AERODYNAMIC HEATING METHOD

The aerodynamic heating subroutine described herein is strictly valid within certain limitations. These are summarized below:

1. In deriving Eckert's relations for digital computing the reference temperature form was used, rather than reference enthalpy. This assumes constant specific heat and Prandtl number.

2. Effects of dissociation and ionization are neglected in the heating relations. This limits the velocity range for which the equations are valid. For example, dissociation of the air molecules behind a normal shock begins at approximately Mach 6, for an altitude of 200,000 ft.

3. Near continuum flow is required, i.e., $H \leq 200{,}000$ ft.

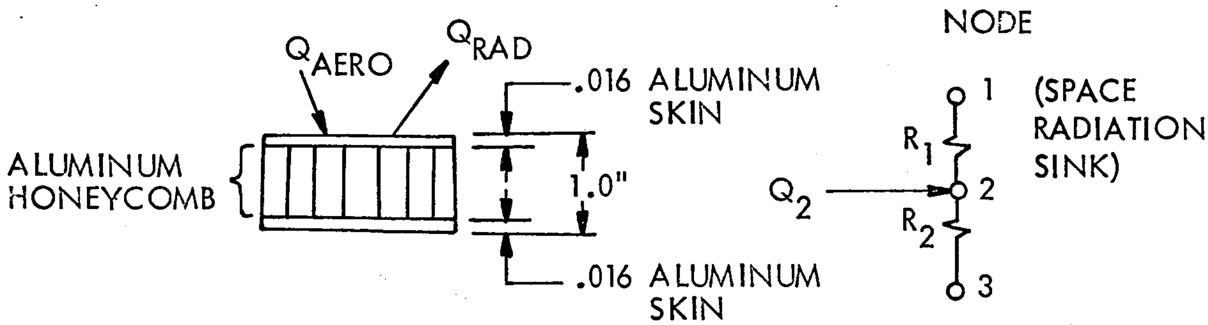
4. Two-dimensional flow along a constant pressure and temperature surface is assumed.

5. Steady, or slowly accelerating, flow is required.

The above assumptions probably limit the accuracy of the method to something like ±10% for a typical ascent trajectory.

EXAMPLE PROBLEM

To illustrate the procedure in using the Eckert heating subroutine, a simple ascent heating problem is set up. Suppose it is desired to calculate the transient temperature response of an aluminum honeycomb skin located at a distance of 45 ft from the nose of the vehicle. The vehicle trajectory (altitude and Mach number vs time) and local flow field parameters are assumed known. A sketch of a unit surface area of the skin and the corresponding thermal network are given below.



The following assumptions are made:

1. The boundary layer is turbulent for the first 135 sec after launch, at which time transition to laminar flow occurs instantaneously.

2. The external skin (node 2) radiates to a free space sink of 50°F.

3. The skin rear face is perfectly insulated.

4. The aluminum properties are:

$$\rho = 0.1 \text{ lb/in}^3$$

$$c_p = 0.2 \text{ Btu/lb°F}$$

$$\epsilon = 0.55$$

$$k_{HONEYCOMB} = 8 \frac{\text{Btu in.}}{\text{hr ft}^2 \text{°F}}$$

$$C_{HONEYCOMB} = 0.03 \frac{\text{Btu}}{\text{°F ft}^2}$$

Capacitors

Lump each skin with half of the honeycomb core

$$C2 = C3 = \rho V c_p + \frac{.03}{2}$$

$$= 0.1 \frac{\text{lb}}{\text{in}^3} \times .016\ (144)\ \text{in}^3 \times 0.2 \frac{\text{Btu}}{\text{lb °F}} + 0.015 = 0.0611 \frac{\text{Btu}}{\text{°F}}$$

Radiation Resistor

$$K1 = \frac{\epsilon FA}{3600} = \frac{0.55\ (1)\ (1)}{3600} = 1.53 \times 10^{-4}$$

Conduction Resistor

$$R2 = \frac{\ell}{kA} = \frac{0.968 \text{ in} \times 3600 \frac{\text{sec}}{\text{hr}}}{8 \frac{\text{Btu in.}}{\text{hr ft}^2 \text{ °F}} \times 1 \text{ ft}^2} = 436 \frac{\text{sec °F}}{\text{Btu}}$$

Eckert K Factor

$$K2_{turbulent} = \frac{0.0296A}{X^{0.2}} = \frac{0.0296(1)}{(45)^{0.2}} = 0.0138$$

$$K2_{laminar} = \frac{0.332A}{X^{0.5}} = \frac{0.332(1)}{(45)^{0.5}} = 0.0496$$

The program input is shown in Figure A-1. The value of the turbulent K factor is stored in the location T(4). Tables 1 and 11 contain, respectively, the vehicle altitude and Mach number vs time. These are assumed values and are typical of large liquid fuel boosters. Tables 3 and 11 contain $\log_{10}$ (Re_∞/XM_∞) and freestream temperature vs altitude for Patrick Air Force Base, Florida. These tables were constructed from the measured data published in Reference 6. Table 4 contains $\log_{10}$ ($Pr^{0.333}$ k) vs temperature, using the data of Reference 7. Thus, tables 3, 4, and 11 contain real properties data, applicable to any Cape Kennedy launch. Actually, the difference between the Patrick Air Force Base Standard Atmosphere and the 1962 U.S. Standard Atmosphere (Reference 8) is so slight that the data in these tables may be used with considerable confidence regardless of the launch site. Tables 5, 6, and 13 list assumed values for the flow field parameters (Re_e/Re_∞, T_e/T_∞, and M_e/M_∞) as a function of freestream Mach number. These values depend primarily on the vehicle geometry. The constants a, b, e, and γ applicable to a turbulent boundary layer are stored in Table 8. In the time block the initial and final times are set at 0 and 160 sec, with a print interval of 5 sec. For a typical ascent heating problem, the computing interval should not exceed 5 sec to prevent temperature oscillations.

The FUNCT subroutine should be self-explanatory. Card 6008 contains the radiation function and, if current time does not equal 140., all other functions are ignored with the exception of the Eckert Heating routine. The K factor is obtained from TN(4). If current time equals 140., the turbulent K factor stored in TN(4) is replaced by the laminar K factor (with a value 0.0496). Also, the constants a and e of table 8 are replaced by their laminar values.

Note that these changes are made at 140 sec, rather than at 135 sec when transition is assumed to occur. This is a result of the finite difference solution which, with a computing interval of $\Delta\theta$ sec, assumes that the value of each independent variable remains constant during the time $\theta - \Delta\theta$. Thus, if the exponent e stored in table 8 is changed at 140 sec, the new value is used by the EAH subroutine during the interval 135-140 sec.

The PRINT subroutine is set up to list the current time, node 2 temperature, node 3 temperature, recovery temperature, and aerodynamic heating rate on a single line of output. The first 4 items will appear in the F-field format, and the heating rate in the E-field format.

TABLE A-1

INPUT FOR AERODYNAMIC HEATING EXAMPLE PROBLEM (1 of 5)

GENERAL PURPOSE DATA SHEET — LOCKHEED-CALIFORNIA COMPANY, A DIVISION OF LOCKHEED AIRCRAFT CORPORATION

PREPARED | NAME | DATE | TITLE | PAGE OF | JOB NO. | CHECKED | WO | EWA | GROUP

Card									Text	Seq.
CID									ECKERT AERØDYNAMIC HEATING EXAMPLE PRØBLEM	1000
DEC04	1	50.	2	70.	4	0.0138			TEMPERATURES	1001
NBK										1999
DEC02	1	1	2	0.	2	2	3	436.	RESISTØRS	2001
NBK										2999
DEC02	2	0.0611	3	0.0611					CAPACITØRS	3001
NBK										3999
DEC	1								ALTITUDE VS TIME	40101
DEC06	0.	0.	20.	2000.	40.	8500.				2
DEC06	60.	24000.	70.	35000.	80.	48500.				3
DEC06	90.	63000.	100.	80000.	110.	100000.				4
DEC06	120.	123000.	130.	150000.	140.	181000.				5
DEC05	150.	212000.	160.	247000.	0.					6
DEC	-1									40107
DEC	3								LØG(RE/XM) VS ALTITUDE FØR PATRICK AFB	40301
DEC06	0.	6.832	5000.	6.775	10000.	6.712				02
	15000.	6.646	20000.	6.582	25000.	6.517				03
	30000.	6.453	35000.	6.382	40000.	6.310				04
	45000.	6.230	50000.	6.134	55000.	6.024				05
	60000.	5.910	65000.	5.790	70000.	5.672				06
	75000.	5.561	80000.	5.447	85000.	5.342				07
	90000.	5.241	95000.	5.127	100000.	5.029				08
	105000.	4.922	110000.	4.815	115000.	4.708				09
	120000.	4.603	125000.	4.507	130000.	4.410				10
DEC06	135000.	4.305	140000.	4.216	145000.	4.125				40311

FORM DX 7924.1

LOCKHEED CALIFORNIA COMPANY

TABLE A-1

INPUT FOR AERODYNAMIC HEATING EXAMPLE PROBLEM (2 of 5)

GENERAL PURPOSE DATA SHEET — LOCKHEED-CALIFORNIA COMPANY, A DIVISION OF LOCKHEED AIRCRAFT CORPORATION

PREPARED | NAME | DATE | TITLE | PAGE OF | JOB NO. | CHECKED | WO | EWA | GROUP

FORM DX 7924.1

1–9	10–14	15–24	25–34	35–44	45–54	55–64	65–69	70–72
DEC06	150000.	4.038	156000.	3.927	175000.	3.628		40312
DEC06	190000.	3.424	200000.	3.280	210000.	3.129		3
DEC06	220000.	2.969	230000.	2.800	240000.	2.621		4
DEC06	250000.	2.429	263000.	2.156	280000.	1.704		5
DEC03	300000.	1.198	0.					6
DEC	-3							40317
DEC	4	LØG (PR**.333 K) VS TEMPERATURE (IN DEG. RANKINE)						40401
DEC06	250.	-5.75	332.	-5.6198	500.	-5.4495		2
DEC06	700.	-5.3388	900.	-5.2691	1000.	-5.2253		3
DEC06	1100.	-5.1981	2300.	-5.0177	5400.	-4.8016		4
DEC06	7200.	-4.7205	9000.	-4.6743	10000.	-4.6476		5
DEC01	0.							6
DEC	-4							40407
DEC	5	REYNØLDS NUMBER RATIØ VS MACH NUMBER						40501
DEC06	0.	1.000	1.0	1.000	1.5	0.925		2
DEC06	2.0	0.772	2.5	0.621	3.0	0.480		3
DEC06	3.5	0.390	4.0	0.320	4.5	0.267		4
DEC06	5.0	0.216	5.8	0.165	6.5	0.129		5
DEC05	7.5	0.095	10.0	0.070	0.			6
DEC	-5							40507
DEC	6	TEMPERATURE RATIØ VS MACH NUMBER						40601
DEC06	0.	1.00	1.0	1.00	1.3	1.02		2
DEC06	2.0	1.07	2.5	1.19	3.0	1.30		3
DEC06	4.0	1.70	4.5	1.90	5.0	2.15		4
DEC06	5.8	2.51	6.5	2.92	7.5	3.48		40605

LOCKHEED CALIFORNIA COMPANY

TABLE A-1

INPUT FOR AERODYNAMIC HEATING EXAMPLE PROBLEM (3 of 5)

FORM DX 7924.1

GENERAL PURPOSE DATA SHEET — LOCKHEED-CALIFORNIA COMPANY, A DIVISION OF LOCKHEED AIRCRAFT CORPORATION

PREPARED | NAME | DATE | TITLE | PAGE OF | JOB NO. | CHECKED | WO | EWA | GROUP

SEQ. (77-80)	1	10	15	25	35	45	55	65	70-72	ID (73-76)
	DEC03		10.	5.00	0.				40606	
	DEC	-6							40607	
	DEC	8	CONSTANTS A,B,E,AND GAMMA						40801	
	DEC04		0.8	0.	0.333	1.4			40802	
	DEC	-8							40803	
	DEC	11	FREESTREAM MACH NUMBER VS TIME						41101	
	DEC06		0.	0.01	20.	0.21	40.	0.44	2	
	DEC06		60.	0.94	70.	1.32	80.	1.74	3	
	DEC06		90.	2.28	100.	2.83	110.	3.50	4	
	DEC06		120.	4.47	130.	4.93	140.	6.02	5	
	DEC05		150.	7.72	160.	8.92	0.		6	
	DEC	-11							41107	
	DEC	12	TEMPERATURE VS ALTITUDE FOR PATRICK AFB						41201	
	DEC06		0.	534.	20000.	471.	40000.	392.	2	
			45000.	377.	50000.	369.	55000.	366.	3	
			60000.	370.	80000.	397.	100000.	414.	4	
			110000.	430.	140000.	484.	160000.	509.	5	
			175000.	509.	190000.	473.	210000.	425.	6	
	DEC06		220000.	400.	240000.	352.	260000.	300.	7	
	DEC05		280000.	298.	300000.	298.	0.		8	
	DEC	-12							41209	
	DEC	13	MACH NUMBER RATIO VS MACH NUMBER						41301	
	DEC06		0.	1.00	1.0	1.00	1.5	0.97	2	
	DEC06		2.0	0.93	2.5	0.85	3.0	0.80	3	
	DEC06		3.5	0.73	4.0	0.68	4.5	0.63	41304	

TABLE A-1

INPUT FOR AERODYNAMIC HEATING EXAMPLE PROBLEM (4 of 5)

FORM DX 7924.1

GENERAL PURPOSE DATA SHEET

LOCKHEED-CALIFORNIA COMPANY
A DIVISION OF LOCKHEED AIRCRAFT CORPORATION

PREPARED | NAME | DATE | TITLE | PAGE OF | JOB NO.

CHECKED | WO | EWA | GROUP

```
SEQ.  1                                                                  ID
DEC06          5.0     0 60      5.8     0.55     6.5     0.52     41305
DEC05          7.5     0 47     10.0     0.40     0                41306
DEC     -13                                                        41307
NBK                                                                49999
DEC03           5.     160.      0.                TIME BLOCK       5001
NBK                                                                 5999

6001 $IBFTC FUNCT    NODECK,NOREF
6002           SUBROUTINE FUNCT
6003           COMMON P,M
6004           DIMENSION P(16000),M(16),TN(1),T(1),R(1),Q(1),RC(1)
6005           EQUIVALENCE (P(1),TN(1)),(P(5),T(1)),(P(9),R(1)),(P(11),C(1)),
6006          1 (P(14),Q(1)),(P(17),RC(1))
6007           REAL M,LIN
6008           R(1)=RAD(1,2,1.53E-4)
6009           IF (M(1).EQ.140.) GO TO 1
6010           GO TO 2
6011      1    TN(4)=0.0496
6012           L8=LOC(8)
6013           P(L8+1)=0.5
6014           P(L8+3)=0.5
6015      2    CALL EAHI(2,TN(4))
6016           RETURN
6017           END
```


LOCKHEED
CALIFORNIA COMPANY

TABLE A-1

INPUT FOR AERODYNAMIC HEATING EXAMPLE PROBLEM (5 of 5)

GENERAL PURPOSE DATA SHEET — LOCKHEED-CALIFORNIA COMPANY, A DIVISION OF LOCKHEED AIRCRAFT CORPORATION

PREPARED | NAME | DATE | TITLE | PAGE OF | JOB NO. | CHECKED | WO | EWA | GROUP

FORM DX 7924.1

```
7001 $IBFTC PRINT    NØDECK,NØREF
7002       SUBRØUTINE PRINT
7003       CØMMØN P,M
7004       DIMENSIØN P(16000),M(16),TN(1),T(1),R(1),C(1),Q(1),RC(1)
7005       EQUIVALENCE (P(1),TN(1)),(P(5),T(1)),(P(9),R(1)),(P(11),C(1)),
7006      1 (P(14),Q(1)),(P(17),RC(1))
7007       REAL M
7008       WRITE (6,10) M(1),T(2),T(3),M(12),Q(2)
7009    10 FØRMAT (1H0,4F18.0,1E18.8)
7010       RETURN
7011       END
```

NOMENCLATURE FOR APPENDIX A

a	Exponent in equation A-11
A	Heated surface area, ft^2
b	Mach number exponent in equation A-11
c	Speed of sound, ft/sec
c_p	Specific heat, Btu/lb°F
C	Thermal capacitance, Btu/°F
e	Prandtl number exponent in equation A-12
h	Heat transfer coefficient, Btu/ft^2 sec°F
H	Altitude, ft
k	Thermal conductivity, Btu/ft sec°F
K	Eckert K factor defined by equations A-13 and A-14
K_W, K_L	Surface slopes
Pr	Prandtl number
q.	Surface heat flux, Btu/ft^2 sec
Q	Surface heat flux, Btu/sec
R	Thermal resistor, sec°F/Btu
Re	Reynolds number
St	Stanton number
T	Temperature, °R
u	Velocity, ft/sec

NOMENCLATURE FOR APPENDIX A (Continued)

V	Volume of lump, in^3
X	Equivalent boundary layer length, ft
α	Local angle of attack, radians
α_R	Vehicle body axis angle of attack, radians
α_o	Vehicle surface angle, radians
β	Angle of attack multiplier
μ	Viscosity, lb/ft sec
γ	Ratio of specific heats
ϵ	Emissivity
θ	Time, sec
ρ	Density, lb/ft^3

Subscripts

e	Boundary layer edge
R	Recovery
w	Wall
∞	Freestream

Superscripts

*	Evaluated at the reference temperature given by equation A-3

APPENDIX B

ABLATION

The ablation subroutine is an extension of the Thermal Analyzer Program to solve problems involving transient and steady-state ablation. The function calculates the amount of material that is ablated (if the temperature is above the ablation temperature) as a function of time as part of the heat balance using the finite difference technique. The amount of material that is ablated is then removed from the network by a switching arrangement, while the normal transient temperature distribution is calculated for the remaining ablation material and back-up structure. The ablation function, like most other functions, must have entries in the data block and FUNCT subroutine with the data block being a table of information that the function needs to perform the ablation process.

The ablation function will only ablate a network that has the nodes at the boundaries of the lumps. The nodes and their information must be put into the table and listed in the capacitor block in the order in which they will ablate, but the node numbers need not be assigned consecutively. The machine calculates the resistors and capacitors in the ablating layer at each cycle, allowing for material removal. All material properties, e.g., thermal conductivity, heat ablation, etc., may be made a function of any variable. The three limitations of the program are:

1. Only ablation from a solid to a gas without a char layer may be considered.
2. Only one-dimensional heat transfer can be handled in the ablation material.
3. Ablation network resistors and capacitors may not be input by the geometric methods discussed on pages 5-6 and 5-8.

Some of the items that make the program very flexible are:

1. Heat of ablation and temperature of ablation may be made a function of temperature, heating rate, enthalpy, or time.

2. The ablating system may be stopped and started an infinite number of times.

3. It can handle the case of any number of layers of different ablation materials.

4. Many ablating systems can be handled at one time, either separately or linked together at a nonablating node.

5. ρ, c_p, and k may be a function of any variable.

6. Radiation from the ablating node with emissivity as a variable can be handled by the program.

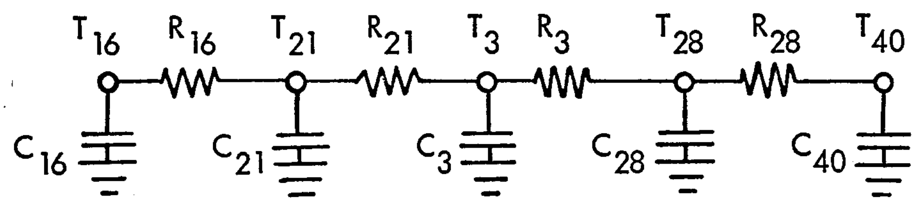
PROGRAM INPUT

The ablation subroutine callout is

CALL ABL(i,N)

where i is the first (surface) ablating node number and N is the designation number of the table which contains the ablator properties used for the calculation of the capacitors and resistors in the ablating network.

To illustrate the use of this function, consider the example of a 4-link ablating chain sketched below.



(NOTE the correspondence between node and resistor numbers. This is an absolute necessity when using the ablation function.)

Node 16 is the first ablating node; i.e., the material is ablated from left to right in the figure. The network to the left of C_{16} is normally a description of the heat input (which would include radiation outward) to the outermost ablating material. The network to the right of C_{28} (starting with node 40) would be the structure to which the ablating material is attached.

LOCKHEED
CALIFORNIA COMPANY

The ablation table contains the list of capacitor-resistor links in the same order that ablation will take place. Note that although the ablation table would be a sufficient description of the ablation network, this ablation network must have been described in the resistor block and capacitor block just as any part of the network was described. However, the initial values assigned to the ablation resistors and capacitors will be replaced by those found by computation using the tabular properties and hence can be input as zero.

The leading line of the table must be entered as 10 zeros, the two trailing columns (columns 9 and 10) must be entered as zeros, and the final line consists of one zero. These bordering zeros result in a convenient matrix reference scheme for the user of the ablation function as well as providing necessary storage for the ablation program. The first 8 entries on the intermediate lines contain, in order, the following information:

1. Node and corresponding resistor designation numbers
2. Ablator thermal conductivity - Btu/hr ft °F
3. Ablator density - lb/ft^3
4. Ablator specific heat - Btu/lb °F
5. Cross-sectional area normal to heat flow - ft^2
6. Initial resistor length - in.
7. Ablation temperature - °F
8. Heat of ablation - Btu/lb

The ablation table (designated as Table 4 in this example) is entered as

0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
16.	K_{16}	ρ_{16}	C_{16}	A_{16}	x_{16}	$T_{16,a}$	$H_{16,a}$	0.	0.
21.	K_{21}	ρ_{21}	C_{21}	A_{21}	x_{21}	$T_{21,a}$	$H_{21,a}$	0.	0.
3.	K_3	ρ_3	C_3	A_3	x_3	$T_{3,a}$	$H_{3,a}$	0.	0.
28.	K_{28}	ρ_{28}	C_{28}	A_{28}	x_{28}	$T_{28,a}$	$H_{28,a}$	0.	0.
0.									

where the various quantities may be identified by reference to the above list. Each entry must be in floating point. During execution, the ninth column will be updated as ablation progresses and will contain the accumulative amount of material (in in.) ablated at that particular node. Consequently, in the PRINT subroutine, a WRITE statement listing the addresses

P(L4+19), P(L4+29), P(L4+39), P(L4+49)

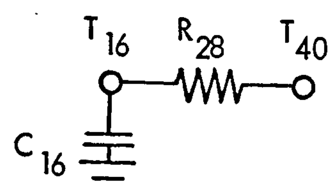
where

L4 = LOC (4)

will cause to be printed the total material ablated at each node as a function of time. To aid in data reduction, each time a new link begins to ablate, and each time a node stops ablating, the regular output block is automatically printed.

There is no restriction on the number of ablating links in an ablating chain or on the number of separate ablating chains in a particular case. Further, there is no restriction on the number of resistors going into the left-most capacitor (in this case C_{16}). However, each interior ablating node must have exactly one resistor to the left and one resistor to the right. If the terminal ablating node is only "half ablating" (input by setting $T_{28,a}$ equal to a fictitiously large number), there is no restriction on the number of resistors attached to the terminal ablating node.

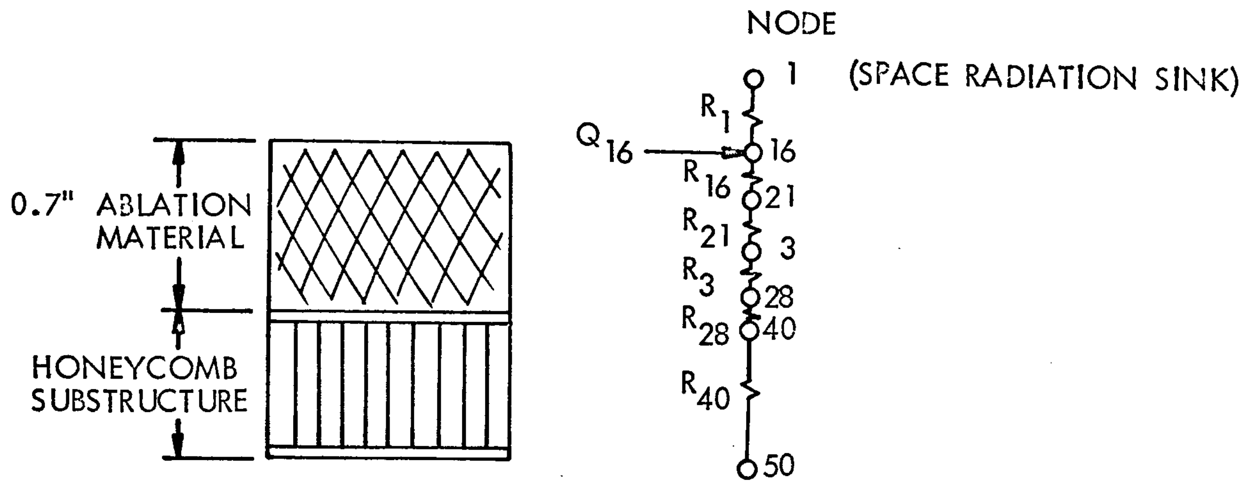
As the material ablates from left to right, the outer boundary is considered to move to the right. Thus, in the example capacitor number 16 continually moves; and if all nodes are completely ablated away, node numbers 21, 3, and 28 and resistor numbers 16, 21, and 3 will vanish from the circuit. Then the circuit will consist merely of



Again we have assumed that the right-hand part of the original C_{28} was non-ablating material so that the ablation stops at this point.

EXAMPLE PROBLEM

To illustrate the procedure in using the ablation subroutine, the following simple problem is set up. Consider the application of a low temperature subliming ablator near the nose region of a highly accelerating solid fuel rocket. The ascent heating pulse is assumed known. A sketch of a unit surface area of the skin and the corresponding thermal network are shown below.



A 4-link ablating chain is set up using the node and resistor designation numbers of the example discussed in the preceding section. The following assumptions are made:

1. The outermost surface of the ablator (node 16) radiates to a free space sink of 50°F. The surface emissivity is 0.8 resulting in a radiation K factor of 2.22×10^{-4}.

2. Conduction resistor 40 connects the substructure skin nodes 40 and 50. R(40) has a value of 2000 sec °F/Btu, and each skin node has a capacitance of 0.10 Btu/°F.

3. The ablator properties and dimensions are:

   Thermal conductivity = 0.08 Btu/hr ft °F at 0°F

   = 0.12 Btu/hr ft °F at 600°F

LOCKHEED CALIFORNIA COMPANY

Density	=	66 lb/ft^3
Specific heat	=	0.35 Btu/lb °F
Cross-sectional area	=	1.0 ft^2
Initial lump thickness	=	0.2 in. for resistors 16, 21, and 3
	=	0.1 in. for resistor 28
Ablation temperature	=	535°F
Heat of Ablation	=	600 Btu/lb

The program input is shown in Table B-1. Dummy values are entered for the ablation material resistors and capacitors since these are computed each cycle by the ABL subroutine. The assumed aerodynamic heat pulse is shown as curve 1 of the data block. The ablation table 4 is entered using the assumed properties listed above. The ablator conductivity is shown in curve 5. The initial and final times are set at 0 and 100 seconds, with a print interval of 2.5 sec.

In the FUNCT subroutine, card 6008 calls for linear interpolation of curve 1 to obtain the aerodynamic heating to node 16. On cards 6010-6013, the values of thermal conductivity listed in table 4 are updated each cycle by interpolating curve 5, using as independent variable the average temperature of the nodes to which the particular resistor is attached. Card 6014 calls in the Ablation subroutine, and card 6015 specifies the space radiation function. The radiation flux is computed and stored in the variable QRAD16 by subtracting the space sink temperature from the wall temperature, and dividing by the space radiation resistor.

The PRINT subroutine calls for the output to be listed in six columns, with the corresponding time appearing by itself to the left of the first line of output. The first line will consist of the temperatures of nodes 16, 21, 3, 28, 40 and 50, in that order. The second line will list the accumulated amount of material ablated at nodes 16, 21, 3, and 28; the aerodynamic heating rate; and the radiation loss to space.

TABLE B-1

INPUT FOR ABLATION EXAMPLE PROBLEM (1 of 3)

GENERAL PURPOSE DATA SHEET — LOCKHEED-CALIFORNIA COMPANY, A DIVISION OF LOCKHEED AIRCRAFT CORPORATION

FORM DX 7924.1

PREPARED | NAME | DATE | TITLE | PAGE OF | CHECKED | JOB NO. | WO | EWA | GROUP

1–8	10	15	20	25	30	35	40	45	50–55	60–65	70–72
CID		ABLATION EXAMPLE PROBLEM									1000
DEC01	1		50.								1001
DEC02	2		70.	50		70.					1002
NBK											1999
DEC02	1	1	16		0.	16	16	21	0.		2001
DEC02	21	21	3		0.	3	3	28	0.		2002
DEC02	28	28	40		0.	40	40	50	2000.		2003
NBK											2999
DEC03	16		0.	21		0.	3		0.		3001
DEC03	28		0.	40		.1	50		.1		3002
NBK											3999
DEC	1		AERODYNAMIC HEATING TO NODE 16								4001
DEC06		0.		0.		15.		0.	25.	3.5	4002
DEC06		35.		12.		40.		16.	50.	19.	4003
DEC06		60.		16.		75.		4.	90.	1.	4004
DEC03		100.		0.		0.					4005
DEC	-1										4006
DEC	4		ABLATION TABLE								4007
DEC06		0.		0.		0.		0.	0.	0.	4008
		0.		0.		0.		0.	16.	0.08	4009
		66.		0.35		1.		0.1	535.	600.	4010
		0.		0.		21.		0.08	66.	0.35	4011
		1.		0.2		535.		600.	0.	0.	4012
		2.		0.08		66.		0.35	1.	0.2	4013
		[illegible]		[illegible]		0.		0.	28.	0.08	4014

LOCKHEED CALIFORNIA COMPANY

TABLE B-1

INPUT FOR ABLATION EXAMPLE PROBLEM (2 of 3)

GENERAL PURPOSE DATA SHEET — LOCKHEED-CALIFORNIA COMPANY, A DIVISION OF LOCKHEED AIRCRAFT CORPORATION

PREPARED | NAME | DATE | TITLE | PAGE OF | JOB NO.
CHECKED | WO | EWA | GROUP

FORM DX 7924.1

```
SEQ.  1-72                                                                   ID
      DEC06        66.       0.35      1.        0.2      535.      600.    4015
      DEC03        0.        0.        0.                                   4016
      DEC      -4                                                           4017
      DEC       5          CΦNDUCTIVITY VS TEMPERATURE                      4018
      DEC05        0.        0.08     600.       0.12      0.               4019
      DEC      -5                                                           4020
      NBK                                                                   4999
      DEC03        2.5       100.      0.                                   5001
      NBK                                                                   5999
6001  $IBFTC FUNCT    NΦDECK,NΦREF
6002        SUBRΦUTINE FUNCT
6003        CΦMMΦN P,M
6004        DIMENSIΦN P(16000),M(16),TN(1),T(1),R(1),C(1),Q(1),RC(1)
6005        EQUIVALENCE (P(1),TN(1)),(P(51),T(1)),(P(101),R(1)),(P(141),C(1)),
6006       1 (P(191),Q(1)),(P(241),RC(1))
6007        REAL M,LIN
6008        Q(16)=LIN(M(1),1)
6009        L4=LΦC(4)
6010        P(L4+12)=LIN((T(16)+T(21))/2.,5)
6011        P(L4+22)=LIN((T(21)+T(3))/2.,5)
6012        P(L4+32)=LIN((T(3)+T(28))/2.,5)
6013        P(L4+42)=LIN((T(28)+T(40)/2.,5)
6014        CALL ABL(16,4)
6015        R(1)=RAD(1,16,2.22E-4)
```


LOCKHEED CALIFORNIA COMPANY

TABLE B-1

INPUT FOR ABLATION EXAMPLE PROBLEM (3 of 3)

GENERAL PURPOSE DATA SHEET — LOCKHEED-CALIFORNIA COMPANY, A DIVISION OF LOCKHEED AIRCRAFT CORPORATION

PREPARED | NAME | DATE | TITLE | PAGE OF | JOB NO. | CHECKED | WO | EWA | GROUP

```
QRAD16=(T(16)-T(1))/R(1)                                                6016
RETURN                                                                  6017
END                                                                     6018

$IBFTC PRINT    NØDECK,NØREF                                            7001
      SUBRØUTINE PRINT                                                  7002
      CØMMØN P,M                                                        7003
      DIMENSIØN P(16000),M(16),TN(1),T(1),R(1),C(1),Q(1),RC(1)          7004
      EQUIVALENCE (P(1),TN(1)),(P(51),T(1)),(P(101),R(1)),(P(141),C(1)),7005
     1 (P(191),Q(1)),(P(241),RC(1))                                     7006
      REAL M                                                            7007
      L4=LØC(4)                                                         7008
      WRITE (6,1) M(1),T(16),T(21),T(3),T(28),T(40),T(50),              7009
     1 P(L4+19),P(L4+29),P(L4+39),P(L4+49),Q(16),QRAD16                 7010
    1 FØRMAT (1H0,7E18.8/(19X,6E18.8))                                  7011
      RETURN                                                            7012
      END                                                               7013
```

FORM DX 7924.1

LOCKHEED CALIFORNIA COMPANY

APPENDIX C

DIGITAL COMPUTER SOLUTION FOR PASSAGE AIR AND WALL TEMPERATURES

The basic approach in the computation of passage air and wall temperatures is the lumped parameter technique widely employed in the electrical analog solution of problems involving partial differential equations. The network for convection at a surface where there is no appreciable variation of the fluid temperature in the flow direction is simply a resistance, R_c, connecting each of the node points representing the surface temperatures of the solid to a node point representing the fluid bulk temperature, where

$$R_c = \frac{1}{h A_c}$$

h = heat transfer coefficient

A_c = area for convective heat transfer per surface temperature node

Such is not the case for the network for convection from a gas to a surface where the convective heat transfer gives rise to appreciable variations of the gas temperature in the flow direction, and the analysis of this situation is therefore presented in somewhat greater detail here. The partial differential equation describing the variation of fluid bulk temperature, T, with distance in the flow direction, X, and time, θ, is

$$\left(\frac{\rho c_v A_c}{p}\right)\left(\frac{\partial T}{\partial \theta}\right) + \left(\frac{W c_p}{p}\right)\left(\frac{\partial T}{\partial X}\right) + h (T - T_i) = 0$$

where

T_i = passage wall temperature

V = fluid mean velocity

W = fluid flow rate

c_p = fluid specific heat at constant pressure

c_v = fluid specific heat at constant volume

h = heat transfer coefficient

p = perimeter of cross-section for heat transfer

A_c = cross-sectional area of passage

The rates of change of flow work and kinetic energy of the fluid are neglected in the preceding equation as these terms are small compared with the rate of change of the internal energy of the fluid and the heat transfer rates as represented by the first and second, and the third terms, respectively.

The straightforward application of the lumped parameter technique entails the substitution of the equivalent difference for the distance derivative, $\left(\frac{\partial T}{\partial X}\right)$, viz.

$$\left(\frac{\partial T}{\partial X}\right) \approx \frac{T_n - T_{n-1}}{(\delta X)}$$

in the equation above to yield

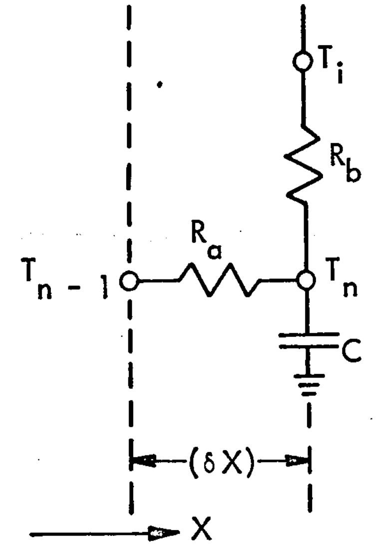
$$\left\{\rho\, c_v\, A_c\, (\delta X)\right\}\left(\frac{d\,T_n}{d\,\theta}\right) + (W\, c_p)\,(T_n - T_{n-1}) + \left\{hp\,(\delta X)\right\}(T_n - T_i) = 0$$

The network representation for this equation is as shown below, where

$C = \rho\, c_v\, A_c\, (\delta X)$

$R_a = \dfrac{1}{W\, c_p}$

$R_b = \dfrac{1}{hp\,(\delta X)}$



Application of this lumped parameter network to the air passage of the problem at hand (and to heat exchanger passages in general where the fluid is a gas) necessitates the use of a very large number of node points for a moderate accuracy in the problem solution. It is shown below from a consideration of a simplified network for a typical element of the complete representation that the capacitor shown in the circuit above may be removed from the circuit without significantly affecting the accuracy of the solution. This in itself does not relieve the requirement for a large number of node points; however, it does permit the term in the original partial differential equation corresponding to this capacitance, $(\rho\, c_v\, A_c/p)\,(\partial T/\partial \theta)$, to be removed. Once this term is removed a more refined approximation than that employed above may be substituted for the second term, $(W\, c_p/p)\,(\partial T/\partial X)$; whence, a smaller number of node points will produce an equivalent accuracy.

A slice of thickness (δX) of the portion of an air passage included between planes of symmetry is illustrated in Figure C-1. Also illustrated is the equivalent network considering variations of the wall temperature in the direction of airflow only. The triangular symbol schematically indicates a cathode follower (functionally a vacuum tube amplifier with unity gain). The purpose of this component is to reproduce the potential (temperature) of a given node point at a second node point without drawing any appreciable current (heat flux) from the first. If cathode followers are not inserted as

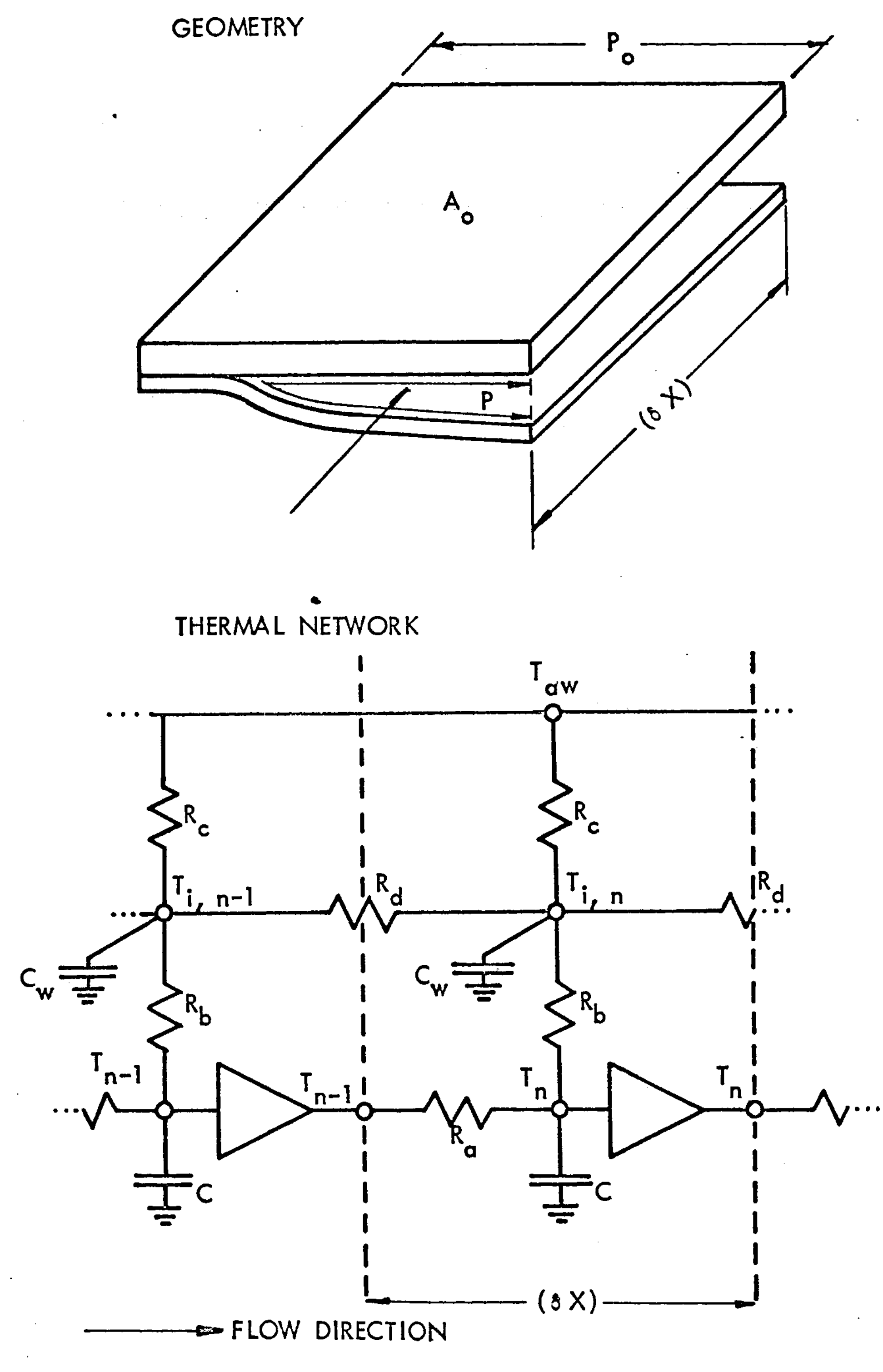


Figure C-1. Physical Geometry and Corresponding Thermal Network for a Portion of an Air Passage indicated, the equations describing the network response will contain terms not present in the lumped parameter equations of the thermal system. The presence of cathode followers is rationalized on a physical basis by noting that cathode followers isolate air node point temperatures from any effects of convective heat transfer at downstream nodes but not from upstream nodes in accordance with the physical facts.

