Art Edwards

UCRL-14754, Rev. 3

TRUMP: A COMPUTER PROGRAM FOR TRANSIENT AND STEADY-STATE TEMPERATURE DISTRIBUTIONS IN MULTIDIMENSIONAL SYSTEMS

Arthur L. Edwards

September 1, 1972

Prepared for U.S. Atomic Energy Commission under contract No. W-7405-Eng-48

LAWRENCE LIVERMORE LABORATORY

University of California/Livermore

9001050123 891212
PDR WASTE PDC
WM-11

TID-4500, UC-32
Mathematics and Computers

LAWRENCE LIVERMORE LABORATORY
University of California/Livermore, California/94550

UCRL-14754, Rev. 3

TRUMP: A COMPUTER PROGRAM FOR TRANSIENT AND STEADY-STATE TEMPERATURE DISTRIBUTIONS IN MULTIDIMENSIONAL SYSTEMS

Arthur L. Edwards

MS. date: September 1, 1972

Supersedes UCRL-14754 Rev. II, dated July 1, 1969

Changes [illegible]

node → [illegible]

region → [illegible]

[illegible] → implicit

Contents

ABSTRACT 1

I. INTRODUCTION 2

II. USE OF TRUMP 4

A. General 4

B. Definition of Problem 5

1. Equations Solved by TRUMP 5

2. Modification of TRUMP 7

C. Development of Calculational Model 9

1. General 9

2. Unit Systems 11

3. Numbering Systems for Materials, Reactants, and Nodes . . 11

4. Geometrical Symmetry and Scale 12

5. Materials 12

6. Chemical Reactants 13

7. Nodes 14

a. General 14

b. Rectangular Coordinates 14

c. Cylindrical Coordinates 14

d. Bodies of Revolution 15

e. Spherical Coordinates 16

8. Internal Heat Generation 17

9. Initial Conditions and System Constants 18

a. Initial Conditions 18

b. System Constants 18

10. Internal Heat and Reactant Transport 18

a. Internal Thermal Connections 18

b. Mass Flow Connections 21

11. Surface Heat Transport 22

a. External Temperatures 22

b. External Thermal Connections 22

12. Special Functions of System Temperatures 24

a. General 24

b. Linear Combinations of Temperature 24

c. Rates of Temperature Change 25

d. Time Integrals of Temperature 26

13. Properties Dependent on Remote Temperatures 26

14. Criteria for Ending the Problem 27

a. Miscellaneous Limits 27

b. Steady State 27

II. USE OF TRUMP (Continued)

15. Control of Method of Calculation 28
16. Control of Accuracy 29
a. Principle Types of Errors 29
b. Modeling Errors 29
c. Spatial Truncation Errors 29
d. Time Truncation Errors 30
e. Temperature Truncation Errors 31
f. Convergence Errors 31
g. Arithmetic Truncation Errors 32
17. Selection of Output Data 32
a. Numerical Output 32
b. Graphical Output 35
c. Punched Output 36
D. Preparation of Input Data 37
1. Card Input Formats 37
2. Data Deck Organization 37
a. General 37
b. Data Decks 38
c. Problem Name Card 38
d. Block Number Cards 38
e. Data End Card 40
3. Block Item Descriptions 40
a. BLOCK 1. Problem Controls, Limits, and Constants (Required) 44
b. BLOCK 2. Material Properties 47
c. BLOCK 3. Chemical Reactant Properties 50
d. BLOCK 4. Node Descriptions 51
e. BLOCK 5. Internal Thermal Connections 52
f. BLOCK 6. External Thermal Connections. 53
g. BLOCK 7. External Temperatures (Boundary Nodes) . . 54
h. BLOCK 8. Variable Heat-Generation Rates 55
i. BLOCK 9. Initial Conditions and Constant Heat-Generation Rates 56
j. BLOCK 10. Mass-Flow Connections 57
k. BLOCK 11. Graphical Output Controls and Nodes . . . 58
l. BLOCK 12. Nodes with Properties Dependent on Remote Temperatures 59
E. Use of Computers 60
1. General 60
2. Choice of Version and Form of TRUMP Program . . . 60
3. Interruption and Restart Procedures 60

E. Use of Computers (Continued)
4. Control of Input and Output Unit Assignments 61
a. Normal I/O Assignments 61
b. Changes in System I/O Assignments 62
c. Changes in Program I/O Assignments 63
F. Evaluation of Calculation Results 64
1. Diagnosis of Input Errors 64
2. Analysis of Calculation Results 64
3. Requirements for Additional Calculations 65
4. Generalization of Results 66
G. Sample Problems 67
1. Heat Regenerator 67
2. Slab Melt 84
3. Nitrocellulose Cylinder 99
III. METHODS OF SOLUTION 124
A. General Sequence of Calculations 124
1. Data Input and Initialization 124
2. Calculations Made at Each Time Step 124
B. Calculation of Properties, Heat and Mass Balance Data . . . 126
1. Evaluation of Tabulated Properties 126
2. Node Heat and Mass Balance 126
a. Thermal Properties 126
b. Heat Balance Data 128
3. Material Heat Balance 128
4. System Heat Balance 129
C. Phase Change 131
D. Chemical Reaction 133
E. Internal Heat Generation 134
F. Internal Heat and Reactant Transport 135
1. Internal Thermal Connections 135
2. Mass Flow Connections 136
a. Heat Transport 136
b. Reactant Transport 139
G. Surface Heat Transport 140
H. Iterative Solution of Implicit Equations 142
1. Overall Temperature Change Equations 142
2. Iterative Procedure 144
3. Convergence Criterion 146
4. Final Corrections for Regular Nodes 146

III. METHODS OF SOLUTION (Continued)
I. Preparation for Next Time Step 148
1. Reclassification of Nodes 148
2. Control of Time Step 148
3. Estimation of Time Derivatives 150
4. Estimation of Interpolation Factor 152
5. Testing Criteria for Ending Problem 153
IV. THE PROGRAM 154
A. General 154
B. TRUMP Subroutines 156
1. General 156
2. Main Program 156
3. KITTY 157
4. HEART 157
5. THERM 158
6. GEN 158
7. CHEM 159
8. FINK 159
9. FLOW 159
10. SURE 160
11. SPECK 161
12. TALLY 161
13. PLOT 162
14. PRIME 162
15. PATCH 163
16. SEEKI 163
17. SEEK2 164
18. REFER 164
19. TIMER 164
C. System Subroutines Used in TRUMP 165
D. Memory Requirements and Problem Size Limits 166
E. Hardware Requirements 170
F. Machine Time Requirements 171
G. TRUMP Glossary 172
1. General Definitions 172
2. TRUMP Control Cards 172
3. Variables and Parameters 174
4. TRUMP Common Blocks 186
5. TRUMP Subroutines and Cliches 187
6. System Subroutines and Statements 188
7. Changes in TRUMP since May 1, 1968 189