R_a, R_b, and C are computed as indicated previously and

T_{aw} = adiabatic wall temperature

h_o = external heat transfer coefficient

A_w = shaded area of the figure above

A_o = p_o (δX) = external area for heat transfer

c_w = specific heat of wall

ρ_w = density of wall

k_w = thermal conductivity of wall

$$R_c = \frac{1}{h_o A_o}$$

$$C_w = \rho_w c_w A_w (\delta X)$$

If the removal of the capacitances representing the rate of change of internal thermal energy of the passage air with respect to time is not to have a significant effect on the solution, an obvious requirement is that these capacitances be very small compared with other capacitances of the system (the passage wall capacitances), i.e.

$$\frac{C}{C_w} = \frac{\rho c_v A_c}{\rho_w c_w A_w} \ll 1$$

An additional consideration is involved as a result of the fact that the effect of changes of passage inlet temperatures are not reflected immediately at downstream points due to the finite time required for the air to traverse the passage. The electrical capacitors representing the thermal capacity of the air, C, the resistances, R_a, and the cathode followers, form an electrical circuit which may be described as a delay line. The delays in this case represent the time required for an air particle to pass from the inlet of the passage to downstream node points. Removal of the capacitors representing the thermal capacity of the air eliminates the delays and is thus tantamount to assuming air particles traverse the passage in zero time. This is justified if the actual time of traverse, θ_L, is very small compared to a suitable measure of the time required for a significant change of the wall temperatures. Such a measure is the time constant, τ, (resistance-capacitance product) of a wall temperature node point for changes of the air temperature. (The term, time constant, is employed in the usual sense here, i.e., the time required for a node point potential to respond within 1/e or 37% of the final steady-state value resulting from a step change of an adjacent node point potential.)

$$\tau = R_b C_w = \frac{\rho_w c_w A_w}{hp}$$

$$\theta_L = \frac{L}{V}$$

where

L = total passage length

The second condition that must be satisfied is then

$$\frac{\theta_L}{\tau} = \frac{L\, hp}{\rho_w c_w A_w V} \ll 1$$

The partial differential equation describing the variation of fluid bulk temperature, T, with distance in the flow direction, X and time, θ, then reduces to

$$\left(\frac{\partial T}{\partial X}\right) + \left(\frac{hp}{W c_p}\right)(T - T_i) = 0$$

Since time does not appear explicitly in this equation it may be treated in the same manner as the ordinary differential equation

$$\left(\frac{dT}{dX}\right) + \left(\frac{hp}{W\, c_p}\right) (T - T_i) = 0$$

insofar as obtaining a solution is concerned with the understanding that instantaneous values of the wall temperature, T_i, will be employed in the solution. Assuming that the wall temperature may be considered uniform over a segment of the passage of length (δX) and noting the boundary condition

$$(T)_{x=0} = T_{n-1}$$

where

T_{n-1} = air temperature at the inlet of the segment

the solution becomes

$$T = T_i + (T_{n-1} - T_i)\, e^{-\frac{hp\, X}{W\, c_p}}$$

Setting $X = (\delta X)$ yields the segment outlet air temperature, T_n

$$T_n = T_i + (T_{n-1} - T_i)\, e^{-\beta}$$

where

$$\beta \triangleq \frac{h\, A}{W\, c_p}$$

$A = p\,(\delta X)$ = effective area of the segment for convective heat transfer

Integrating the heat transfer rate to the wall per unit length over the length of the segment yields the heat transfer rate to the wall segment, Q_i,

$$Q_i = \int_0^{(\delta X)} q \, dX = \int_0^{(\delta X)} hp \, (T - T_i) \, dX$$

$$Q_i = \int_0^{(\delta X)} hp \, (T_{n-1} - T_i) \, e^{-\frac{\beta X}{(\delta X)}} \, dX$$

$$Q_i = \frac{(hA)}{\beta} \, (T_{n-1} - T_i)(1 - e^{-\beta})$$

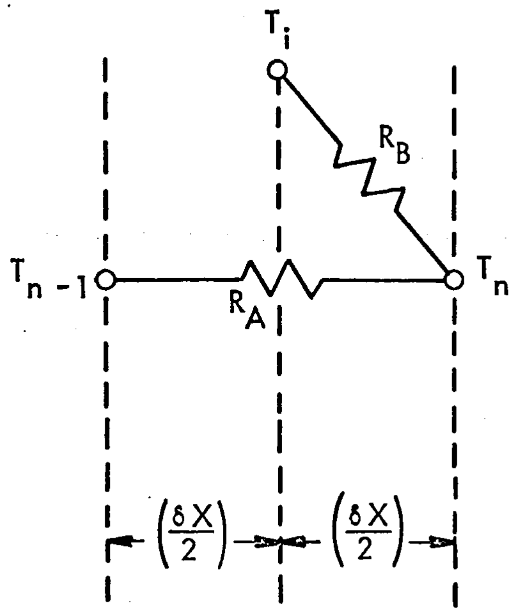
Eliminating the segment inlet temperature, T_{n-1}, between this expression and the expression for the segment outlet temperature yields

$$Q_i = (\dot{W} c_p) \, (e^{\beta} - 1) \, (T_n - T_i)$$

and eliminating the wall temperature, T_i, between the same two expressions yields

$$Q_i = (W c_p) \, (T_{n-1} - T_n)$$

The simplest network satisfying these relationships is shown in the following sketch. (As indicated previously, cathode followers are inserted between this passage segment network and the identical network for the adjacent segments.) Comparison of this network with that previously obtained indicates that, in addition to the absence of a capacitor representing the rate of change of the internal thermal energy of the air, the following differences exist: the wall temperature node location is midway between rather than adjacent to air temperature nodes; and the resistance connecting the segment air outlet temperature node, T_n, to the wall temperature node, T_i, is smaller by the ratio $\beta/(e^{\beta} - 1)$. Note that as (δX) and hence β approach zero this ratio approaches unity, and that the resistances connecting the segment air inlet and outlet temperature nodes are identical.



$$R_A = \frac{1}{W\,c_p}$$

$$R_B = \frac{1}{W\,c_p\,(e^{\beta} - 1)}$$

Since significant temperature gradients may exist in the passage walls in planes normal to as well as in the direction of the airflow, the analysis is extended here to include convection to (N) wall temperature nodes around the periphery of each segment. The partial differential equation becomes

$$\left(\frac{\partial T}{\partial X}\right) + \frac{1}{(W\,c_p)} \sum_{i=1}^{N} (hp)_i\,(T - T_i) = 0$$

and the boundary condition is as before

$$(T)_{x=0} = T_{n-1}$$

The solution is then

$$T = \frac{1}{\alpha}\left(1 - e^{-\frac{\beta X}{(\delta X)}}\right) \sum_{i=1}^{N} (hA)_i\,T_i + T_{n-1}\,e^{-\frac{\beta X}{(\delta X)}}$$

where

$$\alpha \triangleq \sum_{i=1}^{N} (hA)_i$$

$$\beta \triangleq \frac{\alpha}{W c_p}$$

Setting X = (δX) yields the segment outlet air temperature, T_n,

$$T_n = \frac{1}{\alpha}(1 - e^{-\beta}) \sum_{i=1}^{N} (hA)_i T_i + T_{n-1} e^{-\beta}$$

and integrating the heat transfer rate per unit length over the length of the passage yields the following expression for the heat flux to each wall node point.

$$q_i = (hp)_i \int_0^{(\delta X)} (T - T_i)\, dX$$

$$q_i = (hA)_i \left\{ \frac{1}{\alpha}\left(1 + \frac{e^{-\beta}}{\beta} - \frac{1}{\beta}\right) \sum_{i=1}^{N} (hA)_i T_i \right.$$

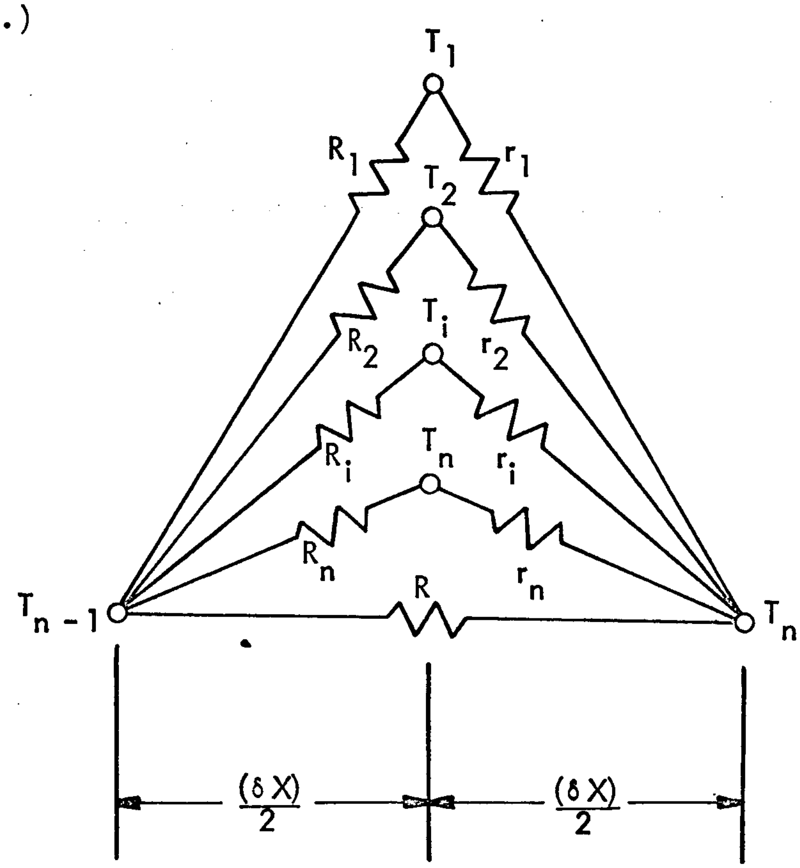
$$\left. + \frac{1}{\beta}(1 - e^{-\beta}) T_{n-1} - T_i \right\}$$

Eliminating the term involving the summation between this expression and the expression for the passage segment air outlet temperature

$$q_i = (hA)_i \left\{ \left(1 - e^{-\beta}\right)^{-1} - \frac{1}{\beta} \right\} (T_n - T_i) + (hA)_i \left\{ \frac{1}{\beta} - e^{-\beta} (1 - e^{-\beta})^{-1} \right\}$$

$$(T_{n-1} - T_i)$$

The simplest network satisfying these relationships is illustrated below. (Again, cathode followers must be inserted between this network and the adjoining networks.)

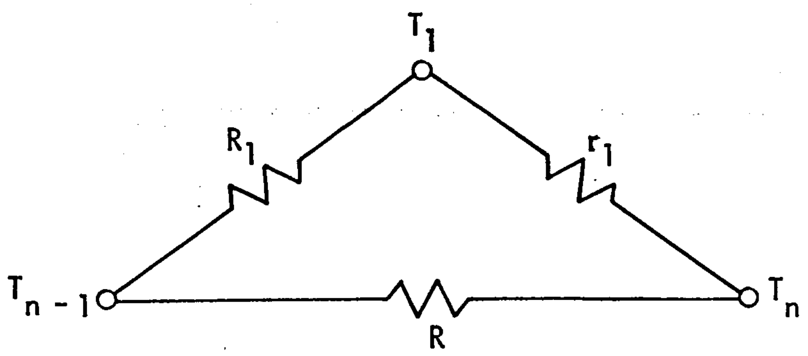


$$R_i = \frac{1}{(hA)_i\left\{\frac{1}{\beta} - e^{-\beta}(1 - e^{-\beta})^{-1}\right\}} = \frac{f_1(\beta)}{(hA)_i}$$

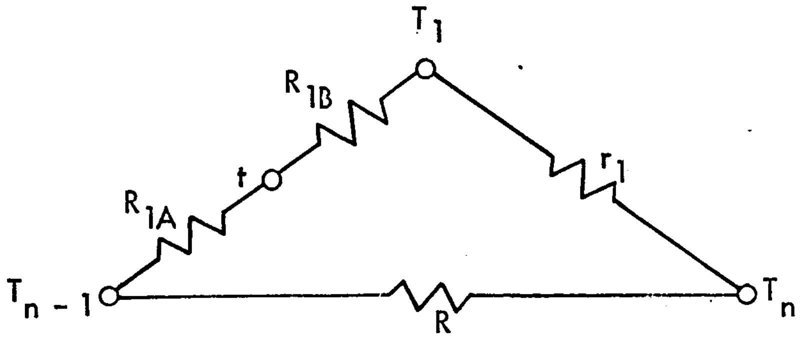
$$r_i = \frac{1}{(hA)_i\left\{(1 - e^{-\beta})^{-1} - \frac{1}{\beta}\right\}} = \frac{f_2(\beta)}{(hA)_i}$$

$$R = \frac{1}{a\left\{(1 - e^{-\beta})^{-1} - \frac{1}{\beta}\right\}} = \frac{f_3(\beta)}{a}$$

It is not immediately apparent that for N = 1 the network above reduces to the network previously presented for a single wall temperature node, and this is therefore demonstrated here. For N = 1 the circuit becomes



The resistor, R_1, will be replaced with two series connected resistors whose series resistance equals that of R_1 and whose ratio is equal to r_1/R as illustrated.



$$R_{1A} + R_{1B} = R_1$$

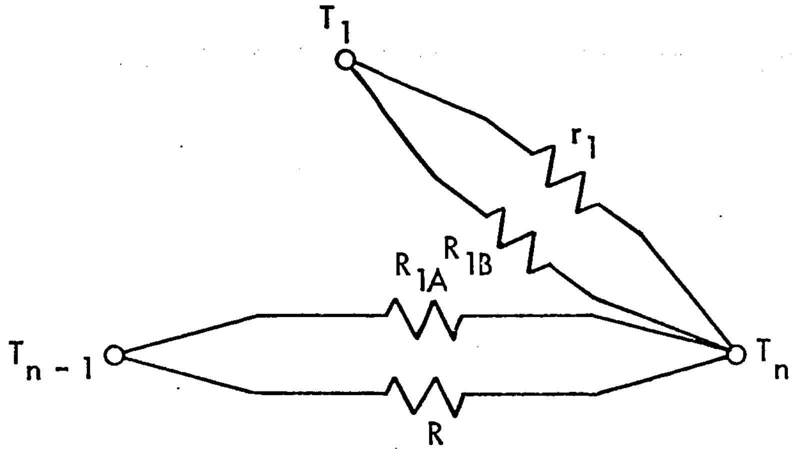
$$\frac{R_{1B}}{R_{1A}} = \frac{r_1}{R}$$

Solving for R_{1A} and R_{1B}

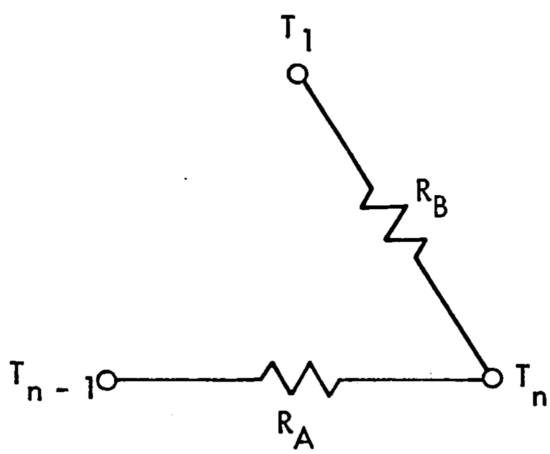
$$R_{1A} = \frac{R_1}{\frac{r_1}{R} + 1}$$

$$R_{1B} = \frac{R_1}{1 + \frac{R}{r_1}}$$

Since there is no current drawn from either node t or node T_n and since $R_{1B}/R_{1A} = r_1/R$, the potentials t and T_n are equal. Hence, the node points t and T_n may be connected to one another without affecting the circuit operation. Thus, the circuit becomes



The resistances R and R_{1A} may now be combined into a single resistance, R_A; and r_1 and R_{1B} may be combined into a single resistance, R_B.



$$R_A = \frac{1}{\frac{1}{R} + \frac{1}{R_{1A}}} = \frac{R\ R_1}{R_1 + r_1 + R}$$

$$R_B = \frac{1}{\frac{1}{r_1} + \frac{1}{R_{1B}}} = \frac{r_1\ R_1}{R_1 + r_1 + R}$$

Substituting the expressions for R_1, r_1 and R_1 yields values of R_A and R_B identical to those obtained in the one wall temperature node analysis, viz.

$$R_A = \frac{1}{W\ c_p}$$

$$R_B = \frac{1}{W\ c_p\ (e^{\beta} - 1)}$$

Since calculation of the factors $f_1(\beta)$, $f_2(\beta)$ and $f_3(\beta)$

$$\left(\frac{1}{\left\{ \frac{1}{\beta} - e^{-\beta} (1 - e^{-\beta})^{-1} \right\}},\quad \frac{1}{\left\{ (1 - e^{-\beta})^{-1} - \frac{1}{\beta} \right\}} \quad \text{and} \quad \frac{e^{\beta} - 1}{\left\{ (1 - e^{-\beta})^{-1} - \frac{1}{\beta} \right\}} \right)$$

is quite tedious, and may be inaccurate for small values of β, these functions are shown in Figures C-2 and C-3.

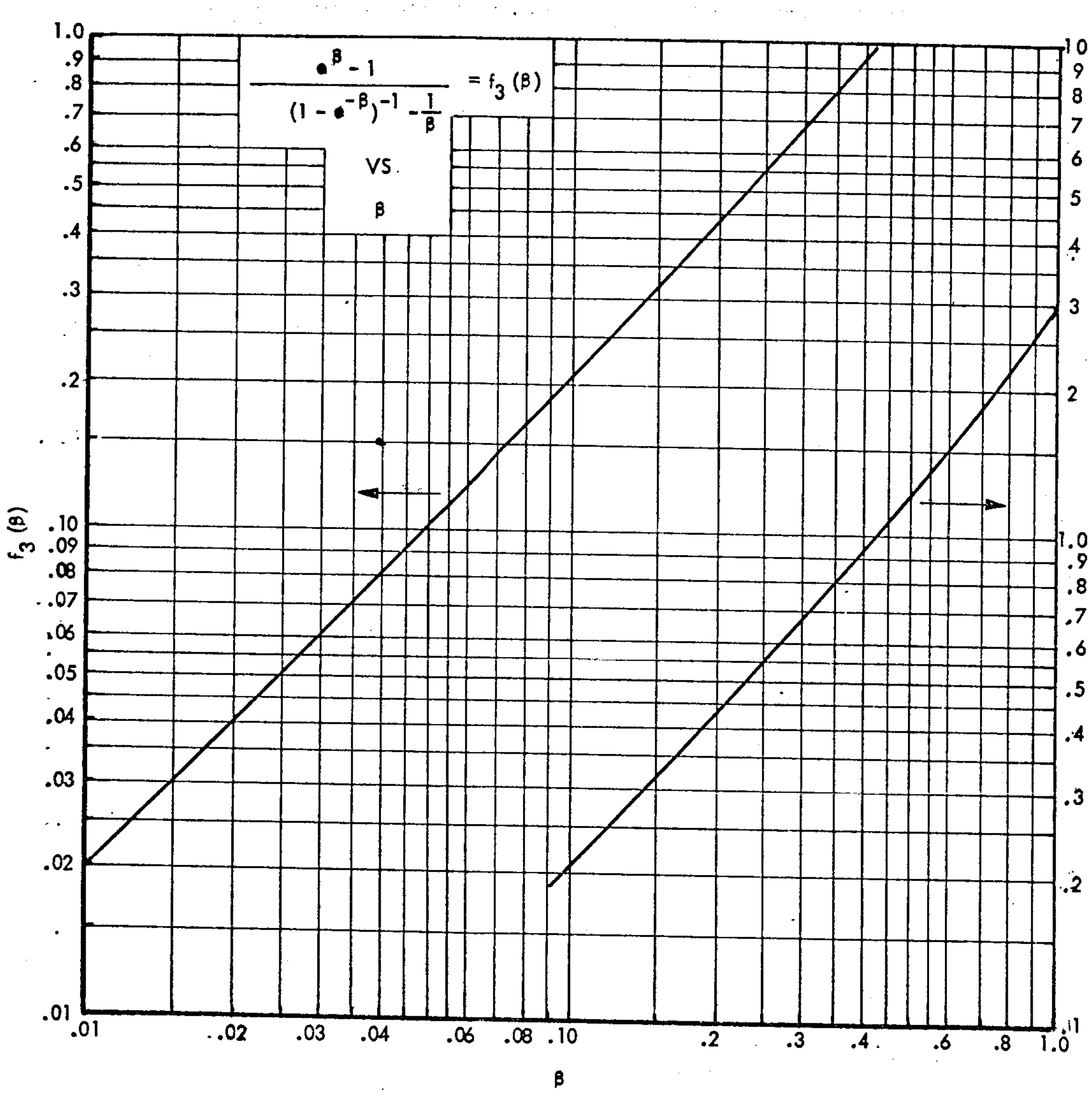


Figure C-2. Function f_3 (β) Required for Passage Air Temperature Solution

LOCKHEED CALIFORNIA COMPANY

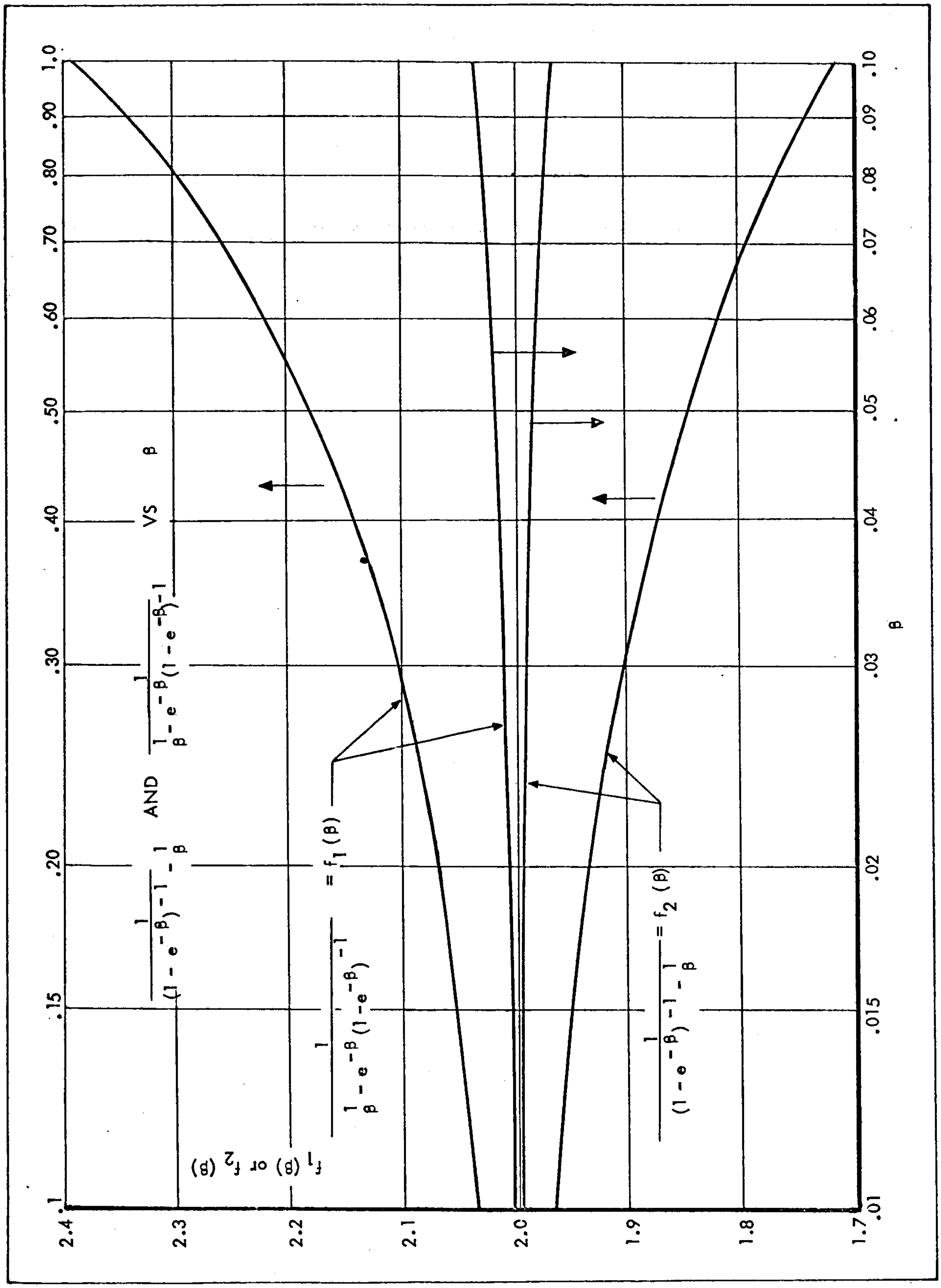


Figure C-3. Functions $f_1(\beta)$ and $f_2(\beta)$ Required for Passage Air Temperature Solution

APPENDIX D

COMPUTER OPERATION

The Thermal Analyzer Program exists in three versions: Version A for regular problems, Version B for maximum-sized problems, and Version C for fluid storage and pressurization problems. Table D-1 summarizes the capacity of the three versions of the program.

TABLE D-1

PROGRAM CAPACITY

Version	"A" (Regular)	"B" (Maximum)	"C" (Pressurization)
Overall Data Storage	16,000	16,000	13,000
Temperatures	~1000	~1000	300
Resistors	~2500	~2500	700
Data Block	~3500	~3500	10,000
FUNCT and PRINT	~2700	~4800	1100

This appendix describes the arrangement of the program deck, computation sequence, and tape usage for Versions A and B. It also contains a brief description of the program decks and a complete listing of the basic program and subroutines. Version C, which is applicable to fluid storage and pressurization problems, is described in detail in Reference 9.

ARRANGEMENT OF THE DECK

The program deck is arranged as follows. The two subroutines, PRINT and FUNCT, are inserted in the program deck between the two cards:

$IEDIT SYSIN1

and

$ENTRY MAN6

The input data — temperatures, resistors, etc. — are inserted between the next two cards:

$DATA

END

Figure D-1 is a representation of the deck setup.

The $IEDIT SYSLB3 card sets up the system to read the required subroutines from a reserved program library tape: Unit A8 on the Stand-Alone 7094, and unit SYSLB3 on the 7040/7094 D.C. Provision is made for the operator to mount the required tape by a $PAUSE card on the Stand-Along 7094, and by a $SETUP card for the 7040/7094 D.C. The required linkage is set up as the programs are read and loaded. Figures D-2 and D-3 show the overlay linkage arrangement for Versions A and B. Both versions use the same basic program and subroutines — the only difference is the way the various subroutines or functions are combined into links. While it is possible to combine the subroutines in a variety of ways, these two versions should cover most situations that may arise. Tables D-2 and D-3 show the order of the program deck for each version. Each subroutine is represented in these tables by its initial card. Version A should be used wherever possible as Version B obtains its additional capacity at considerable expense in execution time.

Additional capacity can be obtained, if necessary, by eliminating certain of the subroutines that are not used in a particular problem. The following subroutines are referenced only by the FUNCT and/or PRINT subroutines, and if not used there, they may be eliminated for a given run.

Subroutine or Function	Deck Name
LIN	LIN6
BIV	BIV6
PAR	PAR6, ATR6*
TANKA	TNK6
RAD	RAD6
SAV	SAV6

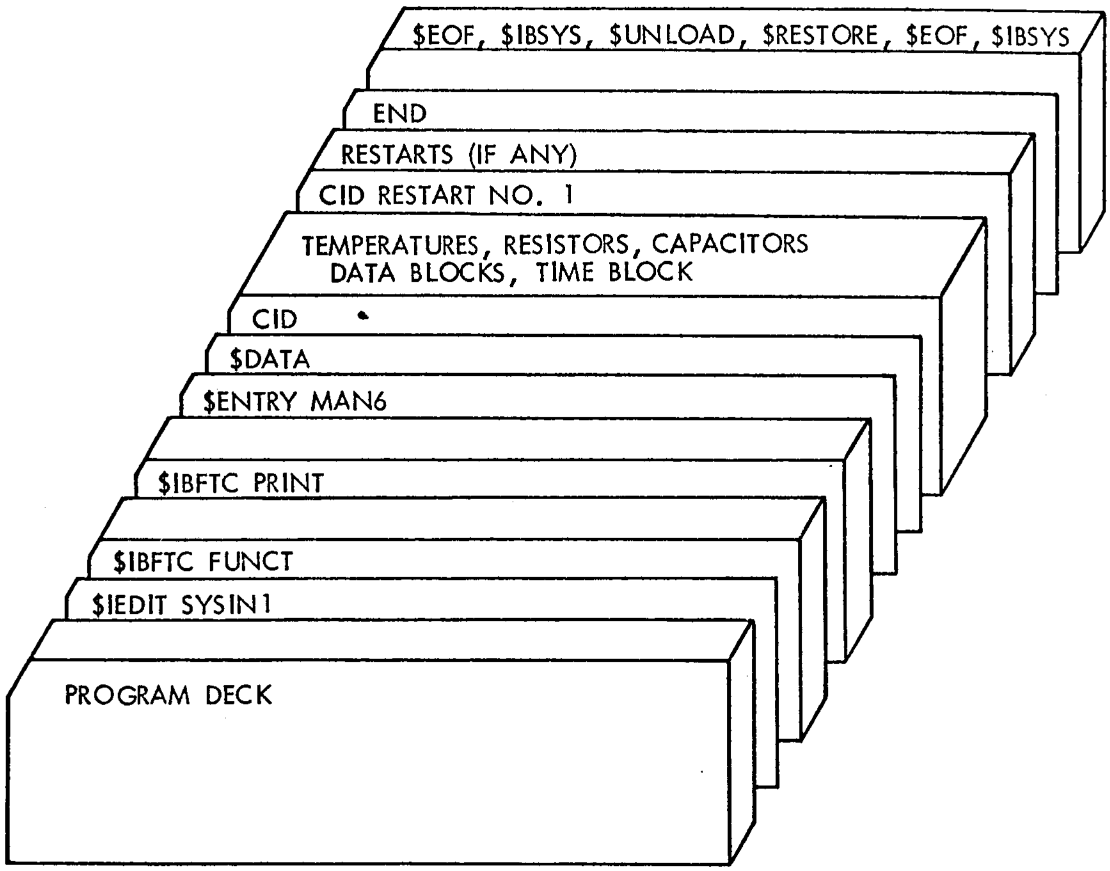


Figure D-1. Deck Setup for Thermal Analyzer Program

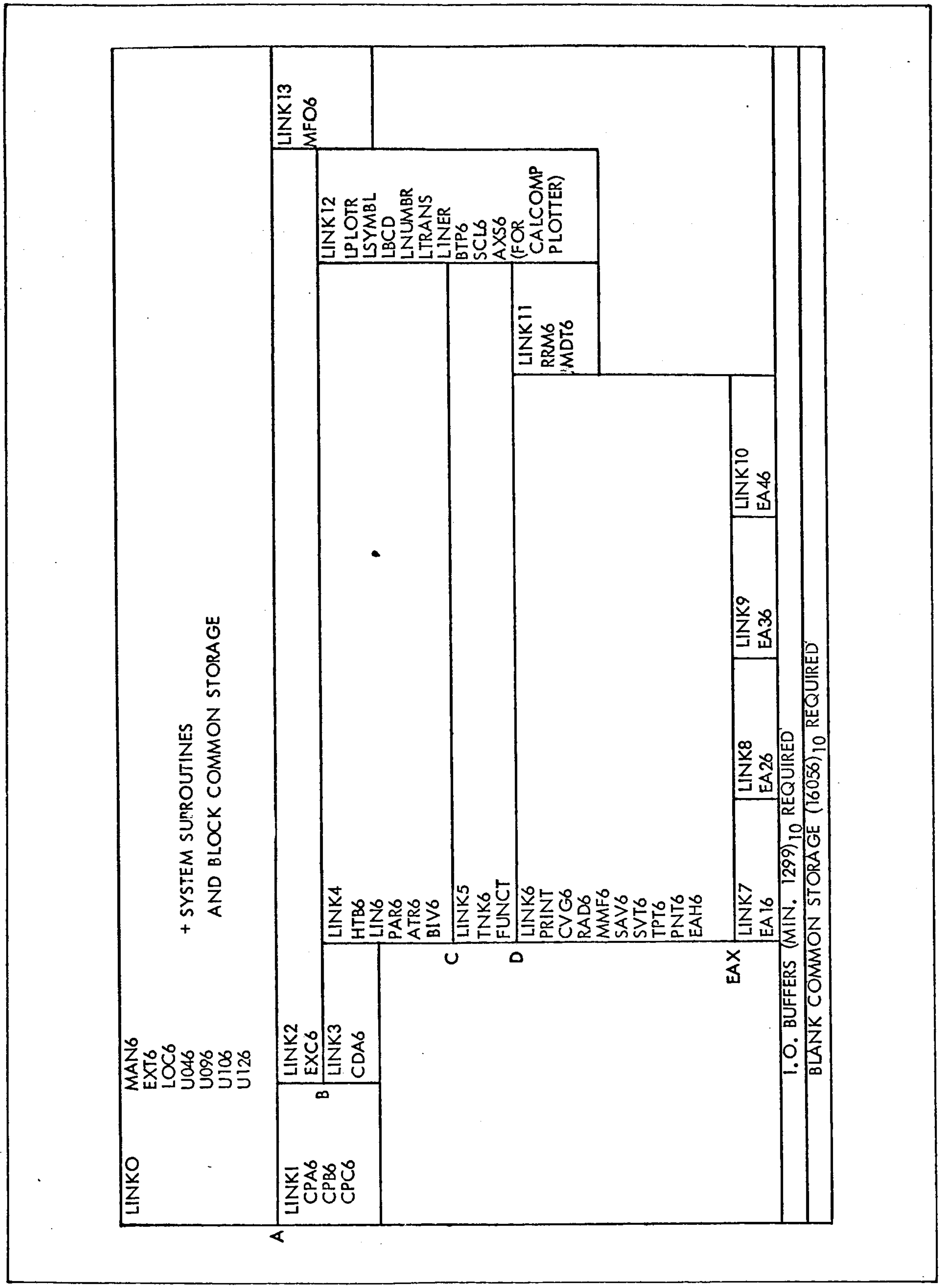


Figure D-2. Program Linkage Arrangement - Version A

LOCKHEED
CALIFORNIA COMPANY

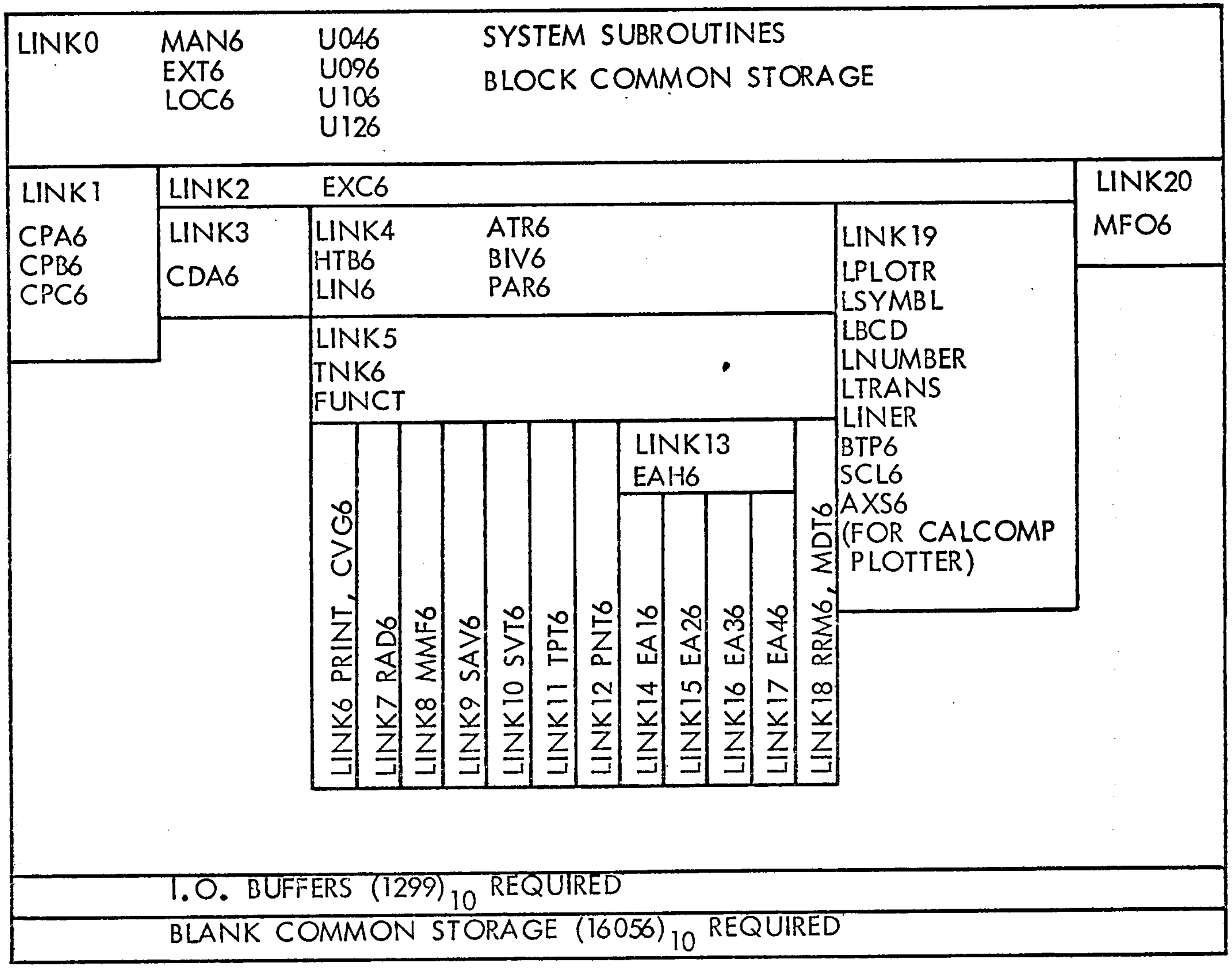


Figure D-3. Program Linkage Arrangement - Version B

TABLE D-2

PROGRAM DECK ORDER - VERSION A

```
$JOB          7,15,10000     5174833780      012266A 43233 DAVID         2266
$ID        JOB 2266A   DAVID   THERMAL ANALYSER - FORTRAN, SHORT JOBS        0001
$ID           7,15,10000     5174833780      012266A 43233 DAVID         2266
$PAUSE     PLEASE MOUNT RESERVE TAPE 290 ON UNIT A8, START
$ATEND        0,77777
$SETUP LB3    T290,DISK,,,1                                                  0003
$ASSIGN       SYSLB3                                                         0004
$ASSIGN       SYSLB2                                                         0005
$ATTACH       B2                                                             0006
$AS           SYSLB2                                                         0007
$ATTACH       A8
$AS           SYSLB3                                                         0009
$EXECUTE      IBJOB                                                          0010
$IBJOB        NOMAP
$FILE         'UNIT16',A(2),READY,NOLIST,INOUT,HIGH,BIN,BLK=457
$IEDIT        SYSLB3,SRCH1                                                   0012
$IBLDR MAN6                                                                  0013
$IBLDR EXT6                                                                  0014
$IBLDR LOC6                                                                  0015
$IBLDR U046                                                                  0016
$IBLDR U096                                                                  0017
$IBLDR U106                                                                  0018
$IBLDR U126                                                                  0019
$ORIGIN       A,SYSLB2                                                       0020
$IBLDR CPA6                                                                  0021
$IBLDR CPB6                                                                  0023
$IBLDR CPC6                                                                  0024
$ORIGIN       A,SYSLB2                                                       0028
$IBLDR EXC6                                                                  0029
$ORIGIN       C,SYSLB2                                                       0030
$IBLDR CDA6                                                                  0031
$ORIGIN       C,SYSLB2                                                       0032
$IBLDR HTB6                                                                  0033
$IBLDR LIN6                                                                  0035
$IBLDR PAR6                                                                  0036
$IBLDR ATR6                                                                  0037
$IBLDR BIV6
$ORIGIN       D,SYSLB2                                                       0039
$IBLDR TNK6                                                                  0040
$INCLUDE      FUNCT                                                          0041
$ORIGIN       E,SYSLB2,REW                                                   0043
$INCLUDE      PRINT                                                          0042
$IBLDR CVG6                                                                  0050
$IBLDR RAD6                                                                  0044
$IBLDR MMF6                                                                  0047
$IBLDR SAV6                                                                  0048
$IBLDR SVT6                                                                  0049
$IBLDR TPT6                                                                  0051
$IBLDR PNT6                                                                 A0051
$IBLDR EAH6                                                                  0057
$ORIGIN       EAX,SYSLB2,REW
$IBLDR EA16                                                                  0053
$ORIGIN       EAX,SYSLB2,REW
$IBLDR EA26                                                                  0054
$ORIGIN       EAX,SYSLB2,REW
$IBLDR EA36                                                                  0055
$ORIGIN       EAX,SYSLB2,REW
$IBLDR EA46                                                                  0056
$ORIGIN       E,SYSLB2,REW                                                   0052
$IBLDR RRM6                                                                  0045
$IBLDR MDT6                                                                  0046
```

TABLE D-2

(CONTINUED)

```
$ORIGIN       C,SYSLB2,REW                                  0079
$INCLUDE      LPLOTR,LSYMBL,LBCD,LNUMBR,LTRANS,LINER        0081
$IBLDR BTP6                                                 0080
$IBLDR SCL6
$IBLDR AXS6
$ORIGIN       A,SYSLB2,REW                                  0082
$IBLDR MFO6                                                 0083
$IEDIT        SYSIN1                                        0084
C
C    INSERT FUNCT AND PRINT ROUTINES HERE
C
$DATA
C     INSERT INPUT DATA HERE, FOLLOWED BY ANY RESTARTS
C
END
$EOF                                                        0086
$IBSYS                                                      0087
$UNLOAD       SYSLB3
$PAUSE    PLEASE REMOVE TAPE 290 FROM UNIT A8, SAVE IT, START
$RESTORE                                                    0089
$EOF                                                        0090
$IBSYS                                                      0091
```

TABLE D-3

PROGRAM DECK ORDER - VERSION B

```
$JOB          7,15,10000     5174833780     012266B 43233 DAVID        2266
$ID       JOB 2266B    DAVID   THERMAL ANALYSER - FORTRAN, LONG JOBS
$ID           7,15,10000     5174833780     012266B 43233 DAVID        2266
$PAUSE     PLEASE MOUNT RESERVE TAPE 290 ON UNIT A8, START
$ATEND        0,77777
$SETUP LB3    T290,DISK,,,1                                                0003
$ASSIGN       SYSLB3                                                       0004
$ASSIGN       SYSLB2                                                       0005
$ATTACH       B2                                                           0006
$AS           SYSLB2                                                       0007
$ATTACH       A8
$AS           SYSLB3                                                       0009
$EXECUTE      IBJOB                                                        0010
$IBJOB        MAP,LOGIC                                                    0011
$FILE         'UNIT16',A(2),READY,NOLIST,INOUT,HIGH,BIN,BLK=457
$IEDIT        SYSLB3,SRCH1                                                 0012
$IBLDR MAN6                                                                0013
$IBLDR EXT6                                                                0014
$IBLDR LOC6                                                                0015
$IBLDR U046                                                                0016
$IBLDR U096                                                                0017
$IBLDR U106                                                                0018
$IBLDR U126                                                                0019
$ORIGIN       A,SYSLB2                                                     0020
$IBLDR CPA6                                                                0021
$IBLDR CPB6                                                                0023
$IBLDR CPC6                                                                0024
$ORIGIN       A,SYSLB2                                                     0028
$IBLDR EXC6                                                                0029
$ORIGIN       C,SYSLB2                                                     0030
$IBLDR CDA6                                                                0031
$ORIGIN       C,SYSLB2                                                     0032
$IBLDR HTB6                                                                0033
$IBLDR LIN6                                                                0035
$IBLDR PAR6                                                                0036
$IBLDR ATR6                                                                0037
$IBLDR BIV6
$ORIGIN       D,SYSLB2                                                     0039
$IBLDR TNK6                                                                0040
$INCLUDE      FUNCT                                                        0041
$ORIGIN       E,SYSLB2,REW                                                 0043
$INCLUDE      PRINT                                                        0042
$IBLDR CVG6                                                                0050
$ORIGIN       E,SYSLB2,REW
$IBLDR RAD6                                                                0044
$ORIGIN       E,SYSLB2,REW
$IBLDR MMF6                                                                0047
$ORIGIN       E,SYSLB2,REW
$IBLDR SAV6                                                                0048
$ORIGIN       E,SYSLB2,REW
$IBLDR SVT6                                                                0049
$ORIGIN       E,SYSLB2,REW
$IBLDR TPT6                                                                0051
$ORIGIN       E,SYSLB2,REW
$IBLDR PNT6                                                               A0051
$ORIGIN       E,SYSLB2,REW
$IBLDR EAH6                                                                0057
$ORIGIN       EAX,SYSLB2,REW
$IBLDR EA16                                                                0053
$ORIGIN       EAX,SYSLB2,REW
$IBLDR EA26                                                                0054
```

TABLE D-3
(CONTINUED)

```
$ORIGIN        EAX,SYSLB2,REW
$IBLDR EA36                                                     0055
$ORIGIN        EAX,SYSLB2,REW
$IBLDR EA46                                                     0056
$ORIGIN        E,SYSLB2,REW                                     0052
$IBLDR RRM6                                                     0045
$IBLDR MDT6                                                     0046

$ORIGIN        C,SYSLB2,REW                                     0079
$INCLUDE       LPLOTR,LSYMBL,LBCD,LNUMBR,LTRANS,LINER           0081
$IBLDR BTP6                                                     0080
$IBLDR SCL6
$IBLDR AXS6
$ORIGIN        A,SYSLB2,REW                                     0082
$IBLDR MF06                                                     0083
$IEDIT         SYSIN1                                           0084
C
C     INSERT FUNCT AND PRINT ROUTINES HERE
C
$DATA
C       INSERT INPUT DATA HERE, FOLLOWED BY ANY RESTARTS
C
END
$EOF                                                            0086
$IBSYS                                                          0087
$UNLOAD        SYSLB3
$PAUSE     PLEASE REMOVE TAPE 290 FROM UNIT A8, SAVE IT, START
$RESTORE                                                        0089
$EOF                                                            0090
$IBSYS                                                          0091
```

Subroutine or Function		Deck Name	
PUNCHT		PNT6	
EAH1	EAH	EA16	EAH6**
EAH2		EA26	
EAH3		EA36	
EAH4		EA46	
RRM		RRM6, MDT6*	
CVG		CVG6***	

*Where two subroutines are listed, the second is called if and only if the first is called

**EAH is called if and only if any one of EAH1, EAH2, EAH3, or EAH4 is called.

***CVG6 (CALL CVG) executes a PRINT, and must therefore be in the same link or a link dependent from PRINT.

To remove a given subroutine it is necessary simply to remove the $IBLDR SUBR. card from the deck, and to remove the $ORIGIN card just ahead of it if and only if the $IBLDR card removed was the only one in that link. That is, two $ORIGIN cards must not appear together.

COMPUTATION SEQUENCE

The $IEDIT SYSIN1 card transfers loading to the standard input tape, and subroutines FUNCT and PRINT are read from it. The system then loads Link 0 into core and transfers control to it. Link 1, the compile phase link, is entered to read and compile the data in the following order:

1. Initial temperatures
2. NBK card
3. Initial resistors and connections
4. NBK card
5. Initial capacitors
6. NBK card
7. Data Blocks

8. NBK card

9. Print Interval, Final Time, Initial Time

10. NBK card

Two passes through the data are made. The first determines the kind of data and the number of items on a card. All cards are read, printed, and stored on tape 4. The second pass converts the data from BCD to the required (floating-point or integer) binary and stores it on the compiled data tape, tape 3.

The execution phase is then loaded, the compiled data is read from tape 3 into core, and control passes to subroutine HTBAL (deck name HTB6) which is the controlling subroutine during execution of the time-history. HTBAL calls the FUNCT subroutine, which in turn calls any subroutines there needed. On the return from FUNCT, HTBAL executes a heat balance for each node listed in the capacitor block, according to one of the two following formulas:

$$T_{k,\theta+\Delta\theta} = \frac{\Delta\theta}{C_k}\left(\sum_j \frac{T_{j,\theta}}{R_j} - T_{k,\theta}\sum_j \frac{1}{R_j} + q_k\right) + T_{k,\theta}$$

where

$T_{k,\theta+\Delta\theta}$ = The temperature of node k at time $\theta + \Delta\theta$

$T_{k,\theta}$ = The temperature of node k at time θ.

$T_{j,\theta}$ = The temperature of node j at time θ.

R_j = The value of resistor j, connecting nodes j and k.

$\sum_j$ = Sum including all nodes connected to node k by a resistor.

C_k = The value of the capacitor at node k.

q_k = The value of the arbitrary heat input to node k.

LOCKHEED CALIFORNIA COMPANY

If the value of the capacitor, C_k, is zero,

$$T_{k,\theta+\Delta\theta} = \frac{\sum_j \frac{T_{j,\theta}}{R_j} + q_k}{\sum_j \frac{1}{R_j}}$$

At the same time, the R-C product of each node having a non-zero capacitor is computed

$$(RC)_k = C_k / \sum_j \frac{1}{R_j}$$

The smallest RC product in the network or the printing interval, whichever is smaller, become the "minimum R-C product," and the time step for the following cycle, $\Delta\theta$, is some fraction (normally, 0.25) of this $(RC)_{min}$.

At this point, the program calls those subroutines requiring both "old" and "new" temperatures: Convergence (CVG), Minimum and Maximum Temperatures (MMF), and Save Current Data (SAV). These routines are used only if they have been called for in the FUNCT routine.

All temperatures in the $T_{\theta+\Delta\theta}$ (new) temperature block are then moved into the T_θ (old) temperature block, thus destroying the old temperatures and making the two blocks identical. Note that this operation specifically includes temperatures not mentioned in the capacitor block, so-called "fixed temperatures." To change the value of a "fixed temperature," it is necessary to change the value on the $T_{\theta+\Delta\theta}$ (new) block. If the old value is changed, that value will be used for that cycle only in the heat balance (as one of $T_{j,\theta}$) and then will be destroyed by the original value left unchanged in the $T_{\theta+\Delta\theta}$ (new) block. $T_{j,\theta+\Delta\theta}$ is stored in P(i), and $T_{j,\theta}$ is stored in (P(max T + i). That is, the new temperature block comes first in the P data storage region.

LOCKHEED
CALIFORNIA COMPANY

The program then calls the PRINT routine and computes the next time to print if it has reached print time. It then computes the current time for the next cycle, calls the plot routine if plotting has been called for, and then returns control to the Function routine for the next cycle. When, in computing the current time for the next cycle, the current time is greater than the final time (specified in the time block), $\Delta\theta$ is chosen so that the current time is equal to the final time for the next cycle, and at the end of the next cycle, control is returned to the executive program in link 0. From there, if plotting has been called for (CALL TPLOT) the plotting routines are called in and executed. Then, if maximum and minimum temperature monitoring has been called for (CALL MMF), the max-min output routine (MFOUT) is called in and executed.