IV. THE PROGRAM (Continued)

H. TRUMP Program Listing 192

ACKNOWLEDGMENTS 256

REFERENCES 257

TRUMP: A COMPUTER PROGRAM FOR TRANSIENT AND STEADY-STATE TEMPERATURE DISTRIBUTIONS IN MULTIDIMENSIONAL SYSTEMS

Abstract

TRUMP solves a general nonlinear parabolic partial differential equation describing flow in various kinds of potential fields, such as fields of temperature, pressure, and electricity and magnetism; simultaneously, it will solve two additional equations representing, in thermal problems, heat production by decomposition of two reactants having rate constants with a general Arrhenius temperature dependence. Steady-state and transient flow in one, two, or three dimensions are considered in geometrical configurations having simple or complex shapes and structures. Problem parameters may vary with spatial positions, time, or primary dependent variables—temperature, pressure, or field strength. Initial conditions may vary with spatial position; and among the criteria that may be specified for ending a problem are upper and lower limits on the size of the primary dependent variable, upper limits on the problem time or on the number of time steps or on the computer time, and attainment of steady state. Solutions may be obtained by use of explicit- or implicit-difference equations, or by an optimized combination of both. TRUMP is primarily written in FORTRAN IV, is subdivided into modular subroutines, and is readily adaptable to most computer systems. The amount of computer time required varies with the user's equipment and needs, but averages from 0.3 to 2 msec per time step for each nodal point and connection between nodal points.

I. Introduction

In research and development related to uses of nuclear energy, a wide variety of problems involving flow in various kinds of potential fields must be solved. Examples include heat flow in temperature fields, mass flow in pressure fields, and current in electrical and magnetic fields; and they often include various types of sources and sinks, and modes of transport other than potential flow. The TRUMP computer program[1] was developed to provide a means of solving such problems by numerical methods on high-speed digital computers.

TRUMP allows solution of a general nonlinear parabolic partial differential equation. Two additional equations, which in thermal problems represent heat production by decomposition of reactants having rate constants with a general Arrhenius temperature dependence, may also be solved simultaneously.

Solutions to steady-state and transient problems may be obtained. Geometric configurations may be quite complex, with flow in one, two or three dimensions; with rectangular, cylindrical, axial, or spherical symmetry; or with arbitrary shape and structure. Initial conditions may vary with spatial position. Material properties, source and sink strengths, boundary conditions, and other problem parameters may vary with spatial position, time, or the primary dependent variable (temperature, pressure, field strength). External sources or sinks, coupled to the system by means of specified boundary conditions, may vary with time. Certain problem parameters at one spatial location may be made to depend on the value of the dependent variable at another spatial location.

Limitations of the program include the number of spatial locations which can be calculated, which may be from several hundred to several thousand in various versions, and the amount of computer time required, which may be from a few seconds up to hours for the largest possible problems.

Input data are efficiently organized to be as simple and compact as possible. Output data include numerical, graphical, and punched card data, with options on frequency and content controlled by the user. Output is organized and labeled to provide the user with maximum information concerning the calculation.

TRUMP has been in use and has been continuously improved and developed since 1965. The language is primarily FORTRAN IV. Some statements and system subroutine calls use the LRLTRAN language employed at LLL. Most of these have equivalent substitutes in other computer systems, or are used in optional parts of the program that are easily removed or modified.

TRUMP has been successfully adapted to a number of computer systems at LLL and elsewhere, including the IBM/7094 (LLL, Bell Labs), the CDC/3600 (LLL, Combustion Engineering), the Univac/1108 (Battelle Northwest, Gulf General Atomics, NBS), the Boroughs/B-5500 (C. F. Braun), the IBM/360 (B. F. Goodrich, NASA Lewis Research Center, Bendix Electrodynamics, Cockerill-Ougee-Providence), the CDC/6600

(LLL, Brookhaven, NUMEC, Bettis Atomic Power Lab), the CDC/6400 (Itek), and the GE/200 (Selas).

The only versions of TRUMP in use now at LLL are TRUMP/6600/STD and TRUMP/6600/DS. Versions for use on the CDC/7600 are being developed.

The report is intended to be a complete users' manual and program description, and supersedes the previous TRUMP report.[1] The TRUMP Thermal Property Data Compilation[2] and the TRUMP/XB2 Thermal Property Data Processing Program[3] are also available. Excellent textbooks, including extensive bibliographies, are available on heat conduction in solids,[4] heat-conduction calculations by finite-difference methods,[5,6] transport phenomena,[7] and radiative transport.[8] The Computer Information Center, LLL, Livermore, provides information and reports on the various computer languages, compiling programs, monitor and time-sharing systems, service routines, and library routines in use at LLL.[9-17]

The results of many TRUMP calculations related to weapons development, space power reactor development, and various Plowshare Projects have been reported in internal LLL documents. Other reports include the thermal effects of the Salmon Experiment, which was an underground nuclear explosion in a salt dome,[18] and a description of the use of TRUMP for calculation of transient laminar fluid flow in porous media (Darcy flow).[19]

Programs have been written to generate geometric input data for the TRUMP program.[20,21,23] A small version of TRUMP is in use on the CHIP system at LLL.[24]

II. Use of TRUMP

A. GENERAL

Use of TRUMP to solve problems can be broadly subdivided into the following steps, each of which will be discussed in detail in the following sections:

(a) Definition of problem: Describing the system geometry, materials, reactants, the equations to be solved, including initial and boundary conditions, modes of heat production and transport, the questions to be answered, and the required accuracy.

(b) Development of calculational model: Determining, if necessary, the thermal analog of a nonthermal problem; simplifying, if required, the problem description; subdivision of the system into discrete nodes; assigning identification numbers to materials, reactants, nodes, and boundary nodes; specifying initial and boundary conditions and modes of heat production for each node, and modes of heat transport between all connected nodes; specifying controls on the method and accuracy of the calculation, the criteria for ending the problem, and the production of output data.

(c) Preparation of input data: punching items for each data block on cards, organizing data blocks into data decks, stacking data decks for multiple-problem computer runs, addition of any necessary TRUMP control cards.

(d) Use of computer facilities: choice of version and form of TRUMP program; submission of job to monitor system; interruption and restart of long problems; card, tape, and output retrieval.

(e) Evaluation of calculation results: checking input data; checking output data; determining cause of any errors, dumps, or unexpected results; checking final results for accuracy and consistency; determining requirements for additional calculations; generalization of results.

B. DEFINITION OF PROBLEM

1. Equations Solved by TRUMP

TRUMP solves a set of simultaneous partial differential transport equations with the four independent variables of spatial coordinates and time, and the three primary dependent variables of temperature and two reactant concentrations. Other dependent variables may include concentration of a phase when phase change is taking place, and several coefficients appearing in the equations which may be functions of time, temperature, and spatial location. For the general three-dimensional case of heat conduction and mass flow, with heat generation and chemical reaction, using standard vector operator notation, the equations are as follows:

$$\rho c \frac{DT}{Dt} = \rho c\left(\frac{\partial T}{\partial t} + \underline{v} \cdot \nabla T\right) = \nabla \cdot k\nabla T + G - \rho Q_a \frac{\partial a}{\partial t} - \rho Q_b \frac{\partial b}{\partial t} \qquad \text{(II.B.1.1)}$$

$$\frac{Da}{Dt} = \frac{\partial a}{\partial t} + \underline{v} \cdot \nabla a = -a^{p_a} \exp\left(Z_a - E_a/RT\right) \qquad \text{(II.B.1.2)}$$

$$\frac{Db}{Dt} = \frac{\partial b}{\partial t} + \underline{v} \cdot \nabla b = -b^{p_b} \exp\left(Z_b - E_b/RT\right), \qquad \text{(II.B.1.3)}$$

where T is temperature in absolute units, t is time, c is specific heat, k is thermal conductivity, $\underline{v}$ is the vector flow velocity, G is heat generation rate per unit volume, a and b are reactant concentrations, p_a and p_b are fraction orders, Q_a and Q_b are heats of reaction, Z_a and Z_b are log collision frequencies, E_a and E_b are activation energies, and R is the gas constant.

Analogous equations are used to describe many types of initial- and boundary-value problems and transport problems.[6,7]

The thermal effect of a phase change, with the phase interface normal to the coordinate direction r, can be expressed as follows:

$$\rho \Delta H_m \left(\frac{\partial r_m}{\partial t}\right) = k_1\left(\frac{\partial T_1}{\partial r}\right)_{r_m} - k_2\left(\frac{\partial T_2}{\partial r}\right)_{r_m} \qquad \text{(II.B.1.4)}$$

$$T_1(r_m) = T_2(r_m) = T_m, \qquad \text{(II.B.1.5)}$$

where ΔH_m is the latent heat of phase change from phase 1 to phase 2, r_m is the location of the interface between phases 1 and 2, and T_m is the temperature at which phase change takes place.

At an interface between materials 1 and 2, normal to the coordinate direction r, the heat balance can be expressed as follows:

$$k_1\left(\frac{\partial T_1}{\partial r}\right)_i = h_i(T_{2,i} - T_{1,i}) = k_2\left(\frac{\partial T_2}{\partial r}\right)_i \qquad \text{(II.B.1.6)}$$

$$h_i = h_{i,o} + h_{i,c}\left[(T_{2,i} - T_{1,i})^2\right]^{p_i/2} + \sigma \mathcal{F}_i (T_{1,i} + T_{2,i})\left(T_{1,i}^2 + T_{2,i}^2\right). \qquad \text{(II.B.1.7)}$$

The expression for interface conductance, h_i, is a general form to include the effects of contact conductance, free and forced convection, and radiation. In Eq. (II.B.1.7), σ is the Stefan-Boltzmann constant, and $\mathcal{F}$ is the overall thermal radiation form factor.

At an external surface normal to the coordinate direction r, the heat balance can be expressed as follows:

$$k\left(\frac{\partial T}{\partial r}\right)_s = U_{s,b}(T_b - T_s), \qquad \text{(II.B.1.8)}$$

where T_b is an external temperature, and the surface conductance $U_{s,b}$ is also calculated by an expression with the form of Eq. (II.B.1.7)

$$U_{s,b} = h_{s,o} + h_{s,c}\left[(T_b - T_s)^2\right]^{p_s/2} + \sigma \mathcal{F}_b (T_s + T_b)\left(T_s^2 + T_b^2\right). \qquad \text{(II.B.1.9)}$$

Extension of the equations above to a general multidimensional Eulerian space, subdivided into discrete elements or nodes, is done as follows:

$$\frac{\partial T_n}{\partial t} = \frac{1}{\rho_n c_n V_n}\left[\sum_k U_{n,k}(T_k - T_n) + \sum_k F_{n,k}(w_{n,k} - w_n) + \sum_b U_{n,b}(T_b - T_n)\right]$$
$$+ \frac{G_n}{\rho_n c_n} - \frac{Q_{a,n}}{c_n}\frac{\partial a_n}{\partial t} - \frac{Q_{b,n}}{c_n}\frac{\partial b_n}{\partial t} \qquad \text{(II.B.1.10)}$$

$$\frac{\partial a_n}{\partial t} = -a_n^{p_a}\exp[Z_{a,n} - E_{a,n}/RT_n] - \frac{1}{\rho_n V_n}\sum_k F_{n,k}(a_{n,k} - a_n) \qquad \text{(II.B.1.11)}$$

$$\frac{\partial b_n}{\partial t} = -b_n^{p_b}\exp(Z_{b,n} - E_{b,n}/RT_n) - \frac{1}{\rho_n V_n}\sum_k F_{n,k}(b_{n,k} - b_n), \qquad \text{(II.B.1.12)}$$

where

$$U_{n,k} = \frac{A_{n,k}}{\dfrac{\Delta r_n}{k_n} + \dfrac{1}{h_{n,k}} + \dfrac{\Delta r_k}{k_k}}, \qquad \text{(II.B.1.13)}$$

and where the summations are over all internal thermal and mass flow connections, and over all surface connections, which each node has. V_n is the node volume. $U_{n,k}$ is the thermal conductance of the connection between nodes n and k with area $A_{n,k}$, distances for heat flow of Δr_n and Δr_k and interface conductance $h_{n,k}$. $F_{n,k}$ is the mass flow

rate between nodes n and k, and $w_{n,k}$ is the enthalpy of the flowing material at the interface. A mass balance is assumed so that the density of node n does not change.

While a node is undergoing phase change, $T_n = T_{m,n}$, the phase change temperature. The left side of Eq. (II.B.1.10) is replaced with

$$-\frac{\Delta H_{m,n}}{c_n}\frac{\partial a_n}{\partial t},$$

where a_n is now the concentration of the low temperature phase. Only a chemical reaction involving concentration b can take place in a node in which phase change takes place.

The actual equations solved in TRUMP are for discrete time steps, obtained from the equations above by substituting $(u' - u)/\Delta t$ for time derivatives of the form $\partial u/\partial t$, where u and u' are values at the beginning and end of the time step Δt, and by use of initial or average values over the time step of all other variable quantities in the equation. The methods used are described in Section III.

The following input parameters may vary with spatial location: ρ, c, k, v, G, Q_a, Q_b, Z_a, Z_b, E_a, E_b, T_m, $\Delta H_{m,n}$, $h_{i,o}$, $h_{i,c}$, p_i, $\mathcal{F}_i$, $h_{s,o}$, $h_{s,c}$, p_s, $\mathcal{F}_{s,b}$, and T_b. At each location, the following parameters may vary with time, with the local temperature, or with a temperature at another specified location: c, k, v, G, Q_a, Q_b, Z_a, Z_b, E_a, E_b, $h_{s,o}$, and $h_{s,c}$. T_b may vary with time at each location. Temperature and time dependencies are generally tabulated, but G may vary exponentially with time, and T_b may vary sinusoidally with time, at each location. Temperature of particular nodes may be made to equal, or be proportional to, linear combinations, including averages and differences, of other temperatures, rates of temperature change, integrals of temperature over time, combinations of these, and a variety of other special functions. The input parameters listed above can then be made functions of the temperatures of these particular nodes, allowing a wide variety of dependencies.