Thereafter, control passes back to the compilation phase, and any restarts are compiled. A restart consists of only one block of input data, followed by an NBK card. However any value that was compiled in the initial case may be changed. There are five flags, each one telling the program that the items following it are to be changed in the initial case. Flag 1 indicates that the cards following are temperature cards. Similarly, resistor and capacitor values are changed by cards following Flags 2 and 3, respectively. Flag 4 indicates a new time block, which must be changed in its entirety. Flag 5 is used to indicate changes to the data block. Note that each sub-block of data (a curve or table) must be changed in its entirety, the restart containing all identifications, values, and if in the original curve, the period. When a data block is changed, the new values are stored starting at the beginning of that data block. Care must be taken not to exceed the storage allotted to that data block in the initial case. If that allotment is exceeded, the remainder will be stored in the next following data block or blocks, thus destroying some information. Each restart may include any or all of the above changes, and each must end with an NBK card. There is no theoretical limit to the number of restarts for a given run, but practical considerations, such as the size of deck, the machine time used, etc., will limit the number.

Following all of the restarts, the last card read by the program should be the "END" card. This card signals the program to exit and avoids an "End of file reading..." diagnostic. It also enables the program to complete plotting and minimum-maximum output before exiting.

The cards following the "end" card are system instructions to restore the system tapes used and allow removal of reserved and special output tapes, as required. Figure D-4 is a simplified flow diagram of the program. It applies to both Versions A and B. Table D-4 is a complete listing of the basic program and subroutines.

DESCRIPTION OF PROGRAM DECKS

A brief description of the function and operation of each of the program decks follows. The subroutines called for in the FUNCT and PRINT routines are discussed in more detail in Section IV.

MAN6	Initiates program and exits to system.
EXT6	Controls the program flow between compile, execution, and post-processing phases; contains block-common storage for information that must not be destroyed between phases.
U046	Defines tape unit 4 as BCD
U096	Defines tape unit 9 as BCD
U106	Defines tape unit 10 as BCD
U126	Defines tapeunit 12 as BCD
LOC6	Locates the beginning point of a data block. Let L = LOC (N), where N is a data block designation number, then P(L+1) is the first point in the data block.
CPA6	Performs first-pass compilation of input data and stores it on unit 4.
CPB6	Performs second-pass compilation of input data, reading unit 4 and writing unit 3.
CPC6	Performs compilation and posting of changes specified in restart blocks, writes revised data on unit 3.
EXC6	Initiates execution phase; contains block common storage for those items that may not be destroyed during the execution phase.
CDA6	Reads compiled data from unit 3 into core storage.
HTB6	Controls execution of the time history of the required problem; performs the basic heat balance on each node; updates

the temperature blocks each cycle; controls the time variables and selects printing cycles.

FUNCT Controls the execution of each cycle except for heat balance and time calculations. Calls such routines as it needs to perform the required operations; performs user's miscellaneous calculations and circuit value changes.

PRINT Writes the required output on the system output tape according to the format(s) specified.

LIN6 Performs linear interpolation for the independent variable and data block specified as Y = LIN(X,N)

BIV6 Performs bivariate linear interpolation for the independent variables and the data block specified as Y = BIV(X1,X2,N) where X1 is the vertical independent variable, and X2 is the horizontal independent variable.

PAR6 Performs parabolic interpolation for the independent variable and the data block specified as Y = PAR(X,N). Data block N must have an odd number of point pairs.

ATR6 Called by PAR6 to assist in parabolic interpolation.

TNK6 Initiates pressurization program, and stores data block numbers for use in pressurization. CALL TANK A (K) results in a diagnostic in versions A and B.

CVG6 Iterates through the whole program, but holds time constant until the temperatures have achieved steady state. CALL CVG (A,N,M), where A is the greatest allowable temperature difference between cycles, N is the upper limit on the number of cycles to be used, M is a print flag signaling to print every M cycle. If M is negative, all capacitors are treated as if zero valued.

RAD6 Calculates the value of a radiation resistor. The function specification R(A) = RAD (B,C,K_{rad}) will cause resistor A to be computed according to the equation

$$R(A) = 1./\left\{.1713*10^{-8} K_{rad} \left[(T_B + 460)^2 + (T_C + 460)^2\right] \left[T_B + T_C + 920\right]\right\}$$

MMF6 Finds local maxima and minima for the temperatures listed in data block N.CALL MMF (θ_i, θ_f, N,S), where θ_i and θ_f give the time range desired, N is the data block containing the temperature numbers, and S means perform this test every Sth cycle.

SAV6 — Initiates a call from HTB6 to SVT6 when the first argument, A, is greater than or equal to the second, B. CALL SAV (A,B).

SVT6 — Writes the current condition of the data onto tape 3, the compiled data tape. This allows a restart to continue from the point at which the previous case stopped, or to rerun a portion of case with new parameters.

TPT6 — Writes time and the values of the temperatures listed in data block N onto tape 11 to be post-processed by BTP6 for the plotter. CALL TPLOT (N).

PNT6 — Punches cards giving the temperature numbers and values suitable for use as an initial temperature block. CALL PUNCHT (θ,A,B) where θ is the time at which temperatures are to be punched, and A and B are the smallest and highest temperature numbers punched. If A = B = 0 all temperatures will be punched.

EA16 — Sets up a call to EAH6 with two arguments, node number j and constant K, i.e., CALL EAH1(j,K)

EA26 — Sets up a call to EAH6 with three arguments, H to be stored CALL EAH2(j,K,H)

EA36 — Sets up a call to EAH6 with four arguments, adding α_o and surface flag to EAH1 arguments in place of H. CALL EAH3(j,K,α_o,F)

EA46 — Sets up a call to EAH6 with five arguments, adding the location for storing H. CALL EAH4(j,K,H,α_o,F)

EAH6 — Computes the aerodynamic heating to node j, given K, α_o, and a flag telling whether the node is upper or lower surface. Eckert's aerodynamic heating equation is used. EAH6 is called from EA16, EA26, EA36, or EA46.

RRM6 — Computes radiation resistors by the same formula as RAD6, except that K_{rad} is computed from a radiation matrix given in data block N, and is dependent on all the temperatures in the matrix network. MDT6 is called from RRM6. R(A) = RRM(B,C,N) where B and C are the nodes directly connected to resistor A, and N is the data block number.

MDT6 — Matrix inversion routine used by RRM6. Maximum matrix size is 15 x 15.

BTP6	Plot post-processor routine reads the information stored by TPT6, sorts it and, for the Calcomp plotter, writes it on a tape for off-line processing. For the S-C 2040 plotter, it displays directly on the CRT.
AXS6	Routine for drawing axes for the Calcomp plotter, called by BTP6.
SCL6	Routine for selecting scales and scaling information for the Calcomp plotter called by BTP6.
MF06	Minimum-maximum temperature post-processor reads information stored on tape by MMF6, sorts it and prints it.

TAPE USAGE

The following list gives the tapes used during compilation and execution of a program, depending in some cases on the requests made in the FUNCT subroutine.

COMPILE PHASE TAPE USAGE

Tape No.	Use
2	Contains program overlay links
3	Store compiled data
4	Store semi-compiled data
5	System standard input
6	System standard output
8	Not used
9	Used by pressurization program
10	Used by pressurization program
11	Not used
12	Materials Library tape (Reserved)
16	Not used

EXECUTION PHASE TAPE USAGE

Tape No.	Use
2	Contains program overlay links.
3	Read compiled data. Rewrite if SAV is called.
4	Not used
5	Not used
6	System Standard Output
8	Specified temperatures for off-line use, Reserve.
9	Min-max intermediate output.
10	Min-max intermediate output.
11	Plot intermediate output
12	Not used
16	Plot output (Calcomp Plotter)

NOTES ON COMPUTER OPERATION

Some general observations regarding computer operation, and some difficulties which frequently arise are discussed in the following paragraphs.

1. When running on the IBM 7094 Stand Alone machine, difficulty has been encountered due to dirty tape heads. In particular, the overlay tape (B-3) is used so heavily on the largest jobs, that within a couple of hours a tape read check is almost certain to occur. This happens when the overlay communication region is trying to reach the next link, which is still on tape, but cannot due to the dirty tape head. Overlay communication remains in a two-instruction loop until operator action intervenes.

2. Versions A and B have a built-in indicator using sense lights to indicate the degree of completion of the time history. The sense lights are used as binary indicators showing 16ths of the time history completed. This enables the operator to determine that the job appears to be progressing properly.

3. On the 7040/7094 D.C. it has been found that not more than one tape may be set up to be deblocked from disk to tape. Therefore, if more than one tape is to be processed off-line or by a later job, all but one such tape must be assigned to a physical tape.

4. Backspacing a simulated (disk) tape is a complicated and time-consuming operation. Therefore, after each link is loaded, the thermal analyzer calls for a rewind of the overlay tape. It might prove more efficient to make the overlay tape physically a tape, but this has not been tested.

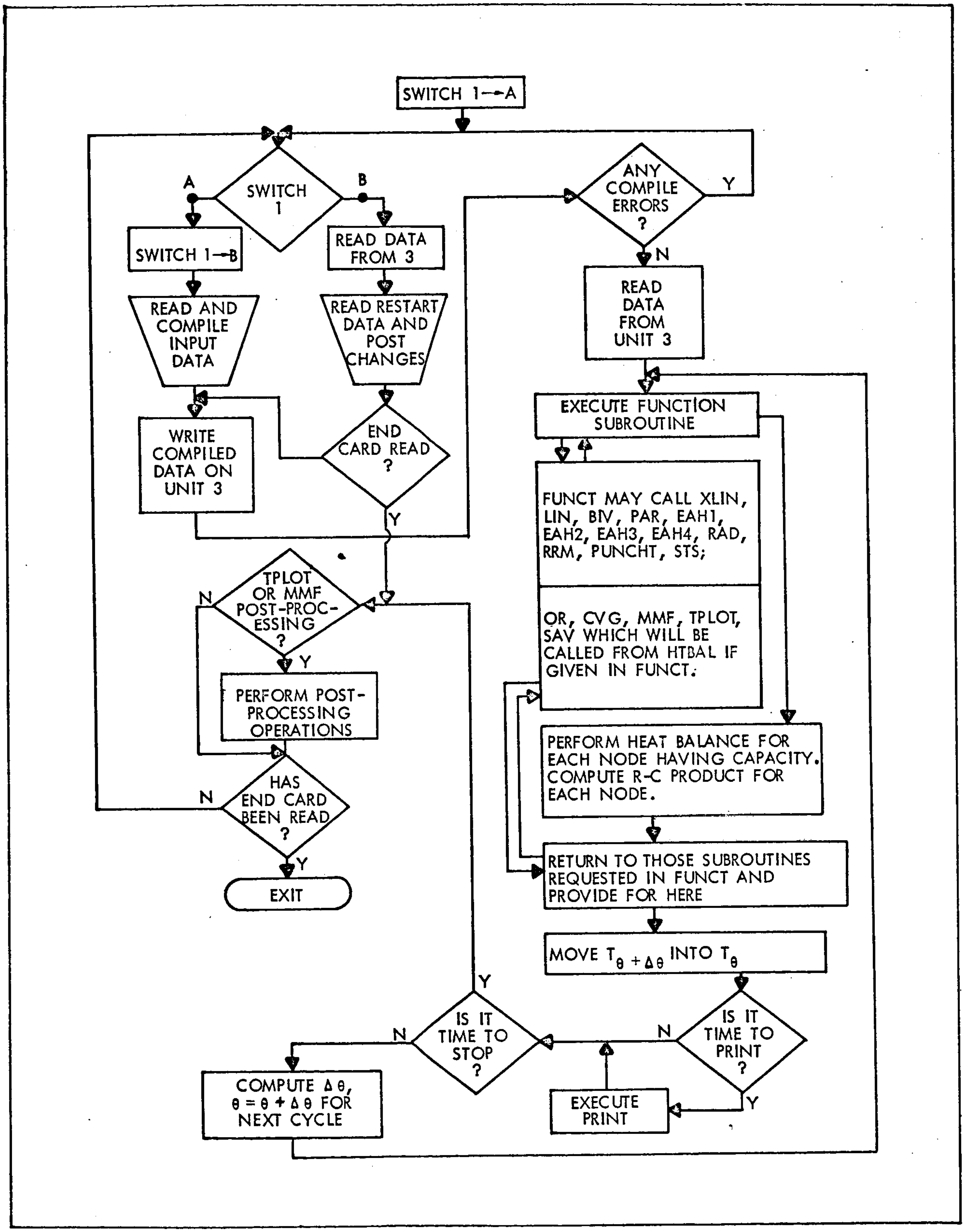


Figure D-4. Simplified Flow Diagram of Computer Program

TABLE D-4. PROGRAM LISTING

```
$IBFTC MAN6                                                                  MAN6000
C      AS OF 12/4/64                                                         MAN6001
       NLAG = 1                                                              MAN6002
       CALL EXETIV (NLAG)                                                    MAN6003
       CALL EXIT                                                             MAN6004
       STOP                                                                  MAN6005
       END                                                                   MAN6006


$IBFTC EXT6                                                                  EXT6000
       SUBROUTINE EXETIV (NLAG)                                              EXT6001
C      AS OF 5/25/65                                                         EXT6002
       COMMON P(16000),M(16),MAXT,MAXR,MAXC,KSP,KST,KURV7,NCVG,NSTS,NFAB     EXT6003
       COMMON NABL(10),NKSP,LMMF,NRMF,NREAD,CC(12)                           EXT6004
       COMMON NR,MAXP,MAXK,MAXS,MAXM                                         EXT6005
       COMMON /CP2/ ERROR,MAXMUM                                             EXT6006
       COMMON /CPITM/ ITIMO                                                  EXT6007
       COMMON /CPPLT/NPLT,NEND,KSKIP                                         EXT6020
       COMMON /EXFLG/NTFLG,NSAV,NFLG                                         EXT6021
       COMMON /EXPLT/ NLFLG                                                  EXT6022
       MAXMUM=16000                                                          EXT6025
       ITIMO = 1                                                             EXT6026
       NEND=0                                                                EXT6027
       NPLT=1                                                                EXT6028
       ERROR=0.                                                              EXT6029
       NTYPE=0                                                               EXT6030
       NTFLG=0                                                               EXT6031
       IR=1                                                                  EXT6032
 8     CALL COMPIL(IR)                                                       EXT6033
       IF (ERROR.EQ.0..AND.NEND.EQ.0) CALL EXCUT(NLAG)                       EXT6035
       IF (ERROR.NE.0.) WRITE (6,100)                                        EXT6036
 100   FORMAT(44H0ERROR HAS OCCURRED SOMEWHERE IN ABOVE CASE.)               EXT6037
       IF (LMMF.GT.2) CALL MFOUT                                             EXT6038
       IF (NPLT.GT.1) NBIT=1                                                 EXT6039
       IF (NBIT.GT.0) CALL BITAP                                             EXT6040
       IF (NEND.GT.0) RETURN                                                 EXT6041
       IR=2                                                                  EXT6042
       GO TO 8                                                               EXT6043
       END                                                                   EXT6045
```

TABLE D-4. (Continued)

```
$IBFTC LOC6
      FUNCTION LOC (KT)                                                 LOC6000
C     AS OF 04/27/65                                                    LOC6001
C                                                                       LOC6002
C   LOC ROUTINE FINDS FIRST LOCATION AHEAD OF LOCATION STARTING TABLE KTLOC6003
C                                                                       LOC6004
      COMMON P, M, MAXT, MAXR, MAXC, KSP, KST                           LOC6005
      COMMON /EXADR/ LCTN,LCTO,LCR,LCC,LCQ,LCRC,LCPS,LCK,LCKA           LOC6006
      COMMON /CP2/ ERROR                                                LOC6007
      DIMENSION P(16000), M(16), NSTRT(1)                               LOC6008
      EQUIVALENCE (P(1), NSTRT(1))                                      LOC6009
      REAL M                                                            LOC6010
      IF (KT.LE.0) GO TO 13                                             LOC6011
   10 J=LCKA                                                            LOC6012
   11 J1=NSTRT(J)/2**18                                                 LOC6013
      J2=MOD(NSTRT(J),2**18)                                            LOC6014
      IF (J1.EQ.KT) GO TO 12                                            LOC6015
      IF (J1.EQ.0)  GO TO 13                                            LOC6016
      J=J+1                                                             LOC6017
      GO TO 11                                                          LOC6018
   12 IF (P(J2).NE.32767.) J2=J2-1                                      LOC6019
      LOC=J2                                                            LOC6020
      RETURN                                                            LOC6021
   13 WRITE (6,21) KT                                                   LOC6022
   21 FORMAT (10HOCURVE NO.I5,25H IS NOT IN ADDRESS TABLE.)             LOC6023
      ERROR =1.                                                         LOC6024
      RETURN                                                            LOC6025
      END                                                               LOC6026
                                                                        LOC6027


$IBMAP U046
        ENTRY    .UN04.                                                 U046
.UN04.  PZE      UNIT04                                                 U 46
UNIT04  FILE     ,UT4,READY,INOUT,BLK=22,BCD,NOLIST                     U 46
        END                                                             U 46
                                                                        U 46


$IBMAP U096
        ENTRY    .UN09.                                                 U096
.UN09.  PZE      UNIT09                                                 U 96
UNIT09  FILE     ,B(3),READY,INOUT,BLK=22,BCD,NOLIST                    U 96
        END                                                             U 96
                                                                        U 96


$IBMAP U106
        ENTRY    .UN10.                                                 U106
.UN10.  PZE      UNIT10                                                 U 06
UNIT10  FILE     ,A(3),READY,INOUT,BLK=22,BCD,NOLIST                    U 06
        END                                                             U 06
                                                                        U 06


$IBMAP U126
        ENTRY    .UN12.                                                 U126
.UN12.  PZE      UNIT12                                                 U 26
UNIT12  FILE     ,B(2),READY,INPUT,BLK=22,BCD,NOLIST                    U 26
        END                                                             U 26
                                                                        U 26
```

TABLE D-4. (Continued)

```
$IBFTC CPA6                                                             CPA6000
      SUBROUTINE COMPIL (IR)                                            CPA6001
C     AS OF 04/27/65                                                    CPA6002
C     8/21/64       CHANGED TO INCLUDE MINMAX                           CPA6003
      COMMON P, M, MAXT, MAXR, MAXC, KSP, KST, KURV 7, NCVG, NSTS, NFAB CPA6004
      COMMON NABL,NKSP,LMMF,NRMF,NREAD,CC                               CPA6005
      COMMON NRECD, NWPD, NWCT, NWST, NWTM                              CPA6006
      COMMON /CP2/ERROR,MAXMUM                                          CPA6007
      COMMON /CPPLT/ NPLT,NEND                                          CPA6008
      COMMON /EXPLT/ NLFLG                                              CPA6009
      DIMENSION CZ(6),CY(6)                                             CPA6010
      DIMENSION A(12), I 1(12), I 2(12), I 3(12), NABL(10), CC(12)      CPA6011
      DIMENSION P(16000), IRLINE(1), KPSD(1), NSTRT(1), M(16), KDAT(1)  CPA6012
      EQUIVALENCE (P(4000), IRLINE(1)), (P(8000), KPSD(1)), (P(13000),  CPA6013
     1  NSTRT(1)), (P(1), KDAT(1))                                      CPA6014
      REAL M, INC, NBK,NET                                              CPA6015
      INTEGER CP                                                        CPA6016
      DATA ENDF/6H   -1 /                                               CPA6017
      DATA NET/6HNET   /                                                CPA6018
      DATA DEC/6HDEC   /                                                CPA6019
      DATA INC/6HINC   /                                                CPA6020
      DATA PER/6HPER   /                                                CPA6021
      DATA NBK/6HNBK   /                                                CPA6022
      DATA CID/6HCID   /                                                CPA6023
      DATA TAP/6HTAP   /                                                CPA6024
      DATA RES/6HRES   /                                                CPA6025
      DATA CAP/6HCAP   /                                                CPA6026
      DATA COD/6HCOD   /                                                CPA6027
      DATA ZERO/6H000000/                                               CPA6028
      DATA (CZ(I),I=1,6)/2H01,2H02,2H03,2H04,2H05,2H06/,BLANK/1H /      CPA6029
      DATA (CY(I),I=1,6)/2H 1,2H 2,2H 3,2H 4,2H 5,2H 6/,ZERZ/2H00/      CPA6030
      DATA ZERY/2H 0/                                                   CPA6031
  209 FORMAT (A3,A2,12A6)                                               CPA6039
  210 FORMAT (5X,12A6)                                                  CPA6040
  211 FORMAT (5X, A3,I2,12A6)                                           CPA6041
  212 FORMAT (1H1,9X,12A6)                                              CPA6042
  215 FORMAT (40H0 KOUNT IS NOT CORRECT IN COMPIL ROUTINE )             CPA6044
  218 FORMAT (I 3, 3X, I 2)                                             CPA6046
  223 FORMAT (7H0CODE= A6,12H IS ILLEGAL.)                              CPA6050
  224 FORMAT (25H0CANNOT FIND CURVE NAMED A6,17H ON LIBRARY TAPE.)      CPA6051
  232 FORMAT (53H0CODE= NET IS ILLEGAL IN THIS VERSION. USE VERSION C.) CPA6053
  233 FORMAT (5H **  ,A3,A2,12A6,23H  ** THIS CARD IGNORED.)            CPA6053
      NCVG = 1                                                          CPA6054
      NSTS = 1                                                          CPA6055
      MATLIB =1                                                         CPA6056
      NSW2=1                                                            CPA6057
      NPLT=1                                                            CPA6058
      NLFLG=1                                                           CPA6059
      NFAB = 1                                                          CPA6061
      NSW2=1                                                            CPA6062
      LINE = 1                                                          CPA6063
      LMMF=1                                                            CPA6064
      NRMF=0                                                            CPA6065
      NREAD=0                                                           CPA6066
      DO 101 I = 1, 10                                                  CPA6067
      NABL(I) = 0                                                       CPA6068
  101 CONTINUE                                                          CPA6069
      GO TO (105, 149, 149), IR                                         CPA6070
  105 REWIND 4                                                          CPA6071
      DO 107  I = 1 , MAXMUM                                            CPA6072
      P(I) = 0.                                                         CPA6073
```

TABLE D-4. (Continued)

```
107   CONTINUE                                                      CPA6074
      KOUNT = 1                                                     CPA6075
110   CP=-1                                                         CPA6076
      IF (MATLIB.EQ.1) READ (5,209) CM,CX,(A(I),I=1,12)             CPA6079
      IF (MATLIB.EQ.2) READ (12,209)CM,CX,(A(I),I=1,12)             CPA6080
1101  DO 1102 I=1,6                                                 CPA6081
1102  IF (CX.EQ.CY(I).OR.CX.EQ.CZ(I)) CP=I                          CPA6082
      IF (CX.EQ.BLANK.OR.CX.EQ.ZERZ.OR.CX.EQ.ZERY) CP=0             CPA6083
      IF (CP.GE.0) GO TO 1103                                       CPA6084
      CP=0                                                          CPA6085
      ERROR=1.                                                      CPA6086
      WRITE (6,215)                                                 CPA6087
1103  IF (LINE.LT.60) GO TO 111                                     CPA6088
109   WRITE (6,212)           (CC(I),I=1,12)                        CPA6089
      LINE = 1                                                      CPA6090
111   IF (CM.EQ.CID) GO TO 117                                      CPA6091
      IF (CM.EQ.TAP) GO TO 150                                      CPA6092
      WRITE (6,211) CM,CP,(A(I),I=1,12)                             CPA6093
      LINE=LINE+1                                                   CPA6094
      IF (CM .EQ. DEC) GO TO 116                                    CPA6095
      IF (CM .EQ. INC) GO TO 118                                    CPA6096
      IF (CM .EQ. PER) GO TO 119                                    CPA6097
      IF (CM .EQ. NBK) GO TO 120                                    CPA6098
      IF (CM.EQ.RES) GO TO 123                                      CPA6099
      IF (CM.EQ.CAP) GO TO 124                                      CPA6100
      IF (CM.EQ.NET) GO TO 170                                      CPA6101
      IF (CM.EQ.COD) GO TO 112                                      CPA6102
115   WRITE (6,223) CM                                              CPA6103
      ERROR=1.                                                      CPA6104
      GO TO 110                                                     CPA6105
112   KOD = 9                                                       CPA6106
      GO TO 125                                                     CPA6107
116   KOD = 1                                                       CPA6108
      GO TO 125                                                     CPA6109
117   CMID=CM                                                       CPA6110
      ICP=CP                                                        CPA6111
      DO 1171 I=1,12                                                CPA6112
      CC(I)=A(I)                                                    CPA6113
1171  CONTINUE                                                      CPA6114
      LINE =1                                                       CPA6115
      WRITE (6,212)           (CC(I),I=1,12)                        CPA6116
      GO TO 110                                                     CPA6117
118   KOD = 3                                                       CPA6118
      GO TO 125                                                     CPA6119
119   KOD = 4                                                       CPA6120
      GO TO 125                                                     CPA6121
120   KOD = 5                                                       CPA6122
      KOUNT = KOUNT + 1                                             CPA6123
      IF (KOUNT - 6) 125, 121, 122                                  CPA6124
121   KOD = 6                                                       CPA6125
      GO TO 125                                                     CPA6126
122   WRITE (6, 215)                                                CPA6127
      ERROR =2.                                                     CPA6128
      RETURN                                                        CPA6129
123   KOD=7                                                         CPA6130
      GO TO 125                                                     CPA6131
124   KOD=8                                                         CPA6132
      GO TO 125                                                     CPA6133
150   IF (MATLIB.EQ.2) GO TO 160                                    CPA6134
      MATLIB=2                                                      CPA6135
      FIRST=0.                                                      CPA6136
```

TABLE D-4. (Continued)

```
      WRITE (6,211) CM,CP,(A(I),I=1,12)                              CPA6137
      LINE=LINE+1                                                    CPA6138
  151 READ (  12  ,209) Z1,MZ2,Z3                                    CPA6139
      IF (Z1.NE.TAP) GO TO 151                                       CPA6140
      IF (Z3.EQ.A(1)) GO TO 110                                      CPA6141
      IF (Z3.NE.ZERO) GO TO 151                                      CPA6142
      REWIND 12                                                      CPA6143
      FIRST=FIRST+1.                                                 CPA6144
      IF (FIRST.EQ.1.) GO TO 151                                     CPA6145
      WRITE (6,224) A(1)                                             CPA6146
      ERROR=1.                                                       CPA6147
  160 MATLIB=1                                                       CPA6148
      GO TO 110                                                      CPA6149
  170 WRITE (6,232) CM                                               CPA6150
      ERROR=1.                                                       CPA6151
  175 READ (5,209) CM,CX,(A(J),J=1,12)                               CPA6152
      IF (CM.EQ.DEC.OR.CM.EQ.CID.OR.CM.EQ.NBK) GO TO 1101            CPA6153
      WRITE (6,233) CM,CX,(A(J),J=1,12)                              CPA6154
      GO TO 175                                                      CPA6155
  125 WRITE (4,218) KOD,CP                                           CPA6163
      WRITE (4, 210) (A(I), I = 1, 12)                               CPA6164
      IF (KOUNT - 6) 110, 130, 122                                   CPA6165
  130 END FILE 4                                                     CPA6166
      REWIND 4                                                       CPA6167
      REWIND 3                                                       CPA6168
      CALL COMP2                                                     CPA6169
      RETURN                                                         CPA6170
  149 CALL RESTRT                                                    CPA6171
      RETURN                                                         CPA6172
      END                                                            CPA6173
```

TABLE D-4. (Continued)

```
$IBFTC CPB6                                                              CPB6000
      SUBROUTINE COMP2                                                   CPB6001
C     AS OF 06/02/65                                                     CPB6002
      COMMON P, M, MAXT, MAXR, MAXC, KSP, KST, KURV 7, NCVG, NSTS, NFAB CPB6003
      COMMON NABL,NKSP,LMMF,NRMF,NREAD,CC                                CPB6004
      COMMON NRECD, NWPD, NWCT, NWST, NWTM                               CPB6005
      COMMON /CP2/ERROR,MAXMUM                                           CPB6006
      DIMENSION BCDB(5)                                                  CPB6007
      DIMENSION A(12), I 1(12), I 2(12), I 3(12), NABL(10), CC(12)       CPB6008
      DIMENSION P(16000), IRLINE(1), KPSD(1), NSTRT(1), M(16), KDAT(1)   CPB6010
      EQUIVALENCE (P(4000), IRLINE(1)), (P(8000), KPSD(1)), (P(13000),   CPB6011
     1  NSTRT(1)), (P(1), KDAT(1))                                       CPB6012
      REAL M, INC, NBK,NET                                               CPB6013
      INTEGER CP                                                         CPB6014
      DATA (BCDB(I),I=1,5)/6HTEMP. ,6HRESIS.,6HCAPAC.,6HCURVE ,6HMISC. /CPB6015
  200 FORMAT (5X, 4(I 5, E 10.2))                                        CPB6016
  201 FORMAT (5X, I 5, E 10.2, I 5)                                      CPB6017
  202 FORMAT (5X, 2(3I 5, E 10.2))                                       CPB6018
  203 FORMAT (5X, 3I 5, E 10.2, I 5)                                     CPB6019
  204 FORMAT (5X, I 5)                                                   CPB6020
  205 FORMAT (5X, 6E 10.2)                                               CPB6021
  206 FORMAT (5X,E10.2,I5)                                               CPB6022
  218 FORMAT (I 3, 3X, I 2)                                              CPB6031
  219 FORMAT (15HOKOD MUST NOT =I5,8H IN THE  A6,7H BLOCK.)              CPB6032
  220 FORMAT (26HOTABLE NO. = 0 IS ILLEGAL.)                             CPB6033
  222 FORMAT (30HO TOO MANY WORDS IN DAT BLOCK   )                       CPB6034
  223 FORMAT (7HOCODE= A6,12H IS ILLEGAL.)                               CPB6035
  224 FORMAT (25HOCANNOT FIND CURVE NAMED A6,17H ON LIBRARY TAPE.)       CPB6036
  230 FORMAT (28HOSOMETHING WRONG AFTER TABLEI6,17H AND BEFORE TABLEI6) CPB6037
  231 FORMAT (25HOSOMETHING WRONG IN TABLE I6)                           CPB6038
  232 FORMAT (7HOERROR,I5,E15.5,22H NEGATIVE NODE NUMBER.)               CPB6039
  233 FORMAT (7HOERROR,3I5,E15.5,26H NEGATIVE RESISTOR NUMBER.)          CPB6040
  235 FORMAT (14HOCAPACITOR NO.I5,22H IS GREATER THAN MAXT=I5)           CPB6041
  236 FORMAT (5HONODE I5,20H HAS NO CONNECTIONS.)                        CPB6042
  234 FORMAT (9HORESISTORI5,15H CONNECTS NODESI5,1H,I5,27H GREATER THAN CPB6043
     1MAX TEMP.NO.=I5)                                                   CPB6044
  240 FORMAT (5X,3I5,3E10.0,I5)                                          CPB6045
  241 FORMAT (5X,3I5,3E10.0,2I5)                                         CPB6046
  242 FORMAT (5X,I5,3E10.0,2I5)                                          CPB6047
  243 FORMAT (5X,I5,3E10.0,3I5)                                          CPB6048
  244 FORMAT (1HO,I6,33H DATA STORAGE EXCEEDED. TOTAL OF I6,             CPB6049
     1 12H DATA ITEMS.)                                                  CPB6050
  250 FORMAT (14H1P STORAGE MAP /14H0 TEMPERATURES,9X,I6,8X,6HTOTAL I6) CPB6051
  251 FORMAT (23H  TEMPERATURES        I6,8X,6HTOTAL I6)                 CPB6052
  252 FORMAT (23H  RESISTORS           I6,8X,6HTOTAL I6)                 CPB6053
  253 FORMAT (23H  CAPACITORS          I6,8X,6HTOTAL I6)                 CPB6054
  254 FORMAT (23H  CURRENTS (Q)        I6,8X,6HTOTAL I6)                 CPB6055
  255 FORMAT (23H  R-C PRODUCTS        I6,8X,6HTOTAL I6)                 CPB6056
  256 FORMAT (23H  PSEUDO-SEQENCE      I6,8X,6HTOTAL I6)                 CPB6057
  257 FORMAT (23H  DATA BLOCKS (CURVES) I6,8X,6HTOTAL I6)                CPB6058
  258 FORMAT (23H  DATA BLOCK ADDRESSES I6,8X,6HTOTAL I6)                CPB6059
      NSW2=1                                                             CPB6060
      NGRC=1                                                             CPB6061
      NGRC1=1                                                            CPB6062
      KGRC=1                                                             CPB6063
      KOUNT = 1                                                          CPB6064
      MAXT = 0                                                           CPB6065
      MAXR = 0                                                           CPB6066
      MAXC = 0                                                           CPB6067
      MAXP = 0                                                           CPB6068
      L = 0                                                              CPB6069
```

TABLE D-4. (Continued)

```
      DO 131 I=1,5000
      KDAT(I)=10
 131  CONTINUE
      IK = 1
      MAXS=1
 132  READ (4, 218) KOD, KOD 1
      IF (NGRC.EQ.2.AND.NGRC1.EQ.2) NGRC=1
      IF (NGRC.EQ.2) NGRC1=2
      GO TO (134, 135, 136, 137, 138, 139), KOUNT
C
C     READ IN INITIAL TEMPERATURES OF ALL NODES
C
 134  GO TO (3,13,9,13,14,13,13,13) ,KOD
 3    READ (4, 200) (I 1(I), A(I), I = 1, KOD 1)
      DO 7 I = 1, KOD 1
      J = I 1(I)
      IF (J - MAXT) 5, 5, 4
 4    MAXT = J
 5    IF (J.EQ.0) WRITE (6,232) J,A(I)
      IF (J.EQ.0) ERROR=1.
 6    P(J) = A(I)
      ITEM = J
      ATEM = A(I)
 7    CONTINUE
      GO TO 132
 9    READ (4, 201) N 1, A 1, N 2
      J = ITEM
      AA = ATEM
      DO 12 I = 1, N 2
      J = J + N 1
      AA = AA + A 1
      IF (J.GT.MAXT) MAXT=J
 11   P(J) = AA
      ITEM = J
      ATEM = AA
 12   CONTINUE
      GO TO 132
 13   WRITE (6,219) KOD,BCDB(KOUNT)
      ERROR =2.
      RETURN
 14   KOUNT = KOUNT + 1
      TEMPX=0.
      DO 141 I=1,MAXT
      IF (KDAT(I).EQ.10) P(I)=TEMPX
      IF (KDAT(I).NE.10) TEMPX=P(I)
 141  CONTINUE
      READ (4, 200) I 1(1)
      WRITE (3) MAXT, NCVG, NSTS
      WRITE (3) (P(I), I = 1, MAXT)
      WRITE (3) MAXT, NCVG, NSTS
      WRITE (3) (P(I), I = 1, MAXT)
      DO 15 I = 1, 5000
      P(I) = 0.
 15   CONTINUE
      WRITE (6,250) MAXT,MAXT
      MAXIM=MAXT+MAXT
      WRITE (6,251) MAXT,MAXIM
      GO TO 132
C
C     READ IN RESISTERS
C
```

TABLE D-4. (Continued)

```
135  GO TO (17,13,24,13,30,13,29,13),KOD                                CPB6131
17   READ (4, 202) (I 1(I), I 2(I), I 3(I), A(I), I = 1, KOD 1)         CPB6132
18   DO 22 I=1,KOD1                                                     CPB6133
     J = I 1(I)                                                         CPB6134
     IF (J.GT.MAXR) MAXR=J                                              CPB6135
19   IF(J.LE.0) WRITE (6,233) I1(I),I2(I),I3(I),A(I)                    CPB6136
     IF(J.LE.0) ERROR=1.                                                CPB6137
20   P(J) = A(I)                                                        CPB6138
     L = L + 1                                                          CPB6139
     IRLINE(L) = I 3(I) + 4096*(I 2(I) + 4096*J)                        CPB6140
     IF (J.GE.2048) IRLINE(L)=-IRLINE(L)                                CPB6141
     ITEM = J                                                           CPB6142
     ITEM 2 = I 2(I)                                                    CPB6143
     ITEM 3 = I 3(I)                                                    CPB6144
     ATEM = A(I)                                                        CPB6145
     IF (ITEM2.LE.MAXT.AND.ITEM3.LE.MAXT) GO TO 22                      CPB6146
     WRITE (6,234) ITEM,ITEM2,ITEM3,MAXT                                CPB6147
     ERROR=1.                                                           CPB6148
22   CONTINUE                                                           CPB6149
     GO TO 132                                                          CPB6150
25   READ (4,241) I1(1),I2(1),I3(1),X,Y,Z,KRVC,N2                       CPB6151
     J=ITEM                                                             CPB6152
     DO 2501 I=1,N2                                                     CPB6153
     J=J+I1(1)                                                          CPB6154
     ITEM2=ITEM2+I2(1)                                                  CPB6155
     ITEM3=ITEM3+I3(1)                                                  CPB6156
     X1=X1+X                                                            CPB6157
     Y1=Y1+Y                                                            CPB6158
     Z1=Z1+Z                                                            CPB6159
     KKRV=KKRV+KCRV                                                     CPB6160
     IF (J.GT.MAXR) MAXR=J                                              CPB6161
     P(J)=X1/(Y1*Z1)                                                    CPB6162
     L=L+1                                                              CPB6163
     IRLINE(L)=ITEM3+4096*(ITEM2+4096*J)                                CPB6164
     IF (J.GE.2048) IRLINE(L)=-IRLINE(L)                                CPB6165
     ITEM=J                                                             CPB6166
     NSTRT(KGRC)=J+4096*KKRV                                            CPB6167
     KGRC=KGRC+1                                                        CPB6168
     IF (ITEM2.LE.MAXT.AND.ITEM3.LE.MAXT) GO TO 2501                    CPB6169
     WRITE (6,234) ITEM,ITEM2,ITME3,MAXT                                CPB6170
     ERROR=1.                                                           CPB6171
2501 CONTINUE                                                           CPB6172
24   GO TO (2401,25),NGRC                                               CPB6173
2401 READ (4,203) I1(1),I2(1),I3(1),A(1),N2                             CPB6174
     J = ITEM                                                           CPB6175
     DO 28 I = 1, N 2                                                   CPB6176
     J = J + I 1(1)                                                     CPB6177
     ITEM 2 = ITEM 2 + I 2(1)                                           CPB6178
     ITEM 3 = ITEM 3 + I 3(1)                                           CPB6179
     ATEM = ATEM + A(1)                                                 CPB6180
     IF (J.GT.MAXR) MAXR=J                                              CPB6181
26   P(J) = ATEM                                                        CPB6182
     L = L + 1                                                          CPB6183
     IRLINE(L) = ITEM 3 + 4096*(ITEM 2 + 4096*J)                        CPB6184
     IF (J.GE.2048) IRLINE(L)=-IRLINE(L)                                CPB6185
     ITEM = J                                                           CPB6186
     IF (ITEM2.LE.MAXT.AND.ITEM3.LE.MAXT) GO TO 28                      CPB6187
     WRITE (6,234) ITEM,ITEM2,ITEM3,MAXT                                CPB6188
     ERROR=1.                                                           CPB6189
28   CONTINUE                                                           CPB6190
     GO TO 132                                                          CPB6191
```

TABLE D-4. (Continued)

```
 29    READ (4,240) I1(1),I2(1),I3(1),X,Y,Z,KCRV
       A(1)=X/(Y*Z)
       NGRC=2
       X1=X
       Y1=Y
       Z1=Z
       NSTRT(KGRC)=I1(1)+4096*KCRV
       KGRC=KGRC+1
       KKRV=KCRV
       GO TO 18
 30    KOUNT = KOUNT + 1
       READ (4, 202) I 1(1)
       WRITE (3) MAXR, NCVG, NSTS
       WRITE (3) (P(I), I = 1, MAXR)
       DO 31 I = 1, MAXR
       P(I) = 0.
 31    CONTINUE
       MAXIM=MAXIM+MAXR
       WRITE (6,252) MAXR,MAXIM
       GO TO 132
C
C     READ IN CAPACITORS
C
 136   GO TO (33,13,46,13,52,13,13,43),KOD
 33    READ (4, 200) (I 1(I), A(I), I = 1, KOD 1)
       NOP = 0
 39    DO 42 I = 1, KOD 1
       J = I 1(I)
       IF (J.GT.MAXT) WRITE (6,235) J,MAXT
       IF (J.GT.MAXT) ERROR=1.
       IF (J.GT.MAXC) MAXC=J
 35    IF (J.LE.0) WRITE (6,232) J,A(I)
       IF (J.LE.0) ERROR=1.
 36    P(J) = A(I)
       ITEM = J
       ATEM = A(I)
       KPSD(IK)=I1(I)+4096*(KKRV+4096*KKRV1)
       KPSD(IK)=-KPSD(IK)
       MAXP = MAXP + 1
       NODCON=0
       IK = IK + 1
       DO 41 K = 1, L
       IDAT 1 = IRLINE(K)/16777216
       IF (IRLINE(K).LT.0) IDAT1=IABS(IDAT1)+2048
       IDAT 2 = MOD(IRLINE(K)/4096, 4096)
       IDAT 3 = MOD(IRLINE(K), 4096)
       IF (IRLINE(K).LT.0) IDAT2=IABS(IDAT2)
       IF (IRLINE(K).LT.0) IDAT3=IABS(IDAT3)
       IF (I 1(I) - IDAT 2) 38, 37, 38
 37    DO 3601 I9=1,KGRC
       KGR=MOD(NSTRT(I9),4096)
       KGK=MOD(NSTRT(I9)/4096,4096)
       IF (KGR.EQ.IDAT1) GO TO 3602
 3601  CONTINUE
       KGK=0
 3602  KPSD(IK)=IDAT3+4096*(IDAT1+4096*KGK)
       IF (KGR.EQ.IDAT1) NSTRT(I9)=0
       MAXP = MAXP + 1
       NODCON=1
       IK = IK + 1
 38    IF (I 1(I) - IDAT 3) 41, 40, 41
```

TABLE D-4. (Continued)

```
40    DO 4001 I9=1,KGRC                                    CPB6253
      KGR=MOD(NSTRT(I9),4096)                              CPB6254
      KGK=MOD(NSTRT(I9)/4096,4096)                         CPB6255
      IF (KGR.EQ.IDAT1) GO TO 4002                         CPB6256
4001  CONTINUE                                             CPB6257
      KGK=0                                                CPB6258
4002  KPSD(IK)=IDAT2+4096*(IDAT1+4096*KGK)                 CPB6259
      IF (KGR.EQ.IDAT1) NSTRT(I9)=0                        CPB6260
      MAXP = MAXP + 1                                      CPB6261
      NODCON=1                                             CPB6262
      IK = IK + 1                                          CPB6263
41    CONTINUE                                             CPB6264
      IF (NODCON.EQ.1) GO TO 42                            CPB6265
      ERROR=1.                                             CPB6266
      WRITE (6,236) I1(I)                                  CPB6267
42    CONTINUE                                             CPB6268
      IF (NOP) 132, 132, 48                                CPB6269
43    READ (4,242) I1(1),X,Y,Z,KKRV,KKRV1                  CPB6270
      A(1)=X*Y*Z                                           CPB6271
      NGRC=2                                               CPB6272
      GO TO 39                                             CPB6273
46    GO TO (47,4701),NGRC                                 CPB6274
47    READ (4,201) N1,A1,N2                                CPB6275
      NOP = NOP + 1                                        CPB6276
      KOD 1 = 1                                            CPB6277
      GO TO 50                                             CPB6278
4701  READ (4,243) N1,X1,Y1,Z1,IK1,IK2,N2                  CPB6279
      KOD1=1                                               CPB6280
      GO TO 49                                             CPB6281
4702  I1(1)=ITEM+N1                                        CPB6282
      X=X+X1                                               CPB6283
      Y=Y+Y1                                               CPB6284
      Z=Z+Z1                                               CPB6285
      A(1)=X*Y*Z                                           CPB6286
      KKRV=KKRV+IK1                                        CPB6287
      KKRV1=KKRV1+IK2                                      CPB6288
      GO TO 39                                             CPB6289
48    IF (NOP - N 2) 49, 132, 132                          CPB6290
49    NOP = NOP + 1                                        CPB6291
      GO TO (50,4702),NGRC                                 CPB6292
50    I 1(1) = ITEM + N 1                                  CPB6293
      A(1) = ATEM + A 1                                    CPB6294
      GO TO 39                                             CPB6295
52    KOUNT = KOUNT + 1                                    CPB6296
      MAXP = MAXP + 1                                      CPB6297
      KPSD(MAXP) = 0                                       CPB6298
      READ (4, 200) I 1(1)                                 CPB6299
      WRITE (3) MAXC, NCVG, NSTS                           CPB6300
      WRITE (3) (P(I), I = 1, MAXC)                        CPB6301
      DO 53 I = 1, MAXC                                    CPB6302
      P(I) = 0.                                            CPB6303
53    CONTINUE                                             CPB6304
      WRITE (3) MAXC, NCVG, NSTS                           CPB6305
      WRITE (3) (P(I), I = 1, MAXC)                        CPB6306
      WRITE (3) MAXC, NCVG, NSTS                           CPB6307
      WRITE (3) (P(I), I = 1, MAXC)                        CPB6308
      WRITE (3) MAXP, NCVG, NSTS                           CPB6309
      WRITE (3) (KPSD(I), I = 1, MAXP)                     CPB6310
      NWPD = MAXP                                          CPB6311
      NRECD = 7                                            CPB6312
      KSP = 2*MAXT + MAXR + 3*MAXC + 1                     CPB6313
```