The accuracy with which any given problem can be solved with TRUMP depends on how well the problem can be modeled by the set of equations above, and the fineness of the spatial subdivisions and time steps used in the calculations.

2. Modification of TRUMP

The choices of equations and allowed functionalities for TRUMP were based on the criteria of simplicity, flexibility, generality, and efficient use of computer memory. TRUMP may easily be modified to handle more specialized or more generalized types of problems. For example, parameters could be made arithmetic functions of time and temperatures instead of tabulated functions; different orders and types of chemical reaction could be used; heat capacity and thermal conductivity could be functions of reactant concentrations; the relations necessary to calculate electrical heat generation with temperature-dependent resistivities and specified applied voltages could be added; geometric coordinates could be specified for each node, and the spatial dependence of parameters specified in tabular or arithmetic form; TRUMP could be coupled with

other computer programs to extend them from isothermal to nonisothermal systems with heat transport, or to use their output data as input data for TRUMP.

TRUMP may also be modified to remove excess subroutines and arrays, thus increasing the memory space available for other arrays. Memory requirements and methods of altering TRUMP are discussed in Section IV.D.

C. DEVELOPMENT OF CALCULATIONAL MODEL

1. General

After the problem has been defined, a suitable calculational model must be developed. In problems involving complex geometries, it is desirable to limit the calculation to the smallest possible region, by eliminating areas weakly connected thermally to the region of interest, and by making use of symmetry. Symmetry planes may be represented by insulated surfaces. In addition, it is often possible to simplify the geometry considerably by replacing complex shapes with simpler shapes of equivalent volume and resistance to heat flow. Several geometric models may be tried for the same system to determine the effects on the results.

The system thus chosen for study must be subdivided into regions composed of different materials, each with specified properties, and which may contain specified chemical reactants. If density, thermal conductivity, or heat capacity vary continuously with spatial location in some region, the degree of subdivision into materials with discrete properties depends on the required accuracy in describing those properties. If values of material properties are uncertain, calculations should be made for a range of values, to determine the effect on the results.

Each region may be subdivided into volume elements, or nodes, each of which may have a different initial temperature and weight fractions of chemical reactants, constant or tabulated heat generation rate, and modes of heat transport in relation to other nodes. Each node on a noninsulated surface may have a different boundary condition. Nodes may be of any size and shape, with any number of internal and external heat transport connections. Each node must contain a representative nodal point, which may be located anywhere within or on the surface of the node. Maximum accuracy in transient problems is obtained if the node shapes and nodal point locations are chosen so that lines joining the nodal points of connected nodes are perpendicularly bisected by the connected area. In steady state problems, the solution is independent of the heat capacity associated with each node, so the nodal points may be located anywhere, without loss of accuracy.

The required fineness of subdivision of each material region into nodes depends on the degree to which properties, initial and boundary conditions, heat generation rates, mass flow rates and resulting temperature distributions vary with spatial location, and the required accuracy in fitting the values. In general, less total effort and machine time are required if calculations are made for a crude subdivision, then successively finer subdivisions of the system, until no appreciable difference is found in the results, rather than beginning with excessively fine subdivisions. Often it is apparent from the first calculation which regions require finer subdivisions and which do not.

Small nodes connected to the system by low conductances may be used to obtain interface and surface temperatures, or as artificial means of finding temperature sums,

differences, weighted averages, interpolated or extrapolated values, or derivatives or integrals with respect to time, without interfering with the solution (see Section II.C.12).

A number of computer programs have been written to subdivide systems of various types of geometry into zones or nodes for various types of calculations. One of these, FED[20,21] has been extensively used to generate geometric input data for TRUMP for general axisymmetric systems. Most existing zoning programs can be easily modified to produce the node and connection descriptions required as input data for TRUMP.

Allowable problem size limits for various versions of TRUMP are summarized in Table 1.

The criteria for ending the calculation must be determined, and may include temperatures falling outside a specified range, time reaching a specified limit, number of time steps or amount of machine time reaching specified limits, or attainment of steady state. These criteria are discussed more fully in Section II.C.14.

The user can choose between several options for the type of difference equations and methods of controlling the size of the time step. A choice must be made between the available options for frequency and type of numerical and graphical output produced during and at the end of the problem (see Section II.C.17), to obtain those results necessary to answer the questions asked.

Table 1. Problem size limits.

Item	Data block	Size-limit parameter	Version of TRUMP 6600	Version of TRUMP 6600/DS
Materials	2	M_2	40	50
Reactants	3	M_3	5	0
Nodes	4	M_4	480	600
Internal thermal connections	5	M_5	1200	1750
External thermal connections	6	M_6	125	150
Boundary nodes	7	M_7	20	20
Heat generation tables	8	M_8	100	150
Initial conditions	9	M_4	480	600
Mass flow connections	10	M_{10}	80	0
Temperature vs time plot nodes	11	M_{11}	12	16
Temperature vs time plot times	–	M_1	200	200
Remotely dependent properties	12	M_{12}	480	600
Table lengths	2, 3, 6, 7, 8, 10	M_9	12	12

2. Unit Systems

In general, any consistent set of units may be used for the input data. However, the unit of time must be chosen so that all time steps required in the calculation are in the range from 10^{-10} to 10^{12}, except for the first time step, which is always 10^{-12}. The temperature scale must be specified for problems involving thermal radiation or chemical reaction, by assigning a value to K_T (KT) in BLOCK 1 of the input data. K_T is 1, 2, 3, or 4 for Centigrade, Kelvin, Fahrenheit, or Rankine temperature scales.

The Stefan-Boltzmann thermal radiation constant, σ (SIGMA), has values of 1.355 (10^{-12}) cal/sec-cm^2-°K if K_T is 1 or 2, or 1.73 (10^{-7}) Btu/hr-ft^2-°R if K_T is 3 or 4, or 1.0 if K_T is 5. If any other unit system is used, an appropriate conversion factor for σ must be included in any specified radiation form factors $\mathcal{F}_{i,j}$ (RINT) or $\mathcal{F}_{s,b}$ (RSURE) in BLOCKs 5 and 6 of the input data deck.

The gas constant R has a value of 1.987 cal/g-mole-°K or 1.987 Btu/lb-mole-°R. If any other unit system is used, an appropriate conversion factor must be included in any specified chemical reactant activation energy E_r (ET) in BLOCK 3 of the input data.

Output quantities appearing on printouts will be in the same unit system as input data. However, options are available in input data BLOCK 11 to make transformations on temperature, node location and time scales of CRT plot output. These options may be used to change units, scale, measure from fixed base values, obtain dimensionless ratios or any combination of these factors.

Typical units are given for many quantities in this report, to clarify the meaning or use of the quantities.