TABLE D-4. (Continued)

```
      KST = KSP + MAXP                                                  CPB6314
      KST 1 = KST                                                       CPB6315
      NKSP = KST - 1                                                    CPB6316
      JJ = 0                                                            CPB6317
      N2=0                                                              CPB6318
      MAXIM=MAXIM+MAXC                                                  CPB6319
      WRITE (6,253) MAXC,MAXIM                                          CPB6320
      MAXIM=MAXIM+MAXC                                                  CPB6321
      WRITE (6,254) MAXC,MAXIM                                          CPB6322
      MAXIM=MAXIM+MAXC                                                  CPB6323
      WRITE (6,255) MAXC,MAXIM                                          CPB6324
      MAXIM=MAXIM+MAXP                                                  CPB6325
      WRITE (6,256) MAXP,MAXIM                                          CPB6326
      GO TO 132                                                         CPB6327
C                                                                       CPB6328
C   READ IN DATA SUB-BLOCKS                                             CPB6329
C                                                                       CPB6330
 137  GO TO (56,13,66,67,68,13,13,13,691) ,KOD                          CPB6331
 56   IF (KOD 1) 57, 57, 64                                             CPB6332
 57   READ (4, 204) NTAB                                                CPB6333
      IF (NTAB) 62, 58, 59                                              CPB6334
 58   WRITE (6, 220)                                                    CPB6335
      GO TO 132                                                         CPB6336
 59   IF (NSW2.EQ.1) GO TO 591                                          CPB6337
      WRITE (6,230) NTABO,NTAB                                          CPB6338
      ERROR=1.                                                          CPB6339
      NSW2=1                                                            CPB6340
      KST=KST1+JJ                                                       CPB6341
 591  NSTRT(MAXS)=KST+2**18*NTAB                                        CPB6342
      NSW2=2                                                            CPB6343
      NTABO=NTAB                                                        CPB6344
      MAXS=MAXS+1                                                       CPB6345
      GO TO 132                                                         CPB6346
 62   IF (NSW2.EQ.2) GO TO 621                                          CPB6347
      WRITE (6,231) NTABO                                               CPB6348
      ERROR=1.                                                          CPB6349
 621  NSW2=1                                                            CPB6350
      IF (NTAB+7) 63,61,63                                              CPB6351
 61   KURV 7 = KST 1 + JJ - KST                                         CPB6352
 63   KST = KST 1 + JJ                                                  CPB6353
      GO TO 132                                                         CPB6354
 64   READ (4, 205) (A(I), I = 1, KOD 1)                                CPB6355
 641  DO 65 I = 1, KOD 1                                                CPB6356
      JJ = JJ + 1                                                       CPB6357
      P(JJ) = A(I)                                                      CPB6358
 65   CONTINUE                                                          CPB6359
      IF (N2.GT.0) GO TO 661                                            CPB6360
      GO TO 132                                                         CPB6361
 66   READ (4,206) AINC,N2                                              CPB6362
      A(1)=P(JJ)                                                        CPB6363
      KOD1=1                                                            CPB6364
 661  A(1)=A(1)+AINC                                                    CPB6365
      N2=N2-1                                                           CPB6366
      GO TO 641                                                         CPB6367
 67   READ (4, 205) A(1)                                                CPB6368
      P(JJ + 1) = 32767.                                                CPB6369
      P(JJ + 2) = A(1)                                                  CPB6370
      JJ = JJ + 2                                                       CPB6371
      GO TO 132                                                         CPB6372
 68   KOUNT = KOUNT + 1                                                 CPB6373
      KST = KST - 1                                                     CPB6374
```


LOCKHEED CALIFORNIA COMPANY

TABLE D-4. (Continued)

```
      NWCT = JJ
      WRITE (3) NWCT, NCVG, NSTS
      WRITE (3) (P(I), I = 1, NWCT)
      NRECD = NRECD + 2
      READ (4, 205) A(1)
      WRITE (3) MAXS, NCVG, NSTS
      WRITE (3) (NSTRT(I), I = 1, MAXS)
      NWST = MAXS
      JJ = 0
      DO 69 I=1,40
      P(I)=0.
   69 CONTINUE
      MAXIM=MAXIM+NWCT
      WRITE (6,257) NWCT,MAXIM
      MAXIM=MAXIM+MAXS
      WRITE (6,258) MAXS,MAXIM
      IF (MAXIM.LT.MAXMUM) GO TO 132
      ERROR=1.
      WRITE (6,244) MAXMUM,MAXIM
      GO TO 132
  691 READ (4,204) ICOD
      KDAT(JJ+1) = 32765
      KDAT(JJ+2) = ICOD
      JJ = JJ+2
      GO TO 132
  138 GO TO (70, 13, 13, 13, 72, 72), KOD
C70   READ (4, 200) (I1(I),A(I),I=1,KOD1)
   70 READ (4, 205) (       A(I),I=1,KOD1)
      DO 71 I = 1, KOD 1
C     JJ = I1(I)
      JJ = JJ+1
      P(JJ+20)=A(I)
   71 CONTINUE
      GO TO 132
   72 KOUNT = KOUNT + 1
C   THE FOLLOWING 5 CARDS GO WITH THE NON-C CARDS ABOVE
      P(1)=P(23)
      P(2)=P(23)
      P(3)=P(22)
      P(4)=P(21)
      IF (P(24).NE.0.) P(7)=P(24)
      P(11)=P(2)
      JJ=16
      WRITE (3) JJ, NCVG, NSTS
      WRITE (3) (P(I), I = 1, JJ)
      NWTM = JJ
      END FILE 3
      NRECD=9
      REWIND 4
  139 REWIND 3
      RETURN
      END
```

TABLE D-4. (Continued)

```
$IBFTC CPC6
       SUBROUTINE RESTRT                                                 CPC6000
C      AS OF 04/27/65                                                    CPC6001
       COMMON P, M, MAXT, MAXR, MAXC, KSP, KST, KURV 7, NCVG, NSTS, NFAB CPC6002
       COMMON NABL,NKSP,LMMF,NRMF,NREAD,CC                               CPC6003
       COMMON NRECD, NWPD, NWCT, NWST, NWTM                              CPC6004
       COMMON /CP2/ERROR,MAXMUM                                          CPC6005
       COMMON /CPPLT/NPLT,NEND                                           CPC6006
       DIMENSION A(12), I 1(12), I 2(12), I 3(12), NABL(10), CC(12)      CPC6007
       DIMENSION A1(6)                                                   CPC6008
       DIMENSION CZ(6),CY(6)                                             CPC6009
       DIMENSION P(16000), IRLINE(1), KPSD(1), NSTRT(1), M(16), KDAT(1)  CPC6010
       EQUIVALENCE (P(4000), IRLINE(1)), (P(8000), KPSD(1)), (P(13000),  CPC6011
      1  NSTRT(1)), (P(1), KDAT(1))                                      CPC6012
       REAL M, INC, NBK,NET                                              CPC6013
       INTEGER CP                                                        CPC6014
       DATA ENDF/6H   -1 /                                               CPC6015
       DATA NET/6HNET    /                                               CPC6016
       DATA DEC/6HDEC    /                                               CPC6017
       DATA INC/6HINC    /                                               CPC6018
       DATA PER/6HPER    /                                               CPC6019
       DATA NBK/6HNBK    /                                               CPC6020
       DATA CID/6HCID    /                                               CPC6021
       DATA TAP/6HTAP    /                                               CPC6022
       DATA COD/6HCOD    /                                               CPC6023
       DATA ZERO/6H000000/                                               CPC6024
       DATA END /6HEND    /                                              CPC6025
       DATA (CY(I),I=1,6)/2H 1,2H 2,2H 3,2H 4,2H 5,2H 6/,ZERZ/2H00/      CPC6026
       DATA (CZ(I),I=1,6)/2H01,2H02,2H03,2H04,2H05,2H06/,BLANK/1H /      CPC6027
       DATA ZERY/2H 0/                                                   CPC6028
 200   FORMAT (5X, 4(I 5, E 10.2))                                       CPC6029
 201   FORMAT (5X, I 5, E 10.2, I 5)                                     CPC6030
 204   FORMAT (5X, I 5)                                                  CPC6031
 205   FORMAT (5X, 6E 10.2)                                              CPC6034
 209   FORMAT (A3,A2,12A6)                                               CPC6035
 210   FORMAT (5X,12A6)                                                  CPC6037
 211   FORMAT (5X, A3,I2,12A6)                                           CPC6038
 212   FORMAT (1H1,9X,12A6)                                              CPC6039
 215   FORMAT (40H0 KOUNT IS NOT CORRECT IN RESTRT ROUTINE )             CPC6040
 218   FORMAT (I 3, 3X, I 2)                                             CPC6042
 222   FORMAT (30H0 TOO MANY WORDS IN DAT BLOCK   )                      CPC6044
 223   FORMAT (7H0CODE= A6,12H IS ILLEGAL.)                              CPC6047
 224   FORMAT (25H0CANNOT FIND CURVE NAMED A6,17H ON LIBRARY TAPE.)      CPC6048
 236   FORMAT (15H0BLOCK NUMBER =I6,29H IS ILLEGAL. RESTART DELETED.   ) CPC6049
 237   FORMAT (14H1END OF CASES.)                                        CPC6052
 241   FORMAT (5X,E10.0,I5)                                              CPC6053
 242   FORMAT (5X,2E10.0,I5)                                             CPC6054
 243   FORMAT (5X,3E10.0,I5)                                             CPC6055
 244   FORMAT (5X,4E10.0,I5)                                             CPC6056
 245   FORMAT (5X,5E10.0,I5)                                             CPC6057
 246   FORMAT (5X,6E10.0,I5)                                             CPC6058
       NRECD=9                                                           CPC6059
       MATLIB =1                                                         CPC6060
       LINE=1                                                            CPC6061
 149   REWIND 4                                                          CPC6062
 139   REWIND 3                                                          CPC6063
       DO 310 I = 1, 16                                                  CPC6064
       M(I) = 0.0                                                        CPC6065
 310   CONTINUE                                                          CPC6066
       M(7) = 0.25                                                       CPC6067
       MAXP = 0                                                          CPC6068
                                                                         CPC6069
```

TABLE D-4. (Continued)

```
      JJ = 1                                                            CPC6070
      DO 75 I = 1, NRECD                                                CPC6071
      READ (3) N 1                                                      CPC6072
      MAXP = MAXP + N 1                                                 CPC6073
      READ (3) (P(J), J = JJ, MAXP)                                     CPC6074
      JJ = MAXP + 1                                                     CPC6075
75    CONTINUE                                                          CPC6076
      READ (3) N 1                                                      CPC6077
      READ (3) (M(I),I=1,N1)                                            CPC6078
      MAXP = MAXP + N 1                                                 CPC6079
      REWIND 3                                                          CPC6080
      IF (MAXP -MAXMUM) 701,701,81                                      CPC6081
81    WRITE (6, 222)                                                    CPC6082
100   ERROR=2.                                                          CPC6083
701   IF (MATLIB.EQ.1) READ (5,209) CM,CX,(A(I),I=1,12)                 CPC6084
      IF (MATLIB.EQ.2) READ (  12  ,209) CM,CX,(A(I),I=1,12)            CPC6085
      CP=-1                                                             CPC6086
1101  DO 1102 I=1,6                                                     CPC6087
1102  IF (CX.EQ.CY(I).OR.CX.EQ.CZ(I)) CP=I                              CPC6088
      IF (CX.EQ.BLANK.OR.CX.EQ.ZERZ.OR.CX.EQ.ZERY) CP=0                 CPC6089
      IF (CP.GE.0) GO TO 1103                                           CPC6090
      WRITE (6,215)                                                     CPC6091
      ERROR=1.                                                          CPC6092
      CP=0                                                              CPC6093
1103  IF (CM.EQ.END) GO TO 900                                          CPC6094
      IF (CM.EQ.TAP) GO TO 150                                          CPC6095
      IF (LINE.LT.60) GO TO 7011                                        CPC6096
      WRITE (6,212)            (CC(I),I=1,12)                           CPC6097
      LINE =1                                                           CPC6098
7011  IF (CM.EQ.CID) GO TO 117                                          CPC6099
      WRITE (6,211) CM,CP,(A(I),I=1,12)                                 CPC6100
      LINE =LINE+1                                                      CPC6101
      IF (CM .EQ. DEC) GO TO 702                                        CPC6102
      IF (CM .EQ. INC) GO TO 703                                        CPC6103
      IF (CM.EQ.COD) GO TO 707                                          CPC6104
      IF (CM .EQ. PER) GO TO 704                                        CPC6105
      IF (CM .EQ. NBK) GO TO 705                                        CPC6106
      WRITE (6,223) CM                                                  CPC6107
      ERROR=1.                                                          CPC6108
      GO TO 701                                                         CPC6109
117   CMID=CM                                                           CPC6110
      ICP=CP                                                            CPC6111
      DO 1171 I=1,12                                                    CPC6112
      CC(I)=A(I)                                                        CPC6113
1171  CONTINUE                                                          CPC6114
      LINE=1                                                            CPC6115
      WRITE (6,212)            (CC(I),I=1,12)                           CPC6116
      GO TO 701                                                         CPC6117
150   IF (MATLIB.EQ.2) GO TO 160                                        CPC6118
      MATLIB=2                                                          CPC6119
      FIRST = 0.                                                        CPC6120
      WRITE (6,211) CM,CP,(A(I),I=1,12)                                 CPC6121
      LINE = LINE+1                                                     CPC6122
151   READ (  12  ,209) Z1, Z2,Z3                                       CPC6123
      IF (Z1.NE.TAP) GO TO 151                                          CPC6124
      IF (Z3.EQ.A(1)) GO TO 701                                         CPC6125
      IF (Z3.NE.ZERO) GO TO 151                                         CPC6126
      REWIND 12                                                         CPC6127
      FIRST =FIRST+1.                                                   CPC6128
      IF (FIRST.EQ.1.) GO TO 151                                        CPC6129
      WRITE (6,224) A(1)                                                CPC6130
```

TABLE D-4. (Continued)

```
      ERROR =1.                                                         CPC6131
  160 MATLIB=1                                                          CPC6132
      GO TO 701                                                         CPC6133
  702 KOD=1                                                             CPC6134
      GO TO 706                                                         CPC6135
  703 KOD = 2                                                           CPC6136
      GO TO 706                                                         CPC6137
  704 KOD = 3                                                           CPC6138
      GO TO 706                                                         CPC6139
  707 KOD = 5                                                           CPC6140
      GO TO 706                                                         CPC6141
  705 KOD = 4                                                           CPC6142
  706 WRITE (4,218) KOD,CP                                              CPC6143
      WRITE (4, 210) (A(I), I = 1, 12)                                  CPC6144
      IF (KOD.LT.4) GO TO 701                                           CPC6145
      IF (KOD.EQ.4) GO TO 708                                           CPC6146
      WRITE (6,215)                                                     CPC6147
      ERROR=1.                                                          CPC6148
  708 END FILE 4                                                        CPC6149
      REWIND 4                                                          CPC6150
  714 READ (4, 218) KOD, KOD 1                                          CPC6151
      GO TO (717, 122, 122, 719), KOD                                   CPC6152
  717 IF (KOD 1) 718, 718, 716                                          CPC6153
  716 WRITE (6, 250)                                                    CPC6154
  250 FORMAT (34H KOD1 IS NOT ZERO IN RESTART BLOCK )                   CPC6155
      ERROR=1.                                                          CPC6156
      GO TO 714                                                         CPC6157
  122 WRITE (6,215)                                                     CPC6158
      ERROR=1.                                                          CPC6159
      GO TO 719                                                         CPC6160
  719 REWIND 4                                                          CPC6161
      REWIND 3                                                          CPC6162
      MAXP = 0                                                          CPC6163
      M(1)=M(2)                                                         CPC6164
      M(11)=M(2)                                                        CPC6165
      JJ = 1                                                            CPC6166
      DO 800 I = 1, NRECD                                               CPC6167
      GO TO (780, 780, 782, 784, 784, 784, 786, 788, 790), I            CPC6168
  780 N 1 = MAXT                                                        CPC6169
      GO TO 798                                                         CPC6170
  782 N 1 = MAXR                                                        CPC6171
      GO TO 798                                                         CPC6172
  784 N 1 = MAXC                                                        CPC6173
      GO TO 798                                                         CPC6174
  786 N 1 = NWPD                                                        CPC6175
      GO TO 798                                                         CPC6176
  788 N 1 = NWCT                                                        CPC6177
      GO TO 798                                                         CPC6178
  790 N 1 = NWST                                                        CPC6179
  798 WRITE (3) N 1, NCVG, NSTS                                         CPC6180
      MAXP = MAXP + N 1                                                 CPC6181
      WRITE (3) (P(J), J = JJ, MAXP)                                    CPC6182
      JJ = MAXP + 1                                                     CPC6183
  800 CONTINUE                                                          CPC6184
      N 1 = 16                                                          CPC6185
      WRITE (3) N 1, NCVG, NSTS                                         CPC6186
      WRITE (3) (M(I),I=1,N1)                                           CPC6187
      END FILE 3                                                        CPC6188
      REWIND 3                                                          CPC6189
      RETURN                                                            CPC6190
  718 READ (4, 204) NBLK                                                CPC6191
```

TABLE D-4. (Continued)

```
      IF (NBLK.EQ.11) GO TO 714                                         CPC6192
      IF (NBLK.GE.1.AND.NBLK.LE.5) GO TO (720,730,740,750,760),NBLK     CPC6193
      WRITE (6,236) NBLK                                                CPC6194
      ERROR=1.                                                          CPC6195
      NBLK =1                                                           CPC6196
      GO TO 714                                                         CPC6197
  720 READ (4, 218) KOD, KOD 1                                          CPC6198
      GO TO (722, 725, 122, 719), KOD                                   CPC6199
  722 IF (KOD 1) 721, 721, 723                                          CPC6200
  721 READ (4, 204) I                                                   CPC6201
      GO TO 714                                                         CPC6202
  723 READ (4, 200) (I 1(I), A(I), I = 1, KOD 1)                        CPC6203
      DO 724 I = 1, KOD 1                                               CPC6204
      J = I 1(I)                                                        CPC6205
      JJ = J + MAXT                                                     CPC6206
      P(JJ) = A(I)                                                      CPC6207
      P(J) = A(I)                                                       CPC6208
      ITEM = I 1(I)                                                     CPC6209
      ATEM = A(I)                                                       CPC6210
  724 CONTINUE                                                          CPC6211
      GO TO 720                                                         CPC6212
  725 READ (4, 201) N 1, A 1, N 2                                       CPC6213
      J = ITEM                                                          CPC6214
      AA = ATEM                                                         CPC6215
      DO 726 I = 1, N 2                                                 CPC6216
      J = J + N 1                                                       CPC6217
      JJ = J + MAXT                                                     CPC6218
      AA = AA + A 1                                                     CPC6219
      P(J) = AA                                                         CPC6220
      P(JJ) = AA                                                        CPC6221
      ITEM = J                                                          CPC6222
      ATEM = AA                                                         CPC6223
  726 CONTINUE                                                          CPC6224
      GO TO 720                                                         CPC6225
  730 READ (4, 218) KOD, KOD 1                                          CPC6226
      GO TO (731, 734, 122, 719), KOD                                   CPC6227
  731 IF (KOD 1) 721, 721, 732                                          CPC6228
  732 READ (4, 200) (I 1(I), A(I), I = 1, KOD 1)                        CPC6229
      DO 733 I = 1, KOD 1                                               CPC6230
      J = I 1(I) + 2*MAXT                                               CPC6231
      P(J) = A(I)                                                       CPC6232
      ITEM = I 1(I)                                                     CPC6233
      ATEM = A(I)                                                       CPC6234
  733 CONTINUE                                                          CPC6235
      GO TO 730                                                         CPC6236
  734 READ (4, 201) N 1, A 1, N 2                                       CPC6237
      J = ITEM                                                          CPC6238
      AA = ATEM                                                         CPC6239
      DO 736 I = 1, N 2                                                 CPC6240
      J = J + N 1                                                       CPC6241
      JJ = J + 2*MAXT                                                   CPC6242
      AA = AA + A 1                                                     CPC6243
      P(JJ) = AA                                                        CPC6244
      ITEM = J                                                          CPC6245
      ATEM = AA                                                         CPC6246
  736 CONTINUE                                                          CPC6247
      GO TO 730                                                         CPC6248
  740 READ (4, 218) KOD, KOD 1                                          CPC6249
      GO TO (741, 744, 122, 719), KOD                                   CPC6250
  741 IF (KOD 1) 721, 721, 742                                          CPC6251
  742 READ (4, 200) (I 1(I), A(I), I = 1, KOD 1)                        CPC6252
```


LOCKHEED
CALIFORNIA COMPANY

TABLE D-4. (Continued)

```
      DO 743 I = 1, KOD 1
      J = I 1(I) + 2*MAXT + MAXR
      P(J) = A(I)
      ITEM = I 1(I)
      ATEM = A(I)
743   CONTINUE
      GO TO 740
744   READ (4, 201) N 1, A 1, N 2
      J = ITEM
      AA = ATEM
      DO 746 I = 1, N 2
      J = J + N 1
      JJ = J + 2*MAXT + MAXR
      AA = AA + A 1
      P(JJ) = AA
      ITEM = J
      ATEM = AA
746   CONTINUE
      GO TO 740
750   JJ = 5
751   READ (4, 218) KOD, KOD 1
      GO TO (752, 122, 122, 719), KOD
752   IF (KOD 1) 721, 721, 753
753   READ (4, 205) (        A(I),I=1,KOD1)
      DO 754 I = 1, KOD 1
      JJ = JJ-1
      IF (JJ.EQ.1) M(7)=A(I)
      IF (JJ.EQ.1) GO TO 754
      M(JJ) = A(I)
754   CONTINUE
      GO TO 751
760   READ (4, 218) KOD, KOD 1
      IF (KOD.EQ.4) GO TO 719
759   READ (4, 204) NTAB
      IF (NTAB) 760, 714, 761
761   L = LOC (NTAB)
762   READ (4, 218) KOD, KOD 1
      GO TO (764,770,763,719,773) ,KOD
763   L = L + 1
      READ (4, 205) A(1)
      P(L) = A(1)
      GO TO 762
764   IF (KOD 1) 759, 759, 766
766   READ (4, 205) (A(I), I = 1, KOD 1)
      KOD2=KOD1
      DO 769 I = 1, KOD 1
      L = L + 1
      P(L) = A(I)
769   CONTINUE
      GO TO 762
770   IF (KOD1.LE.0) KOD1=KOD2
      GO TO (7701,7702,7703,7704,7705,7706),KOD1
7701  READ (4,241) (A1(I),I=1,KOD1),INCR
      GO TO 771
7702  READ (4,242) (A1(I),I=1,KOD1),INCR
      GO TO 771
7703  READ (4,243) (A1(I),I=1,KOD1),INCR
      GO TO 771
7704  READ (4,244) (A1(I),I=1,KOD1),INCR
      GO TO 771
7705  READ (4,245) (A1(I),I=1,KOD1),INCR
```


LOCKHEED
CALIFORNIA COMPANY

TABLE D-4. (Continued)

```
      GO TO 771                                                         CPC6314
 7706 READ (4,246) (A1(I),I=1,KOD1),INCR                                CPC6315
      GO TO 771                                                         CPC6316
  771 DO 772 I=1,KOD1                                                   CPC6317
      A(I)=A(I)+A1(I)                                                   CPC6318
      L=L+1                                                             CPC6319
      P(L)=A(I)                                                         CPC6320
  772 CONTINUE                                                          CPC6321
      INCR=INCR-1                                                       CPC6322
      IF (INCR.GT.0) GO TO 771                                          CPC6323
      GO TO 762                                                         CPC6324
  773 READ (4,204) ICOD                                                 CPC6325
      KDAT (L+1) = 32765                                                CPC6326
      KDAT (L+2) = ICOD                                                 CPC6327
      L=L+2                                                             CPC6328
      GO TO 762                                                         CPC6329
  900 REWIND 3                                                          CPC6330
      REWIND 4                                                          CPC6331
      WRITE (6,237)                                                     CPC6332
      NEND=1                                                            CPC6333
      RETURN                                                            CPC6334
      END                                                               CPC6335


$IBFTC EXC6                                                             EXC6000
      SUBROUTINE EXCUT(NLAG)                                            EXC6001
C     AS OF 05/19/65                                                    EXC6002
      COMMON P(16000),M(16),MAXT,MAXR,MAXC,KSP,KST,KURV7,NCVG           EXC6003
      COMMON NSTS,NFAB,NABL(10),NKSP                                    EXC6004
      DIMENSION KP(1)                                                   EXC6005
      EQUIVALENCE (P(1),KP(1))                                          EXC6006
      COMMON /CP2/ERROR                                                 EXC6007
      COMMON /EXADR/LCTN,LCTO,LCR,LCC,LCQ,LCRC,LCPS,LCK,LCKA            EXC6008
      COMMON /EXPLT/ NLFLG                                              EXC6009
      COMMON /EXFLG/NTFLG,NSAV,NFLG                                     EXC6010
      COMMON /EXSTA/ NC,NCUR1,NCURV(20)                                 EXC6011
      COMMON /EXCVG/ NFLAG,ITMAX,ITER,NODE,PM1,PM,DTO,ATEM,NTEM,MTEM    EXC6012
      NFLG=0                                                            EXC6013
      NSAV=0                                                            EXC6014
      NLFLG=0                                                           EXC6015
      REWIND 11                                                         EXC6016
      CALL CDATA                                                        EXC6017
      CALL HTBAL(NLAG)                                                  EXC6018
      RETURN                                                            EXC6019
      END                                                               EXC6020
```

TABLE D-4. (Continued)

```
$IBFTC CDA6                                                              CDA6000
      SUBROUTINE CDATA                                                   CDA6001
C     AS OF 04/27/65                                                     CDA6002
      COMMON P(16000),M(16),MAXT,MAXR,MAXC,KSP,KST,KURV7,NCVG,           CDA6003
     1 NSTS,NFAB,NABL(10),NKSP,LMMF,NRMF,NREAD,CC(12)                    CDA6004
      COMMON NR,MAXP,NWCT,MAXS,MAXM                                      CDA6005
      COMMON /CP2/ERROR                                                  CDA6006
      COMMON /EXADR/ LCTN,LCTO,LCR,LCC,LCQ,LCRC,LCPS,LCK,LCKA            CDA6007
      REAL M                                                             CDA6008
      LCTN=1                                                             CDA6009
      READ(3) MAXT                                                       CDA6010
      LCTO=LCTN+MAXT                                                     CDA6011
      L=LCTO-1                                                           CDA6012
      READ(3) (P(I),I=LCTN,L)                                            CDA6013
      READ(3) MAXT                                                       CDA6014
      LCR=LCTO+MAXT                                                      CDA6015
      L=LCR-1                                                            CDA6016
      READ (3) (P(I),I=LCTO,L)                                           CDA6017
      READ (3) MAXR                                                      CDA6018
      LCC=LCR+MAXR                                                       CDA6019
      L=LCC-1                                                            CDA6020
      READ (3) (P(I),I=LCR,L)                                            CDA6021
      READ (3) MAXC                                                      CDA6022
      LCQ=LCC+MAXC                                                       CDA6023
      L=LCQ-1                                                            CDA6024
      READ (3) (P(I),I=LCC,L)                                            CDA6025
      READ (3) MAXC                                                      CDA6026
      LCRC=LCQ+MAXC                                                      CDA6027
      L=LCRC-1                                                           CDA6028
      READ (3) (P(I),I=LCQ,L)                                            CDA6029
      READ (3) MAXC                                                      CDA6030
      LCPS=LCRC+MAXC                                                     CDA6031
      L=LCPS-1                                                           CDA6032
      READ (3) (P(I),I=LCRC,L)                                           CDA6033
      READ (3) MAXP                                                      CDA6034
      LCK=LCPS+MAXP                                                      CDA6035
      L=LCK-1                                                            CDA6036
      READ (3) (P(I),I=LCPS,L)                                           CDA6037
      READ (3) NWCT                                                      CDA6038
      LCKA=LCK+NWCT                                                      CDA6039
      L=LCKA-1                                                           CDA6040
      READ (3) (P(I),I=LCK,L)                                            CDA6041
      READ (3) MAXS                                                      CDA6042
      L=LCKA+MAXS-1                                                      CDA6043
      READ (3) (P(I),I=LCKA,L)                                           CDA6044
      READ (3) MAXM                                                      CDA6045
      READ (3) (M(I),I=1,MAXM)                                           CDA6046
      IF (M(7).LE.0.) M(7)=.25                                           CDA6047
      RETURN                                                             CDA6048
      END                                                                CDA6049
```

TABLE D-4. (Continued)

```
$IBFTC HTB6
      SUBROUTINE HTBAL (NLAG)
C     AS OF 06/12/65
      COMMON P, M, MAXT, MAXR, MAXC, KSP, KST, KURV 7, NCVG, NSTS, NFAB
      COMMON NABL,NKSP,LMMF,NRMF,NREAD
      COMMON /EXFLG/NTFLG,NSAV,NFLG
      COMMON / CP2/ ERROR
      COMMON / EXPLT/ NLFLG
      COMMON /CPITM/ ITIMO
      DIMENSION P(16000), M(16), KPSD(1), NABL(10)
      EQUIVALENCE (P(1), KPSD(1))
      REAL M,LIN
      ITIME = ITIMO
 24   IR=1
      T16TH = (M(3)-M(2))/16.
      T16TS = T16TH+M(2)
      CALL SLITE (0)
      IF (M(16).LE.0.) M(16)=0.
      IF (M(16).GT.0.) GO TO 29
      DO 28 I=KSP,NKSP
      IF (KPSD(I)) 25,29,26
 25   NN=-KPSD(I)
      ND= MOD(NN,4096)
      K1= NN/2**24
      K2= MOD(NN/2**12,4096)
      IF (K1.EQ.0.OR.K2.EQ.0) GO TO 28
      RHO = LIN(P(ND),K1)
      CP = LIN(P(ND),K2)
      NC = ND+2*MAXT+MAXR
      P(NC) = P(NC)*CP*RHO
      GO TO 28
 26   ND = MOD(KPSD(I),4096)
      NR = MOD(KPSD(I)/4096,4096)
      K3 = KPSD(I)/2**24
      IF (K3.EQ.0) GO TO 28
      J = NR+2*MAXT
      FACT=LIN(P(ND),K3)
      P(J)=P(J)/FACT
 28   CONTINUE
 29   CONTINUE
 40   GO TO (1, 4, 4), NSTS
 1    NPRINT = 1
 2    I 1 = 0
      I 2 = MAXT
      I 3 = I 2 + MAXT
      I 4 = I 3 + MAXR
      I 5 = I 4 + MAXC
      I 6 = I 5 + MAXC
      SUM 1 = 0.
      SUM 2 = 0.
      SUM 3 = 0.
 3    CALL FUNCT
      IF (ERROR.GT.0.) GO TO 770
      IF (NSTS.EQ.2) IR=2
      IF (NCVG.EQ.2) IR=2
 4    NFLAG = 1
      IF (M(1).LT.T16TS) GO TO 203
      T16TS=T16TS+T16TH
      DO 202 J=1,4
      I=5-J
      CALL SLITET (I,LIT)
```

TABLE D-4. (Continued)

```
      GO TO (202,201),LIT                                               HTB6061
201   CONTINUE                                                          HTB6062
      CALL SLITE (I)                                                    HTB6063
      GO TO 203                                                         HTB6064
202   CONTINUE                                                          HTB6065
203   CONTINUE                                                          HTB6066
      RCM = M(4)                                                        HTB6067
      M(9) = 0.                                                         HTB6068
      J = KSP                                                           HTB6069
44    NN = MOD(KPSD(J),4096)                                            HTB6070
      NN = ISIGN(NN,KPSD(J))                                            HTB6071
      IF (NN.EQ.0) GO TO 50                                             HTB6072
      GO TO 14                                                          HTB6073
5     NN = MOD(KPSD(J),4096)                                            HTB6074
      NN = ISIGN(NN,KPSD(J))                                            HTB6075
      IF (NN) 6,  7, 19                                                 HTB6076
6     IF (J.LT.NKSP) GO TO 8                                            HTB6077
7     NFLAG = 2                                                         HTB6078
8     P(NRC) = ABS(P(NC))/SUM2*SIGN(1.,P(NRC))                          HTB6079
      IF (SUM2.EQ.0.) WRITE (6,2003) N2                                 HTB6080
      IF (SUM2.EQ.0.) ERROR=2.                                          HTB6081
2003  FORMAT (23H0SUM(1./R)=0. FOR NODE I5)                             HTB6082
      IF (M(16).LE.0. ) P(NRC)=ABS(P(NRC))                              HTB6083
      IF (IR.EQ.2) GO TO 10                                             HTB6084
      IF (P(NC).NE.0.) GO TO 11                                         HTB6085
10    P(NTNK) = (SUM 1 + P(NQ))/SUM 2                                   HTB6086
      GO TO 13                                                          HTB6087
11    P(NTNK)=M(5)/ABS(P(NC))*(SUM1-SUM 3 + P(NQ)) + P(NTOK)            HTB6088
      IF (P(NRC).NE.0.) GO TO 132                                       HTB6089
2001  FORMAT (23H0  R-C PRODUCT FOR NODE I 4, 6H    IS E 20.8)          HTB6090
130   WRITE (6, 2001) N 2, P(NRC)                                       HTB6091
      ERROR=2.                                                          HTB6092
      GO TO 13                                                          HTB6093
132   IF (ABS(P(NRC)).GE.RCM) GO TO 13                                  HTB6094
12    M(9) = N 2                                                        HTB6095
      RCM = ABS(P(NRC))                                                 HTB6096
13    SUM 1 = 0.                                                        HTB6097
      SUM 2 = 0.                                                        HTB6098
      SUM 3 = 0.                                                        HTB6099
      GO TO (14, 72, 99), NFLAG                                         HTB6100
14    N 2 = - NN                                                        HTB6101
      NTNK = N 2 + I 1                                                  HTB6102
      NC = N 2 + I 4                                                    HTB6103
      NTOK = N 2 + I 2                                                  HTB6104
      NQ = N 2 + I 5                                                    HTB6105
      NRC = N 2 + I 6                                                   HTB6106
      J = J + 1                                                         HTB6107
19    NN = MOD(KPSD(J)/4096,4096)                                       HTB6108
      IDAT = MOD(KPSD(J), 4096)                                         HTB6109
      NR = NN + I 3                                                     HTB6110
      NTOM = IDAT + I 2                                                 HTB6111
      J = J + 1                                                         HTB6112
22    SUM 1 = SUM 1 + P(NTOM)/P(NR)                                     HTB6113
      SUM 2 = SUM 2 + 1./P(NR)                                          HTB6114
      SUM 3 = SUM 3 + P(NTOK)/P(NR)                                     HTB6115
      GO TO 5                                                           HTB6116
50    WRITE (6, 52)                                                     HTB6117
52    FORMAT (35H SOMETHING ROTTEN IN HTBAL ROUTINE    )                HTB6118
      GO TO 770                                                         HTB6119
72    M(6) = RCM                                                        HTB6120
      M(16)=M(16)+1.0                                                   HTB6121
```


LOCKHEED

TABLE D-4. (Continued)

```
      M(14)=M(5)                                                        HTB6122
      DO 724 I=KSP,NKSP                                                 HTB6123
      IF (KPSD(I)) 721,725,722                                          HTB6124
  721 NN =-KPSD(I)                                                      HTB6125
      ND = MOD(NN,4096)                                                 HTB6126
      K1 = NN/2**24                                                     HTB6127
      K2 = MOD(NN/2**12,4096)                                           HTB6128
      IF (K1.EQ.0.OR.K2.EQ.0) GO TO 724                                 HTB6129
      RHO = LIN(P(ND),K1)                                               HTB6130
      CP = LIN(P(ND),K2)                                                HTB6131
      J = ND+MAXT                                                       HTB6132
      RHO1 = LIN(P(J),K1)                                               HTB6133
      CP1 = LIN(P(J),K2)                                                HTB6134
      NC = ND+2*MAXT+MAXR                                               HTB6135
      P(NC) = P(NC)*CP/CP1*RHO/RHO1                                     HTB6136
      GO TO 724                                                         HTB6137
  722 ND = MOD(KPSD(I),4096)                                            HTB6138
      NR = MOD(KPSD(I)/4096,4096)                                       HTB6139
      K3 = KPSD(I)/2**24                                                HTB6140
      IF (K3.EQ.0) GO TO 724                                            HTB6141
      J = NR+2*MAXT                                                     HTB6142
      J1 = ND+MAXT                                                      HTB6143
      FACT=LIN(P(ND),K3)                                                HTB6144
      FACT1=LIN(P(J1),K3)                                               HTB6145
      P(J)=P(J)*FACT/FACT1                                              HTB6146
  724 CONTINUE                                                          HTB6147
  725 CONTINUE                                                          HTB6148
      IF (NSAV.NE.0) CALL SAVT3                                         HTB6149
      ITIMO = ITIME                                                     HTB6151
      GO TO (56, 99, 56), NSTS                                          HTB6152
   56 GO TO (108,60,60,108),NCVG                                        HTB6153
  108 GO TO (73,109,111,109),LMMF                                       HTB6154
  109 LMMF=3                                                            HTB6155
      CALL MMF(0.,0.,1,1)                                               HTB6156
      LMMF=4                                                            HTB6157
      GO TO 73                                                          HTB6158
   60 GO TO (115,112,111,115),LMMF                                      HTB6159
  111 WRITE (6,2005) LMMF                                               HTB6160
 2005 FORMAT(7H  LMMF=I5,4X,36H  WRONG VALUE AT THIS POINT IN HTBAL  )  HTB6161
      GO TO 770                                                         HTB6162
  112 LMMF=4                                                            HTB6163
  115 CALL CVG(-1.,1,1)                                                 HTB6164
      M(5) = M(7)*M(6)                                                  HTB6165
      M(15) = M(5)                                                      HTB6166
      GO TO (50, 3, 3, 76), NCVG                                        HTB6167
   73 J = MAXT + 1                                                      HTB6168
      DO 74 I = 1, MAXT                                                 HTB6169
      P(J) = P(I)                                                       HTB6170
      J = J + 1                                                         HTB6171
   74 CONTINUE                                                          HTB6172
      IF (M(8)) 79, 75, 79                                              HTB6173
   75 IF (NPRINT) 78, 77, 77                                            HTB6174
   79 M(8) = 0.                                                         HTB6175
   77 CALL PRINT                                                        HTB6176
      IF (NPRINT) 78, 97, 76                                            HTB6177
   76 M(11) = M(11) + M(4)                                              HTB6178
   78 M(5) = M(7)*M(6)                                                  HTB6179
      IF (M(5).GT.0.) GO TO 80                                          HTB6180
      WRITE(6,2002) M(5)                                                HTB6181
      GO TO 770                                                         HTB6182
 2002 FORMAT (16H1 DELTA THETA =  E15.6,24H MUST BE GREATER THAN 0.)    HTB6183
```

TABLE D-4. (Continued)

```
 80     M(15) = M(5)                                                      HTB6184
        PM 1 = M(1)                                                       HTB6185
        IF (NLFLG.GT.1) CALL TPLOT(-1)                                    HTB6186
        M(1) = M(1) + M(5)                                                HTB6187
        IF (M(11) - M(3)) 83, 82, 82                                      HTB6188
 82     M(11) = M(3)                                                      HTB6189
        LPT = 2                                                           HTB6190
        GO TO 84                                                          HTB6191
 83     LPT = 1                                                           HTB6192
 84     IF (M(1) - M(11)) 85, 90, 87                                      HTB6193
 85     NPRINT = - 1                                                      HTB6194
        GO TO 3                                                           HTB6195
 87     M(5) = M(11) - PM 1                                               HTB6196
        M(1) = M(11)                                                      HTB6197
 90     GO TO (92, 93), LPT                                               HTB6198
 92     NPRINT = 1                                                        HTB6199
        GO TO 3                                                           HTB6200
 93     NPRINT = 0                                                        HTB6201
        GO TO 3                                                           HTB6202
 770    ERROR=2.                                                          HTB6203
        WRITE (6,2004) ERROR                                              HTB6204
 2004   FORMAT (36HOERROR HAS OCCURRED. JOB TERMINATED. 6H LEVELF4.1/     HTB6205
       1 28HOLAST TIME POINT CALCULATED.)                                 HTB6206
        CALL PRINT                                                        HTB6207
        NLAG=2                                                            HTB6208
        RETURN                                                            HTB6209
 97     IF (NLFLG.GT.1) CALL TPLOT(-1)                                    HTB6210
        GO TO (96,111,111,94),LMMF                                        HTB6211
 94     NLAG=3                                                            HTB6212
        GO TO 91                                                          HTB6213
 96     NLAG=2                                                            HTB6214
 91     GO TO (95,98,98),NFAB                                             HTB6215
 95     GO TO (99,50,98),NSTS                                             HTB6216
 98     GO TO 99                                                          HTB6217
 99     RETURN                                                            HTB6218
        END                                                               HTB6219
```

TABLE D-4. (Continued)

```
$IBFTC BIV6                                                         BIV6000
      FUNCTION BIV (XV,XH,K,ERR   )                                 BIV6001
C     AS OF 06/12/65                                                BIV6002
      COMMON P(16000),M(16)                                         BIV6003
      COMMON /CP2/ERROR                                             BIV6004
      DIMENSION KP(1)                                               BIV6005
      EQUIVALENCE (P(1),KP(1))                                      BIV6006
      REAL M                                                        BIV6006
      L=LOC(K)+1                                                    BIV6007
      KFLL=0                                                        BIV6008
      KOD=32765                                                     BIV6009
      IF (KOD.EQ.KP(L)) KOD=KP(L+1)                                 BIV6010
      IF (KOD.NE.32765)  L=L+2                                      BIV6011
      LV = P(L)/1000.+.1                                            BIV6012
      LH = AMOD(P(L),1000.)+.1                                      BIV6013
      IF (P(L+1).GT.P(L+2)) GO TO 20                                BIV6014
      IF (XH.LT.P(L+1)) GO TO 50                                    BIV6015
      DO 15 J=2,LH                                                  BIV6016
      L1=L+J-1                                                      BIV6017
      IF(P(L1).GT.P(L1+1)) GO TO 50                                 BIV6018
      IF (XH.LE.P(L1+1)) GO TO 30                                   BIV6019
   15 CONTINUE                                                      BIV6020
   50 ERROR=2.                                                      BIV6021
      WRITE (6,90) K,XV,XH                                          BIV6022
      RETURN                                                        BIV6023
   20 IF (XH.GT.P(L+1)) GO TO 50                                    BIV6024
      DO 25 J=2,LH                                                  BIV6025
      L1=L+J-1                                                      BIV6026
      IF (P(L1).LT.P(L1+1)) GO TO 50                                BIV6027
      IF (XH.GE.P(L1+1)) GO TO 30                                   BIV6028
   25 CONTINUE                                                      BIV6029
      GO TO 50                                                      BIV6030
   30 L2=L+LH+1                                                     BIV6031
      L3=L2+LH+1                                                    BIV6032
      IF (P(L2).GT. P(L3)) GO TO 40                                 BIV6033
      IF (XV.LT.P(L2)) GO TO 50                                     BIV6034
      DO 35 I=1,LV                                                  BIV6035
      L2= L+I*(LH+1)                                                BIV6036
      L3= L2+LH+1                                                   BIV6037
      IF (P(L2).GT.P(L3)) GO TO 50                                  BIV6038
      IF (XV.LE.P(L3)) GO TO 60                                     BIV6039
   35 CONTINUE                                                      BIV6040
      GO TO 50                                                      BIV6041
   40 IF (XV.GT.P(L2)) GO TO 50                                     BIV6042
      DO 45 I=1,LV                                                  BIV6043
      L2= L+I*(LH+1)                                                BIV6044
      L3= L2+LH+1                                                   BIV6045
      IF (P(L2).LT.P(L3)) GO TO 50                                  BIV6046
      IF (XV.GE.P(L3)) GO TO 60                                     BIV6047
   45 CONTINUE                                                      BIV6048
      GO TO 50                                                      BIV6049
   60 J1=L2+L1-L                                                    BIV6050
      J2=J1+1                                                       BIV6051
      J3=L3+L1-L                                                    BIV6052
      J4=J3+1                                                       BIV6053
      IF (KOD.EQ.32765.OR.KOD.EQ.0) GO TO 80                        BIV6054
   61 KFLG=0                                                        BIV6055
      IF (P(J1).EQ.0.)KFLG=KFLG+1                                   BIV6056
      IF (P(J2).EQ.0.)KFLG=KFLG+2                                   BIV6057
      IF (P(J3).EQ.0.)KFLG=KFLG+4                                   BIV6058
      IF (P(J4).EQ.0.)KFLG=KFLG+8                                   BIV6059
```

TABLE D-4. (Continued)

```
      IF (KFLG.EQ.0) GO TO 70                                           BIV6060
      KFLL=KFLG                                                         BIV6061
      GO TO (62,64,66,62,62,50,62,64,50,64,64,68,62,64,50),KFLG         BIV6062
   62 J1=J1+1                                                           BIV6063
      J2=J2+1                                                           BIV6064
      J3=J3+1                                                           BIV6065
      J4=J4+1                                                           BIV6066
      L1=L1+1                                                           BIV6067
      GO TO 61                                                          BIV6068
   64 J1=J1-1                                                           BIV6069
      J2=J2-1                                                           BIV6070
      J3=J3-1                                                           BIV6071
      J4=J4-1                                                           BIV6072
      L1=L1-1                                                           BIV6073
      GO TO 61                                                          BIV6074
   66 J1=J1+LH+1                                                        BIV6075
      J2=J2+LH+1                                                        BIV6076
      J3=J3+LH+1                                                        BIV6077
      J4=J4+LH+1                                                        BIV6078
      L3=L3+LH+1                                                        BIV6079
      GO TO 61                                                          BIV6080
   68 J1=J1-(LH+1)                                                      BIV6081
      J2=J2-(LH+1)                                                      BIV6082
      J3=J3-(LH+1)                                                      BIV6083
      J4=J4-(LH+1)                                                      BIV6084
      L3=L3-(LH+1)                                                      BIV6085
      GO TO 61                                                          BIV6086
   70 Z1=ABS(P(J1))+(ABS(P(J2))-ABS(P(J1)))/(P(L1+1)-P(L1))             BIV6087
     1  *(XH-P(L1  ))                                                   BIV6088
      Z2=ABS(P(J3))+(ABS(P(J4))-ABS(P(J3)))/(P(L1+1)-P(L1))             BIV6089
     1  *(XH-P(L1  ))                                                   BIV6090
   75 L4=L3-LH-1                                                        BIV6091
      BIV= Z1+(Z2-Z1)/(P(L3)-P(L4))      *(XV-P(L4))                    BIV6092
      IF (KOD.EQ.0.OR.KOD.EQ.32765) GO TO 79                            BIV6093
      IF (TIME.EQ.M(1)) GO TO 76                                        BIV60931
      IF (TIME1.EQ.M(11)) GO TO 79                                      BIV60932
      TIME=M(1)                                                         BIV60933
   76 IF (TIME1.NE.M(11)) WRITE (6,92)                                  BIV60934
      TIME1=M(11)                                                       BIV60935
      WRITE (6,93) K,KOD,XH,XV,BIV                                      BIV60936
      IF (KFLL.NE.0) WRITE (6,91) K,KOD ,XV,XH,BIV                      BIV60937
   79 RETURN                                                            BIV6094
   80 Z1=    (P(J1))+(    (P(J2))-    (P(J1)))/(P(L1+1)-P(L1))          BIV6095
     1  *(XH-P(L1  ))                                                   BIV6096
      Z2=    (P(J3))+(    (P(J4))-    (P(J3)))/(P(L1+1)-P(L1))          BIV6097
     1  *(XH-P(L1  ))                                                   BIV6098
      GO TO 75                                                          BIV6099
   90 FORMAT (24HOOFF BIVARIATE CURVE NO. I5,16H. VERTICAL I.V.=E12.4,  BIV6100
     1 18H, HORIZONTAL I.V.=E12.4,1H.)                                  BIV6101
   91 FORMAT (29HOEXTRAPOLATION USED FOR CURVE I5, 7H, FLAG=I3,         BIV6102
     1 16H. VERTICAL I.V.=E12.4,18H, HORIZONTAL I.V.=E12.4,             BIV6103
     2 11H, DEP.VAR.=E12.4)                                             BIV6104
   92 FORMAT (47HODERIVED VALUES USED IN THE FOLLOWING CURVES...  )     BIV6106
   93 FORMAT (20H BIVARIATE CURVE NO. I6,8H    COD = I6,                BIV6107
     1 11H    X(HOR) = E12.4,12H    X(VERT) = E12.4,12H    Y VALUE =E12.4) BIV6108
      END                                                               BIV6109
```