3. Numbering Systems for Materials, Reactants, and Nodes

Each material, chemical reactant, system node, and boundary node must be identified by a nonzero integer number, for use in cross-referencing data in different blocks of the input data deck. The numbering system for each group is independent of all others. Within each group, all items must have unique numbers. The only limitation on the size of the numbers is that they must fit within the 5-column fields on the data cards, i.e., all numbers from -9999 to 99999, except zero, are allowed. Numbers need not be in sequence, and the choice of numbers has no effect on the running time of the problem.

In complex geometries, it is often convenient to use a node numbering convention in which different groups of digits in the node number represent coordinates or zones in different directions in the system. See Section II.C.17b for the use of node numbers as CRT plot node locations.

The size of the input data deck can be considerably reduced if groups of nodes or connectors have identical descriptions, or when K_d is 2 or 3, differ only by a constant incremental increase in radius d_r (DRAD). Such sequences can be specified with only one card, or two cards when d_r is incremented, if the nodes involved are numbered in an arithmetic sequence. When d_r is constant the first item in the sequence is described, and the number of additional items, and the incremental change in node numbers specified. When d_r is incremented, the first item in the sequence is described normally. On the next card (following any tables associated with the first item, in BLOCK 6), the

second item in the sequence is described, with a minus sign preceding d_r (DRAD), and the number of additional items in the sequence, and the incremental change in node number specified. This procedure may be used in BLOCKs 4, 5, 6, 8, 9, 10, and 12 of the input data deck.

Materials and chemical reactants may also be assigned names consisting of up to 5 Hollerith characters. These are used in the program only to make identification of the materials and chemical reactants easier in the input card decks and in printed output data.

4. Geometric Symmetry and Scale

A symmetry indicator K_d (KD) may be specified in data BLOCK 1, to control the way in which node volumes and thermal connection areas are calculated from input data in BLOCKs 4, 5, and 6. Details of these calculations are given in Sections II.C.7, II.C.10, and II.C.11 below. The choices of K_d are 1, 2, or 3, for systems with no symmetry center or axis, systems with an axis of symmetry, or systems with a center of symmetry, respectively. Simple examples of these three types are systems in which nodes are bounded by the orthogonal surfaces of a rectangular, cylindrical, or spherical coordinate system, respectively.

A geometric scale factor S_d (SCALE) may also be specified in data BLOCK 1. All linear dimensions specified in BLOCKs 4, 5, 6, and 10 will be multiplied by S_d, so that areas will scale as S_d^2 and volumes as S_d^3.

If different parts of the system are most conveniently described by use of different values of K_d or S_d, the input data may be subdivided into two or more sections. Each section may consist of type A data blocks (see Section II.D.2.d.) preceded by a BLOCK 1 with the desired values of K_d and S_d specified.

5. Materials

Each material must be assigned an identification number N_n (MAT) (see Section II.C.3) and a Hollerith name (AMAT). Values must be specified for density ρ_m (DENS, typical units: g/cm^3); specific heat c_m (CAPT, typical units: cal/g-°C), which may be a tabulated function of time or temperature (TVARC); and thermal conductivity k_m (CONT, typical units: cal/sec-cm-°C), which may also be a tabulated function of time or temperature (TVARK).

All tabulated input quantities in TRUMP (CAPT, CONT, QT, ZT, ET, GT, FLOWT, TEMPB, and HSURT) must be in ascending order of time or temperature. Consecutive values of time or temperature may be identical, however.

Latent heat effects may be described in two ways. The temperature T_m (TMELT) and latent heat ΔH_m (HMELT, typical units: cal/g), may be specified, or the latent heat included as the area below a narrow peak in a table of specific heat vs temperature. Both methods may be used if a material has latent heat effects at more than one temperature.

If a material contains one or two chemical reactants, the identification numbers of the reactants must be specified. This data is specified in BLOCK 2 of the input data deck. Chemical reactants are described separately in BLOCK 3. If HMELT is specified, only one chemical reactant may be specified.

The TRUMP Thermal Property Data Compilation[2] lists densities, transition temperatures, heats of transition, and tables of specific heat and thermal conductivity vs temperature for over 1000 materials. The compilation includes the material data list, in the format required for TRUMP input data, an alphabetical index of material names and descriptions, a list of data sources, a material classification system, and a data quality-indicator system. The entire compilation is available on cards and magnetic tape.

The TRUMP/XB2 computer program[3] was developed to process the material data list. In addition to certain processing options also included in TRUMP, TRUMP/XB2 has procedures for producing graphs of specific heat and thermal conductivity vs temperature on the LRL cd80 plotting system.

6. Chemical Reactants

A chemical reactant is a material component that releases or absorbs heat by decomposition. The rate constant may have a modified Arrhenius temperature dependence.

$$a \xrightarrow{K_a} \text{products} + Q_a \text{ cal/g} \tag{II.C.6.1}$$

$$K_a = \exp\ (Z_a - E_a/RT) \tag{II.C.6.2}$$

$$\frac{\partial a}{\partial t} = -K_a\, a^{p_a} \tag{II.C.6.3}$$

$$\left(\frac{\partial T}{\partial t}\right)_{\text{reaction}} = -\frac{Q_a}{c_v}\left(\frac{\partial a}{\partial t}\right), \tag{II.C.6.4}$$

where R is the gas constant (1.987 cal/g-mole-°K), and p_a (CORD) is the reaction order. T is an absolute temperature. Each chemical reactant must be assigned an identification number N_r (KEM), and may be assigned a Hollerith name (AKEM). Values must be specified for heat of reaction Q_r (QT, typical units: cal/g), log collision frequency Z_r (ZT, typical units: log $\sec^{-1}$), and activation energy E_r (ET, typical units: cal/g-mole-°K), each of which may be a tabulated function of time or temperature (QT vs TVARC, ZT vs TVARZ, and ET vs TVARE). A reaction order p_a (CORD) must also be specified, if p_a is not 1.0. Thus, quite general temperature dependence of the rate constant, and any reaction order, may be obtained; these data are specified in BLOCK 3 of the input data deck. Identification numbers of chemical reactants contained in materials are specified in BLOCK 2. Nodes of materials containing chemical reactants must have initial values of reactant weight fractions specified in BLOCKs 1 or 9, or both, of the data deck (see Section II.D.3).

Simple modifications can be made in the program to obtain autocatalytic reactions, reactions involving two components, and reversible reactions.

It should be noted that the only effect of chemical reaction is release or absorption of heat, and no change is caused in other properties except indirectly as a result of

temperature changes. The program could be modified to make other properties functions of reactant concentration.

7. Nodes

a. General

Each node must be assigned an identification number N_n (NODE) for use in referring to the node in various input data blocks. The identification number of the material of which the node consists (NODMAT) must be specified. The node may be classified "special" by specifying a non-zero value of K_s (KS) (see Section II.C.15).

Three dimensional factors, d_ℓ (DLONG), d_w (DWIDE), and d_r (DRAD), must be specified in input data BLOCK 4 for nodes with finite volume. The volume of the node will be calculated as follows:

$$V_n = \alpha\, d_\ell\, d_w\, d_r^\beta\, S_d^3\,, \qquad \text{(II.C.7.1)}$$

where α is 1, 2π, or 4π, and β is 1, 1, or 2, if K_d (KD) is 1, 2, or 3, respectively. S_d (SCALE) is the scale factor. Any dimensional factors that result in the correct volume according to Eq. (II.C.7.1) may be specified. However, the value of d_r (DRAD) determines the node location used for graphical output when KD is 2 or 3. When KD is 1, the node location is determined by the node number N_n (NODE).

Separate computer programs such as FED[21] may be used to generate the geometric data required as input data for TRUMP. The methods of calculating volumes for several special types of geometries are described below.

b. Rectangular Coordinates

The volume of a node bounded by the orthogonal surfaces of a rectangular coordinate system with axes x, y, and z is as follows:

$$V_n = (x_2 - x_1)(y_2 - y_1)(z_2 - z_1). \qquad \text{(II.C.7.2)}$$

Equation (II.C.7.2) may also be used for nodes with irregular or curved surfaces, if the coordinates represent average values for each surface and are measured in three orthogonal directions. If KD is 1 and the heat flow is two-dimensional or one-dimensional, either one or two, respectively, of the BLOCK 4 dimensional factors d_ℓ, d_w, or d_r may be assigned an arbitrary value, such as 1.

c. Cylindrical Coordinates

The volume of a differential volume element bounded by the orthogonal surfaces of a cylindrical coordinate system with coordinates r, z (linear dimensions), and θ (angle measured in radians), is as follows:

$$\Delta V = r\Delta r\ \Delta z\ \Delta\theta\ . \tag{II.C.7.3}$$

For a finite-volume element, or node, this becomes:

$$V_n = (1/2)\left(r_2^2 - r_1^2\right)(z_2 - z_1)(\theta_2 - \theta_1) \tag{II.C.7.4}$$

or

$$V_n = 2\pi\ \bar{r}\ (r_2 - r_1)(z_2 - z_1)[\ (\theta_2 - \theta_1)/(2\pi)]\ , \tag{II.C.7.5}$$

where

$$\bar{r} = (1/2)(r_2 + r_1). \tag{II.C.7.6}$$

If KD is 2, the BLOCK 4 dimensional factors may conveniently be specified as follows:

$$\left.\begin{aligned} d_r &= \bar{r} = (1/2)(r_2 + r_1) \\ d_w &= (r_2 - r_1) \\ d_\ell &= (z_2 - z_1)(\theta_2 - \theta_1)/(2\pi) \end{aligned}\right\} \tag{II.C.7.7}$$

where

$$1/(2\pi) = 0.15915494.$$

If θ extends from 0 to 2π, d_ℓ becomes $(z_2 - z_1)$. In the case of one-dimensional radial heat flow, the axial length $(z_2 - z_1)$ may arbitrarily be made 1, so that d_ℓ is 1.

d. Bodies of Revolution

The volume of a differential volume element formed by rotating an area ΔA (measured in a plane containing the axis of revolution) through an angle $\Delta\theta$ (measured in radians) about the axis of revolution is as follows:

$$\Delta V = r\ \Delta A\ \Delta\theta\ , \tag{II.C.7.8}$$

where r is the radius of area ΔA from the axis of revolution. For a finite volume element, or node, this becomes:

$$V_n = 2\pi\ \bar{r}\ A\ [\ (\theta_2 - \theta_1)/2\pi]\ , \tag{II.C.7.9}$$

where $\bar{r}$ is the radius of the centroid of area A from the axis. If area A can be expressed as the product of two factors L_1 and L_2, and KD is 2, the BLOCK 4 dimensional factors may conveniently be specified as follows:

$$\left.\begin{aligned} d_r &= \bar{r} \\ d_w &= L_1 \\ d_\ell &= L_2(\theta_2 - \theta_1)/(2\pi), \end{aligned}\right\} \qquad \text{(II.C.7.10)}$$

where

$$1/(2\pi) = 0.15915494.$$

If the body of revolution is complete, i.e., θ extends from 0 to 2π, then d_ℓ becomes L_2.

e. Spherical Coordinates

The volume of a differential volume element bounded by the orthogonal surfaces of a spherical coordinate system with distance from the origin r, angle in the x-y plane θ (measured in radians) and angle from the positive z axis ϕ (also measured in radians), is as follows:

$$\Delta V = r^2 \sin\phi \, \Delta r \, \Delta\theta \, \Delta\phi \, . \qquad \text{(II.C.7.11)}$$

For a finite volume element, or node, this becomes:

$$V_n = (1/3)\left(r_2^3 - r_1^3\right)(\cos\phi_1 - \cos\phi_2)(\theta_2 - \theta_1) \qquad \text{(II.C.7.12)}$$

or

$$V_n = 4\pi(\bar{r})^2(r_2 - r_1)\,[\sin(\bar{\phi})/2]\,(\phi_2 - \phi_1)[(\theta_2 - \theta_1)/(2\pi)], \qquad \text{(II.C.7.13)}$$

where

$$\bar{r} = r_{av}\left\{1 + (1/12)[(r_2 - r_1)/r_{av}]^2\right\}^{1/2} \qquad \text{(II.C.7.14)}$$

$$r_{av} = (1/2)(r_2 + r_1) \qquad \text{(II.C.7.15)}$$

$$\sin\bar{\phi} = \sin(\phi_{av})\,\sin[(\phi_2 - \phi_1)/2]/[(\phi_2 - \phi_1)/2] \qquad \text{(II.C.7.16)}$$

$$\phi_{av} = (1/2)(\phi_2 + \phi_1). \qquad \text{(II.C.7.17)}$$

If $(r_2 - r_1)$ is less than $r_{av}/6$, r_{av} may be substituted for $\bar{r}$ with an error of less than 0.1%. If $(\phi_2 - \phi_1)$ is less than 0.15 radian (8.6°), ϕ_{av} may be substituted for $\bar{\phi}$ with an error of less than 0.1%. For a solid spherical zone of inner radius zero and outer radius r_2, $\bar{r}$ reduces to $r_{av}/\sqrt{3}$, or 0.57735 r_{av}.

If KD is 3, the BLOCK 4 dimensional factors may conveniently be specified as follows:

$$d_r = \begin{cases} \bar{r}, & r_2 - r_1 > r_{av}/6 \\ r_{av}, & r_2 - r_1 < r_{av}/6 \end{cases} \tag{II.C.7.18}$$

$$d_w = r_2 - r_1 \tag{II.C.7.19}$$

$$d_\ell = [(\cos\phi_1 - \cos\phi_2)/2][(\theta_2 - \theta_1)/(2\pi)] \tag{II.C.7.20}$$

or

$$d_\ell = [\sin(\bar{\phi})/2]\,(\phi_2 - \phi_1)[(\theta_2 - \theta_1)/(2\pi)] \tag{II.C.7.21}$$

The expressions for d_ℓ are considerably simplified if θ extends from 0 to 2π, or if ϕ extends from 0 to π. If the node is a complete spherical shell, d_ℓ is 1.