LOCKHEED
CALIFORNIA COMPANY

TABLE D-4. (Continued)

```
$IBFTC LIN6                                                              LIN6000
      REAL FUNCTION LIN (X, N)                                           LIN6001
C     AS OF 5/25/65                                                      LIN6002
C                                                                        LIN6003
C    Y=LINEAR FUNCTION OF X GIVEN BY CURVE N                             LIN6004
C                                                                        LIN6005
      COMMON P, M                                                        LIN6006
      COMMON /CP2/ ERROR                                                 LIN6007
      DIMENSION P(16000), M(16) ,KP(1)                                   LIN6008
      EQUIVALENCE (P(1),KP(1))                                           LIN6009
      REAL M                                                             LIN6010
      KOD=32765                                                          LIN6011
      PER = 32767.                                                       LIN6012
      LL = LOC(N)                                                        LIN6013
      LL = LL + 1                                                        LIN6014
      NFLAG = 1                                                          LIN6015
      IF (KP(LL  ).NE.KOD) GO TO 1                                       LIN6016
      KOD=KP(LL+1)                                                       LIN6017
      LL=LL+2                                                            LIN6018
 1    IF (P(LL-1).NE.PER) GO TO 3                                        LIN6019
 2    PER = P(LL)                                                        LIN6020
      LL = LL + 1                                                        LIN6021
      NFLAG = 2                                                          LIN6022
 3    LU = LL                                                            LIN6023
      J = LL                                                             LIN6024
 5    LU = LU + 2                                                        LIN6025
      IF (P(LU) - P(J)) 7, 6, 6                                          LIN6026
 6    J = LU                                                             LIN6027
      GO TO 5                                                            LIN6028
 7    LU = LU - 1                                                        LIN6029
      LV=LU+2                                                            LIN6030
 8    IF (X - P(LL)) 9, 10, 10                                           LIN6031
 9    PERX=-PER                                                          LIN6032
      GO TO (40, 18), NFLAG                                              LIN6033
 10   IF (P(LU - 1) - X) 12, 25, 25                                      LIN6034
 12   PERX= PER                                                          LIN6035
      GO TO (40, 18), NFLAG                                              LIN6036
 18   IF (ABS((P(LU-1)-P(LL))/PER-1.).GT.PER*1.E-6) GO TO 40             LIN6037
      DO 20 J = LL, LV, 2                                                LIN6038
      P(J) = P(J) + PERX                                                 LIN6039
 20   CONTINUE                                                           LIN6040
      GO TO 8                                                            LIN6041
 25   LL = LL + 2                                                        LIN6042
      DO 28 J = LL, LU, 2                                                LIN6043
      IF (X - P(J)) 27, 27, 28                                           LIN6044
 27   IF (KOD.NE.32765) GO TO 272                                        LIN6045
 271  LIN=(P(J+1)-P(J-1))/(P(J)-P(J-2))*(X-P(J-2))+P(J-1)                LIN6046
      GO TO 50                                                           LIN6047
 272  LIN=(ABS(P(J+1))-ABS(P(J-1)))/(P(J)-P(J-2))*(X-P(J-2))+ABS(P(J-1))LIN6048
      IF (KOD.EQ.0) GO TO 50                                             LIN6049
 76   FORMAT (47HODERIVED VALUES USED IN THE FOLLOWING CURVES...    )    LIN6050
      IF (TIME.EQ.M(1)) GO TO 80                                         LIN6051
      IF(TIME1.EQ.M(11)) GO TO 50                                        LIN60511
      TIME=M(1)                                                          LIN6052
 80   IF (TIME1.NE.M(11)) WRITE (6,76)                                   LIN60521
      TIME1=M(11)                                                        LIN60522
      WRITE (6,77) N,KOD,X,LIN                                           LIN6053
 77   FORMAT (11H CURVE NO.= I6, 8H   COD = I5,25H   INDEPENDENT VARIABLLIN6054
     1E = E12.4,23H   DEPENDENT VARIABLE = E12.4)                        LIN6055
      GO TO 50                                                           LIN6056
 28   CONTINUE                                                           LIN6057
```

TABLE D-4. (Continued)

```
      LL = LL - 2                                                       LIN6058
 40   WRITE (6, 42) X, N                                                LIN6059
      WRITE (6, 73) (P(I), I = LL, LU)                                  LIN6060
 73   FORMAT(29HOCURVE POINT PAIRS FOLLOW.... /(1X,3(2E18.8,4X)))       LIN6061
 42   FORMAT (27HOINDEP VAR FOR LIN INTERP.= E 20.8, 7H CURVE= I 4 ,    LIN6062
     1 16H      OFF CURVE     )                                         LIN6063
      IF (ERROR.EQ.0.) WRITE (6, 75)                                    LIN6064
 75   FORMAT (30HO  LAST TIME POINT CALCULATED   )                      LIN6065
      M(8)=1.                                                           LIN6066
      IF (ERROR.EQ.0.) ERROR=2.                                         LIN6067
 50   RETURN                                                            LIN6068
      END                                                               LIN6069


$IBFTC ATR6
      FUNCTION ATRP 1 (C, X)
C     AS OF 12/4/64
      DIMENSION C(1)
      I = 5
      IF (C(1) - X) 3, 3, 1
 1    IF (C(I + 2) - C(I)) 16, 17, 17
 17   I = I + 4
      GO TO 1
 16   J = I + 3
      ATRP 1 = C(2)
 18   WRITE (6, 101) X, (C(K), K = 1, J)
 101  FORMAT (13H OFF CURVE X= E 16.8/(1H  2E 16.8))
      RETURN
 3    IF (X - C(I)) 4, 4, 5
 4    TELA = (C(I - 1) - C(I - 3))/(C(I - 2) - C(I - 4))
      TELB = (C(I + 1) - C(I - 1))/(C(I) - C(I - 2))
      ATRP 1 = C(I - 3) + (X - C(I - 4))*TELA + (X - C(I - 4))*(X - C(I
     1  - 2))*(TELB - TELA)/(C(I) - C(I - 4))
      RETURN
 5    IF (C(I + 2) - C(I)) 6, 6, 7
 6    ATRP 1 = C(I + 1)
      J = I + 2
      GO TO 18
 7    I = I + 4
      GO TO 3
      END
```

TABLE D-4. (Continued)

```
$IBFTC PAR6
      FUNCTION PAR (X, N)
C      AS OF 04/27/65
C
C     Y=PARABOLIC FUNCTION OF X GIVEN BY CURVE N
C
      COMMON P, M
      DIMENSION P(16000), M(16), C(8)
      REAL M
      PER = 32767.
      LL = LOC(N)
      LL = LL + 1
      NFLAG = 1
      IF (P(LL - 1) - PER) 3, 2, 3
 2    PER = P(LL)
      LL = LL + 1
      NFLAG = 2
 3    LU = LL
      J = LL
 5    LU = LU + 2
      IF (P(LU) - P(J)) 7, 6, 6
 6    J = LU
      GO TO 5
 7    LU = LU - 1
 8    IF (X - P(LL)) 9, 10, 10
 9    GO TO (40, 14), NFLAG
 10   IF (P(LU - 1) - X) 12, 25, 25
 12   GO TO (40, 18), NFLAG
 14   DO 15 J = LL, LU, 2
      P(J) = P(J) - PER
 15   CONTINUE
      GO TO 8
 18   DO 20 J = LL, LU, 2
      P(J) = P(J) + PER
 20   CONTINUE
      GO TO 8
 25   LL = LL + 4
      DO 28 J = LL, LU, 4
      IF (X - P(J)) 27, 27, 28
 27   LL = J - 4
      GO TO 44
 28   CONTINUE
      LL = LL - 4
 40   WRITE (6, 42) X, N
 42   FORMAT (27H0INDEP VAR FOR PAR INTERP.= E 20.8, 7H CURVE= I 4 ,
     1 16H      OFF CURVE    )
      WRITE (6, 43) (P(I), I = LL, LU)
 43   FORMAT (16H0CURVE POINTS =  4E 18.8)
      CALL PRINT
      CALL EXIT
      STOP
 44   LU = LL + 5
      I = 0
      DO 45 J = LL, LU
      I = I + 1
      C(I) = P(J)
 45   CONTINUE
      C(I + 1) = 0.
      PAR = ATRP 1(C, X)
      RETURN
      END
```


LOCKHEED CALIFORNIA COMPANY

TABLE D-4. (Continued)

```
$IBFTC MMF6                                                             MMF6000
      SUBROUTINE MMF (T1,T2,NN,MM)                                      MMF6001
C     AS OF 04/27/65                                                    MMF6002
      COMMON P,M,MAXT,MAXR,MAXC,KSP,KST,KURV7,NCVG,NSTS,NFAB            MMF6003
      COMMON NABL,NKSP,LMMF,NRMF,NREAD                                  MMF6004
      DIMENSION P(16000),M(16),NABL(10),KNOD(1)                         MMF6005
      EQUIVALENCE (P(1),KNOD(1))                                        MMF6006
      REAL M                                                            MMF6007
  301 FORMAT (7O12)                                                     MMF6008
      GO TO (1,2,3,770),LMMF                                            MMF6009
    1 NRMF=0                                                            MMF6010
      REWIND 10                                                         MMF6011
      REWIND 9                                                          MMF6012
      NP=0                                                              MMF6013
      NREAD=0                                                           MMF6014
      KCY=0                                                             MMF6015
      LMMF=2                                                            MMF6016
    2 NRMF=NRMF+1                                                       MMF6017
      WRITE (10,301) T1,T2,NN,MM                                        MMF6018
      L=LOC(NN)                                                         MMF6019
  100 L=L+1                                                             MMF6020
      NDN=P(L)                                                          MMF6021
      IF (NDN) 770,770,101                                              MMF6022
  101 KNOD(L)=KCY+4096*(NP+4096*NDN)                                    MMF6023
      NDN=KNOD(L)/16777216                                              MMF6024
      NP=MOD(KNOD(L)/4096,4096)                                         MMF6025
      KCY=MOD(KNOD(L),4096)                                             MMF6026
      GO TO 100                                                         MMF6027
    3 REWIND 10                                                         MMF6028
      DO 60 I=1,NRMF                                                    MMF6029
      READ (10,301) TO,TF,NT,MC                                         MMF6030
      IF (M(1)-TF) 4,5,60                                               MMF6031
    4 IF (M(1)-TO) 60,5,5                                               MMF6032
    5 L=LOC(NT)                                                         MMF6033
   11 L=L+1                                                             MMF6034
      NDN=KNOD(L)/16777216                                              MMF6035
      NP=MOD(KNOD(L)/4096,4096)                                         MMF6036
      KCY=MOD(KNOD(L),4096)                                             MMF6037
      IF (NDN) 60,60,21                                                 MMF6038
   21 KCY=KCY+1                                                         MMF6039
      IF (KCY-MC) 50,6,6                                                MMF6040
    6 KCY=0                                                             MMF6041
      NDO=MAXT+NDN                                                      MMF6042
      IF (NP) 22,22,30                                                  MMF6043
   22 IF (P(NDN)-P(NDO)) 25,50,24                                       MMF6044
   24 NP=1                                                              MMF6045
      GO TO 50                                                          MMF6046
   25 NP=2                                                              MMF6047
      GO TO 50                                                          MMF6048
   30 IF (P(NDN)-P(NDO)) 33,50,31                                       MMF6049
   31 GO TO (50,32),NP                                                  MMF6050
   32 NPP=1                                                             MMF6051
      GO TO 40                                                          MMF6052
   33 GO TO (34,50),NP                                                  MMF6053
   34 NPP=2                                                             MMF6054
   40 PM1=M(1)-M(5)                                                     MMF6055
      NREAD=NREAD+1                                                     MMF6056
      WRITE (9,301) TO,TF,NT,NP,NDN,P(NDO),PM1                          MMF6057
      NP=NPP                                                            MMF6058
   50 KNOD(L)=KCY+4096*(NP+4096*NDN)                                    MMF6059
      GO TO 11                                                          MMF6060
```

TABLE D-4. (Continued)

```
 60    CONTINUE                                                     MMF6061
       REWIND 10                                                    MMF6062
 770   RETURN                                                       MMF6063
       END                                                          MMF6064
```

```
$IBFTC RAD6                                                         RAD6000
       FUNCTION RAD (N 1, N 2, CV)                                  RAD6001
C      AS OF 04/27/65                                               RAD6002
C                                                                   RAD6003
C     RADIATION WITH CONSTANT OR VARIABLE FACTOR CV                 RAD6004
C                                                                   RAD6005
       COMMON P, M, MAXT                                            RAD6006
       DIMENSION P(16000), M(16)                                    RAD6007
       REAL M                                                       RAD6008
       SIGK = CV*.1713 E - 8                                        RAD6009
       J = MAXT + N 1                                               RAD6010
       T 1 = P(J) + 459.6                                           RAD6011
       J = MAXT + N 2                                               RAD6012
       T 2 = P(J) + 459.6                                           RAD6013
       RAD = 1./(SIGK*(T 1*T 1 + T 2*T 2)*(T 1 + T 2))              RAD6014
       RETURN                                                       RAD6015
       END                                                          RAD6016
```

```
$IBFTC SAV6                                                         SAV6000
       SUBROUTINE SAV(X,Y)                                          SAV6001
C      AS OF 04/27/65                                               SAV6002
       COMMON P(16000),M(16),MAXT,MAXR,MAXC                         SAV6003
       COMMON /EXFLG/NTFLG,NSAV,NFLG                                SAV6004
       REAL M                                                       SAV6005
       NSAV=0                                                       SAV6006
       IF (X.LT.Y.OR.NFLG.GT.0) GO TO 20                            SAV6007
       NFLG=1                                                       SAV6008
       NSAV=1                                                       SAV6009
 20    RETURN                                                       SAV6010
       END                                                          SAV6011
```

```
$IBFTC EA16                                                         EA16000
       SUBROUTINE EAH1 (NODE,QK)                                    EA16001
C      AS OF 05/07/65                                               EA16002
       COMMON    P,M,MAXT,MAXR,MAXC,KSP,KST                         EA16003
       DIMENSION P(16000),M(16)                                     EA16004
       REAL M                                                       EA16005
       K=2*MAXT+MAXR+MAXC+NODE                                      EA16006
       KOD=1                                                        EA16007
       ALPHO=0.                                                     EA16008
       MF=0                                                         EA16009
       CALL ECHERT (KOD,NODE,ALPHO,MF,HJ,TER)                       EA16010
       P(K)=  HJ*QK*TER                                             EA16011
       RETURN                                                       EA16012
       END                                                          EA16013
```


LOCKHEED
CALIFORNIA COMPANY

TABLE D-4. (Continued)

```
$IBFTC EA26
      SUBROUTINE EAH 2 (NODE, QK, VAR)
C     AS OF 04/27/65
      COMMON P, M, MAXT, MAXR, MAXC, KSP, KST
      DIMENSION P(16000), M(16)
      REAL M
      K = 2*MAXT + MAXR + MAXC + NODE
      KOD = 1
      ALPHO = 0.
      MF = 0
      CALL ECHERT (KOD, NODE, ALPHO, MF, HJ, TER)
      P(K) = HJ*QK*TER
      VAR = HJ
      RETURN
      END


$IBFTC EA36
      SUBROUTINE EAH 3 (NODE, QK, ALPHO, MF)
C     AS OF 04/27/65
      COMMON P, M, MAXT, MAXR, MAXC, KSP, KST
      DIMENSION P(16000), M(16)
      REAL M
      KOD = 2
      CALL ECHERT (KOD, NODE, ALPHO, MF, HJ, TER)
      K = 2*MAXT + MAXR + MAXC + NODE
      P(K) = HJ*QK*TER
      RETURN
      END


$IBFTC EA46
      SUBROUTINE EAH 4 (NODE, QK, VAR, ALPHO, MF)
C     AS OF 04/27/65
      COMMON P, M, MAXT, MAXR, MAXC, KSP, KST
      DIMENSION P(16000), M(16)
      REAL M
      K = 2*MAXT + MAXR + MAXC + NODE
      KOD = 2
      CALL ECHERT (KOD, NODE, ALPHO, MF, HJ, TER)
      P(K) = HJ*QK*TER
      VAR = HJ
      RETURN
      END
```

TABLE D-4. (Continued)

```
$IBFTC EAH6                                                               EAH6000
      SUBROUTINE ECHERT (KOD, NODE, ALPHO, K, HJ, TER)                    EAH6001
C     AS OF 04/27/65                                                      EAH6002
C                                                                         EAH6003
C    AERODYNAMIC HEATING USING ECHERT FORMULA                             EAH6004
C                                                                         EAH6005
      COMMON P, M, MAXT, MAXR, MAXC, KSP, KST, KURV 7                     EAH6006
      DIMENSION P(16000), M(16)                                           EAH6007
      REAL M, LIN                                                         EAH6008
      GO TO (1, 2), KOD                                                   EAH6009
 1    BETHA = 1.                                                          EAH6010
      GO TO 15                                                            EAH6011
 2    ALPHR = LIN(M(1), 2)                                                EAH6012
      IF (K) 3, 3, 4                                                      EAH6013
 3    ALPHA = ALPHO - ALPHR                                               EAH6014
      GO TO 5                                                             EAH6015
 4    ALPHA = ALPHR - ALPHO                                               EAH6016
 5    IF (KURV 7 - 3) 7, 7, 6                                             EAH6017
 6    BETHA = LIN(ALPHA, 7)                                               EAH6018
      GO TO 15                                                            EAH6019
 7    J = LOC(7)                                                          EAH6020
      IF (ALPHA) 8, 8, 10                                                 EAH6021
 8    BETHA = P(J + 2)*ALPHA + 1.                                         EAH6022
      GO TO 15                                                            EAH6023
 10   BETHA = P(J + 1)*ALPHA + 1.                                         EAH6024
 15   QMAX = LIN(M(1), 11)                                                EAH6025
      S = LIN(QMAX, 5)                                                    EAH6026
      H = LIN(M(1), 1)                                                    EAH6027
      ALRDXM = LIN(H, 3)                                                  EAH6028
      TLDTM = LIN(QMAX, 6)                                                EAH6029
      TMAX = LIN(H, 12)                                                   EAH6030
      TL = TLDTM*TMAX                                                     EAH6031
      QMLDMM = LIN(QMAX, 13)                                              EAH6032
      QML = QMLDMM*QMAX                                                   EAH6033
      J = LOC(8)                                                          EAH6034
      A = P(J + 1)                                                        EAH6035
      B = P(J + 2)                                                        EAH6036
      QL = P(J + 3)                                                       EAH6037
      GAMMA = P(J + 4)                                                    EAH6038
      QN = 0.71**QL*((GAMMA - 1.)/2.)*QML**2                              EAH6039
      J = MAXT + NODE                                                     EAH6040
      TLPDTL = 0.5*(1. + (P(J) + 459.6)/TL) + 0.22*QN                     EAH6041
      TLP = TLPDTL*TL                                                     EAH6042
      TR = TL*(1. + QN)                                                   EAH6043
      M(12) = TR                                                          EAH6044
      ALPRK = LIN(TLP, 4)                                                 EAH6045
      PRK = 10.0**ALPRK                                                   EAH6046
      RLDXM = 10.0**ALRDXM                                                EAH6047
      HJ = BETHA*(S*RLDXM*QMAX/TLPDTL**1.69)**A*QMAX**B*PRK               EAH6048
      TER = TR - P(J) - 459.6                                             EAH6049
      RETURN                                                              EAH6050
      END                                                                 EAH6051
```

TABLE D-4. (Continued)

```
$IBFTC CVG6
      SUBROUTINE CVG (A, N, M 2)
C      AS OF 04/26/65
C
C    CONVERGE ROUTINE FINDS EQUILIBRIUM STARTING TEMPERATURES
C
      COMMON P, M, MAXT, MAXR, MAXC, KSP, KST, KURV 7, NCVG, NSTS, NFAB
      COMMON NABL, NKSP
      DIMENSION P(16000), M(16), NABL(10), KPSD(1)
      EQUIVALENCE (P(1), KPSD(1))
      COMMON /EXCVG/ NFLAG,ITMAX,ITER,NODE,PM1,PM,DTO,ATEM,NTEM,MTEM
      COMMON /CP2/ ERROR
      REAL M
 1    GO TO (2, 8, 8, 6), NCVG
 2    ITMAX = 0
      ITER = 0
      NODE = 0
      NFLAG = 1
      PM 1 = M(1)
      PM2=M(6)
      PM3=M(9)
      DTO = 0.
      ATEM = A
      NTEM = N
 7    MTEM = M 2
      IF (MTEM) 4, 3, 5
 3    WRITE (6, 50)
 50   FORMAT (49H0 NO. OF ITERATIONS BETWEEN PRINTS IS ZERO IN CVG )
 51   ERROR = 2.
      RETURN
 4    NCVG = 2
      MTEM =IABS(MTEM)
 61   M(1)=PM1
      M(6)=PM2
      M(9)=PM3
 6    RETURN
 5    NCVG = 3
      GO TO 61
 8    IF (A) 9, 9, 7
 9    DIFM = 0.
 20   ITER = ITER + 1
      ITMAX = ITMAX + 1
      JJ = NKSP - 1
      DO 11 I = KSP, JJ
      IF (KPSD(I)) 10, 11, 11
 10   JN=MOD(KPSD(I),4096)
      JN=IABS(JN)
      J = MAXT + JN
      DIF = ABS(P(JN) - P(J))
      IF (DIF.LT.DTO) GO TO 303
 300  DTMX=0.1*(P(J)+460.)
      IF (DTMX.LT.0.) DTMX=0.
      IF (DIF.LE.DTMX) GO TO 302
      DIF=SIGN(DTMX,(P(JN)-P(J)))
      P(JN)=P(J)+DIF
      GO TO 303
 302  P(JN)=(P(JN)+P(J))/2.
 303  DIF=ABS(DIF)
      IF (DIF.LE.DIFM) GO TO 11
      DIFM=DIF
      NODE=JN
```

TABLE D-4. (Continued)

```
11    CONTINUE                                                        CVG6061
      DTO = DIFM                                                      CVG6062
      J = MAXT + 1                                                    CVG6063
307   DO 308 I = 1, MAXT                                              CVG6064
      P(J) = P(I)                                                     CVG6065
      J = J + 1                                                       CVG6066
308   CONTINUE                                                        CVG6067
      IF (NFLAG.EQ.1) GO TO 55                                        CVG6068
      IF (DIFM.GE.ATEM) GO TO 14                                      CVG6069
13    NCVG = 4                                                        CVG6070
55    ITER = 0                                                        CVG6071
      M(1) = ITMAX                                                    CVG6072
      M(9)=NODE                                                       CVG6073
      M(6)=DTO                                                        CVG6074
56    CALL PRINT                                                      CVG6075
      NFLAG = 2                                                       CVG6076
      GO TO 6                                                         CVG6077
14    IF (ITMAX - NTEM) 15, 16, 16                                    CVG6078
15    IF (ITER - MTEM) 6, 55, 55                                      CVG6079
16    M(1) = NTEM                                                     CVG6080
      CALL PRINT                                                      CVG6081
      WRITE (6,60) ITMAX,DTO,NODE                                     CVG6082
60    FORMAT (33H0TEMPERATURES DID NOT CONVERGE IN I7,12H ITERATIONS. CVG6083
     1  8H MAX DT= E12.4, 8H AT NODE I5)                              CVG6084
      GO TO 51                                                        CVG6085
      END                                                             CVG6086


$IBFTC STS6                                                           STS6000
      SUBROUTINE STS (QK)                                             STS6001
C     AS OF 04/27/65                                                  STS6002
C                                                                     STS6003
C    STEADY STATE FINDS EQUIL. STARTING TEMPS. WITH HEAT BALANCE EQ.  STS6004
C                                                                     STS6005
      COMMON P, M, MAXT, MAXR, MAXC, KSP, KST, KURV 7, NCVG, NSTS     STS6006
      DIMENSION P(16000), M(16)                                       STS6007
      REAL M                                                          STS6008
1     GO TO (2, 10, 10), NSTS                                         STS6009
2     NSTS = 2                                                        STS6010
      NLAG = 1                                                        STS6011
4     CALL HTBAL (NLAG)                                               STS6012
      DIFM = 0.                                                       STS6013
      J = MAXT + 1                                                    STS6014
      DO 6 I = 1, MAXT                                                STS6015
      DIF = ABS(P(I) - P(J))                                          STS6016
      J = J + 1                                                       STS6017
      IF (DIFM - DIF) 5, 6, 6                                         STS6018
5     DIFM = DIF                                                      STS6019
6     CONTINUE                                                        STS6020
      J = MAXT + 1                                                    STS6021
      DO 8 I = 1, MAXT                                                STS6022
      P(J) = P(I)                                                     STS6023
      J = J + 1                                                       STS6024
8     CONTINUE                                                        STS6025
      IF (DIFM - QK) 9, 4, 4                                          STS6026
9     NSTS = 3                                                        STS6027
10    RETURN                                                          STS6028
      END                                                             STS6029
```


LOCKHEED
CALIFORNIA COMPANY

TABLE D-4. (Continued)

```
$IBFTC RRM6                                                                   RRM6000
      FUNCTION RRM (N 1, N 2, N)                                              RRM6001
C     AS OF 06/12/65                                                          RRM6002
      COMMON P, M, MAXT                                                       RRM6003
      COMMON /CP2/ERROR                                                       RRM6003
      DIMENSION P(16000), M(16), A(15, 15)                                    RRM6004
      REAL M                                                                  RRM6005
      L = LOC(N)                                                              RRM6006
      KORD = P(L + 1)                                                         RRM6007
      IF (KORD.GT.15) GO TO 60                                                RRM60071
      L = L + 2                                                               RRM6008
      K = L + KORD*KORD                                                       RRM6009
      NSB 1 = 0                                                               RRM6010
      NSB 2 = 0                                                               RRM6011
      KOUNT = 0                                                               RRM6012
      K 2 = K + 4*KORD - 1                                                    RRM6013
      J = 0                                                                   RRM6014
      DO 18 I = K, K 2, 4                                                     RRM6015
      J = J + 1                                                               RRM6016
      NODE = P(I)                                                             RRM6017
      IF (NODE - N 1) 5, 4, 5                                                 RRM6018
  4   A 1 = P(I + 1)                                                          RRM6019
      E 1 = P(I + 2)                                                          RRM6020
      RHO 1 = P(I + 3)                                                        RRM6021
      NSB 1 = J                                                               RRM6022
      GO TO 17                                                                RRM6023
  5   IF (NODE - N 2) 18, 6, 18                                               RRM6024
  6   A 2 = P(I + 1)                                                          RRM6025
      E 2 = P(I + 2)                                                          RRM6026
      RHO 2 = P(I + 3)                                                        RRM6027
      NSB 2 = J                                                               RRM6028
 17   KOUNT = KOUNT + 1                                                       RRM6029
      IF (KOUNT - 2) 18, 20, 19                                               RRM6030
 18   CONTINUE                                                                RRM6031
 19   WRITE (6, 70) N 1, N 2, N                                               RRM6032
 70   FORMAT (7HO NODE  I 4, 3HOR  I 4, 24HDOES NOT APPEAR IN TABLE I6)       RRM6033
      GO TO 80                                                                RRM6035
 20   A 1 = A 1*A 2                                                           RRM6037
      E 1 = E 1*E 2                                                           RRM6038
      RHO 1 = RHO 1*RHO 2                                                     RRM6039
      LTEM = L                                                                RRM6040
      K = 0                                                                   RRM6041
      K 2 = 0                                                                 RRM6042
      DO 42 I = 1, KORD                                                       RRM6043
      IF (I - NSB 1) 22, 21, 22                                               RRM6044
 21   L = L + KORD                                                            RRM6045
      GO TO 42                                                                RRM6046
 22   K = K + 1                                                               RRM6047
      DO 40 J = 1, KORD                                                       RRM6048
      IF (J - NSB 2) 24, 39, 24                                               RRM6049
 24   K 2 = K 2 + 1                                                           RRM6050
      A(K, K 2) = P(L)                                                        RRM6051
 39   L = L + 1                                                               RRM6052
 40   CONTINUE                                                                RRM6053
      K 2 = 0                                                                 RRM6054
 42   CONTINUE                                                                RRM6055
      NN = KORD - 1                                                           RRM6056
      CALL MDETR (NN, C, A)                                                   RRM6057
      DIJ = C                                                                 RRM6058
      L = LTEM                                                                RRM6059
      DO 50 I = 1, KORD                                                       RRM6060
```

TABLE D-4. (Continued)

```
      DO 50 J = 1, KORD                                                 RRM6061
      A(I, J) = P(L)                                                    RRM6062
      L = L + 1                                                         RRM6063
   50 CONTINUE                                                          RRM6064
      CALL MDETR (KORD, C, A)                                           RRM6065
      QK = (0.4811 E.- 12*E 1*A 1)/(0.1713 E - 8*RHO 1)*ABS(DIJ/C)      RRM6066
      JN1=MAXT+N1                                                       RRM6067
      JN2=MAXT+N2                                                       RRM6068
      T1=P(JN1)+460.                                                    RRM6069
      T2=P(JN2)+460.                                                    RRM6070
      RRM=1./(0.1713E-8*QK*(T1*T1+T2*T2)*(T1+T2))                       RRM6071
      RETURN                                                            RRM6072
   60 WRITE (6,61) N,KORD                                               RRM6073
   61 FORMAT (10HOTABLE NO. I6,17H HAS MATRIX ORDER I6,                 RRM6074
     1 30H, MAXIMUM IS 15. CASE DELETED. )                              RRM60741
   80 ERROR=2.                                                          RRM6075
      RETURN                                                            RRM6076
      END                                                               RRM6077
```

```
$IBFTC MDT6                                                             MDT6000
      SUBROUTINE MDETR (N, C, A)                                        MDT6001
C     AS OF 12/4/64                                                     MDT6002
      DIMENSION A(15, 15), B(16)                                        MDT6003
      LN = N - 1                                                        MDT6004
      DO 18 I = 1, LN                                                   MDT6005
      MI = I + 1                                                        MDT6006
      DO 18 K = MI, N                                                   MDT6007
      IF (A(I, I)) 19, 3, 19                                            MDT6008
    3 K 1 = I + 1                                                       MDT6009
      DO 6 K 2 = K 1, N                                                 MDT6010
      IF (A(I, K 2)) 4, 6, 4                                            MDT6011
    4 DO 5 JJ = 1, N                                                    MDT6012
      B(JJ) = A(JJ, K 2)                                                MDT6013
      A(JJ, K 2) = A(JJ, I)                                             MDT6014
      A(JJ, I) = - B(JJ)                                                MDT6015
    5 CONTINUE                                                          MDT6016
      GO TO 19                                                          MDT6017
    6 CONTINUE                                                          MDT6018
      C = 0.0                                                           MDT6019
      GO TO 100                                                         MDT6020
   19 AA = A(K, I)/A(I, I)                                              MDT6021
      DO 18 J = I, N                                                    MDT6022
   17 A(K, J) = A(K, J) - AA*A(I, J)                                    MDT6023
   18 CONTINUE                                                          MDT6024
      B(1) = 1.0                                                        MDT6025
      DO 32 L = 1, N                                                    MDT6026
      B(L + 1) = B(L)*A(L, L)                                           MDT6027
   32 CONTINUE                                                          MDT6028
      C = B(N + 1)                                                      MDT6029
  100 RETURN                                                            MDT6030
      END                                                               MDT6031
```

TABLE D-4. (Continued)

```
$IBFTC SVT6                                                                     SVT6000
      SUBROUTINE SAVT3                                                          SVT6001
C     AS OF 06/04/65                                                            SVT6002
      COMMON P(16000),M(16),MAXT,MAXR,MAXC,NDUMY(20),DUMMY(12),NR,              SVT6003
     1 MAXP,MAXK,MAXS,MAXM                                                      SVT6004
      COMMON /EXFLG/NTFLG,NSAV,NFLG                                             SVT6005
      DIMENSION LX(9)                                                           SVT6006
      REAL M                                                                    SVT6007
      NSAV=0                                                                    SVT6008
      REWIND 3                                                                  SVT6009
      LX(1)=MAXT                                                                SVT6010
      LX(2)=MAXT                                                                SVT6011
      LX(3)=MAXR                                                                SVT6012
      LX(4)=MAXC                                                                SVT6013
      LX(5)=MAXC                                                                SVT6014
      LX(6)=MAXC                                                                SVT6015
      LX(7)=MAXP                                                                SVT6016
      LX(8)=MAXK                                                                SVT6017
      LX(9)=MAXS                                                                SVT6018
      L2=0                                                                      SVT6019
      DO 10 I=1,9                                                               SVT6020
      L1=L2+1                                                                   SVT6021
      L2=L2+LX(I)                                                               SVT6022
      WRITE (3) LX(I),L1,L2                                                     SVT6023
      WRITE (3) (P(J),J=L1,L2)                                                  SVT6024
   10 CONTINUE                                                                  SVT6025
      L1=1                                                                      SVT6026
      L2=MAXM                                                                   SVT6027
      WRITE (3) MAXM,L1,L2                                                      SVT6028
      WRITE (3) (M(I),I=L1,L2)                                                  SVT6029
      END FILE 3                                                                SVT6030
      REWIND 3                                                                  SVT6031
      RETURN                                                                    SVT6032
      END                                                                       SVT6033
```

TABLE D-4. (Continued)

```
$IBFTC TPT6                                                             TPT6000
      SUBROUTINE TPLOT(NMB)                                             TPT6001
C     AS OF 06/05/65                                                    TPT6002
C     FOR THERMAL ANALYSER ROUTINE                                      TPT6003
C     THIS ROUTINE PUTS ALL POINTS OF CURVE WITH RESPECT TO TIME RANGE, TPT6004
C     AND NMB  ON BINARY TAPE                                           TPT6005
C  ROUTINE WILL ONLY HANDLE 20 CURVES WITH 20 NODES PER CURVE           TPT6006
      COMMON /CPPLT/NPLT,NEND                                           TPT6007
      COMMON /EXSTA/ NC,NCUR1,NCURV                                     TPT6008
      COMMON / EXPLT/ NLFLG                                             TPT6009
      COMMON P,M,MAXT,MAXR,MAXC                                         TPT6010
      DIMENSION P(16000),M(16) ,T(20) ,NCURV(20),NODE(20)               TPT6011
      REAL M                                                            TPT6012
C         NMB = CURVE NUMBER                                            TPT6013
C         K   = NUMBER OF TEMPERATURE CURVES TO BE PLOTTED PER CURVE NMB.TPT6014
C         P(L+1) = INTIAL TIME--THETAI                                  TPT6015
C         P(L+2) = FINAL TIME --THETAF                                  TPT6016
C         P(L+3) = TIME INCREMENT--DELTA THETA                          TPT6017
      IF (NLFLG.LE.1) NLFLG=1                                           TPT6018
      NPLT=2                                                            TPT6019
      GO TO (1 , 2 , 5 )  ,  NLFLG                                      TPT6020
    1 IF (NMB .LE. 0)  GO TO 30                                         TPT6021
      NCUR1 = NMB                                                       TPT6022
      NLFLG = 2                                                         TPT6023
      NC = 1                                                            TPT6024
      IF (NC.LE.20) GO TO 3                                             TPT6025
      WRITE (6,102) NC                                                  TPT6026
  102 FORMAT ( 1H0,I5,26H  IS TOO MANY PLOT CURVES.)                    TPT6027
      GO TO 30                                                          TPT6028
    3 NCURV(NC) = NMB                                                   TPT6029
      GO TO 30                                                          TPT6030
    2 IF (NMB .LE. 0)  GO TO 4                                          TPT6031
      IF (NCUR1 .EQ. NMB)  GO TO 4                                      TPT6032
      NC = NC + 1                                                       TPT6033
      NCURV(NC) = NMB                                                   TPT6034
      GO TO 30                                                          TPT6035
    4 NLFLG = 3                                                         TPT6036
    5 IF (NMB .GT. 0)  GO TO 30                                         TPT6037
      TIME = M(1)                                                       TPT6038
      DO 28 I1 = 1 , NC                                                 TPT6039
      NTAB= NCURV(I1)                                                   TPT6040
      K = 0                                                             TPT6041
      L = LOC(NTAB)                                                     TPT6042
      N = L+4                                                           TPT6043
   13 IF (TIME.LT.P(L+1).OR .TIME.GT.P(L+2)) GO TO 28                   TPT6044
   15 J = P(N)                                                          TPT6045
      IF (J.EQ.0) GO TO 17                                              TPT6046
      N = N+1                                                           TPT6047
      K = K+1                                                           TPT6048
      IF (K.LE.20) GO TO 16                                             TPT6049
      WRITE (6,101) NTAB                                                TPT6050
      P(N-1)=0.                                                         TPT6051
      GO TO 17                                                          TPT6052
  101 FORMAT (15H0PLOT CURVE NO. I5,25H HAS MORE THAN 20 POINTS.)       TPT6053
   16 NODE(K) = J                                                       TPT6054
      NN = J + MAXT                                                     TPT6055
      T(K) = P(NN)                                                      TPT6056
      GO TO 15                                                          TPT6057
   17 IF (K.LT.1) GO TO 28                                              TPT6058
   25 WRITE (11)NTAB,K,(NODE(I),I=1,K),P(L+1),P(L+2),P(L+3),M(1),       TPT6059
     1 (T(I),I=1,K)                                                     TPT6060
```


LOCKHEED
CALIFORNIA COMPANY

TABLE D-4. (Continued)

```
      P(L+1) = P(L+1)+P(L+3)                                            TPT6061
   28 CONTINUE                                                          TPT6062
   30 RETURN                                                            TPT6063
      END                                                               TPT6064


$IBFTC PNT6                                                             PNT6000
      SUBROUTINE PUNCHT(TIME,N1,N2)                                     PNT6001
C     AS OF 04/27/65                                                    PNT6002
      COMMON P(16000),M(16),MAXT,NDUM(22),CC(12)                        PNT6003
      REAL M                                                            PNT6004
      IF (M(1).NE.TIME) RETURN                                          PNT6005
      NA=N1+MAXT                                                        PNT6006
      IF (N1.EQ.0) NA=1+MAXT                                            PNT6007
      NB=N2+MAXT                                                        PNT6008
      IF (N2.EQ.0) NB=MAXT+MAXT                                         PNT6009
      DO 10 I=NA,NB                                                     PNT6010
      IF (I.EQ.NA.OR.I.EQ.NB) GO TO 9                                   PNT6011
      IF (P(I).EQ.P(I-1)) GO TO 10                                      PNT6012
 9    IX=I-MAXT                                                         PNT6013
      ISEQ=IX+10000                                                     PNT6014
      PUNCH 11,IX,P(I),CC(1),CC(2),TIME,ISEQ                            PNT6015
 10   CONTINUE                                                          PNT6016
      RETURN                                                            PNT6017
 11   FORMAT (5HDEC01,I5,F10.2, 7X,A6,A3,5H - ATF7.0,8H SECONDS,        PNT6018
     1  11X,I5,8X)                                                      PNT6019
      END                                                               PNT6020


$IBFTC TNK6                                                             TNK6001
      SUBROUTINE TANKA(N)                                               TNK6002
      WRITE (6,10)                                                      TNK6003
      RETURN                                                            TNK6004
 10   FORMAT (42HOCALL TO TANKA IS ILLEGAL IN THIS VERSION.)            TNK6005
      END                                                               TNK6006
```

TABLE D-4. (Continued)

```
$IBFTC BTP6                                                             BTP6000
      SUBROUTINE BITAP                                                  BTP6001
C     AS OF 05/07/65                                                    BTP6002
C     ROUTINE FOR CAL-COMP PLOTTER                                      BTP6003
C     FOR THERMAL ANALYSER                                              BTP6004
      COMMON /CPPLT/NPLT,NEND,KSKIP                                     BTP6005
      COMMON P,M,NX(23),CC(12)                                          BTP6006
      DIMENSION P(16000) , M(16) , XP(1) , YP(1) , TEMP(20) , KURV(20) ,BTP6007
     1 XS(2) , YS(2) , TITLE(3) , XTITLE(3) , YTITLE(4) , BCD(10) ,     BTP6008
     2 TNODE(20) , KNOD(20) , TITLEN(2) , BUFFER(914)                   BTP6009
      EQUIVALENCE (P(10000),XP(1)),(P(11000),YP(1))                     BTP6010
      REAL M                                                            BTP6011
      KSKIP = 1                                                         BTP6012
      NOK=5                                                             BTP6013
      HT = .14                                                          BTP6014
      ERR = 0.0                                                         BTP6015
      NRECD = 0                                                         BTP6016
      NTRECD = 0                                                        BTP6017
      IDONE = 8888                                                      BTP6018
      WRITE (11) IDONE,KSKIP, (P(I), I = 1 , 6)                         BTP6019
C   NTAB = TABLE (CURVE) NUMBER                                         BTP6020
C   NC = NUMBER OF DIFFERENT CURVES WITH POINTS                         BTP6021
C     NT  = NUMBER OF TEMPERATURE POINTS PER CURVE NUMBER               BTP6022
C   TIMI = INITIAL TIME                                                 BTP6023
C   TIMF = FINAL TIME                                                   BTP6024
C   DTIM = DELTA TIME                                                   BTP6025
C   TIM = CURRENT TIME                                                  BTP6026
C   TEMP(I)...I=1,NT.. = TEMPERATURES                                   BTP6027
      DATA (TITLE(I),I=1,3) /6HCURVE ,6HNUMBER,6H=      /               BTP6028
      DATA (XTITLE(I),I=1,3)/6HTIME I,6HN SEC.,6H       /               BTP6029
      DATA (YTITLE(I),I=1,4)/6HTEMPER,6HATURE ,6HIN DEG,6HREES F/       BTP6030
      DATA (TITLEN(I),I=1,2) /6H  = NC,  6HDE    /                      BTP6031
      DATA (BCD(I),I=1,10)  /0150000000000,0160000000000,0170000000000, BTP6032
     10320000000000,0350000000000,0360000000000,0370000000000,0560000000BTP6033
     2000,0120000000000,0330000000000/                                  BTP6034
      H = .42                                                           BTP6035
C                                                                       BTP6036
      PNO = 0.0                                                         BTP6037
      REWIND 11                                                         BTP6038
      IEND = 7777                                                       BTP6039
      YRANGE = 8.0                                                      BTP6040
      XRANGE = 10.0                                                     BTP6041
  1   NTRECD = NTRECD + NRECD + 1                                       BTP6042
      IF (IEND .EQ. IDONE)  GO TO 93                                    BTP6043
      NRECD = 1                                                         BTP6044
      M2 = 200                                                          BTP6045
      NC = 1                                                            BTP6046
      KC = 0                                                            BTP6047
      DO 2 I = 1 , NTRECD                                               BTP6048
  2   READ (11)  NTAB                                                   BTP6049
      KURV(NC) = NTAB                                                   BTP6050
      IF (NTAB .EQ. IEND .OR. NTAB .EQ. IDONE)  GO TO 93                BTP6051
  6   READ (11)  NTAB                                                   BTP6052
      IF (NTAB .EQ. IDONE)  IEND = 8888                                 BTP6053
      IF (NTAB     .EQ. IEND)  GO TO 11                                 BTP6054
      NRECD = NRECD + 1                                                 BTP6055
      DO 8 I = 1 , NC                                                   BTP6056
      IF (KURV(I) .EQ. NTAB)  GO TO 6                                   BTP6057
  8   CONTINUE                                                          BTP6058
      NC = NC + 1                                                       BTP6059
      KURV(NC) = NTAB                                                   BTP6060
```

TABLE D-4. (Continued)

```
      GO TO 6
 11   REWIND 11
      KC = KC + 1
      IF (KC .GT. NC)  GO TO 1
      YS(2) = 0.
      YS(1) = 1.E16
      N = 0
      NFLAG = 1
      K = 0
      DO 3 I = 1 , NTRECD
 3    READ (11)  NTAB
      BACKSPACE 11
      DO 16 KK = 1 , NRECD
      READ (11) NTAB,NT,(KNOD(I),I=1,NT),TIMI,TIMF,DTIM,TIM,
     1 (TEMP(I),I=1,NT)
      IF (KURV(KC) .NE. NTAB)  GO TO 16
      GO TO (12,13) , NFLAG
 12   XS(2) = TIMF
      XS(1) = TIMI
      DO 15 I=1,NT
 15   TNODE(I) = KNOD(I)
      NFLAG = 2
      NODES = NT
 13   K = K + 1
      P(K)  = TIM
      N = N + M2 + K
      DO 14 I = 1 , NT
      P(N) = TEMP(I)
      IF (TEMP(I) .GT. YS(2))  YS(2) = TEMP(I)
      IF (TEMP(I) .LT. YS(1))  YS(1) = TEMP(I)
      N = N + M2
 14   CONTINUE
      N = 0
 16   CONTINUE
      IF (YS(1).NE.YS(2)) GO TO 17
      YS(1)=YS(1)-1.
      YS(2)=YS(2)+1.
 17   YS(1)=YS(1)-(YS(2)-YS(1))/10.
      YS(2)=YS(2)+(YS(2)-YS(1))/10.
      REWIND 11
      NPTS = K
      MM = 0
      N = 0
      IF (KSKIP .GT. 1)  GO TO 20
C  DETERMINE SCALE AND DRAW AXIS
      CALL PLOTS (BUFFER,914)
      KSKIP = 2
 20   CALLASCALE (XS,2,XRANGE,XMIN,DX,1,ERX)
      CALLASCALE (YS,2,YRANGE,YMIN,DY,1,ERY)
      L=2.3  -(ERX-XMIN)/10.
      IF (L.LT.0) L=0
 24   CALL XYAXIS(1.,1.,XRANGE,1.,XMIN,DX,L,5.,.3,.21,0.,XTITLE,18,0.)
      L=2.3  -(ERY-YMIN)/10.
      IF (L.LT.0) L=0
      CALL XYAXIS(1.,1.,YRANGE,1.,YMIN,DY,L,.1,3.,.21,90.,YTITLE,18,90.)
      YN = 6.0
      PNO = PNO + 1.
      KX = 0
 25   KX = KX + 1
      N = N + M2
      B = BCD(KX)
```


LOCKHEED

TABLE D-4. (Continued)

```
      MM = MM + 1                                                       BTP6122
      DO 30 I = 1 , NPTS                                                BTP6123
      K = N + I                                                         BTP6124
      X = ( P(I) - XMIN) /DX + 1.0                                      BTP6125
      Y = (P(K) - YMIN) /DY +1.0                                        BTP6126
      XP(I) = X                                                         BTP6127
      YP(I) = Y                                                         BTP6128
      CALL SYMBL4 (X,Y,H,B,0.0,1)                                       BTP6129
   30 CONTINUE                                                          BTP6130
      CALL LINE (XP,YP,NPTS,1)                                          BTP6131
      X = XRANGE + 1.5                                                  BTP6132
      YN = YN - .5                                                      BTP6133
      CALL SYMBL4 (X,YN,H,B,0.0,1)                                      BTP6134
      CALL SYMBL4 (X,YN,HT,TITLEN,0.0,12)                               BTP6135
      X = X + 1.2                                                       BTP6136
      Y = TNODE(MM)                                                     BTP6137
      CALL NUMBER (X,YN,HT,Y,0.0,-1)                                    BTP6138
      IF (NODES .GT. MM .AND. KX .LT. NOK)GO TO 25                      BTP6139
      Y = 9.5                                                           BTP6140
      X = XRANGE/2. - 3.                                                BTP6141
      Z = X + 1.8                                                       BTP6142
      CALL SYMBL4 (X,Y,HT,TITLE,0.0,18)                                 BTP6143
      CALL NUMBER (Z,Y,HT,  PNO ,0.0,-1)                                BTP6144
      CALL SYMBL4(X,Y-0.3,HT, CC  ,0.,61)                               BTP6145
      XSPACE = XRANGE + 7.0                                             BTP6146
      CALL PLOT (XSPACE,0.0,-3)                                         BTP6147
      IF (NODES .EQ. MM)  GO TO 11                                      BTP6148
      GO TO 24                                                          BTP6149
   93 IF (KSKIP .LE. 1)  GO TO 95                                       BTP6150
      IF (NEND.EQ.0) GO TO 95                                           BTP6151
   94 PRINT 222,PNO                                                     BTP6152
  222 FORMAT(37H0 REMOVE PLOT TAPE AND MARK AS HAVING F5.0 ,8H   PLOTS )BTP6153
      CALL TRWEND                                                       BTP6154
      PAUSE 55555                                                       BTP6155
      WRITE(6,222) PNO                                                  BTP6156
   95 REWIND 11                                                         BTP6157
      RETURN                                                            BTP6158
      END                                                               BTP6159
```

TABLE D-4. (Continued)

```
$IBFTC SCL6
      SUBROUTINE ASCALE (X, N, S, YMIN, DY, K, YMAX)
C
C     X - THE GIVEN ARRAY OF VALUES TO BE SCALED, AND THE OUTPUT
C         SCALED VALUES
C     N - NO. OF X VALUES
C     S - NO. OF INCREMENTS
C     YMIN - GENERATED MINIMUM X VALUE, ROUNDED DOWN
C     DY    - GENERATED INCREMENT
C     K     - SPACING BETWEEN X VALUE STORAGES
C     YMAX = GENERATED MAXIMUN X VALUE, NOT ROUNDED
C
      DIMENSION X(2)
      YMAX = X(1)
      YMIN = X(1)
      NP = N*K
      DO 6 I = K, NP, K
      IF (YMAX .LT. X(I)) YMAX = X(I)
      IF (YMIN .GT. X(I)) YMIN = X(I)
 6    CONTINUE
      IF (YMIN .NE. YMAX) GO TO 20
      DX = 1.0
      YMIN = YMIN - (S/2.0)*DX
      GO TO 36
 20   DX = (YMAX - YMIN)/S
      NA = 0
      IF (DX - 1.) 25, 36, 30
 25   DX = DX*10.
      NA = NA - 1
      IF (DX .LT. 1. .OR. DX .GE. 10.) GO TO 25
      GO TO 35
 30   IF (DX .GE. 1. .AND. DX .LT. 10.) GO TO 35
      DX = DX/10.
      NA = NA + 1
      GO TO 30
 35   IF (DX .GT. 4.) DY = 10.**(NA + 1)
      IF (DX .LE. 4.) DY = 4.*10.**NA
      IF (DX .LE. 2.) DY = 2.*10.**NA
 36   RNDFTR = -0.9
      IYMIN = YMIN/DY + RNDFTR
      YMIN = FLOAT(IYMIN)*DY
      DO 40 I = 1, NP, K
      X(I) = (X(I)-YMIN)/DY
 40   CONTINUE
      RETURN
      END
```

TABLE D-4. (Continued)

```
$IBFTC AXS6
      SUBROUTINE XYAXIS(XORG,YORG,ALNGTH,SCDT,AMIN,DA,N,XLAB,
     1  YLAB,HLAB,ALAB,HEAD,NC,AXANG)
C
C     XORG - X COORDINATE OF FIRST POINT ON AXIS
C     YORG - Y COORDINATE OF FIRST POINT ON AXIS
C     ALNGTH-LENGTH OF AXIS IN INCHES - TOTAL
C     SCDT  -LENGTH IN INCHES BETWEEN TICK MARKS
C     AMIN  -LABEL VALUE OF FIRST POINT ON AXIS
C     DA    -SCALE VALUE PER INCH OF LABEL VALUES
C     N     -NUMBER OF DECIMAL PLACES FOR AXIS LABEL VALUES
C     XLAB  -X COORDINATE OF AXIS LABEL
C     YLAB  -Y COORDINATE OF AXIS LABEL
C     HLAB  -HEIGHTH OF AXIS LABEL
C     ALAB  -ANGLE OF AXIS LABEL
C     HEAD  -AXIS LABEL
C     NC    -NUMBER OF CHARACTERS IN AXIS LABEL
C     AXANG -ANGLE OF AXIS
C
      DIMENSION HEAD(3)
      DATA DATK, HNUM/0.07, 0.14/
      DATA  PLUG/0.5E-07/
      XCON = 1.0
      YCON = HNUM/2.0
      IF ( AXANG .NE. 0.0 )  GO TO 50
      XCON = 2.0
      YCON = 0.25
   50 RLAB = AXANG * 0.01745329
      DELNUM = DA*SCDT
      SRSCDT = SIN(RLAB)*SCDT
      CRSCDT = COS(RLAB)*SCDT
      CR = COS(RLAB)
      SR = SIN(RLAB)
      WNUM = HNUM*6.0/7.0
      ICNT = 0
      ANUM = AMIN
  100 NA = N + 3
      IF ( ANUM .EQ.  0.0)  ANUM = 0.0
      IF (ANUM .LT. 0.0) NA = NA + 1
      I = 0
  105 I = I + 1
      IF (ABS(ANUM) .LT. 10.0**I) GO TO 115
      NA = NA + 1
      GO TO 105
  115 ADIST = FLOAT(ICNT)*SCDT
      IF (ADIST .GT. ALNGTH) GO TO 200
      XC = XORG + ADIST*CR - FLOAT(NA)*WNUM/XCON  + 0.05
      YC = YORG + ADIST*SR - YCON
      IF (YC .GT. YLAB .AND. YLAB .GT. (YC - SRSCDT) .OR.  XC .GT. XLAB
     1  .AND. XLAB .GT. (XC - CRSCDT)) CALL SYMBL 4 (XLAB, YLAB, HLAB,
     2  HEAD, ALAB, NC)
      XNUM = ANUM + PLUG
      CALL NUMBER (XC,YC,HNUM,XNUM,0.0,N)
      ICNT = ICNT + 1
      ANUM = ANUM + DELNUM
      GO TO 100
  200 IF (YLAB .GE.(YORG+ALNGTH)) CALL SYMBL4(XLAB,YLAB,HLAB,HEAD ,
     1  ALAB,NC)
      I = 3
  205 XC = XORG + FLOAT(ICNT - 1)*CRSCDT
      YC = YORG + FLOAT(ICNT - 1)*SRSCDT
```

TABLE D-4. (Continued)

```
CALL PLOT (XC, YC, I)
I = I - 1
IF (I .LT. 1) I = 1
XC = XC - SR*DATK
YC = YC - CR*DATK
CALL PLOT (XC, YC, I)
XC = XC + SR*DATK
YC = YC + CR*DATK
CALL PLOT (XC, YC, I)
ICNT = ICNT - 1
IF (ICNT .LE. 0) RETURN
GO TO 205
END
```

TABLE D-4. (Continued)

```
$IBFTC MF06                                                              MF06000
      SUBROUTINE MFOUT                                                   MF06001
C     AS OF 04/27/65                                                     MF06002
      COMMON P,M,MAXT,MAXR,MAXC,KSP,KST,KURV7,NCVG,NSTS,NFAB             MF06003
      COMMON NABL,NKSP,LMMF,NRMF,NREAD                                   MF06004
      DIMENSION P(16000),M(16),NABL(10),TEML(1),TIML(1),TEMS(1),TIMS(1)  MF06005
      DIMENSION NLNO(1500),KNOD(1)                                       MF06006
      EQUIVALENCE (P(1),KNOD(1),TEML(1)),(P(3000),TIML(1)),(P(6000),     MF06007
     1 TEMS(1)),(P(9000),TIMS(1))                                        MF06008
      REAL M                                                             MF06009
  301 FORMAT (7O12)                                                      MF06010
      REWIND 10                                                          MF06011
      REWIND 9                                                           MF06012
      IF (NREAD) 1,1,2                                                   MF06013
    1 WRITE (6,204)                                                      MF06014
      GO TO 770                                                          MF06015
    2 KK=0                                                               MF06016
      DO 102 I=1,NRMF                                                    MF06017
      READ (10,301) TO,TF,NT,MC                                          MF06018
      L=LOC (NT)                                                         MF06019
   99 L=L+1                                                              MF06020
      KK=KK+1                                                            MF06021
      NDN=KNOD(L)/16777216                                               MF06022
      IF (NDN) 101,101,100                                               MF06023
  100 NLNO(KK)=NDN                                                       MF06024
      GO TO 99                                                           MF06025
  101 NLNO(KK)=0                                                         MF06026
  102 CONTINUE                                                           MF06027
      REWIND 10                                                          MF06028
      L=0                                                                MF06029
      DO 40 I=1,NRMF                                                     MF06030
      READ (10,301) TO,TF,NT,MC                                          MF06031
      WRITE (6,200) TO,TF                                                MF06032
      WRITE (6,205)                                                      MF06033
  205 FORMAT (1H0,5X,8HNODE NO.,6X,5HMAX T,13X,5HMIN T,11X,              MF06034
     1 11HTIME OF MAX,7X,11HTIME OF MIN  )                               MF06035
    3 L=L+1                                                              MF06036
      JJ=1                                                               MF06037
      KK=1                                                               MF06038
      ND1=NLNO(L)                                                        MF06039
      IF (ND1) 40,40,4                                                   MF06040
    4 DO 30 J=1,NREAD                                                    MF06041
      READ (9,301) T1,T2,NT2,MMT,ND2,TEM,TIME                            MF06042
      IF (NT-NT2) 30,10,30                                               MF06043
   10 IF (ND1-ND2) 30,11,30                                              MF06044
   11 GO TO (13,16),MMT                                                  MF06045
   13 TEML(JJ)=TEM                                                       MF06046
      TIML(JJ)=TIME                                                      MF06047
      JJ=JJ+1                                                            MF06048
      GO TO 30                                                           MF06049
   16 TEMS(KK)=TEM                                                       MF06050
      TIMS(KK)=TIME                                                      MF06051
      KK=KK+1                                                            MF06052
   30 CONTINUE                                                           MF06053
      TEML(JJ)=0.                                                        MF06054
      TIML(JJ)=0.                                                        MF06055
      TEMS(KK)=0.                                                        MF06056
      TIMS(KK)=0.                                                        MF06057
      IF (JJ-KK) 31,32,32                                                MF06058
   31 JJ=KK-1                                                            MF06059
      GO TO 34                                                           MF06060
```

TABLE D-4. (Continued)

```
32    JJ=JJ-1                                                       MF06061
34    WRITE (6,202) ND1,(TEML(K),TEMS(K),TIML(K),TIMS(K),K=1,JJ)    MF06062
      REWIND 9                                                      MF06063
      GO TO 3                                                       MF06064
40    CONTINUE                                                      MF06065
200   FORMAT (28H1    RANGE OF MINIMUM-MAXIMUM //4X,2E18.8   )      MF06066
202   FORMAT (4H0     I10,4E18.8/(14X,4E18.8))                      MF06067
204   FORMAT (43H      NO MAXIMUMS AND MINIMUMS WERE RECORDED )     MF06068
      REWIND 10                                                     MF06069
770   RETURN                                                        MF06070
      END                                                           MF06071
```

TABLE D-4. (Continued)

```
$IBFTC XLN6                                                              XLN6000
          FUNCTION XLIN(X1,X2,N)                                        XLN6001
C      AS OF 6/17/65                                                    XLN6002
C                                                                       XLN6003
C      XLIN = AREA UNDER CURVE N FROM X1 TO X2                          XLN6004
C                                                                       XLN6005
       COMMON P, M                                                      XLN6006
       COMMON /CP2/ ERROR                                               XLN6007
       DIMENSION P(16000), M(16) ,KP(1)                                 XLN6008
       EQUIVALENCE (P(1),KP(1))                                         XLN6009
       REAL M                                                           XLN6010
       KOD=32765                                                        XLN6011
       PER = 32767.                                                     XLN6012
       SUM=0.                                                           XLN60121
       NJC=1                                                            XLN60121
       X=X1                                                             XLN60121
       LL = LOC(N)                                                      XLN6013
       LL = LL + 1                                                      XLN6014
       NFLAG = 1                                                        XLN6015
       IF (KP(LL  ).NE.KOD) GO TO 1                                     XLN6016
       KOD=KP(LL+1)                                                     XLN6017
       LL=LL+2                                                          XLN6018
  1    IF (P(LL-1).NE.PER) GO TO 3                                      XLN6019
  2    PER = P(LL)                                                      XLN6020
       LL = LL + 1                                                      XLN6021
       NFLAG = 2                                                        XLN6022
  3    LU = LL                                                          XLN6023
       J = LL                                                           XLN6024
  5    LU = LU + 2                                                      XLN6025
       IF (P(LU) - P(J)) 7, 6, 6                                        XLN6026
  6    J = LU                                                           XLN6027
       GO TO 5                                                          XLN6028
  7    LU = LU - 1                                                      XLN6029
       LV=LU+2                                                          XLN6030
  8    IF (X - P(LL)) 9, 10, 10                                         XLN6031
  9    PERX=-PER                                                        XLN6032
       GO TO (40, 18), NFLAG                                            XLN6033
 10    IF (P(LU - 1) - X) 12, 25, 25                                    XLN6034
 12    PERX= PER                                                        XLN6035
       GO TO (40, 18), NFLAG                                            XLN6036
 18    IF (ABS((P(LU-1)-P(LL))/PER-1.).GT.PER*1.E-6) GO TO 40           XLN6037
       DO 20 J = LL, LV, 2                                              XLN6038
       P(J) = P(J) + PERX                                               XLN6039
 20    CONTINUE                                                         XLN6040
       GO TO 8                                                          XLN6041
 25    LL = LL + 2                                                      XLN6042
 26    DO 30 J = LL, LU, 2                                              XLN6043
       IF (X - P(J)) 27, 27, 28                                         XLN6044
 27    IF (KOD.NE.32765) GO TO 272                                      XLN6045
 271   XLN=(P(J+1)-P(J-1))/(P(J)-P(J-2))*(X-P(J-2))+P(J-1)              XLN6046
       GO TO 50                                                         XLN6047
 272   XLN=(ABS(P(J+1))-ABS(P(J-1)))/(P(J)-P(J-2))*(X-P(J-2))+ABS(P(J-1))XLN6048
       IF (KOD.EQ.0) GO TO 50                                           XLN6049
 76    FORMAT (47HODERIVED VALUES USED IN THE FOLLOWING CURVES...    )  XLN6050
       IF (TIME.EQ.M(1)) GO TO 80                                       XLN6051
       IF(TIME1.EQ.M(11)) GO TO 50                                      XLN60511
       TIME=M(1)                                                        XLN6052
 80    IF (TIME1.NE.M(11)) WRITE (6,76)                                 XLN60521
       TIME1=M(11)                                                      XLN60522
       WRITE (6,77) N,KOD,X,LIN                                         XLN6053
 77    FORMAT (11H CURVE NO.= I6, 8H   COD = I5,25H   INDEPENDENT VARIABLXLN6054
```

TABLE D-4. (Continued)

```
     1E = E12.4,23H   DEPENDENT VARIABLE = E12.4)                      XLN6055
      GO TO 50                                                         XLN6056
 28   SUM=SUM+(P(J)-P(J-2))*(P(J-1)+P(J+1))/2.                         XLN60561
 30   CONTINUE                                                         XLN6057
      LL = LL - 2                                                      XLN6058
 40   WRITE (6, 42) X, N                                               XLN6059
      WRITE (6, 73) (P(I), I = LL, LU)                                 XLN6060
 73   FORMAT(29H0CURVE POINT PAIRS FOLLOW.... /(1X,3(2E18.8,4X)))      XLN6061
 42   FORMAT (27H0INDEP VAR FOR LIN INTERP.= E 20.8, 7H CURVE= I 4 ,   XLN6062
     1 16H        OFF CURVE    )                                       XLN6063
      IF (ERROR.EQ.0.) WRITE (6, 75)                                   XLN6064
 75   FORMAT (30H0  LAST TIME POINT CALCULATED   )                     XLN6065
      M(8)=1.                                                          XLN6066
      IF (ERROR.EQ.0.) ERROR=2.                                        XLN6067
      RETURN                                                           XLN6068
 50   IF (NJC.EQ.2) GO TO 85                                           XLN6069
      NJC=2                                                            XLN6070
      SUM1=SUM+(X-P(J-2))*(P(J-1)+XLN)/2.                              XLN6071
      X=X2                                                             XLN6072
      SUM=0.                                                           XLN6073
      GO TO 26                                                         XLN6074
 85   SUM2=SUM+(X-P(J-2))*(P(J-1)+XLN)/2.                              XLN6075
      XLIN=(SUM2-SUM1)/(X2-X1)                                         XLN6076
      RETURN                                                           XLN6077
      END                                                              XLN6078
```

APPENDIX E

DATA DIAGNOSTIC PROGRAM

A special program has been written, based extensively on the compile phase of the Thermal Analyzer Program, which scans input data and notes any errors that it finds. The purpose of this program is to allow a short diagnostic run for a large data deck before submitting it to execute.

PROGRAM OPERATION

The program actually compiles the data, printing notes on any errors it finds as it goes along. A much more rigorous examination is made in this routine than in the regular Thermal Analyzer compile phase.

In the Thermal Analyzer several errors can cause the program to terminate before completing its error check. The coding covering most of these errors has been changed in this program to allow the error check to continue. In particular, optional error exits have been selected so that bad data that would cause termination of the run under normal circumstances will simply be noted and the program will continue execution.

To simplify the program set-up on the computer, the special tapes used for the Thermal Analyzer have been eliminated from this job. All the program decks are contained in the card deck submitted to the computer. Therefore the program library tape used with the Thermal Analyzer is not used with this program. Any curves which are requested from the material properties library tape will merely be noted. The tape is not searched, and therefore is not needed. However, the program cannot make sure that the requested curve actually is on the tape; if it is not, a diagnostic will result later from the Thermal Analyzer run.

Since the problem is not executed, those diagnostics peculiar to the execution phase of the Thermal Analyzer do not appear in this program. For example, a condition producing a time step of zero can be detected only in the execution phase. An independent variable that goes beyond the range of the independent variable specified in a curve for interpolation can only be detected in the execution phase. Such errors will not be noted in the diagnostic program.

Since the Thermal Analyzer has versions with different storage allocations, some of the errors do not apply to all versions. For example, if the data exceeds 14000 storages but not 16000, it cannot be run on version C, but may be run on versions A or B. Where such restrictions apply, the diagnostic will so state.

PROGRAM SET-UP

Figure E-1 shows the deck set-up for the diagnostic program, and Table E-1 is a list of the diagnostic comments that could occur.

The input data is set-up just as for the Thermal Analyzer:

1. Temperature block cards
2. NBK card
3. Resistor block cards
4. NBK card
5. Capacitor block cards
6. NBK card
7. Data Sub-block cards
8. NBK card
9. Time block cards
10. NBK card
11. Restart } Repeat as often as needed.
12. NBK card }
13. END

When the data deck is ready, it is inserted into the diagnostic program deck following the $DATA card; it is input data for this program, too. Following the last data card, which is the "END" card, are the required program termination cards to return control to IB monitor system, $EOF and $IBSYS.

Table E-2 is a complete listing of the eleven data diagnostic program subroutines.

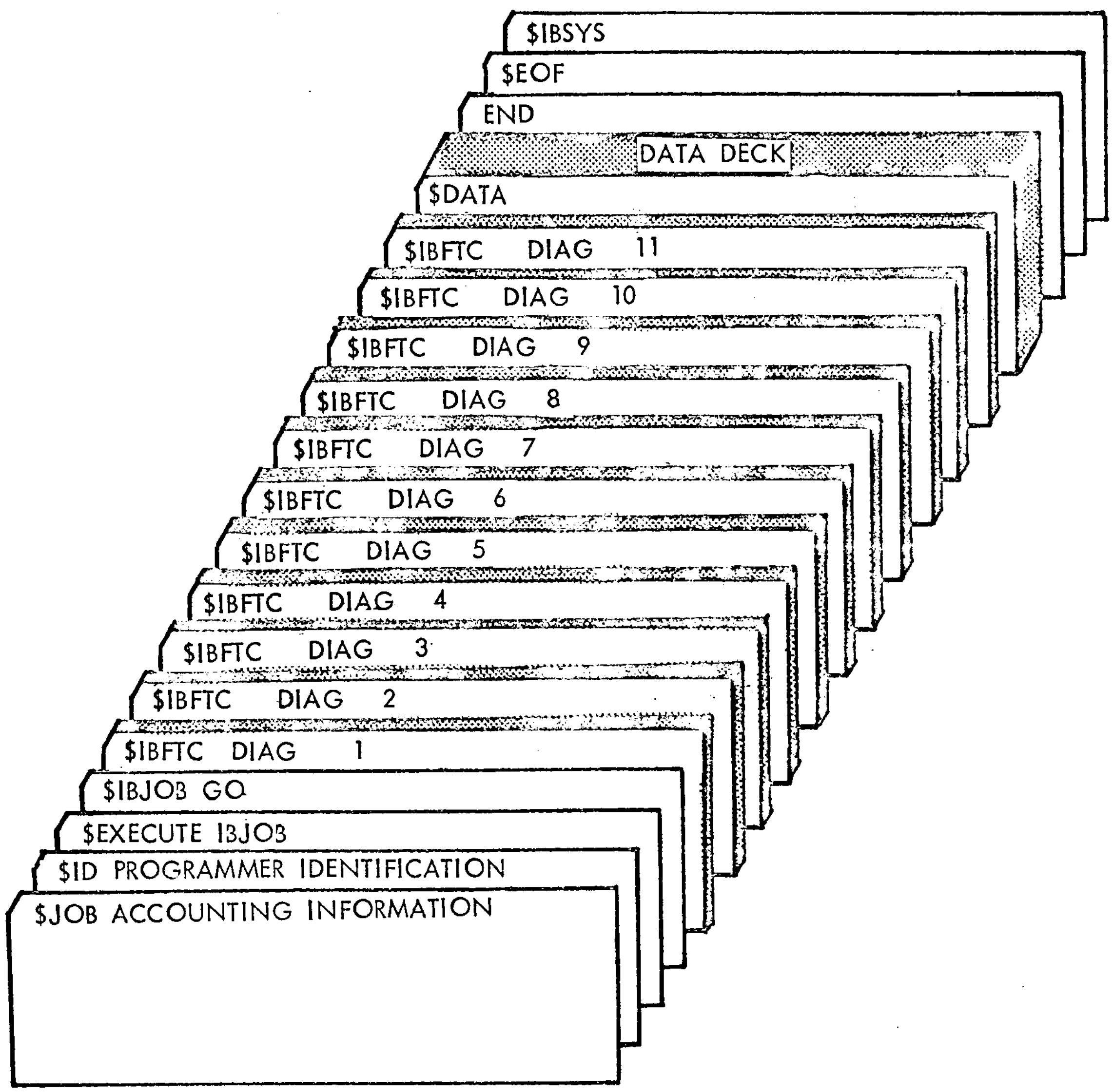


Figure E-1 Source Program Deck Setup For Data Diagnostic Program

TABLE E-1

DIAGNOSTICS

DECK NAME: DIAG

SUBROUTINE: COMPIL

```
215   FORMAT ( 66H0 COUNT IN COLS. 4 AND 5 IS NOT AN INTEGER LESS THAN OCOMP025
     1R EQUAL TO 6 )                                                   COMP0251
223   FORMAT (7H0CODE= A6,12H IS ILLEGAL.)                             COMP021
214   FORMAT (56H0 BLOCK COUNT IN COMPIL ROUTINE IS AN UNREASONABLE VALUCOMP024
     1E )                                                              COMP0241
```

DECK NAME: DIAG

SUBROUTINE: SREAD

```
210   FORMAT (1H0 71H AN ILLEGAL CHARACTER HAS BEEN REPLACED BY A ZERO I2266277
     1N THE FOLLOWING CARD /1H 24I3)                                    2266278
211   FORMAT (1H0100H AN ILLEGAL CHARACTER HAS BEEN REPLACED BY A ZERO I2266279
     1N THE FOLLOWING PRESSURIZATION PROGRAM INPUT CARD /1H 7F10.2)     2266280
201   FORMAT (10H0NTYPE OF I5,12H IS ILLEGAL.)                          2266270
202   FORMAT ( 9H0NRAD OF I5,12H IS ILLEGAL.)                           2266271
203   FORMAT ( 9H0NCIR OF I5,12H IS ILLEGAL.)                           2266272
204   FORMAT (10H0NSECT OF I5,12H IS ILLEGAL.)                          2266273
205   FORMAT ( 9H0NTNK OF I5,12H IS ILLEGAL.)                           2266274
206   FORMAT ( 8H0IGC OF I5,12H IS ILLEGAL.)                            2266275
207   FORMAT (11H0IPRESS OF I5,12H IS ILLEGAL.)                         2266276
```

DECK NAME: DIAG

SUBROUTINE: LOC

```
21    FORMAT (10H0CURVE NO.I5,25H IS NOT IN ADDRESS TABLE.)             LOC 535
```

TABLE E-1 DIAGNOSTICS (Continued)

DECK NAME: DIAG10
SUBROUTINE: COMP2

```
350  FORMAT (1H080H ILLEGAL CHARACTER IN ONE OF THE FOLLOWING NODE NOS.COM2032
      1 OR TEMPS. REPLACED BY ZERO./5X,I5,F10.2,I5,F10.2,I5,F10.2,I5,F10.COM2033
                                                                     COM20331
      22)
232  FORMAT (7H0ERROR,I5,E15.5,22H NEGATIVE NODE NUMBER.)             COMP030
351  FORMAT (1H087H ILLEGAL CHARACTER HAS BEEN REPLACED BY 0 IN THE FOLCOM2034
      1LOWING INC CARD, TEMPERATURE BLOCK./1H 5HINC  I5,E10.2,I5)     COM2035
352  FORMAT (1H045H FOLLOWING CARD ILLEGAL IN TEMPERATURE BLOCK  /5XA3,COM2036
                                                                     COM2037
      1I2,12A6)
353  FORMAT (1H042H FOLLOWING CARD ILLEGAL IN RESISTOR BLOCK  /5XA3,I2,COM2038
                                                                     COM2039
      112A6)
2531 FORMAT (1H087H AN ILLEGAL CHARACTER HAS BEEN REPLACED BY A ZERO INCOM2040
      1 THE FOLLOWING CARD, RESISTOR BLOCK)                           COM2041
233  FORMAT (7H0ERROR,3I5,E15.5,26H NEGATIVE RESISTOR NUMBER.)        COMP030
234  FORMAT (9H0RESISTORI5,15H CONNECTS NODESI5,1H,I5,27H GREATER THAN COM20301
                                                                     COM20302
      1MAX TEMP.NO.=I5)
358  FORMAT (1H043H FOLLOWING CARD ILLEGAL IN CAPACITOR BLOCK /5XA3,I2,COM2045
                                                                     COM2046
      112A6)
259  FORMAT (1H0 88H AN ILLEGAL CHARACTER HAS BEEN REPLACED BY A ZERO ICOM2047
      1N THE FOLLOWING CARD, CAPACITOR BLOCK)                         COM2048
235  FORMAT (14H0CAPACITOR NO.I5,22H IS GREATER THAN MAXT=I5)         COM2030
264  FORMAT (1H043H FOLLOWING CARD ILLEGAL IN DATA SUB-BLOCKS /5XA3,I2,COM2053
                                                                     COM2054
      112A6)
265  FORMAT (1H0 87H AN ILLEGAL CHARACTER HAS BEEN REPLACED BY A ZERO ICOM2055
      1N THE FOLLOWING CARD, DATA SUB-BLOCK)                          COM2056
220  FORMAT (26H0TABLE NO. = 0 IS ILLEGAL.)                           COMP029
230  FORMAT (28H0SOMETHING WRONG AFTER TABLEI6,17H AND BEFORE TABLEI6) COMP021
231  FORMAT (25H0SOMETHING WRONG IN TABLE I6)                         COMP021
244  FORMAT (1H0,I6,33H DATA STORAGE EXCEEDED. TOTAL OF I6,           COM20311
                                                                     COM20312
      1 12H DATA ITEMS.)                                             DIAG0312
245  FORMAT (30H0DATA WILL FIT VERSION A OR B.)
269  FORMAT (1H0 37H FOLLOWING CARD ILLEGAL IN TIME BLOCK /1H A3,I2,12ACOM2060
                                                                     COM2061
      16)
270  FORMAT (1H0 83H AN ILLEGAL CHARACTER HAS BEEN REPLACED BY A ZERO ICOM2062
      1N THE FOLLOWING CARD, TIME BLOCK  /1H A3,I2,F10.2,F10.2,F10.2,F10.COM2063
                                                                     COM2064
      22,F10.2,F10.2)
```


LOCKHEED CALIFORNIA COMPANY

TABLE E-1 DIAGNOSTICS (Continued)

DECK NAME: DIAG11

SUBROUTINE: RESTRT

```
222  FORMAT (30H0 TOO MANY WORDS IN DAT BLOCK   )                         COMP030
215  FORMAT (40H0 KOUNT IS NOT CORRECT IN RESTRT ROUTINE )                 COMP025
223  FORMAT (7H0CODE= A6,12H IS ILLEGAL.)                                  COMP021
280  FORMAT (1H0 44H FOLLOWING CARD OUT OF PLACE IN THIS RESTART)          22660221
250  FORMAT (43H KOUNT HAS BEEN INCLUDED ON BLOCK FLAG CARD )              COMP356
252  FORMAT (1H0 70H AN ILLEGAL CHARACTER HAS BEEN REPLACED BY ZERO IN 22660223
    1THE FOLLOWING CARD  /5X,5HDEC  I5)                                    22660224
236  FORMAT (15H0BLOCK NUMBER =I6,29H IS ILLEGAL. RESTART DELETED.  )  REST021
253  FORMAT (1H0 64H FOLLOWING CARD ILLEGAL IN TEMPERATURE BLOCK OF PRE22660225
    1VIOUS RESTART)                                                        22660226
254  FORMAT (1H0 88H TERMINATING CARD OF ONE OF THE BLOCKS IN PREVIOUS 22660227
    1RESTART CONTAINS AN ILLEGAL CHARACTER).                               22660228
255  FORMAT (1H0113H AN ILLEGAL CHARACTER HAS BEEN REPLACED BY A ZERO I22660229
    1N THE FOLLOWING CARD OF THE TEMP BLOCK OF THE PREVIOUS RESTART)   2266023
257  FORMAT (1H0 61H FOLLOWING CARD ILLEGAL IN RESISTOR BLOCK OF PREVIO22660232
    1US RESTART)                                                           22660233
259  FORMAT (1H0116HAN ILLEGAL CHARACTER HAS BEEN REPLACED BY A ZERO IN22660235
    1 THE FOLLOWING CARD OF THE RESISTOR BLOCK OF THE PREVIOUS RESTART)22660236
260  FORMAT (1H0 62H FOLLOWING CARD ILLEGAL IN CAPACITOR BLOCK OF PREVI22660237
    1OUS RESTART)                                                          22660238
261  FORMAT (1H0112H AN ILLEGAL CHARACTER HAS BEEN REPLACED BY A ZERO I22660239
    1N THE FOLLOWING CARD OF THE CAP BLOCK OF THE PREVIOUS RESTART)    22660240
262  FORMAT (1H0 57H FOLLOWING CARD ILLEGAL IN TIME BLOCK OF PREVIOUS R22660241
    1ESTART)                                                               22660242
263  FORMAT (1H0113H AN ILLEGAL CHARACTER HAS BEEN REPLACED BY A ZERO I22660243
    1N THE FOLLOWING CARD OF THE TIME BLOCK OF THE PREVIOUS RESTART)   22660244
264  FORMAT (1H0115H AN ILLEGAL CHARACTER HAS BEEN REPLACED BY A ZERO I22660245
    1N THE FOLLOWING CARD IN A DATA SUB-BLOCK OF THE PREVIOUS RESTART) 22660246
```

TABLE E-2

DATA DIAGNOSTIC PROGRAM LISTING

```
$IBFTC DIAG1
C      AS OF 12/4/64
       NLAG = 1                                                          2088M003
       CALL EXETIV (NLAG)                                                2088M004
       CALL EXIT                                                         2088M005
       STOP                                                              2088M006
       END                                                               2088M007
```

```
$IBMAP DIAG2
        ENTRY    .UN04.
.UN04.  PZE      UNIT04
UNIT04  FILE     ,UT4,READY,INOUT,BLK=22,BCD,NOLIST
        END
$IBMAP DIAG4
        ENTRY    .UN09.
.UN09.  PZE      UNIT09
UNIT09  FILE     ,B(3),READY,INOUT,BLK=22,BCD,NOLIST
        END
$IBMAP DIAG3
        ENTRY    .UN10.
.UN10.  PZE      UNIT10
UNIT10  FILE     ,A(3),READY,INOUT,BLK=22,BCD,NOLIST
        END
```

TABLE E-2. (Continued)

```
$IBFTC DIAG5
      SUBROUTINE EXETIV (NLAG)                                          EXETV002
C     AS OF 5/25/65
      COMMON P(14000),M(16),MAXT,MAXR,MAXC,KSP,KST,KURV7,NCVG,NSTS,NFAB EXETV010
      COMMON NABL(10),NKSP,LMMF,NRMF,NREAD,CC(12)                       EXETV020
      COMMON NR,MAXP,MAXK,MAXS,MAXM                                     EXET 020
      COMMON /CP2/ ERROR,MAXMUM                                         EXET021
      COMMON /CPITM/ ITIMO                                              EXET006
      COMMON /CPNET/NTYPE,TKRAD,TKLEN,TEMPZ,NRAD,NCIR,NSECT,NTNK,
     1 TKTHK(5),TKWAL1,TKWAL2,INNOD,KOUT,
     2          FMCO,FMNO,FMOLC,VGAS,IGC,IPRESS,FMOLN,FMIN,FMAX,PMIN,
     3          PMAX,FMOLX,TSAT,ANGLE,TX,PR,TANKV,
     4          MM(50), NUMTAB(100),                 NEXW(50),
     5          RADI(4),NSEX(50),HELB(14)       ,LLFLOW,HELT,HELP,HELW
     6          ,ISTRAT,SAREA,IOPTSX(5)
      COMMON /CONTR/  TLIQ,SCLIQ,TGAS,SCGAS,SCALL,TLSMAX,SCINT,TSGAS,
     1COMUN(5),DMCC,TLMAX,TALL,FMLO,GAVITY,FKG,FKL,ZC,PC,ZN,FMLU,FMLA,
     2          ZX,CPX,TLA,GAMMA,RHOL,RHOG,CPL,CPG,RLAM,MCNT,PN,FMC,FMN,R
      COMMON/CHP/SMCV,SMCA,SMNV,SMNA,SDMCC,SMLAS,SMLUS,SUMEX,HELWS,GOLD,
     1            ANGOLD,FMG,FML,TS,DELTS,DELVS,FMGAS
      COMMON /CPPLT/NPLT,NEND,KSKIP
      COMMON /EXFLG/NTFLG,NSAV,NFLG                                     HTB 006
      COMMON /EXPLT/ NLFLG
      DIMENSION VS(100),FLA(250)
      EQUIVALENCE (P(13651),VS),(P(13751),FLA)
      MAXMUM = 13650                                                    DIAG007
      ITIMO = 1                                                         EXET0069
      NEND=0
      NPLT=1
      ERROR=0.                                                          EXET 025
      NTYPE=0                                                           EXET 025
      NTFLG=0                                                           EXET 026
    5 IR=NLAG                                                           EXETV030
    8 CALL COMPIL(IR)                                                   EXETV060
      NLAG=IR                                                           EXETV070
      NLAG=2
      IF (NEND.GT.0) RETURN                                             EXET0772
      IF (ERROR.GT.1.) ERROR=0.                                         EXET0773
      IF (1.EQ.1) GO TO 5                                               EXET078
      RETURN                                                            EXET079
      END                                                               EXET099
```


LOCKHEED CALIFORNIA COMPANY

TABLE E-2 (Continued)

```
$IBFTC DIAG6
      SUBROUTINE COMPIL (IR)                                            COMP003
C     AS OF 04/27/65
C     8/21/64      CHANGED TO INCLUDE MINMAX                            COMP004
      COMMON P, M, MAXT, MAXR, MAXC, KSP, KST, KURV 7, NCVG, NSTS, NFAB COMP004
      COMMON NABL,NKSP,LMMF,NRMF,NREAD,CC                               COMP005
      COMMON NRECD, NWPD, NWCT, NWST, NWTM                              COMP006
      COMMON /CP2/ERROR,MAXMUM                                          COMP006
      COMMON /CPPLT/ NPLT,NEND                                          COMP006
      COMMON /EXPLT/ NLFLG                                              COMP006
      DIMENSION CZ(6),CY(6)                                             COMP007
      DIMENSION A(12), I 1(12), I 2(12), I 3(12), NABL(10), CC(12)      COMP007
      DIMENSION P(14000), IRLINE(1), KPSD(1), NSTRT(1), M(16), KDAT(1)  COMP008
      EQUIVALENCE (P(4000), IRLINE(1)), (P(8000), KPSD(1)), (P(13000),  COMP009
     1 NSTRT(1)), (P(1), KDAT(1))                                       COMP010
      REAL M, INC, NBK,NET                                              COMP0
      INTEGER CP                                                        COMP011
      DATA ENDF/6H    -1 /                                              COMP011
      DATA NET/6HNET   /                                                COMP011
      DATA DEC/6HDEC   /                                                COMP011
      DATA INC/6HINC   /                                                COMP011
      DATA PER/6HPER   /                                                COMP011
      DATA NBK/6HNBK   /                                                COMP011
      DATA CID/6HCID   /                                                COMP011
      DATA TAP/6HTAP   /                                                COMP011
      DATA RES/6HRES   /                                                COMP011
      DATA CAP/6HCAP   /                                                COMP011
      DATA COD/6HCOD   /                                                COMP011
      DATA ZERO/6H000000/                                               COMP011
      DATA (CZ(I),I=1,6)/2H01,2H02,2H03,2H04,2H05,2H06/,BLANK/1H /      COMP011
      DATA (CY(I),I=1,6)/2H 1,2H 2,2H 3,2H 4,2H 5,2H 6/,ZERZ/2H00/      COMP011
      DATA ZERY/2H 0/                                                   COMP011
  200 FORMAT (5X, 4(I 5, E 10.2))                                       COMP015
  201 FORMAT (5X, I 5, E 10.2, I 5)                                     COMP016
  202 FORMAT (5X, 2(3I 5, E 10.2))                                      COMP017
  203 FORMAT (5X, 3I 5, E 10.2, I 5)                                    COMP018
  204 FORMAT (5X, I 5)                                                  COMP019
  205 FORMAT (5X, 6E 10.2)                                              COMP020
  208 FORMAT (1H1,A3,I2,12A6)                                           COMP021
  209 FORMAT (A3,A2,12A6)                                               COMP021
  210 FORMAT (5X,12A6)                                                  COMP021
  211 FORMAT (5X, A3,I2,12A6)                                           COMP022
  212 FORMAT (1H1,9X,12A6)                                              COMP023
  214 FORMAT (56H0 BLOCK COUNT IN COMPIL ROUTINE IS AN UNREASONABLE VALUCOMP024
     1E )                                                               COMP0241
  215 FORMAT ( 66H0 COUNT IN COLS. 4 AND 5 IS NOT AN INTEGER LESS THAN OCOMP025
     1R EQUAL TO 6 )                                                    COMP0251
  216 FORMAT (I 3, A 6)                                                 COMP026
  218 FORMAT (I 3, 3X, I 2)                                             COMP027
  219 FORMAT (38H0 KOD IS NOT CORRECT IN COMPIL ROUTINE )               COMP028
  220 FORMAT (26H0TABLE NO. = 0 IS ILLEGAL.)                            COMP029
  222 FORMAT (30H0 TOO MANY WORDS IN DAT BLOCK   )                      COMP030
  223 FORMAT (7H0CODE= A6,12H IS ILLEGAL.)                              COMP021
  224 FORMAT (25H0CANNOT FIND CURVE NAMED A6,17H ON LIBRARY TAPE.)      COMP021
  230 FORMAT (28H0SOMETHING WRONG AFTER TABLEI6,17H AND BEFORE TABLEI6) COMP021
  231 FORMAT (25H0SOMETHING WRONG IN TABLE I6)                          COMP021
  232 FORMAT (5X,3H***,5X,6HCURVE A6,36H FROM LIBRARY TAPE WILL APPEAR HDIAG022
     1ERE.)
      NCVG = 1                                                          COMP031
      NSTS = 1                                                          COMP032
      MATLIB =1                                                         COMP032
```


LOCKHEED
CALIFORNIA COMPANY

TABLE E-2 -(Continued)

```
      NSW2=1                                                  2266032
      NPLT=1                                                  2266032
      NLFLG=1                                                 2266032
      NETSW=0                                                 COMP032
      NFAB = 1                                                COMP033
      NSW2=1                                                  COMP034
      LINE = 1                                                COMP034
      LMMF=1                                                  COMP034
      NRMF=0                                                  COMP034
      NREAD=0                                                 COMP034
      DO 101 I = 1, 10                                        COMP035
      NABL(I) = 0                                             COMP036
101   CONTINUE                                                COMP037
      GO TO (105, 149, 149), IR                               COMP038
105   REWIND 4                                                COMP039
      DO 107  I = 1 , MAXMUM                                  COMP040
      P(I) = 0.                                               COMP0401
107   CONTINUE                                                COMP0402
      KOUNT = 1                                               COMP042
110   CP=-1                                                   COMP043
      IF (KOUNT.EQ.2.AND.NETSW.EQ.2) GO TO 180                COMP0431
      IF (KOUNT.EQ.3.AND.NETSW.EQ.1) GO TO 190                COMP0431
      IF (MATLIB.EQ.1) READ (5,209) CM,CX,(A(I),I=1,12)       COMP044
1101  DO 1102 I=1,6                                           COMP045
1102  IF (CX.EQ.CY(I).OR.CX.EQ.CZ(I)) CP=I                    COMP0451
      IF (CX.EQ.BLANK.OR.CX.EQ.ZERZ.OR.CX.EQ.ZERY) CP=0       COMP046
      IF (CP.GE.0) GO TO 1103                                 COMP0461
      CP=0                                                    COMP0462
      ERROR=1.                                                COMP0462
      WRITE (6,215) CX                                        COMP0462
1103  IF (LINE.LT.60) GO TO 111                               COMP0463
109   WRITE (6,212)          (CC(I),I=1,12)                   COMP047
      LINE = 1                                                COMP048
111   IF (CM.EQ.CID) GO TO 117                                COMP049
      IF (CM.EQ.TAP) GO TO 150                                COMP0491
      WRITE (6,211) CM,CP,(A(I),I=1,12)                       COMP0492
      LINE=LINE+1                                             COMP0492
      IF (CM .EQ. DEC) GO TO 116                              COMP050
      IF (CM .EQ. INC) GO TO 118                              COMP051
      IF (CM .EQ. PER) GO TO 119                              COMP052
      IF (CM .EQ. NBK) GO TO 120                              COMP053
      IF (CM.EQ.RES) GO TO 123                                COMP050
      IF (CM.EQ.CAP) GO TO 124                                COMP050
      IF (CM.EQ.NET) GO TO 170                                COMP053
      IF (CM.EQ.COD) GO TO 112                                COMP050
115   WRITE (6,223) CM                                        COMP054
      ERROR=1.                                                COMP054
      GO TO 110                                               COMP055
112   KOD = 9                                                 COMP0551
      GO TO 125                                               COMP0552
116   KOD = 1                                                 COMP056
      GO TO 125                                               COMP057
117   CMID=CM                                                 COMP057
      ICP=CP                                                  COMP0571
      DO 1171 I=1,12                                          COMP0572
      CC(I)=A(I)                                              COMP0573
1171  CONTINUE                                                COMP0574
      LINE =1                                                 COMP0575
      WRITE (6,212)          (CC(I),I=1,12)                   COMP0575
      GO TO 110                                               COMP0576
118   KOD = 3                                                 COMP058
      GO TO 125                                               COMP059
```


LOCKHEED

TABLE E-2 (Continued)

```
119  KOD = 4                                            COMP060
     GO TO 125                                          COMP061
120  KOD = 5                                            COMP062
     KOUNT = KOUNT + 1                                  COMP063
     IF (KOUNT - 6) 125, 121, 122                       COMP064
121  KOD = 6                                            COMP065
     GO TO 125                                          COMP066
122  WRITE (6,214)                                      COMP067
     ERROR =2.                                          COMP068
     RETURN                                             COMP069
123  KOD=7                                              COMP0691
     GO TO 125                                          COMP0692
124  KOD=8                                              COMP0693
     GO TO 125                                          COMP0694
150  WRITE (6,211) CM,CP,(A(I),I=1,12)                  DIAG0702
     WRITE (6,232) A(1)                                 DIAG0703
     LINE = LINE +2                                     DIAG0704
     GO TO 110                                          COMP0710
170  NETSW=2                                            COMP0711
     CALL INSIDE                                        COMP0712
     GO TO 110                                          COMP0712
180  READ (10,209) CM,CX,(A(I),I=1,12)                  COMP0713
     IF (A(1).NE.ENDF) GO TO 1101                       COMP0714
     NETSW=1                                            COMP0715
     REWIND 10                                          COMP0715
     GO TO 110                                          COMP0716
190  READ ( 9,209) CM,CX,(A(I),I=1,12)                  COMP0717
     IF (A(1).NE.ENDF) GO TO 1101                       COMP0718
     NETSW=0                                            COMP0719
     REWIND 9                                           COMP0719
     GO TO 110                                          COMP0720
125  WRITE (4,218) KOD,CP                               COMP0721
     WRITE (4, 210) (A(I), I = 1, 12)                   COMP073
     IF (KOUNT - 6) 110, 130, 122                       COMP074
130  END FILE 4                                         COMP075
     REWIND 4                                           COMP076
     REWIND 3                                           COMP077
     CALL COMP2                                             078
     RETURN                                                 079
149  CALL RESTRT
     RETURN
     END                                                    080
```