8. Internal Heat Generation

Each node in the system may have a volumetric heat-generation rate g_n (GONE, GG, or GT, typical units: cal/sec-cm^3). This rate may be constant or specified in BLOCKS 1 or 9, or both, of the data deck, or it may be a tabulated function of time or temperature (GT or TVARG), or it may decay exponentially from a specified initial value $g_{o,n}$ (GT) with a half life $t_{g,n}$ (TVARG). Variable heat-generation rates are specified in BLOCK 8 of the data deck. An average heat-generation rate, $\bar{g}_n$, is used for each time step Δt.

In the case of exponential decay, $\bar{g}_n$ is exactly

$$\bar{g}_n = g_{o,r}\, e^{-\gamma t}\,(1 - e^{-\gamma\Delta t})/\gamma\Delta t, \tag{II.C.8.1}$$

where $\gamma = \ln 2/t_{g,n}$. For small values of $\gamma\Delta t(<10^{-5})$, the following equation is used:

$$\bar{g}_n = g_{o,n}\, e^{-\gamma t}\,(1 - \gamma\Delta t/2). \tag{II.C.8.2}$$

The resulting temperature changes in materials with temperature-dependent specific heat will be accurate only if the time steps are sufficiently small to limit the change in specific heat to a small percentage. It may be necessary to specify SMALL in BLOCK 1 to accomplish this.

9. Initial Conditions and System Constants

a. Initial Conditions

Initial temperatures and weight fractions of chemical reactants, if present, and constant heat-generation rates, if any, may be specified in two ways. Constant values of initial temperature T_0 (TONE), weight fractions of two chemical reactants a_0 (ALONE) and b_0 (BONE), and heat generation rate g_0 (GONE) may be specified in BLOCK 1 of the input data deck. These values will be assigned to all nodes subsequently described in a BLOCK 4, unless other values are subsequently specified for individual nodes in BLOCK 9. The corresponding BLOCK 1 values will be substituted for blank input data fields in BLOCK 9. BLOCK 1 may be used more than once in a data deck to specify different sets of constants for use in later data blocks.

b. System Constants

For convenience, several other constants may be specified in BLOCK 1, to be substituted for blank data fields in other data blocks read in after BLOCK 1. These include a constant mass-flow rate w_0 (FONE), an external convection coefficient h_0 (HSURE), an external radiation form factor F_0 (RONE), and an external free-convection exponent P_0 (PONE). These replace blank data fields for FLOWN, HSURE, RSURE, and POWER, respectively. BLOCK 1 may be used more than once in a data deck to specify different sets of constants for use in later data blocks.

10. Internal Heat and Reactant Transport

a. Internal Thermal Connections

(1) General—Each internal thermal connection between nodes must be described in input data BLOCK 5 by specifying the two node numbers N_i (NOD1) and N_j (NOD 2), two connector lengths d_i (DEL1) and d_j (DEL2), and two interface dimensional factors d_ℓ (DLONG) and d_r (DRAD). An interface conductance $h_{i,j}$ (HINT) may be specified, to include the effect of conduction through any surface films and conduction and convection across any space between the connected surfaces. A radiative form factor $\mathcal{F}_{i,j}$ (RINT) may be specified to include the effect of the geometric view factor and surface emissivities for radiative transfer between the connected surfaces. See Section II.C.2 for restrictions on the units of $\mathcal{F}_{i,j}$. If neither $h_{i,j}$ nor $\mathcal{F}_{i,j}$ is specified, the program will use a value of 10^{12} for $h_{i,j}$, in effect equivalent to perfect thermal contact at the interface.

The conductance of the thermal connection $U_{i,j}$ (TRAN, typical units: cal/sec-°C) is calculated as follows:

$$U_{i,j} = A_{i,j} \left[(d_i/\bar{k}_i) + (d_j/\bar{k}_j) + (\bar{H}_{i,j})^{-1} \right]^{-1}, \qquad \text{(II.C.10.1)}$$

where

$$A_{i,j} = \alpha\, d_\ell\, d_r^\beta S_d^2 \tag{II.C.10.2}$$

and

$$\bar{H}_{i,j} = h_{i,j} + \sigma \mathscr{F}_{i,j} (\bar{T}_i + \bar{T}_j)\left(\bar{T}_i^2 + \bar{T}_j^2\right) \tag{II.C.10.3}$$

or

$$\bar{H}_{i,j} = h_{i,j} \left|\bar{T}_j - \bar{T}_j\right|^{|\mathscr{F}_{i,j}|}. \tag{II.C.10.4}$$

In Eq. (II.C.10.1), $\bar{k}_i$ and $\bar{k}_j$ are the estimated average thermal conductivities of nodes N_i and N_j during a time step, evaluated at the estimated average temperature $\bar{T}_i$ and $\bar{T}_j$, respectively (see Sections III.F.1 and III.I.4).

In Eq. (II.C.10.2), α is 1, 2π, or 4π, and β is 1, 1, or 2, if K_d (KD) is 1, 2, or 3, respectively. S_d (SCALE) is the scale factor.

Equation (II.C.10.4) is used in place of Eq. (II.C.10.3) if the specified value of $\mathscr{F}_{i,j}$ is negative. Positive values are used for the temperature difference and exponent. This form of conductance is often used for heat transfer by free convection.

Any number of connections may be specified between any pair of nodes. If an interface conductance has a temperature or time dependency that cannot be modeled as described above, a very thin node may be placed in series between the two nodes. The thin node may be a material with a thermal conductivity tabulated vs temperature or time, to obtain the desired conductance.

Any desired connector lengths and area factors which result in the correct conductance of the connection, according to Eq. (II.C.10.1), may be specified. The values are independent of the dimensional factors specified in input data BLOCK 4 for the nodes. This method allows systems of any shape and with any type of interconnection to be described. Separate computer programs such as FED[21] may be used to generate the geometric input data required as input data for TRUMP.

(2) Selection of Connector Lengths and Interface Areas—The selection of connector lengths depends on the location of the representative point (or nodal point) within each node. These locations may be chosen arbitrarily, but, for greatest accuracy in transient calculations, should usually be at the geometric centers of the nodes. Exceptions are nodes for which the amount of heat flow actually depends on surface temperatures, requiring that the nodal points be located on the surface. This includes nodes with surface connections of any type except a specified flux, and nodes with internal connections involving radiative transport or using the free convection option, Eq. (II.C.10.4). Care must be taken to avoid thermal connections for which $U_{i,j}$ is very large in comparison with the conductances of other thermal connections involving nodes N_i and N_j. This

causes numerical difficulties in calculating the heat balance. Such connections may be eliminated by either lumping the two nodes into a single node or by moving the nodal points further apart, whichever causes the least error.