TABLE E-2 (Continued)

```
$IBFTC DIAG7
      SUBROUTINE INSIDE                                                 1911002
C     AS OF 6/30/65
      COMMON P(14000)
      COMMON /CPNET/NTYPE,TKRAD,TKLEN,TEMPZ,NRAD,NCIR,NSECT,NTNK,
     1 TKTHK(5),TKWAL1,TKWAL2,INNOD,KCUT,
     2          FMCO,FMNO,FMOLC,VGAS,IGC,IPRESS,FMOLN,FMIN,FMAX,PMIN,
     3          PMAX,FMOLX,TSAT,ANGLE,TX,PR,TANKV,
     5          RADI(4),HELB(14),LLFLOW,HELT,HELP,HELW
     6          ,ISTRAT,SAREA,IOPTSX(5),FMULTF(5),STRINP(20)
     7           ,GSOLD (2),TTCPBL(2),DLIQ   ,NLSURF,ICALL
      EQUIVALENCE  (FMULTF(1), FLT),(FMULTF(2),FLKL)
      DIMENSION VS(250),DUMMY(500)
     1   ,NUMTAB(100),NEXW(125),NSEX(125),MM(125)
      DIMENSION LNO(250),NLNNS(125),NGNNS(125)
      EQUIVALENCE (P(13251),VS),(P(13501),DUMMY)
     1,(P(12526),NUMTAB),(P(12626),NEXW),(P(12751),NSEX),(P(12876),MM)
      EQUIVALENCE(P(12276),LNO),(P(12151),NLNNS),(P(12026),NGNNS)
      REAL LEN                                                          1911007
      PI = 3.14159265                                                   1911004
      V=0.                                                              INSD
      FLA=0.                                                            INSD
      CALL SREAD                                                        1911005
      INODE=INNOD
      REWIND 9
      REWIND 10
C     TANKV = VOLUME OF TANK
      VOLX=PI*TKRAD**2                                                  INSD007
      IF (NTYPE.EQ.2) TANKV=1.33333*VOLX*TKRAD                          INSD0071
      IF (NTYPE.EQ.1) TANKV=VOLX*(1.33333*TKRAD+TKLEN)                  INSD0072
      ITIM= 1                                                           1911008
      SUM =0.0                                                          1911009
      DO 10 I=1,NTNK                                                    1911010
      SUM =SUM+ TKTHK(I)                                                1911011
   10 CONTINUE                                                          1911012
      FNRAD =NRAD                                                       1911013
      FNCIR =NCIR                                                       1911014
      FNSECT= NSECT                                                     1911015
      KEYNOD=  -1                                                       1911016
C     IF CYLINDER NTYPE=1 ,IF SPHERE  NTYPE=2                           1911017
      PI2= 3.14159265*2.0                                               1911018
      NRES =0                                                           1911019
C     COMPUTE  RADIAL RESISTORS FOR HEMISPHERE                          1911021
   15 KK = NRAD/4                                                       1911022
      M = 0                                                             1911023
      KI = (NCIR+NTNK +1)                                               1911024
      JJ =  NRAD                                                        1911025
      J1 =  NCIR + NTNK +1                                              1911026
      II = NCIR +NTNK                                                   1911027
      DO   50  KI= 1,KK                                                 1911028
      K = KI-1                                                          1911029
      DO   40  JI= 1,JJ                                                 1911030
      J=JI -1                                                           1911031
      DO   30   I= 1,II,1                                               1911032
      M = M+1                                                           1911033
      NODE =    M-1   +INODE                                            1911034
   29 NRES =NRES+1                                                      1911056
      NODE1= NODE+1                                                     1911057
      WRITE(10,8002) NRES,NODE,NODE1,FLA                                1911058
 8002 FORMAT(5HDEC01,3I5,E10.3,50X)                                     INSD059
   30 CONTINUE                                                          1911060
```


LOCKHEED
CALIFORNIA COMPANY

TABLE E-2 (Continued)

```
      M= M+1                                                    1911061
   40 CONTINUE                                                  1911062
   50 CONTINUE                                                  1911063
      M = 0                                                     1911064
      II =II+1                                                  1911065
      DO  80 KI =1,KK                                           1911066
      K = KI-1                                                  1911067
      DO  70 JI =1,JJ                                           1911068
      J = JI-1                                                  1911069
      DO  60 I  =1,II                                           1911070
      M  =  M  + 1                                              1911071
      NODE =          M +INODE - 1                              1911072
      NODE1 = NODE +NCIR +NTNK+ 1                               1911073
      IF(JI.EQ.JJ) NODE1 = NODE-(NRAD-1)*(NCIR+NTNK+1)          1911074
      IF(I.LE.NCIR+1) GO TO 59                                  INSD075
      IF(I.NE.II) GO TO 60                                      INSD085
   59 NRES = NRES + 1                                           1911089
      WRITE(10,8002) NRES,NODE,NODE1,FLA                        1911090
   60 CONTINUE                                                  1911091
   70 CONTINUE                                                  1911092
   80 CONTINUE                                                  1911093
      M = 0                                                     1911094
      KK =KK-1                                                  1911095
      IF (KK*JJ*II.EQ.0)  GO TO 115
      DO 110 KI=1,KK                                            1911096
      K = KI-1                                                  1911097
      DO 100 JI=1,JJ                                            1911098
      J = JI-1                                                  1911099
      DO 90 I=1,II                                              1911100
      M=M+1
      NODE =    M-1+INODE                                       1911101
      NODE1=    NODE +(NRAD)*(NCIR + NTNK + 1)                  1911102
      IF(I.LE.NCIR+1) GO TO 89                                  INSD103
      IF(I.NE.II) GO TO 90                                      INSD112
   89 NRES =  NRES + 1                                          1911116
      WRITE(10,8002) NRES,NODE,NODE1,FLA                        1911117
   90 CONTINUE                                                  1911118
  100 CONTINUE                                                  1911119
  110 CONTINUE                                                  1911120
C     CALCULATE  CONDENSOR BLOCK LIST FOR SPHERE                1911121
  115 CONTINUE
      M =0                                                      1911122
      KK = NRAD/4                                               1911123
      JJ = NRAD                                                 1911124
      II = NCIR + NTNK + 1                                      1911125
      DO  140  KI= 1,KK                                         1911126
      K =KI-1                                                   1911127
      CON =-SIN(FLOAT(K)*PI2/FNRAD) +SIN(FLOAT(K+1)*PI2/FNRAD)  1911128
      DO  130  JI= 1,JJ                                         1911129
      J = JI-1                                                  19111291
      DO  120  I =1,II                                          1911130
      M = M+1                                                   1911131
      NODE = M -1 +INODE                                        1911132
  119 WRITE(9,8003) NODE,V                                      1911147
 8003 FORMAT(5HDEC01,I5,E10.3,60X)                              INSD148
  120 CONTINUE                                                  1911149
  130 CONTINUE                                                  1911150
  140 CONTINUE                                                  1911151
C     TEST  KEYNODE                                             1911152
C                                                               1911153
C                                                               1911154
```

TABLE E-2 (Continued)

```
C                                                                       1911155
      IF( KEYNOD.EQ.-1)GO TO 200                                        1911156
      JJ = NRAD                                                         1911157
      II = NCIR +NTNK + 1                                               1911158
      M =0                                                              1911159
C     CALCULATE CONNECTING RESISTORS SPHERE TO SPHERE AND SPHERE TO     1911160
C     CYLINDER                                                          1911161
      DO 170 JI =1,JJ                                                   1911162
      J= JI-1                                                           1911163
      DO 160 I  =1,II                                                   1911164
      M= M+1                                                            1911165
      NODE = M-1 +KEYNOD                                                1911166
                NODE1 = M-1 +INODE                                      1911168
      IF(I.LE.NCIR+1) GO TO 149                                         INSD169
      IF(I.NE.II) GO TO 160                                             INSD178
  149 NRES=NRES+1                                                       INSD183
      WRITE(10,8002) NRES,NODE,NODE1,FLA                                1911184
  160 CONTINUE                                                          1911185
  170 CONTINUE                                                          1911186
  190 NEND =-1                                                          1911187
      WRITE( 9,8002) NEND                                               1911188
      WRITE(10,8002) NEND                                               1911189
      END FILE  9                                                       1911190
      END FILE  10                                                      1911191
      REWIND 9                                                          INSD1911
      REWIND 10                                                         INSD1911
      RETURN                                                            1911192
  200 IF(NTYPE.EQ.1) GO TO 201                                          1911193
      KEYNOD=  INODE                                                    1911194
      INODE = (NCIR + NTNK+1)*NRAD**2/4 +INODE                          1911195
      GO TO 15                                                          1911196
  201 KEYNOD = INODE                                                    1911197
      INODE = (NCIR + NTNK+1)*NRAD**2/4 + INODE                         1911198
C     CALCULATE RESISTORS AND CONDENSORS OF CYLINDER BODY               1911199
      M = 0                                                             1911200
      II = NCIR +NTNK                                                   1911201
      JJ =   NRAD                                                       1911202
      KK =  NSECT                                                       1911203
      DO 230 KI =1,KK                                                   1911204
      K = KI-1                                                          1911205
      DO 220 JI =1,JJ                                                   1911206
      J = JI-1                                                          1911207
      DO 210 I  =1,II                                                   1911208
      M = M +1                                                          1911209
      NODE  = M-1 +INODE                                                1911210
      NODE1= NODE +1                                                    1911211
  209 NRES = NRES + 1                                                   1911233
      WRITE(10,8002) NRES,NODE,NODE1,FLA                                1911234
  210 CONTINUE                                                          1911235
      M = M+1                                                           1911236
  220 CONTINUE                                                          1911237
  230 CONTINUE                                                          1911238
      II =II+1                                                          1911239
      M   = 0                                                           1911240
      DO  270 KI =1,KK                                                  1911241
      K = KI-1                                                          1911242
      DO  260 JI =1,JJ                                                  1911243
      J = JI-1                                                          1911244
      DO  250  I =1,II                                                  1911245
      M = M+1                                                           1911246
      NODE = M-1 +INODE                                                 1911247
      NODE1=    NODE +NCIR +NTNK + 1                                    1911248
```

TABLE E-2 (Continued)

```
  249 IF(JI.EQ.JJ) NODE1 = NODE1 -NRAD*(NCIR + NTNK+1)
      NRES =NRES +1                                         1911264
      WRITE(10,8002) NRES,NODE,NODE1,FLA                    1911265
  250 CONTINUE                                              1911266
  260 CONTINUE                                              1911267
  270 CONTINUE                                              1911268
      KK = KK-1                                             1911269
      LEN =TKLEN /FNSECT                                    1911270
      M =0                                                  1911271
      IF (KK*JJ*II.EQ.0)  GO TO 305                         1911272
      DO 300 KI =1,KK
      K = KI-1                                              1911273
      DO 290 JI =1,JJ                                       1911274
      J=JI-1                                                1911275
      DO 280 I  =1,II                                       1911276
      M=M+1                                                 1911277
      NODE = M-1 +INODE
      NODE1 = NODE +(NRAD   )*(NCIR+NTNK+1)                 1911278
      IF(I.LE.NCIR+1) GO TO 279                             1911279
      IF(I.NE.II) GO TO 280                                 INSD280
  279 NRES = NRES +1                                        INSD287
      WRITE(10,8002) NRES,NODE,NODE1,FLA                    1911291
  280 CONTINUE                                              1911292
  290 CONTINUE                                              1911293
  300 CONTINUE                                              1911294
C     CALCULATE CONDENSORS FOR CYLINDER                     1911295
  305 CONTINUE                                              1911296
      KK =KK+1
      M = 0                                                 1911297
      DO 330 KI=1,KK                                        1911298
      K = KI-1                                              1911299
      DO 320 JI=1,JJ                                        1911300
      J = JI-1                                              1911301
      DO 310 I =1,II                                        1911302
      M =M+1                                                1911303
      NODE = M -1 +INODE                                    1911304
  309 WRITE( 9,8003) NODE,V                                 1911305
  310 CONTINUE                                              1911319
  320 CONTINUE                                              1911320
  330 CONTINUE                                              1911321
      II = NCIR + NTNK +1                                   1911322
      JJ = NRAD                                             1911323
      M = 0                                                 1911324
      DO  350 JI =1,JJ                                      1911325
      J  = JI-1                                             1911326
      DO  340 I =1,II                                       1911327
      M  = M+1                                              1911328
      NODE  = M-1 +KEYNOD                                   1911329
      NODE1 = M-1 +INODE                                    1911330
      IF(I.LE.NCIR+1) GO TO 349                             1911331
      IF(I.NE.II) GO TO 340                                 INSD332
  349 NRES=NRES+1                                           INSD340
      WRITE(10,8002) NRES ,NODE, NODE1,FLA                  INSD344
  340 CONTINUE                                              1911345
  350 CONTINUE                                              1911346
      KEYNOD= INODE +(NCIR+NTNK+1)*NRAD *(NSECT-1)          1911347
      INODE = INODE +(NCIR+NTNK+1)*NRAD *NSECT              1911348
      GO  TO 15                                             1911349
      END                                                   1911350
                                                            1911352
```

TABLE E-2 (Continued)

```
$IBFTC DIAG8
      SUBROUTINE SREAD
C     AS OF 05/18/65
C     IF KOUT=1       --SOME AUXILIARY PRINT
C     IF KOUT=2       --ALL  AUXILIARY PRINT
      COMMON P(14000)
      COMMON /CPNET/NTYPE,TKRAD,TKLEN,TEMPZ,NRAD,NCIR,NSECT,NTNK,
     1 TKTHK(5),TKWAL1,TKWAL2,INNOD,KOUT,
     2           FMCO,FMNO,FMOLC,VGAS,IGC,IPRESS,FMOLN,FMIN,FMAX,PMIN,
     3           PMAX,FMOLX,TSAT,ANGLE,TX,PR,TANKV,
     4           MM(50), NUMTAB(100),                  NEXW(50),
     5           RADI(4),NSEX(50),HELB(14)         ,LLFLOW,HELT,HELP,HELW
     6           ,ISTRAT,SAREA,IOPTSX(5)
      DIMENSION VS(100),FLA(250)
      EQUIVALENCE (P(13651),VS),(P(13751),FLA)
      EQUIVALENCE (IOPTSX(1),IRSTRT),(IOPTSX(2),IPTLIQ),
     1             (IOPTSX(3),LIBTAP),(IOPTSX(4),MTIME )
     2            ,(IOPTSX(5),KUT    )
C     MTIME = MACHINE TIME FOR THIS SEGMENT (MINUTES) IF RUN IN SEGMENTS
C     KUT   = TIME(IN SECONDS) AT WHICH CURRENT SEGMENT IS TO BE CUT
      DIMENSION STRINP(7),FMULTF(5)
      DIMENSION IERROR(1)
      DATA IERROR(1)/32/, NERROR/1/
      CALL FXPPT (IERROR,NERROR)
      CALL FSWTON(3,32,33,45)
      CALL FXNPRT(1,32)
      READ(5,10)  NTYPE,NRAD,NCIR,NSECT,NTNK,INNOD,IGC,IPRESS,KOUT
     1           ,(IOPTSX(I),I=1,4),KUT1,KUT2,KUT3
      IF (IERROR(1).GT.0) GO TO 1
      WRITE (6,210) NTYPE,NRAD,NCIR,NSECT,NTNK,INNOD,IGC,IPRESS,KOUT
     1              ,(IOPTSX(I),I=1,4),KUT1,KUT2,KUT3
      IERROR(1) =IABS(IERROR(1))
    1 CONTINUE
      MTIME = MTIME *60
      KUT         = 1000000*KUT1 + 1000*KUT2 + KUT3
      IF (NTYPE.LT.1.OR.NTYPE.GT.2)        GO TO 101
    2 IF (NRAD.LE.0.OR.MOD(NRAD,4).NE.0)  GO TO 102
    3 IF (NCIR.LT.1)                       GO TO 103
    4 IF (NSECT.LT.1)                      GO TO 104
    5 IF (NTNK.LT.0. OR.NTNK.GT.5)         GO TO 105
    6 IF (IGC.LT.1.OR.IGC.GT.2)            GO TO 106
    7 IF (IPRESS.LT.1.OR.IPRESS.GT.2)      GO TO 107
    8 READ (5,20) TKRAD,TKLEN,TEMPZ,TKWAL1,TKWAL2
      IF (IERROR(1) .GT.0) GO TO 11
      WRITE (6,211) TKRAD,TKLEN,TEMPZ,TKWAL1,TKWAL2
      IERROR(1) =IABS(IERROR(1))
   11 CONTINUE
      READ (5,20)  (TKTHK(I),I=1,5)
      IF (IERROR(1).GT.0) GO TO 12
      WRITE (6,211) (TKTHK(I),I=1,5)
      IERROR(1) =IABS (IERROR(1))
   12 CONTINUE
      READ (5, 20)  PR,VGAS,TSAT,FMOLC,FMOLN,FMOLX,TX
      IF (IERROR(1).GT.0) GO TO 13
      WRITE (6,211) PR,VGAS,TSAT,FMOLC,FMOLN,FMOLX,TX
      IERROR(1) =IABS(IERROR(1))
   13 CONTINUE
      READ (5, 20)  FMIN,FMAX,PMIN,PMAX
      IF (IERROR(1).GT.0) GO TO 14
      WRITE (6,211) FMIN,FMAX,PMIN,PMAX
      IERROR(1) =IABS(IERROR(1))
```


LOCKHEED
CALIFORNIA COMPANY

TABLE E-2 (Continued)

```
 14    CONTINUE
       READ (5,20)  (RADI(I),I=1,4)                                       2266229
       IF (IERROR(1).GT.0) GO TO 15
       WRITE (6,211) (RADI(I),I=1,4)                                      2266230
       IERROR(1) =IABS(IERROR(1))                                         2266231
 15    CONTINUE                                                           2266232
       READ (5,20) (HELB(I),I=1, 8)                                       2266233
       IF (IERROR(1).GT.0) GO TO 16                                       2266001
       WRITE (6,211) (HELB(I),I=1,8)                                      2266234
       IERROR(1) =IABS(IERROR(1))                                         2266235
 16    CONTINUE                                                           2266236
       READ (5,20) (STRINP(I),I=1,7)                                      2266237
       IF (IERROR(1).GT.0) GO TO 17
       WRITE (6,211) (STRINP(I),I=1,7)
       IERROR(1) = IABS(IERROR(1))
 17     CONTINUE
       READ (5,20) (FMULTF(I),I=1,5)
       IF (IERROR(1).GT.0) GO TO 18
       WRITE (6,211) (FMULTF(I),I=1,5)
       IERROR(1) = IABS(IERROR(1))
 18    CONTINUE
       WRITE (6,601)
      1  NTYPE,NRAD,NCIR,NSECT,NTNK,INNOD,IGC,IPRESS,
      2  TKRAD,TKLEN,TEMPZ,TKWAL1,TKWAL2,(TKTHK(I),I=1,5),
      3  PR,VGAS,TSAT,FMOLC,FMOLN,FMOLX,TX,
      4  FMIN,FMAX,PMIN,PMAX,(RADI(I),I=1,4),
      5  (HELB(I),I=1, 8)
       HELB(5) = HELB(5)+460.
       HELT = HELB(5)
       HELP = HELB(7)                                                     2266002
       HELB(11) = HELT                                                    2266004
       HELB(13)=(NCIR+NTNK)*(NRAD*(2+NSECT))+
      1(NCIR+NTNK+1)*(NRAD*(3+2*NSECT)+2*(NRAD*(NRAD/4-1)))
       HELB(12)=(NCIR+NTNK+1)*(NRAD *(NSECT+2*NRAD/4))
 601   FORMAT(29H1PRESSURIZATION PROGRAM INPUT  //
      1 80H NTYPE    NRAD      NCIR      NSECT      NTNK      INODE      IG
      2C      IPRESS     /1H ,I3,7I10/ 50HOTKRAD     TKLEN      TEMPZ      T
      3KWAL 1    TKWAL 2    /XF7.3,4F10.3/23HOINSULATION THICKNESSES/
      4XF7.3,4F10.3/,
      5 70HO PR      V GAS      T SAT    MOL C     MOL N     MOL X
      6T X       /XF7.3,6F10.3/35HOF MIN     F MAX      P MIN      P MAX
      7/XF7.3,3F10.3/,
      8 40HORADI F    RADI S    RADI E1    RADI E2
      A  /X F7.3,E11.3,2F10.3/
      9 20HOHELIUM BOTTLE INPUT    /    XF7.1,6F10.3/ F7.3/1H1)
       TSAT = TSAT+460.
       TEMPZ = TEMPZ+460.
       TX = TX+460.
       LLFLOW = 0
       RETURN
 10    FORMAT(24I3)
 20    FORMAT(7F10.0)
 101   WRITE (6,201)  NTYPE
       GO TO 2
 102   WRITE (6,202)  NRAD                                                2266250
       GO TO 3
 103   WRITE (6,203)  NCIR                                                2266252
       GO TO 4
 104   WRITE (6,204)  NSECT                                               2266254
       GO TO 5
 105   WRITE (6,205)  NTNK                                                2266256
```

TABLE E-2 (Continued)

```
      GO TO 6                                                           2266258
  106 WRITE (6,206)  IGC
      GO TO 7                                                           2266260
  107 WRITE (6,207)  IPRESS
      GO TO 8                                                           2266262
  201 FORMAT (10HONTYPE OF I5,12H IS ILLEGAL.)                          2266270
  202 FORMAT ( 9HONRAD OF I5,12H IS ILLEGAL.)                           2266271
  203 FORMAT ( 9HONCIR OF I5,12H IS ILLEGAL.)                           2266272
  204 FORMAT (10HONSECT OF I5,12H IS ILLEGAL.)                          2266273
  205 FORMAT ( 9HONTNK OF I5,12H IS ILLEGAL.)                           2266274
  206 FORMAT ( 8HOIGC OF I5,12H IS ILLEGAL.)                            2266275
  207 FORMAT (11HOIPRESS OF I5,12H IS ILLEGAL.)                         2266276
  210 FORMAT (1H0 71H AN ILLEGAL CHARACTER HAS BEEN REPLACED BY A ZERO I2266277
     IN THE FOLLOWING CARD /1H 24I3)                                    2266278
  211 FORMAT (1H0100H AN ILLEGAL CHARACTER HAS BEEN REPLACED BY A ZERO I2266279
     IN THE FOLLOWING PRESSURIZATION PROGRAM INPUT CARD /1H 7F10.2)     2266280
      END
```

```
$IBFTC DIAG9
      FUNCTION LOC (KT)                                                 LOC  002
C     AS OF 04/27/65
C                                                                       LOC  003
C   LOC ROUTINE FINDS FIRST LOCATION AHEAD OF LOCATION STARTING TABLE KTLOC  004
C                                                                       LOC  005
      COMMON P, M, MAXT, MAXR, MAXC, KSP, KST                           LOC  006
      COMMON /EXADR/ LCTN,LCTO,LCR,LCC,LCQ,LCRC,LCPS,LCK,LCKA           LOC 5207
      COMMON /CP2/ ERROR                                                LOC 5207
      DIMENSION P(14000), M(16), NSTRT(1)                               LOC  007
      EQUIVALENCE (P(1), NSTRT(1))                                      LOC  008
      REAL M                                                            LOC  009
      IF (KT.LE.0) GO TO 13                                             LOC 523
   10 J=LCKA                                                            LOC 524
   11 J1=NSTRT(J)/2**18                                                 LOC 5241
      J2=MOD(NSTRT(J),2**18)                                            LOC 525
      IF (J1.EQ.KT) GO TO 12                                            LOC 527
      IF (J1.EQ.0)  GO TO 13                                            LOC 528
      J=J+1                                                             LOC 529
      GO TO 11                                                          LOC 530
   12 IF (P(J2).NE.32767.) J2=J2-1                                      LOC 531
      LOC=J2                                                            LOC 532
      RETURN                                                            LOC 533
   13 WRITE (6,21) KT                                                   LOC 534
   21 FORMAT (10HOCURVE NO.I5,25H IS NOT IN ADDRESS TABLE.)             LOC 535
      ERROR =1.                                                         LOC 536
      RETURN                                                            LOC 537
      END                                                               LOC 538
```

TABLE E-2 (Continued)

```
$IBFTC DIAG10
      SUBROUTINE COMP2                                                  COM2002
C     AS OF 06/02/65
      COMMON P, M, MAXT, MAXR, MAXC, KSP, KST, KURV 7, NCVG, NSTS, NFAB COMP004
      COMMON NABL,NKSP,LMMF,NRMF,NREAD,CC                               COMP005
      COMMON NRECD, NWPD, NWCT, NWST, NWTM                              COMP006
      COMMON /CP2/ERROR,MAXMUM                                          COM2006
      DIMENSION BCDB(5)                                                 COM2007
      DIMENSION A(12), I 1(12), I 2(12), I 3(12), NABL(10), CC(12)      COMP007
      DIMENSION P(14000), IRLINE(1), KPSD(1), NSTRT(1), M(16), KDAT(1)  COMP008
      EQUIVALENCE (P(4000), IRLINE(1)), (P(8000), KPSD(1)), (P(13000),  COMP009
     1 NSTRT(1)), (P(1), KDAT(1))                                       COMP010
      REAL M, INC, NBK,NET                                              COMP0
      INTEGER CP                                                        COMP011
      DATA (BCDB(I),I=1,5)/6HTEMP. ,6HRESIS.,6HCAPAC.,6HCURVE ,6HMISC. /COM2011
  200 FORMAT (5X, 4(I 5, E 10.2))                                       COMP015
  201 FORMAT (5X, I 5, E 10.2, I 5)                                     COMP016
  202 FORMAT (5X, 2(3I 5, E 10.2))                                      COMP017
  203 FORMAT (5X, 3I 5, E 10.2, I 5)                                    COMP018
  204 FORMAT (5X, I 5)                                                  COMP019
  205 FORMAT (5X, 6E 10.2)                                              COMP020
  206 FORMAT (5X,E10.2,I5)                                              COM2021
  208 FORMAT (1H1,A3,I2,12A6)                                           COMP021
  209 FORMAT (A3,I2,12A6)                                               COMP021
  210 FORMAT (5X,12A6)                                                  COMP021
  211 FORMAT (5X, A3,I2,12A6)                                           COMP022
  212 FORMAT (5H1      12A 6)                                           COMP023
  214 FORMAT (29H0 INPUT MUST BE BAD, QUIT JOB )                        COMP024
  215 FORMAT (40H0 KOUNT IS NOT CORRECT IN  COM2  ROUTINE )             COMP025
  216 FORMAT (I 3, A 6)                                                 COMP026
  218 FORMAT (I 3, 3X, I 2)                                             COMP027
  219 FORMAT (15H0KOD MUST NOT =I5,8H IN THE  A6,7H BLOCK.)             COM2028
  220 FORMAT (26H0TABLE NO. = 0 IS ILLEGAL.)                            COMP029
  222 FORMAT (30H0 TOO MANY WORDS IN DAT BLOCK   )                      COMP030
  223 FORMAT (7H0CODE= A6,12H IS ILLEGAL.)                              COMP021
  224 FORMAT (25H0CANNOT FIND CURVE NAMED A6,17H ON LIBRARY TAPE.)      COMP021
  230 FORMAT (28H0SOMETHING WRONG AFTER TABLEI6,17H AND BEFORE TABLEI6) COMP021
  231 FORMAT (25H0SOMETHING WRONG IN TABLE I6)                          COMP021
  232 FORMAT (7H0ERROR,I5,E15.5,22H NEGATIVE NODE NUMBER.)              COMP030
  233 FORMAT (7H0ERROR,3I5,E15.5,26H NEGATIVE RESISTOR NUMBER.)         COMP030
  234 FORMAT (9H0RESISTORI5,15H CONNECTS NODESI5,1H,I5,27H GREATER THAN COM20301
     1MAX TEMP.NO.=I5)                                                  COM20302
  235 FORMAT (14H0CAPACITOR NO.I5,22H IS GREATER THAN MAXT=I5)          COM2030
  236 FORMAT (5H0NODE I5,20H HAS NO CONNECTIONS.)                       COMP030
  240 FORMAT (5X,3I5,3E10.0,I5)                                         COM2031
  241 FORMAT (5X,3I5,3E10.0,2I5)                                        COM2031
  242 FORMAT (5X,I5,3E10.0,2I5)                                         COM2031
  243 FORMAT (5X,I5,3E10.0,3I5)                                         COM2031
  244 FORMAT (1H0,I6,33H DATA STORAGE EXCEEDED. TOTAL OF I6,            COM20311
     1 12H DATA ITEMS.)                                                 COM20312
  245 FORMAT (30H0DATA WILL FIT VERSION A OR B.)                        DIAG0312
  250 FORMAT (14H1P STORAGE MAP /14H0 TEMPERATURES,9X,I6,8X,6HTOTAL I6) COM20313
  251 FORMAT (23H  TEMPERATURES          I6,8X,6HTOTAL I6)              COM20313
  252 FORMAT (23H  RESISTORS             I6,8X,6HTOTAL I6)              COM20313
  253 FORMAT (23H  CAPACITORS            I6,8X,6HTOTAL I6)              COM20313
  254 FORMAT (23H  CURRENTS (Q)          I6,8X,6HTOTAL I6)              COM20313
  255 FORMAT (23H  R-C PRODUCTS          I6,8X,6HTOTAL I6)              COM20313
  256 FORMAT (23H  PSEUDO-SEQENCE        I6,8X,6HTOTAL I6)              COM20313
  257 FORMAT (23H  DATA BLOCKS (CURVES)  I6,8X,6HTOTAL I6)              COM20313
  258 FORMAT (23H  DATA BLOCK ADDRESSES  I6,8X,6HTOTAL I6)              COM20313
  350 FORMAT (1H080H ILLEGAL CHARACTER IN ONE OF THE FOLLOWING NODE NOS.COM2032
```


LOCKHEED CALIFORNIA COMPANY

TABLE E-2 (Continued)

```
     1 OR TEMPS. REPLACED BY ZERO./5X,I5,F10.2,I5,F10.2,I5,F10.2,I5,F10.COM2033
     22)                                                              COM20331
 351  FORMAT (1H087H ILLEGAL CHARACTER HAS BEEN REPLACED BY 0 IN THE FOLCOM2034
     1LOWING INC CARD, TEMPERATURE BLOCK./1H 5HINC  I5,E10.2,I5)      COM2035
 352  FORMAT (1H045H FOLLOWING CARD ILLEGAL IN TEMPERATURE BLOCK  /5XA3,COM2036
     1I2,12A6)                                                        COM2037
 353  FORMAT (1H042H FOLLOWING CARD ILLEGAL IN RESISTOR BLOCK  /5XA3,I2,COM2038
     112A6)                                                           COM2039
 2551 FORMAT (1H087H AN ILLEGAL CHARACTER HAS BEEN REPLACED BY A ZERO INCOM2040
     1 THE FOLLOWING CARD, RESISTOR BLOCK)                            COM2041
 354  FORMAT (1H A3,I2,3I5,F10.2,3I5,F10.2)                           COM2042
 356  FORMAT (1H A5,3I5,3F10.2,2I5)                                   COM2043
 357  FORMAT (1H A5,3I5,E10.2,I5)                                     COM2044
 358  FORMAT (1H043H FOLLOWING CARD ILLEGAL IN CAPACITOR BLOCK /5XA3,I2,COM2045
     1I2A6)                                                           COM2046
 259  FORMAT (1H0 88H AN ILLEGAL CHARACTER HAS BEEN REPLACED BY A ZERO ICOM2047
     1N THE FOLLOWING CARD, CAPACITOR PLOCK)                          COM2048
 260  FORMAT (1H A3,I2,I5,F10.2,I5,F10.2,I5,F10.2,I5,F10.2)           COM2049
 261  FORMAT (1H A5,4F10.2,2I5)                                       COM2050
 262  FORMAT (1H 5HINC  I5,F10.2,I5)                                 COM2051
 263  FORMAT (1H 5HINC  I5,3F10.2,3I5)                                COM2052
 264  FORMAT (1H043H FOLLOWING CARD ILLEGAL IN DATA SUB-BLOCKS /5XA3,I2,COM2053
     112A6)                                                           COM2054
 265  FORMAT (1H0 87H AN ILLEGAL CHARACTER HAS BEEN REPLACED BY A ZERO ICOM2055
     1N THE FOLLOWING CARD, DATA SUB-BLOCK)                           COM2056
 266  FORMAT (1H 5HDEC  I5)                                           COM2057
 267  FORMAT (1H A3,I2,F10.2,F10.2,F10.2,F10.2,F10.2,F10.2)           COM2058
 268  FORMAT (1H A5,F10.2,I5)                                         COM2059
 269  FORMAT (1H0 37H FOLLOWING CARD ILLEGAL IN TIME BLOCK /1H A3,I2,12ACOM2060
     16)                                                              COM2061
 270  FORMAT (1H0 83H AN ILLEGAL CHARACTER HAS BEEN REPLACED BY A ZERO ICOM2062
     1N THE FOLLOWING CARD, TIME BLOCK  /1H A3,I2,F10.2,F10.2,F10.2,F10.COM2063
     22,F10.2,F10.2)                                                  COM2064
      DIMENSION CODE(8)                                               COM2071
      DATA CODE /6HDEC   ,6H      ,6HINC   ,6HPER   ,6HNBK   ,6HNBK   ,6COM2072
     1HRES   ,6HCAP   /                                               COM2073
      DIMENSION IERROR(1)                                             COM2074
      DATA IERROR(1)/32/,NERROR/1/                                    COM2075
      CALL FXPPT(IERROR,NERROR)                                       COM2076
      CALL FSWTON(3,32,33,45)
      CALL FXNPRT(1,32)
      MAXAB = 16000                                                   DIAG0761
      NSW2=1                                                          COMP077
      NGRC=1                                                          COM2078
      NGRC1=1                                                         COM2078
      KGRC=1                                                          COM2078
      KOUNT = 1                                                       COMP078
      MAXT = 0                                                        COMP079
      MAXR = 0                                                        COMP080
      MAXC = 0                                                        COMP081
      MAXP = 0                                                        COMP082
      L = 0                                                           COMP083
      DO 131 I=1,5000                                                 22660831
      KDAT(I)=10                                                      22660832
 131  CONTINUE                                                        22660833
      IK = 1                                                          COMP084
      MAXS=1                                                          COMP085
 132  READ (4, 218) KOD, KOD 1                                        COMP086
      IERROR(1)=IABS(IERROR(1))
      IF (NGRC.EQ.2.AND.NGRC1.EQ.2) NGRC=1                            COM20861
      IF (NGRC.EQ.2) NGRC1=2                                          COM20862
      GO TO (134, 135, 136, 137, 138, 139), KOUNT                     COMP087
```


LOCKHEED
CALIFORNIA COMPANY

TABLE E-2 (Continued)

```
C                                                                     COMP088
C     READ IN INITIAL TEMPERATURES OF ALL NODES                       COMP089
C                                                                     COMP090
  134 GO TO (3,13,9,13,14,13,13,13) ,KOD                              COM2091
    3 READ (4, 200) (I 1(I), A(I), I = 1, KOD 1)                      COMP092
      IF ( IERROR(1).GT.0) GO TO 500                                  COM20921
      WRITE (6, 350) ( I1(I), A(I), I=1, KOD1)                        COM20922
      IERROR(1) =IABS(IERROR(1))                                      COM20923
  500 CONTINUE                                                        COM20924
      DO 7 I = 1, KOD 1                                               COMP093
      J = I 1(I)                                                      COMP094
      IF (J - MAXT) 5, 5, 4                                           COMP095
    4 MAXT = J                                                        COMP096
    5 IF (J.EQ.0) WRITE (6,232) J,A(I)                                COMP097
      IF (J.EQ.0) ERROR=1.                                            COMP0971
    6 P(J) = A(I)                                                     COMP098
      ITEM = J                                                        COMP099
      ATEM = A(I)                                                     COMP100
    7 CONTINUE                                                        COMP101
      GO TO 132                                                       COMP102
    9 READ (4, 201) N 1, A 1, N 2                                     COMP103
      IF (IERROR(1).GT.0) GO TO 501                                   COM21031
      WRITE (6,351) N1,A1,N2                                          COM21032
      IERROR(1) =IABS(IERROR(1))                                      COM21033
  501 CONTINUE                                                        COM21034
      J = ITEM                                                        COMP104
      AA = ATEM                                                       COMP105
      DO 12 I = 1, N 2                                                COMP106
      J = J + N 1                                                     COMP107
      AA = AA + A 1                                                   COMP108
      IF (J.GT.MAXT) MAXT=J                                           COMP109
   11 P(J) = AA                                                       COMP111
      ITEM = J                                                        COMP112
      ATEM = AA                                                       COMP113
   12 CONTINUE                                                        COMP114
      GO TO 132                                                       COMP115
   13 READ (4,210) (A(I), I=1,12)                                     COM21160
      WRITE (6,352) CODE(KOD),KOD1, (A(I),I=1,12)                     COM21161
      ERROR = 2.                                                      COM21162
      IERROR(1)=IABS(IERROR(1))                                       COM21163
      GO TO 132                                                       COM21164
   14 KOUNT = KOUNT + 1                                               COMP118
      TEMPX=0.                                                        22661181
      DO 141 I=1,MAXT                                                 22661182
      IF (KDAT(I).EQ.10) P(I)=TEMPX                                   22661183
      IF (KDAT(I).NE.10) TEMPX=P(I)                                   22661184
  141 CONTINUE                                                        22661185
      READ (4, 200) I 1(1)                                            COMP119
      IERROR(1) =IABS(IERROR(1))                                      COM21191
      WRITE (3) MAXT, NCVG, NSTS                                      COMP120
      WRITE (3) (P(I), I = 1, MAXT)                                   COMP121
      WRITE (3) MAXT, NCVG, NSTS                                      COMP122
      WRITE (3) (P(I), I = 1, MAXT)                                   COMP123
      DO 15 I = 1, 5000                                               COMP124
      P(I) = 0.                                                       COMP125
   15 CONTINUE                                                        COMP126
      WRITE (6,250) MAXT,MAXT                                         COM21261
      MAXIM=MAXT+MAXT                                                 COM21262
      WRITE (6,251) MAXT,MAXIM                                        COM21263
      GO TO 132                                                       COMP127
C                                                                     COMP128
C     READ IN RESISTERS                                               COMP129
C                                                                     COMP130
  135 GO TO (17,16,24,16,30,16,29,16), KOD                            COM21310
   16 READ (4,210) (A(I),I=1,12)                                      22661311
```

TABLE E-2 (Continued)

```
      WRITE (6,353) CODE(KOD),KOD1, (A(I),I=1,12)                          COM21312
      IERROR(1) =IABS(IERROR(1))                                            COM21313
      GO TO 132                                                             COM21314
17    READ (4, 202) (I 1(I), I 2(I), I 3(I), A(I), I = 1, KOD 1)           COMP132
      IF (IERROR(1).GT.0) GO TO 175                                         COM21321
      WRITE (6,2531)                                                        COM21322
      WRITE (6,354) CODE(KOD), KOD1,(I1(I),I2(I),I3(I),A(I),I=1,KOD1)      COM21323
      IERROR(1) =IABS(IERROR(1))                                            COM21324
175   CONTINUE                                                              COM21325
18    DO 22 I=1,KOD1                                                        COM2133
      J = I 1(I)                                                            COMP134
      IF (J.GT.MAXR) MAXR=J                                                 COMP135
19    IF(J.LE.0) WRITE (6,233) I1(I),I2(I),I3(I),A(I)                       COMP137
      IF(J.LE.0) ERROR=1.                                                   COMP1371
20    P(J) = A(I)                                                           COMP138
      L = L + 1                                                             COMP139
      IRLINE(L) = I 3(I) + 4096*(I 2(I) + 4096*J)                           COMP140
      IF (J.GE.2048) IRLINE(L)=-IRLINE(L)                                   COM21401
      ITEM = J                                                              COMP141
      ITEM 2 = I 2(I)                                                       COMP142
      ITEM 3 = I 3(I)                                                       COMP143
      ATEM = A(I)                                                           COMP144
      IF (ITEM2.LE.MAXT.AND.ITEM3.LE.MAXT) GO TO 22                         COM21441
      WRITE (6,234) ITEM,ITEM2,ITEM3,MAXT                                   COM21442
      ERROR=1.                                                              COM21443
22    CONTINUE                                                              COMP145
      GO TO 132                                                             COMP146
25    READ (4,241) I1(1),I2(1),I3(1),X,Y,Z,KRVC,N2                          COM21461
      IF (IERROR(1).GT.0) GO TO 2502                                        COM21462
      WRITE (6,2531)                                                        COM21463
      WRITE (6,356) CODE(3),I1(1),I2(1),I3(1),X,Y,Z,KRVC,N2                 COM21464
      IERROR(1) =IABS(IERROR(1))                                            COM21465
2502  CONTINUE                                                              COM21466
      J=ITEM                                                                COM21462
      DO 2501 I=1,N2                                                        COM21463
      J=J+I1(1)                                                             COM21464
      ITEM2=ITEM2+I2(1)                                                     COM21464
      ITEM3=ITEM3+I3(1)                                                     COM21464
      X1=X1+X                                                               COM21464
      Y1=Y1+Y                                                               COM21464
      Z1=Z1+Z                                                               COM21464
      KKRV=KKRV+KCRV                                                        COM21464
      IF (J.GT.MAXR) MAXR=J                                                 COM21465
      P(J)=X1/(Y1*Z1)                                                       COM21465
      L=L+1                                                                 COM21465
      IRLINE(L)=ITEM3+4096*(ITEM2+4096*J)                                   COM21466
      IF (J.GE.2048) IRLINE(L)=-IRLINE(L)                                   COM21467
      ITEM=J                                                                COM21467
      NSTRT(KGRC)=J+4096*KKRV                                               COM21467
      KGRC=KGRC+1                                                           COM21468
      IF (ITEM2.LE.MAXT.AND.ITEM3.LE.MAXT) GO TO 2501                       COM21469
      WRITE (6,234) ITEM,ITEM2,ITME3,MAXT                                   COM2147
      ERROR=1.                                                              COM21471
2501  CONTINUE                                                              COM21472
      GO TO (2401,25),NGRC                                                  COM21478
2401  READ (4,203) I1(1),I2(1),I3(1),A(1),N2                                COM21479
      IF (IERROR(1).GT.0) GO TO 2402                                        COM21480
      WRITE (6,2531)                                                        COM21481
      WRITE (6,357) CODE(3),I1(1),I2(1),I3(1),A(1),N2                       COM21482
      IERROR(1) =IABS(IERROR(1))                                            COM21482
2402  CONTINUE                                                              COM21483
```

TABLE E-2 (Continued)

```
      J = ITEM                                                          COM21484
      DO 28 I = 1, N 2                                                  COMP149
      J = J + I 1(1)                                                    COMP150
      ITEM 2 = ITEM 2 + I 2(1)                                          COMP151
      ITEM 3 = ITEM 3 + I 3(1)                                          COMP152
      ATEM = ATEM + A(1)                                                COMP153
      IF (J.GT.MAXR) MAXR=J                                             COM2154
26    P(J) = ATEM                                                       COMP156
      L = L + 1                                                         COMP157
      IRLINE(L) = ITEM 3 + 4096*(ITEM 2 + 4096*J)                       COMP158
      IF (J.GE.2048) IRLINE(L)=-IRLINE(L)                               COM21581
      ITEM = J                                                          COMP159
      IF (ITEM2.LE.MAXT.AND.ITEM3.LE.MAXT) GO TO 28                     COM21591
      WRITE (6,234) ITEM,ITEM2,ITEM3,MAXT                               COM21592
      ERROR=1.                                                          COM21593
28    CONTINUE                                                          COMP160
      GO TO 132                                                         COMP161
29    READ (4,240) I1(1),I2(1),I3(1),X,Y,Z,KCRV                         COM21161
      IF (IERROR(1).GT.0) GO TO 2901                                    COM21611
      WRITE (6,2531)                                                    COM21612
      WRITE (6,356) CODE(7),I1(1),I2(1),I3(1),X,Y,Z,KCRV                COM21613
      IERROR(1) =IABS(IERROR(1))                                        COM21614
2901  CONTINUE                                                          COM21615
      A(1)=X/(Y*Z)                                                      COM21162
      NGRC=2                                                            COM21162
      X1=X                                                              COM21162
      Y1=Y                                                              COM21162
      Z1=Z                                                              COM21162
      NSTRT(KGRC)=I1(1)+4096*KCRV                                       COM21163
      KGRC=KGRC+1                                                       COM21164
      KKRV=KCRV                                                         COM21164
      GO TO 18                                                          COM21165
30    KOUNT = KOUNT + 1                                                 COMP162
      READ (4, 202) I 1(1)                                              COMP163
      IERROR(1) =IABS(IERROR(1))                                        COM21631
      WRITE (3) MAXR, NCVG, NSTS                                        COMP164
      WRITE (3) (P(I), I = 1, MAXR)                                     COMP165
      DO 31 I = 1, MAXR                                                 COMP166
      P(I) = 0.                                                         COMP167
31    CONTINUE                                                          COMP168
      MAXIM=MAXIM+MAXR                                                  COM21681
      WRITE (6,252) MAXR,MAXIM                                          COM21682
      GO TO 132                                                         COMP169
C                                                                       COMP170
C     READ IN CAPACITORS                                                COMP171
C                                                                       COMP172
136   GO TO (33,32,46,32,52,32,32,43), KOD                              COM2174
32    READ (4,210) (A(I),I=1,12)                                        COM21741
      WRITE (6,358) CODE(KOD),KOD1, (A(I),I=1,12)                       COM21742
      IERROR(1) =IABS(IERROR(1))                                        COM21743
      GO TO 132                                                         COM21744
33    READ (4, 200) (I 1(I), A(I), I = 1, KOD 1)                        COMP174
      IF (IERROR(1).GT.0) GO TO 3301                                    COM21745
      WRITE (6,259)                                                     COM21746
      WRITE (6,260) CODE(KOD), KOD1, (I1(I),A(I),I=1,KOD1)              COM21747
      IERROR(1) =IABS(IERROR(1))                                        COM21748
3301  CONTINUE                                                          COM21749
      NOP = 0                                                           COMP175
39    DO 42 I = 1, KOD 1                                                COMP176
      J = I 1(I)                                                        COMP177
      IF (J.GT.MAXT) WRITE (6,235) J,MAXT                               COM2178
```

TABLE E-2 (Continued)

```
      IF (J.GT.MAXT) ERROR=1.                                           COM2178
      IF (J.GT.MAXC) MAXC=J                                             COMP178
   35 IF (J.LE.0) WRITE (6,232) J,A(I)                                  COMP180
      IF (J.LE.0) ERROR=1.                                              COMP1801
   36 P(J) = A(I)                                                       COMP181
      ITEM = J                                                          COMP182
      ATEM = A(I)                                                       COMP183
      KPSD(IK)=I1(I)+4096*(KKRV+4096*KKRV1)                             COM2184
      KPSD(IK)=-KPSD(IK)                                                COM21841
      MAXP = MAXP + 1                                                   COMP185
      NODCON=0
      IK = IK + 1                                                       COMP186
      DO 41 K = 1, L                                                    COMP187
      IDAT 1 = IRLINE(K)/16777216                                       COMP188
      IF (IRLINE(K).LT.0) IDAT1=IABS(IDAT1)+2048                        COM21881
      IDAT 2 = MOD(IRLINE(K)/4096, 4096)                                COMP189
      IDAT 3 = MOD(IRLINE(K), 4096)                                     COMP190
      IF (IRLINE(K).LT.0) IDAT2=IABS(IDAT2)                             COM21901
      IF (IRLINE(K).LT.0) IDAT3=IABS(IDAT3)                             COM21901
      IF (I 1(I) - IDAT 2) 38, 37, 38                                   COMP191
   37 DO 3601 I9=1,KGRC                                                 COM2192
      KGR=MOD(NSTRT(I9),4096)                                           COM21921
      KGK=MOD(NSTRT(I9)/4096,4096)                                      COM21921
      IF (KGR.EQ.IDAT1) GO TO 3602                                      COM21922
 3601 CONTINUE                                                          COM21923
      KGK=0                                                             COM21924
 3602 KPSD(IK)=IDAT3+4096*(IDAT1+4096*KGK)                              COM21925
      IF (KGR.EQ.IDAT1) NSTRT(I9)=0                                     COM21926
      MAXP = MAXP + 1                                                   COMP193
      NODCON=1
      IK = IK + 1                                                       COMP194
   38 IF (I 1(I) - IDAT 3) 41, 40, 41                                   COMP195
   40 DO 4001 I9=1,KGRC                                                 COM2196
      KGR=MOD(NSTRT(I9),4096)                                           COM21961
      KGK=MOD(NSTRT(I9)/4096,4096)                                      COM21961
      IF (KGR.EQ.IDAT1) GO TO 4002                                      COM21962
 4001 CONTINUE                                                          COM21963
      KGK=0                                                             COM21964
 4002 KPSD(IK)=IDAT2+4096*(IDAT1+4096*KGK)                              COM21965
      IF (KGR.EQ.IDAT1) NSTRT(I9)=0                                     COM21966
      MAXP = MAXP + 1                                                   COMP197
      NODCON=1
      IK = IK + 1                                                       COMP198
   41 CONTINUE                                                          COMP199
      IF (NODCON.EQ.1) GO TO 42
      ERROR=1.                                                          COMP1991
      WRITE (6,236) I1(I)                                               COMP1992
   42 CONTINUE                                                          COMP200
      IF (NOP) 132, 132, 48                                             COMP201
   43 READ (4,242) I1(1),X,Y,Z,KKRV,KKRV1                               COM22011
      IF (IERROR(1).GT.0) GO TO 4201                                    COM22011
      WRITE (6,259)                                                     COM22012
      WRITE (6,261) CODE(KOD),I1(1),X,Y,Z,KKRV,KKRV1                    COM22013
      IERROR(1) =IABS(IERROR(1))                                        COM22014
 4201 CONTINUE                                                          COM22015
      A(1)=X*Y*Z                                                        COM22012
      NGRC=2                                                            COM22013
      GO TO 39                                                          COM22014
   46 GO TO (47,4701),NGRC                                              COM2202
   47 READ (4,201) N1,A1,N2                                             COM22021
      IF (IERROR(1).GT.0) GO TO 471                                     COM22022
      WRITE (6,358)                                                     COM22023
```

TABLE E-2 (Continued)

```
      WRITE (6,262) N1,A2,N2                                    COM22024
      IERROR(1) =IABS(IERROR(1))                                COM22025
  471 CONTINUE                                                  COM22026
      NOP = NOP + 1                                             COMP203
      KOD 1 = 1                                                 COMP204
      GO TO 50                                                  COMP205
 4701 READ (4,243) N1,X1,Y1,Z1,IK1,IK2,N2                       COM22051
      IF (IERROR(1).GT.0) GO TO 4705                            COM22052
      WRITE (6,358)                                             COM22053
      WRITE (6,263) N1,X1,Y1,Z1,IK1,IK2,N2                      COM22054
      IERROR(1) =IABS(IERROR(1))                                COM22055
 4705 CONTINUE                                                  COM22056
      KOD1=1                                                    COM22052
      GO TO 49                                                  COM22053
 4702 I1(1)=ITEM+N1                                             COM22054
      X=X+X1                                                    COM22055
      Y=Y+Y1                                                    COM22055
      Z=Z+Z1                                                    COM22055
      A(1)=X*Y*Z                                                COM22056
      KKRV=KKRV+IK1                                             COM22056
      KKRV1=KKRV1+IK2                                           COM22056
      GO TO 39                                                  COM22057
   48 IF (NOP - N 2) 49, 132, 132                               COMP206
   49 NOP = NOP + 1                                             COMP207
      GO TO (50,4702),NGRC                                      COM22071
   50 I 1(1) = ITEM + N 1                                       COMP208
      A(1) = ATEM + A 1                                         COMP209
      GO TO 39                                                  COMP210
   52 KOUNT = KOUNT + 1                                         COMP211
      MAXP = MAXP + 1                                           COMP212
      KPSD(MAXP) = 0                                            COMP213
      READ (4, 200) I 1(1)                                      COMP214
      IERROR(1) =IABS(IERROR(1))                                COM22141
      WRITE (3) MAXC, NCVG, NSTS                                COMP215
      WRITE (3) (P(I), I = 1, MAXC)                             COMP216
      DO 53 I = 1, MAXC                                         COMP217
      P(I) = 0.                                                 COMP218
   53 CONTINUE                                                  COMP219
      WRITE (3) MAXC, NCVG, NSTS                                COMP220
      WRITE (3) (P(I), I = 1, MAXC)                             COMP221
      WRITE (3) MAXC, NCVG, NSTS                                COMP222
      WRITE (3) (P(I), I = 1, MAXC)                             COMP223
      WRITE (3) MAXP, NCVG, NSTS                                COMP224
      WRITE (3) (KPSD(I), I = 1, MAXP)                          COMP225
      NWPD = MAXP                                               COMP226
      NRECD = 7                                                 COMP227
      KSP = 2*MAXT + MAXR + 3*MAXC + 1                          COMP228
      KST = KSP + MAXP                                          COMP229
      KST 1 = KST                                               COMP230
      NKSP = KST - 1                                            COMP231
      JJ = 0                                                    COMP232
      N2=0                                                      COM22321
      MAXIM=MAXIM+MAXC                                          COM22322
      WRITE (6,253) MAXC,MAXIM                                  COM22323
      MAXIM=MAXIM+MAXC                                          COM22324
      WRITE (6,254) MAXC,MAXIM                                  COM22325
      MAXIM=MAXIM+MAXC                                          COM22326
      WRITE (6,255) MAXC,MAXIM                                  COM22327
      MAXIM=MAXIM+MAXP                                          COM22328
      WRITE (6,256) MAXP,MAXIM                                  COM22329
      GO TO 132                                                 COMP233
C                                                               COMP234
C     READ IN DATA SUB-BLOCKS                                   COMP235
C                                                               COMP236
  137 GO TO (56,55,66,67,68,55,55,55,691) ,KOD                  COM2237
```


LOCKHEED
CALIFORNIA COMPANY

TABLE E-2 (Continued)

```
 55   READ (4,210) (A(I),I=1,12)                                          COM22371
      WRITE (6,264) CODE(KOD), KOD1, (A(I),I=1,12)                        COM22372
      IERROR(1) =IABS(IERROR(1))                                          COM22373
      GO TO 132                                                           COM22374
 56   IF (KOD 1) 57, 57, 64                                               COMP238
 57   READ (4, 204) NTAB                                                  COMP239
      IF (IERROR(1).GT.0) GO TO 5701                                      COM22391
      WRITE (6,265)                                                       COM22392
      WRITE (6,266) NTAB                                                  COM22393
      IERROR(1) =IABS(IERROR(1))                                          COM22394
 5701 CONTINUE                                                            COM22395
      IF (NTAB) 62, 58, 59                                                COMP240
 58   WRITE (6, 220)                                                      COMP241
      GO TO 132                                                           COMP242
 59   IF (NSW2.EQ.1) GO TO 591                                            COMP243
      WRITE (6,230) NTABO,NTAB                                            COMP2431
      ERROR=1.                                                            COMP2432
      NSW2=1                                                              COMP2433
      KST=KST1+JJ                                                         COMP2434
 591  NSTRT(MAXS)=KST+2**18*NTAB                                          COM22435
      NSW2=2                                                              COMP2436
      NTABO=NTAB                                                          COMP2437
      MAXS=MAXS+1                                                         COMP244
      GO TO 132                                                           COMP246
 62   IF (NSW2.EQ.2) GO TO 621                                            COMP247
      WRITE (6,231) NTABO                                                 COMP2471
      ERROR=1.                                                            COMP2472
 621  NSW2=1                                                              COMP2473
      IF (NTAB+7) 63,61,63                                                COMP2474
 61   KURV 7 = KST 1 + JJ - KST                                           COMP248
 63   KST = KST 1 + JJ                                                    COMP249
      GO TO 132                                                           COMP250
 64   READ (4, 205) (A(I), I = 1, KOD 1)                                  COMP251
      IF (IERROR(1).GT.0) GO TO 6401                                      COM22511
      WRITE (6,265)                                                       COM22512
      WRITE (6,267) CODE(KOD),KOD1,(A(I),I=1,KOD1)                        COM22513
      IERROR(1) =IABS(IERROR(1))                                          COM22514
 6401 CONTINUE                                                            COM22515
 641  DO 65 I = 1, KOD 1                                                  COM2252
      JJ = JJ + 1                                                         COMP253
      P(JJ) = A(I)                                                        COMP254
 65   CONTINUE                                                            COMP255
      IF (N2.GT.0) GO TO 661                                              COM22551
      GO TO 132                                                           COMP256
 66   READ (4,206) AINC,N2                                                COM22561
      IF (IERROR(1).GT.0) GO TO 6601                                      COM22562
      WRITE (6,265)                                                       COM22563
      WRITE (6,268) CODE(KOD),AINC,N2                                     COM22564
      IERROR(1) =IABS(IERROR(1))                                          COM22565
 6601 CONTINUE                                                            COM22566
      A(1)=P(JJ)                                                          COM22562
      KOD1=1                                                              COM22562
 661  A(1)=A(1)+AINC                                                      COM22563
      N2=N2-1                                                             COM22564
      GO TO 641                                                           COM22565
 67   READ (4, 205) A(1)                                                  COMP257
      IF (IERROR(1).GT.0) GO TO 6701                                      COM22571
      WRITE (6,265)                                                       COM22572
      WRITE (6,268) CODE(KOD),A(1)                                        COM22573
      IERROR(1) =IABS(IERROR(1))                                          COM22574
 6701 CONTINUE                                                            COM22575
```

TABLE E-2 (Continued)

```
      P(JJ + 1) = 32767.                                              COMP258
      P(JJ + 2) = A(1)                                                COMP259
      JJ = JJ + 2                                                     COMP260
      GO TO 132                                                       COMP261
   68 KOUNT = KOUNT + 1                                               COMP262
      KST = KST - 1                                                   COMP263
      NWCT = JJ                                                       COMP264
      WRITE (3) NWCT, NCVG, NSTS                                      COMP265
      WRITE (3) (P(I), I = 1, NWCT)                                   COMP266
      NRECD = NRECD + 2                                               COMP267
      READ (4, 205) A(1)                                              COMP268
      IERROR(1) =IABS(IERROR(1))                                      COM22681
      WRITE (3) MAXS, NCVG, NSTS                                      COMP269.
      WRITE (3) (NSTRT(I), I = 1, MAXS)                               COMP270
      NWST = MAXS                                                     COMP271
      JJ = 0                                                          COMP272
      DO 69 I=1,40                                                    COM22721
      P(I)=0.                                                         COM22722
   69 CONTINUE                                                        COM22723
      MAXIM=MAXIM+NWCT                                                COM22724
      WRITE (6,257) NWCT,MAXIM                                        COM22725
      MAXIM=MAXIM+MAXS                                                COM22726
      WRITE (6,258) MAXS,MAXIM,MAXIM                                  COM22727
      IF (MAXIM.LT.MAXMUM) GO TO 132                                  COM22731
      ERROR=1.                                                        COM22732
      WRITE (6,244) MAXMUM,MAXIM                                      COM22733
      IF(MAXIM.LE.16000) WRITE (6,245)                                DIAG2733
      IF(MAXIM.GT.16000) WRITE (6,244) MAXAB,MAXIM                    DIAG2734
      GO TO 132                                                       COM22734
  691 READ (4,204) ICOD                                               COM22731
      KDAT(JJ+1) = 32765                                              COM22732
      KDAT(JJ+2) = ICOD                                               COM22733
      JJ = JJ+2                                                       COM22734
      GO TO 132                                                       COM22735
  138 GO TO (70,73,73,73,72,72) ,KOD                                  COM2274
   73 READ (4,210) (A(I), I=1,12)                                     COM22741
      WRITE (6,269) CODE(KOD),KOD1,(A(I),I=1,12)                      COM22742
      IERROR(1) =IABS(IERROR(1))                                      COM22743
      GO TO 132                                                       COM22744
C70   READ (4, 200) (I1(I),A(I),I=1,KOD1)                             COM2275
   70 READ (4, 205) (       A(I),I=1,KOD1)                            COM2275
      IF (IERROR(1).GT.0) GO TO 7001                                  COM22751
      WRITE (6,270) CODE(KOD), KOD1, (A(I),I=1,KOD1)                  COM22752
      IERROR(1) =IABS(IERROR(1))                                      COM22753
 7001 CONTINUE                                                        COM22754.
      DO 71 I = 1, KOD 1                                              COMP276
C     JJ = I1(I)                                                      COM2277
      JJ = JJ+1                                                       COM2277
      P(JJ+20)=A(I)                                                   COMP278
   71 CONTINUE                                                        COMP279
      GO TO 132                                                       COMP280
   72 KOUNT = KOUNT + 1                                               COMP281
C    THE FOLLOWING 5 CARDS GO WITH THE NON-C CARDS ABOVE
      P(1)=P(23)                                                      COM22810
      P(2)=P(23)                                                      COM22810
      P(3)=P(22)                                                      COM22810
      P(4)=P(21)                                                      COM22810
      IF (P(24).NE.0.) P(7)=P(24)                                     COM22810
      P(11)=P(2)                                                      COM22810
      JJ=16                                                           COM22811
      WRITE (3) JJ, NCVG, NSTS                                        COMP282


      WRITE (3) (P(I), I = 1, JJ)                                     COMP283
      NWTM = JJ                                                       COMP284
      END FILE 3                                                      COMP285
      NRECD=9                                                         COM22851
      REWIND 4                                                        COMP286
  139 REWIND 3                                                        COMP287
      RETURN                                                          COMP288
      END
```

TABLE E-2 (Continued)

```
$IBFTC DIAG11
      SUBROUTINE RESTRT                                               REST0002
C     AS OF 04/27/65
      COMMON P, M, MAXT, MAXR, MAXC, KSP, KST, KURV 7, NCVG, NSTS, NFAB COMP004
      COMMON NABL,NKSP,LMMF,NRMF,NREAD,CC                             COMP005
      COMMON NRECD, NWPD, NWCT, NWST, NWTM                            COMP006
      COMMON /CP2/ERROR,MAXMUM                                        REST006
      COMMON /CPPLT/NPLT,NEND
      DIMENSION A(12), I 1(12), I 2(12), I 3(12), NABL(10), CC(12)    COMP007
      DIMENSION A1(6)                                                 REST007
      DIMENSION CZ(6),CY(6)                                           REST007
      DIMENSION P(14000), IRLINE(1), KPSD(1), NSTRT(1), M(16), KDAT(1) COMP008
      EQUIVALENCE (P(4000), IRLINE(1)), (P(8000), KPSD(1)), (P(13000), COMP009
     1 NSTRT(1)), (P(1), KDAT(1))                                     COMP010
      REAL M, INC, NBK,NET                                            COMP0
      INTEGER CP                                                      COMP011
      DATA ENDF/6H    -1 /                                            COMP011
      DATA NET/6HNET   /                                              COMP011
      DATA DEC/6HDEC   /                                              COMP011
      DATA INC/6HINC   /                                              COMP011
      DATA PER/6HPER   /                                              COMP011
      DATA NBK/6HNBK   /                                              COMP011
      DATA CID/6HCID   /                                              COMP011
      DATA TAP/6HTAP   /                                              COMP011
      DATA COD/6HCOD   /                                              REST011
      DATA ZERO/6H000000/                                             COMP011
      DATA END /6HEND    /                                            REST011
      DATA (CY(I),I=1,6)/2H 1,2H 2,2H 3,2H 4,2H 5,2H 6/,ZERZ/2H00/    REST011
      DATA (CZ(I),I=1,6)/2H01,2H02,2H03,2H04,2H05,2H06/,BLANK/1H /    REST011
      DATA ZERY/2H 0/                                                 REST011
      DIMENSION CODE(4)                                               22660111
      DATA CODE/6HDEC    ,6HINC    ,6HPER    ,6HNBK    /              22660112
  200 FORMAT (5X, 4(I 5, E 10.2))                                     COMP015
  201 FORMAT (5X, I 5, E 10.2, I 5)                                   COMP016
  202 FORMAT (5X, 2(3I 5, E 10.2))                                    COMP017
  203 FORMAT (5X, 3I 5, E 10.2, I 5)                                  COMP018
  204 FORMAT (5X, I 5)                                                COMP019
  205 FORMAT (5X, 6E 10.2)                                            COMP020
  208 FORMAT (1H1,A3,I2,12A6)                                         COMP021
  209 FORMAT (A3,A2,12A6)                                             REST021
  210 FORMAT (5X,12A6)                                                COMP021
  211 FORMAT (5X, A3,I2,12A6)                                         COMP022
  212 FORMAT (1H1,9X,12A6)                                            REST023
  214 FORMAT (29H0 INPUT MUST BE BAD, QUIT JOB )                      COMP024
  215 FORMAT (40H0 KOUNT IS NOT CORRECT IN RESTRT ROUTINE )           COMP025
  216 FORMAT (I 3, A 6)                                               COMP026
  218 FORMAT (I 3, 3X, I 2)                                           COMP027
  219 FORMAT (38H0 KOD IS NOT CORRECT IN RESTRT ROUTINE )             COMP028
  220 FORMAT (26H0TABLE NO. = 0 IS ILLEGAL.)                          COMP029
  222 FORMAT (30H0 TOO MANY WORDS IN DAT BLOCK   )                    COMP030
  223 FORMAT (7H0CODE= A6,12H IS ILLEGAL.)                            COMP021
  224 FORMAT (25H0CANNOT FIND CURVE NAMED A6,17H ON LIBRARY TAPE.)    COMP021
  230 FORMAT (28H0SOMETHING WRONG AFTER TABLEI6,17H AND BEFORE TABLEI6) COMP021
  231 FORMAT (25H0SOMETHING WRONG IN TABLE I6)                        COMP021
  236 FORMAT (15H0BLOCK NUMBER =I6,29H IS ILLEGAL. RESTART DELETED.  ) REST021
  237 FORMAT (14H1END OF CASES.)                                      REST021
  241 FORMAT (5X,E10.0,I5)                                            REST021
  242 FORMAT (5X,2E10.0,I5)                                           REST021
  243 FORMAT (5X,3E10.0,I5)                                           REST021
  244 FORMAT (5X,4E10.0,I5)                                           REST021
  245 FORMAT (5X,5E10.0,I5)                                           REST021
```

TABLE E-2 (Continued)

```
  246  FORMAT (5X,6E10.0,I5)                                                REST021
  280  FORMAT (1H0 44H FOLLOWING CARD OUT OF PLACE IN THIS RESTART)         22660221
  251  FORMAT (1H A3,I2,12A6)                                               22660222
  252  FORMAT (1H0 70H AN ILLEGAL CHARACTER HAS BEEN REPLACED BY ZERO IN 22660223
      1THE FOLLOWING CARD  /5X,5HDEC  I5)                                   22660224
  253  FORMAT (1H0 64H FOLLOWING CARD ILLEGAL IN TEMPERATURE BLOCK OF PRE22660225
      1VIOUS RESTART)                                                       22660226
  254  FORMAT (1H0 88H TERMINATING CARD OF ONE OF THE BLOCKS IN PREVIOUS 22660227
      1RESTART CONTAINS AN ILLEGAL CHARACTER)                               22660228
  255  FORMAT (1H0113H AN ILLEGAL CHARACTER HAS BEEN REPLACED BY A ZERO I22660229
      1N THE FOLLOWING CARD OF THE TEMP BLOCK OF THE PREVIOUS RESTART)     2266023
  256  FORMAT (1H 5X,A3,I2,I5,F10.2,I5)                                     22660231
  257  FORMAT (1H0 61H FOLLOWING CARD ILLEGAL IN RESISTOR BLOCK OF PREVIO22660232
      1US RESTART)                                                          22660233
  258  FORMAT (1H A3,I2,6F10.2)                                             22660234
  259  FORMAT (1H0116HAN ILLEGAL CHARACTER HAS BEEN REPLACED BY A ZERO IN22660235
      1 THE FOLLOWING CARD OF THE RESISTOR BLOCK OF THE PREVIOUS RESTART)22660236
  260  FORMAT (1H0 62H FOLLOWING CARD ILLEGAL IN CAPACITOR BLOCK OF PREVI22660237
      1OUS RESTART)                                                         22660238
  261  FORMAT (1H0112H AN ILLEGAL CHARACTER HAS BEEN REPLACED BY A ZERO I22660239
      IN THE FOLLOWING CARD OF THE CAP BLOCK OF THE PREVIOUS RESTART)      22660240
  262  FORMAT (1H0 57H FOLLOWING CARD ILLEGAL IN TIME BLOCK OF PREVIOUS R22660241
      1ESTART)                                                              22660242
  263  FORMAT (1H0113H AN ILLEGAL CHARACTER HAS BEEN REPLACED BY A ZERO I22660243
      IN THE FOLLOWING CARD OF THE TIME BLOCK OF THE PREVIOUS RESTART)     22660244
  264  FORMAT (1H0115H AN ILLEGAL CHARACTER HAS BEEN REPLACED BY A ZERO I22660245
      1N THE FOLLOWING CARD IN A DATA SUB-BLOCK OF THE PREVIOUS RESTART) 22660246
  265  FORMAT (1H A3,I2,F10.2,I5)                                           22660247
  266  FORMAT (1H A3,I2,2F10.2,I5)                                          22660248
  267  FORMAT (1H A3,I2,3F10.2,I5)                                          22660249
  268  FORMAT (1H A3,I2,4F10.2,I5)                                          22660250
  269  FORMAT (1H A3,I2,5F10.2,I5)                                          22660251
  270  FORMAT (1H A3,I2,5F10.2,I5)                                          22660252
  271  FORMAT (5X,3H***,5X,6HCURVE A6,36H FROM LIBRARY TAPE WILL APPEAR HDIAG0252
      1ERE.)
       DIMENSION IERROR(1)                                                  2266024
       DATA IERROR(1) /32/,NERROR/1/                                        2266025
       CALL FXPPT (IERROR,NERROR)                                           2266026
       CALL FSWTON(3,32,33,45)
       CALL FXNPRT(1,32)
       NRECD=9
       MATLIB =1                                                            CEST032
       LINE=1                                                               COMP032
  149  REWIND 4                                                             COMP032
  139  REWIND 3                                                             COMP288
       DO 310 I = 1, 16                                                     COMP301
       M(I) = 0.0                                                           COMP302
  310  CONTINUE                                                             COMP303
       M(7) = 0.25                                                          COMP304
       MAXP = 0                                                             COMP305
       JJ = 1                                                               COMP306
       DO 75 I = 1, NRECD                                                   COMP307
       READ (3) N 1                                                         COMP308
       MAXP = MAXP + N 1                                                    COMP309
       READ (3) (P(J), J = JJ, MAXP)                                        COMP310
       JJ = MAXP + 1                                                        COMP311
   75  CONTINUE                                                             COMP312
       READ (3) N 1                                                         COMP313
       READ (3) (M(I),I=1,N1)                                               COMP314
       REWIND 3                                                             REST315
       IF (MAXP -MAXMUM) 701,701,81                                         COMP317
   81  WRITE (6, 222)                                                       COMP318
                                                                            COMP322
```

TABLE E-2 (Continued)

```
 100   ERROR=2.                                                        COMP323
 701   IF (MATLIB.EQ.1) READ (5,209) CM,CX,(A(I),I=1,12)              REST325
       CP=-1                                                           REST3251
 1101  DO 1102 I=1,6                                                   REST3251
 1102  IF (CX.EQ.CY(I).OR.CX.EQ.CZ(I)) CP=I                            REST3251
       IF (CX.EQ.BLANK.OR.CX.EQ.ZERZ.OR.CX.EQ.ZERY) CP=0               COMP3251
       IF (CP.GE.0) GO TO 1103                                         REST3251
       WRITE (6,215)                                                   REST3251
       ERROR=1.                                                        REST3251
       CP=0                                                            REST3251
 1103  IF (CM.EQ.END) GO TO 900                                        REST3251
       IF (CM.EQ.TAP) GO TO 150                                        REST3251
       IF (LINE.LT.60) GO TO 7011                                      REST3252
       WRITE (6,212)             (CC(I),I=1,12)                        REST3253
       LINE =1                                                         REST3254
 7011  IF (CM.EQ.CID) GO TO 117                                        REST3255
       WRITE (6,211) CM,CP,(A(I),I=1,12)                               REST3256
       LINE =LINE+1                                                    REST3257
       IF (CM .EQ. DEC) GO TO 702                                      COMP332
       IF (CM .EQ. INC) GO TO 703                                      COMP333
       IF (CM.EQ.COD) GO TO 707                                        REST333
       IF (CM .EQ. PER) GO TO 704                                      COMP334
       IF (CM .EQ. NBK) GO TO 705                                      COMP335
       WRITE (6,223) CM                                                REST336
       ERROR=1.                                                        REST3361
       GO TO 701                                                       REST3362
 117   CMID=CM                                                         REST3363
       ICP=CP                                                          REST3364
       DO 1171 I=1,12                                                  REST3365
       CC(I)=A(I)                                                      REST3366
 1171  CONTINUE                                                        REST3367
       LINE=1                                                          REST3367
       WRITE (6,212)             (CC(I),I=1,12)                        REST3368
       GO TO 701                                                       REST3369
 150   WRITE (6,211) CM,CP,(A(I),I=1,12)                               DIAG3372
       WRITE (6,271) A(1)                                              DIAG3373
       LINE = LINE + 2                                                 DIAG3374
       GO TO 701                                                       REST338
 702   KOD=1                                                           REST3381
       GO TO 706                                                       REST3382
 703   KOD = 2                                                         COMP339
       GO TO 706                                                       COMP340
 704   KOD = 3                                                         COMP341
       GO TO 706                                                       COMP342
 707   KOD = 5                                                         REST3421
       GO TO 706                                                       REST3422
 705   KOD = 4                                                         COMP343
 706   WRITE (4,218) KOD,CP                                            REST347
       WRITE (4, 210) (A(I), I = 1, 12)                                COMP348
       IF (KOD.LT.4) GO TO 701                                         REST349
       IF (KOD.EQ.4) GO TO 708                                         REST3491
       WRITE (6,215)                                                   REST3492
       ERROR=1.                                                        REST3493
 708   END FILE 4                                                      COMP350
       REWIND 4                                                        COMP351
 714   READ (4, 218) KOD, KOD 1                                        COMP352
       IERROR(1) =IABS(IERROR(1))                                      22663521
       GO TO (717,1221,1221,719) ,KOD                                  2266353
 1221  READ (4,210) (A(I),I=1,12)                                      22663531
       WRITE (6,280)                                                   22663532
       WRITE (6,251) CODE(KOD),KOD1,(A(I),I=1,12)                      22663533
       IERROR(1) =IABS(IERROR(1))                                      22663534
       GO TO 714                                                       22663535
 717   IF (KOD 1) 718, 718, 716                                        COMP354
 716   WRITE (6, 250)                                                  COMP355
```

TABLE E-2 (Continued)

```
  250 FORMAT (43H KOUNT HAS BEEN INCLUDED ON BLOCK FLAG CARD )              COMP356
      ERROR=1.                                                              REST357
      GO TO 714                                                             REST3571
  719 REWIND 4                                                              COMP358
      REWIND 3                                                              COMP359
      MAXP = 0                                                              COMP360
      M(1)=M(2)                                                             REST360
      M(11)=M(2)                                                            REST360
      JJ = 1                                                                COMP361
      DO 800 I = 1, NRECD                                                   COMP362
      GO TO (780, 780, 782, 784, 784, 784, 786, 788, 790), I                COMP363
  780 N 1 = MAXT                                                            COMP364
      GO TO 798                                                             COMP365
  782 N 1 = MAXR                                                            COMP366
      GO TO 798                                                             COMP367
  784 N 1 = MAXC                                                            COMP368
      GO TO 798                                                             COMP369
  786 N 1 = NWPD                                                            COMP370
      GO TO 798                                                             COMP371
  788 N 1 = NWCT                                                            COMP372
      GO TO 798                                                             COMP373
  790 N 1 = NWST                                                            COMP374
  798 WRITE (3) N 1, NCVG, NSTS                                             COMP375
      MAXP = MAXP + N 1                                                     COMP376
      WRITE (3) (P(J), J = JJ, MAXP)                                        COMP377
      JJ = MAXP + 1                                                         COMP378
  800 CONTINUE                                                              COMP379
      N 1 = 16                                                              COMP380
      WRITE (3) N 1, NCVG, NSTS                                             COMP381
      WRITE (3) (M(I),I=1,N1)                                               COMP382
      END FILE 3                                                            COMP383
      REWIND 3                                                              COMP384
      RETURN                                                                COMP385
  718 READ (4, 204) NBLK                                                    COMP387
      IF (IERROR(1).GT.0) GO TO 7181                                        22663871
      WRITE (6,252) NBLK                                                    22663872
      IERROR(1) =IABS(IERROR(1))                                            22663873
 7181 CONTINUE                                                              22663874
      IF (NBLK.EQ.11) GO TO 714                                             22663875
      IF (NBLK.GE.1.AND.NBLK.LE.5) GO TO (720,730,740,750,760),NBLK         REST388
      WRITE (6,236) NBLK                                                    REST3881
      ERROR=1.                                                              REST3882
      NBLK =1                                                               REST3883
      GO TO 714                                                             REST3884
  720 READ (4, 218) KOD, KOD 1                                              COMP389
      IERROR(1) =IABS(IERROR(1))                                            22663891
      GO TO (722,725,7201,719), KOD                                         2266390
 7201 READ (4,210) (A(I),I=1,12)                                            22663901
      WRITE (6,253)                                                         22663902
      WRITE (6,251) CODE(KOD), KOD1, (A(I),I=1,12)                          22663903
      GO TO 720                                                             22663904
  722 IF (KOD 1) 721, 721, 723                                              COMP391
  721 READ (4, 204) I                                                       COMP392
      IF (IERROR(1).GT.0) GO TO 7211                                        22663921
      WRITE (6,254)                                                         22663922
      IERROR(1) =IABS(IERROR(1))                                            22663923
 7211 CONTINUE                                                              22663924
      GO TO 714                                                             COMP393
  723 READ (4, 200) (I 1(I), A(I), I = 1, KOD 1)                            COMP394
      IF (IERROR(1).GT.0) GO TO 7231                                        22663941
      WRITE (6,255)                                                         22663942
      WRITE (6,258) CODE(KOD), KOD1, (A(I),I=1,KOD1)                        22663943
```

TABLE E-2 (Continued)

```
      IERROR(1) =IABS(IERROR(1))                                        22663944
 7231 CONTINUE                                                          22663945
      DO 724 I = 1, KOD 1                                               COMP395
      J = I 1(I)                                                        COMP396
      JJ = J + MAXT                                                     COMP397
      P(JJ) = A(I)                                                      COMP398
      P(J) = A(I)                                                       COMP399
      ITEM = I 1(I)                                                     COMP400
      ATEM = A(I)                                                       COMP401
  724 CONTINUE                                                          COMP402
      GO TO 720                                                         COMP403
  725 READ (4, 201) N 1, A 1, N 2                                       COMP404
      IF (IERROR(1).GT.0) GO TO 7251                                    22664041
      WRITE (6,255)                                                     22664042
      WRITE (6,256) CODE(KOD), KOD1, N1, A1,N2                          22664043
      IERROR(1) =IABS(IERROR(1))                                        22664044
 7251 CONTINUE                                                          22664045
      J = ITEM                                                          COMP405
      AA = ATEM                                                         COMP406
      DO 726 I = 1, N 2                                                 COMP407
      J = J + N 1                                                       COMP408
      JJ = J + MAXT                                                     COMP409
      AA = AA + A 1                                                     COMP410
      P(J) = AA                                                         COMP411
      P(JJ) = AA                                                        COMP412
      ITEM = J                                                          COMP413
      ATEM = AA                                                         COMP414
  726 CONTINUE                                                          COMP415
      GO TO 720                                                         COMP416
  730 READ (4, 218) KOD, KOD 1                                          COMP417
      IERROR(1) =IABS(IERROR(1))                                        22664171
      GO TO (731, 734,7301,719), KOD                                    2266418
 7301 READ (4,210)                                                      22664181
      WRITE (6,257)                                                     22664182
      WRITE (6,251) CODE(KOD), KOD1, (A(I),I=1,12)                      22664183
      IERROR(1) =IABS(IERROR(1))                                        22664184
      GO TO 730
  731 IF (KOD 1) 721, 721, 732                                          COMP419
  732 READ (4, 200) (I 1(I), A(I), I = 1, KOD 1)                        COMP420
      IF (IERROR(1).GT.0) GO TO 7321                                    22664201
      WRITE (6,259)                                                     22664202
      WRITE (6,258) CODE(KOD), KOD1, (A(I),I=1,KOD1)                    22664203
      IERROR(1) =IABS(IERROR(1))                                        22664204
 7321 CONTINUE                                                          22664205
      DO 733 I = 1, KOD 1                                               COMP421
      J = I 1(I) + 2*MAXT                                               COMP422
      P(J) = A(I)                                                       COMP423
      ITEM = I 1(I)                                                     COMP424
      ATEM = A(I)                                                       COMP425
  733 CONTINUE                                                          COMP426
      GO TO 730                                                         COMP427
  734 READ (4, 201) N 1, A 1, N 2                                       COMP428
      IF (IERROR(1).GT.0) GO TO 7341                                    22664281
      WRITE (6,259)                                                     22664282
      WRITE (6,256)  CODE(KOD), KOD1, N1,A1,N2                          22664283
      IERROR(1) =IABS(IERROR(1))                                        22664284
 7341 CONTINUE                                                          22664285
      J = ITEM                                                          COMP429
      AA = ATEM                                                         COMP430
      DO 736 I = 1, N 2                                                 COMP431
      J = J + N 1                                                       COMP432
      JJ = J + 2*MAXT                                                   COMP433
      AA = AA + A 1                                                     COMP434
      P(JJ) = AA                                                        COMP435
      ITEM = J                                                          COMP436
```

TABLE E-2 (Continued)

```
      ATEM = AA                                                         COMP437
  736 CONTINUE                                                          COMP438
      GO TO 730                                                         COMP439
  740 READ (4, 218) KOD, KOD 1                                          COMP440
      IERROR(1)=IABS(IERROR(1))                                         22664401
      GO TO (741,744,7401,719), KOD                                     2266441
 7401 READ (4,210) (A(I),I=1,12)                                        22664411
      WRITE (6,260)                                                     22664412
      WRITE (6,251) CODE(KOD), KOD1, (A(I),I=1,12)                      22664413
      IERROR(1) =IABS(IERROR(1))                                        22664414
      GO TO 740                                                         22664415
  741 IF (KOD 1) 721, 721, 742                                          COMP442
  742 READ (4, 200) (I 1(I), A(I), I = 1, KOD 1)                        COMP443
      IF (IERROR(1).GT.0) GO TO 7411                                    22664431
      WRITE (6,261)                                                     22664432
      WRITE (6,258) CODE(KOD), KOD1, (A(I),I=1,KOD1)                    22664433
      IERROR(1) =IABS(IERROR(1))                                        22664434
 7411 CONTINUE                                                          22664435
      DO 743 I = 1, KOD 1                                               COMP444
      J = I 1(I) + 2*MAXT + MAXR                                        COMP445
      P(J) = A(I)                                                       COMP446
      ITEM = I 1(I)                                                     COMP447
      ATEM = A(I)                                                       COMP448
  743 CONTINUE                                                          COMP449
      GO TO 740                                                         COMP450
  744 READ (4, 201) N 1, A 1, N 2                                       COMP451
      IF (IERROR(1).GT.0) GO TO 7441                                    22664511
      WRITE (6,261)                                                     22664512
      WRITE (6,256) CODE(KOD),KOD1,N1,A1,N2                             22664513
      IERROR(1) =IABS(IERROR(1))                                        22664514
 7441 CONTINUE                                                          22664515
      J = ITEM                                                          COMP452
      AA = ATEM                                                         COMP453
      DO 746 I = 1, N 2                                                 COMP454
      J = J + N 1                                                       COMP455
      JJ = J + 2*MAXT + MAXR                                            COMP456
      AA = AA + A 1                                                     COMP457
      P(JJ) = AA                                                        COMP458
      ITEM = J                                                          COMP459
      ATEM = AA                                                         COMP460
  746 CONTINUE                                                          COMP461
      GO TO 740                                                         COMP462
  750 JJ = 5                                                            COMP463
  751 READ (4, 218) KOD, KOD 1                                          COMP464
      IERROR(1) =IABS(IERROR(1))                                        22664641
      GO TO (752,7511,7511,719), KOD                                    2266465
 7511 READ (4,210) (A(I),I=1,12)                                        22664651
      WRITE (6,262)                                                     22664652
      WRITE (6,251) CODE(KOD), KOD1,(A(I),I=1,KOD1)                     22664653
      IERROR(1) =IABS(IERROR(1))                                        22664654
      GO TO 750                                                         22664655
  752 IF (KOD 1) 721, 721, 753                                          COMP466
  753 READ (4, 205) (     A(I),I=1,KOD1)                                REST467
      IF (IERROR(1).GT.0) GO TO 7531                                    22664671
      WRITE (6,263)                                                     22664672
      WRITE (6,258) CODE(KOD), KOD1, (A(I),I=1,KOD1)                    22664673
      IERROR(1) =IABS(IERROR(1))                                        22664674
 7531 CONTINUE                                                          22664675
      DO 754 I = 1, KOD 1                                               COMP468
      JJ = JJ-1                                                         REST469
      IF (JJ.EQ.1) M(7)=A(I)                                            REST4691
```

TABLE E-2 (Continued)

```
      IF (JJ.EQ.1) GO TO 754                                              REST4692
      M(JJ) = A(I)                                                        COMP470
754   CONTINUE                                                            COMP471
      GO TO 751                                                           COMP472
760   READ (4, 218) KOD, KOD 1                                            COMP473
      IERROR(1) =IABS(IERROR(1))                                          22664731
      IF (KOD.EQ.4) GO TO 719                                             22664732
759   READ (4, 204) NTAB                                                  COMP474
      IF (IERROR(1).GT.0) GO TO 7591                                      22664741
      WRITE (6,264)                                                       22664742
      WRITE (6,256) CODE(KOD), KOD1, NTAB                                 22664743
      IERROR(1) =IABS(IERROR(1))                                          22664744
7591  CONTINUE                                                            22664745
      IF (NTAB) 760, 714, 761                                             COMP475
761   L = LOC (NTAB)                                                      REST476
762   READ (4, 218) KOD, KOD 1                                            COMP477
      IERROR(1) =IABS(IERROR(1))                                          22664771
      GO TO (764,770,763,719,773) ,KOD                                    REST478
763   L = L + 1                                                           COMP479
      READ (4, 205) A(1)                                                  COMP480
      IF (IERROR(1).GT.0) GO TO 7631                                      22664801
      WRITE (6,264)                                                       22664802
      WRITE (6,258) CODE(KOD),KOD1, A(1)                                  22664803
      IERROR(1) =IABS(IERROR(1))                                          22664804
7631  CONTINUE                                                            22664805
      P(L) = A(1)                                                         COMP481
      GO TO 762                                                           COMP482
764   IF (KOD 1) 759, 759, 766                                            COMP483
766   READ (4, 205) (A(I), I = 1, KOD 1)                                  COMP484
      IF (IERROR(1).GT.0) GO TO 7661                                      22664841
      WRITE (6,264)                                                       22664842
      WRITE (6,258) CODE(KOD),KOD1, (A(I),I=1,KOD1)                       22664843
      IERROR(1) =IABS(IERROR(1))                                          22664844
7661  CONTINUE                                                            22664845
      KOD2 = KOD1                                                         22664846
      DO 769 I = 1, KOD 1                                                 COMP485
      L = L + 1                                                           COMP486
      P(L) = A(I)                                                         COMP487
769   CONTINUE                                                            COMP488
      GO TO 762                                                           COMP489
770   IF (KOD1.LE.0) KOD1=KOD2                                            REST4890
      GO TO (7701,7702,7703,7704,7705,7706),KOD1
7701  READ (4,241) (A1(I),I=1,KOD1),INCR
      IF (IERROR(1).GT.0) GO TO 771                                       2266500
      WRITE (6,264)                                                       22665001
      WRITE (6,265) CODE(KOD), KOD1, (A(I),I=1,KOD1),INCR                 22665002
      GO TO 7707                                                          22665003
7702  READ (4,242) (A1(I),I=1,KOD1),INCR
      IF (IERROR(1).GT.0) GO TO 771                                       22665011
      WRITE (6,264)                                                       22665012
      WRITE (6,266) CODE (KOD), KOD1, (A(I),I=1,KOD1),INCR                22665013
      GO TO 7707                                                          22665014
7703  READ (4,243) (A1(I),I=1,KOD1),INCR
      IF (IERROR(1).GT.0) GO TO 771                                       2266502
      WRITE (6,264)                                                       22665021
      WRITE (6,267)  CODE(KOD),KOD1,(A(I),I=1,KOD1),INCR                  22665022
      GO TO 7707                                                          22665023
7704  READ (4,244) (A1(I),I=1,KOD1),INCR
      IF (IERROR(1).GT.0) GO TO 771                                       2266502
      WRITE (6,264)                                                       22665021
      WRITE (6,268) CODE(KOD),KOD1,(A(I),I=1,KOD1),INCR                   22665022
```

LOCKHEED CALIFORNIA COMPANY

TABLE E-2 (Continued)

```
      GO TO 7707                                                  22665023
 7705 READ (4,245) (A1(I),I=1,KOD1),INCR
      IF (IERROR(I).GT.0) GO TO 771                               2266503
      WRITE (6,264)                                               22665031
      WRITE (6,269) CODE(KOD),KOD1,(A(I),I=1,KOD1),INCR           22665032
      GO TO 7707                                                  22665033
 7706 READ (4,246) (A1(I),I=1,KOD1),INCR
      IF (IERROR(1).GT.0) GO TO 771                               2266504
      WRITE (6,264)                                               22665041
      WRITE (6,270) CODE(KOD),KOD1,(A(I),I=1,KOD1),INCR           22665042
      GO TO 7707                                                  22665043
 7707 IERROR(1) =IABS(IERROR(I))                                  2266505
 771  DO 772 I=1,KOD1                                             REST4893
      A(I)=A(I)+A1(I)                                             REST4894
      L=L+1                                                       REST4894
      P(L)=A(I)                                                   REST4895
 772  CONTINUE                                                    REST4896
      INCR=INCR-1                                                 REST4897
      IF (INCR.GT.0) GO TO 771                                    REST4898
      GO TO 762                                                   REST4899
 773  READ (4,204) ICOD                                           REST490
      KDAT (L+1) = 32765                                          REST4901
      KDAT (L+2) = ICOD                                           REST4902
      L=L+2                                                       REST4903
      GO TO 762                                                   REST4904
 900  REWIND 3                                                    REST4905
      REWIND 4                                                    REST491
      WRITE (6,237)                                               REST4911
      NEND=1
      RETURN
      END                                                         REST494
```


LOCKHEED
CALIFORNIA COMPANY

REFERENCES

1. Linneman, E.R. — *Computer Program for the Computation of Three-Dimensional Configuration Factors*, Lockheed-California Company, LR 18905, June, 1965

2. *An Introduction to Spacecraft Thermal Control*, Lockheed-California Company, LR 18901, July, 1965

3. McAdams, W.H. — *Heat Transmission*, McGraw-Hill Book Company, Inc., 1954.

4. Nevelli, B.A. — *Computer Program for the Calculation of Incident Orbital Radiant Heat Flux*, Lockheed-California Company, LR 18904, June, 1964

5. Eckert, E.R.G. — *Engineering Relations for Heat Transfer and Friction in a High Velocity Laminar and Turbulent Boundary Layer Flow Over Surfaces with Constant Temperature and Pressure*, ASME Transactions, August, 1956

6. Smith, O. — *A Reference Atmosphere for Patrick Air Force Base, Florida*, NASA TN D-595, March, 1961

7. Krieth, F. — *Principles of Heat Transfer*, International Textbook Company, 1961

8. *U.S. Standard Atmosphere, 1962*, prepared under sponsorship of NASA, USAF, and the U.S. Weather Bureau, U.S. Printing Office, December, 1962

9. Hirasawa, P.S., Josephine Laue, and I. Shuldiner — *Thermal Analyzer Computer Program for the Solution of Fluid Storage and Pressurization Problems*, Lockheed-California Company, LR 18903, July, 1965