In general, a thermal connection between two nodes with different thermal conductivities has average distances $\bar{d}_i$ and $\bar{d}_j$ to the interface, and average areas in each node $\bar{A}_i$ and $\bar{A}_j$. In transient problems, the locations of isothermal surfaces may change with time, and the heat flow paths may also change. The estimates of average lengths and areas must be based on some assumption about the direction of heat flow or about the component of heat flow perpendicular to the interface or in the direction of a line or curve connecting the nodal points. If the nodes have the same thermal conductivity, the interface location has no effect on the conductance of the connection, which then depends only on the total heat-flow path length and average area. In terms of the average areas and distances defined above, the conductance of a general connection is as follows:

$$U_{i,j} = \left[\bar{d}_i/\bar{k}_i\,\bar{A}_i + \bar{d}_j/\bar{k}_j\,\bar{A}_j + (\bar{H}_{i,j}\,A_{i,j})^{-1}\right]^{-1}. \qquad \text{(II.C.10.5)}$$

Since a particular interface area $A_{i,j}$ must be specified, the corresponding connector lengths should be specified as follows:

$$d_i = \bar{d}_i\,A_{i,j}/\bar{A}_i\,, \qquad \text{(II.C.10.6)}$$

$$d_j = \bar{d}_j\,A_{i,j}/\bar{A}_j\,. \qquad \text{(II.C.10.7)}$$

In the case of simple radial flow in cylindrical or spherical geometry, the average areas $\bar{A}_i$ and $\bar{A}_j$ should be either log mean or geometric mean values, respectively, between the areas at the nodal points and the area at the interface. If the thermal conductivity of both nodes is the same, $A_{i,j}$ may be the appropriate mean area between the nodal point locations, and d_i and d_j may be the actual radial distances from the nodal points to any arbitrary interface location.

In one-dimensional systems, thermal conductances will be accurately described if all node interfaces are perpendicular to the direction of heat flow, and the appropriate mean areas are used. In two-dimensional or three-dimensional systems, thermal conductances will be accurately described only if all node interfaces are perpendicular to the lines joining the two nodal points and intersect that line at an appropriate mean position (arithmetic mean of non-divergent coordinates, log mean of cylindrical radius, or geometric mean of spherical radius). This may be difficult to achieve but is an ideal to be approximated as closely as practicable.

(3) Connector Lengths and Areas in Simple Coordinate Systems—In rectangular coordinates, connector lengths and areas for thermal connections in the x, y, and z directions are, respectively,

$$\left.\begin{aligned} d_i &= \Delta x_i\,,\ d_j = \Delta x_j,\ A_{i,j} = \Delta y\,\Delta z \\ d_i &= \Delta y_i\,,\ d_j = \Delta y_j,\ A_{i,j} = \Delta z\,\Delta x \\ d_i &= \Delta z_i\,,\ d_j = \Delta z_j,\ A_{i,j} = \Delta x\,\Delta y \end{aligned}\right\} \qquad \text{(II.C.10.8)}$$

In cylindrical coordinates, connector lengths and areas for thermal connections in the r, θ, and z directions are, respectively,

$$\left.\begin{aligned} d_i &= \Delta r_i\,,\ d_j = \Delta r_j\,,\ A_{i,j} = r\,\Delta z\,\Delta\theta \\ d_i &= \bar{r}_i\Delta\theta_i\,,\ d_j = \bar{r}_j\Delta\theta_j,\ A_{i,j} = \Delta r\,\Delta z \\ d_i &= \Delta z_i\,,\ d_j = \Delta z_j\,,\ A_{i,j} = \bar{r}\,\Delta r \end{aligned}\right\} \qquad \text{(II.C.10.9)}$$

where the $\bar{r}$ values are arithmetic means.

In spherical coordinates, connector lengths and areas for thermal connections in the r, θ, and ϕ directions are, respectively,

$$\left.\begin{aligned} d_i &= \Delta r_i\,,\ d_j = \Delta r_j\,,\ A_{i,j} = r^2 \sin\bar{\phi}\Delta\phi\Delta\theta \\ d_i &= r_i \sin\phi_i\Delta\theta_i\,,\ d_j = r_j \sin\phi_j\Delta\theta_j\,,\ A_{i,j} = r_{av}\Delta r\Delta\phi \\ d_i &= r_i\Delta\phi_i\,,\ d_j = r_j\Delta\phi_j\,,\ A_{i,j} = r_{av}\Delta r \sin\phi\Delta\theta \end{aligned}\right\} \qquad \text{(II.C.10.10)}$$

where r_{av} and $\bar{\phi}$ are as defined in Eqs. (II.C.7.15) through (II.C.7.17). The dimensional factors d_ℓ and d_r may be chosen as desired to give the correct interface areas according to Eq. (II.C.10.2).

b. Mass Flow Connections

Heat may be transported through the system by mass flow along any paths connecting nodes of the same material. Each mass-flow connection between a source node N_i (NODF1) and a sink node N_j (NODF2) must have a specified mass-flow rate $F_{i,j}$ that may be constant (FLOWN, typical units: g/sec) or a tabulated function of time or temperature (FLOWT vs TVARFL). In addition, relative distances from the nodal points to the interface, d_i (DELF1) and d_j (DELF2), must be specified and are used to calculate by linear interpolation the heat content of the material flowing through the interface. These data are specified in BLOCK 10 of the input data deck.

Where mass flows into the system, a zero-volume node may be used as the source node. A constant inlet temperature may be obtained by giving the zero-volume source node the desired initial temperature. A time-dependent inlet temperature may be obtained by giving the zero-volume node an external connection to a boundary node with the desired temperature variation with time.

No sink node may be a zero-volume node. Where mass flows out of the system, no connection need be specified. The form of the equations used is equivalent to the assumption that each mass-flow stream entering or leaving a node through a mass-flow connection interface is accompanied by an equal and opposite mass-flow stream at the temperature of the node, maintaining constant mass in the node. Thus, mass effectively enters the system at the temperature of the upstream node of each flow path and leaves at the temperature of the downstream node of each flow path.

The mass flow across each connection carries with it the total sensible and latent heat content of the flowing material, and any reactants contained in the material. The effect on concentrations of low-temperature phases, when phase changes are occurring, depends solely on the net heat balance around each node. Therefore these concentrations are not calculated as a direct mass-transport effect, as are reactant concentrations.

11. Surface Heat Transport

a. External Temperatures

For convenience in specifying boundary conditions that involve heat transfer between the system and its surroundings, external temperatures may be specified. Each such external temperature must be assigned an identification number N_b (NODB) (independent of numbering systems used for materials, reactants, and nodes) and have a specified constant or time-dependent temperature T_b. T_b may be a tabulated function of time (TEMPB vs TIMEB) or vary sinusoidally with time with a specified mean value $T_{b,0}$ (TEMPB(1)), amplitude ΔT_b (TEMPB(2)), period t_b (TIMEB(1)), and phase advance Δt_b (TIMEB(2)), as follows:

$$T_b(t) = T_{b,0} + \Delta T_b \sin\left[2\pi\left(\frac{t + \Delta t_b}{t_b}\right)\right]. \qquad \text{(II.C.11.1)}$$

For convenience, the term "boundary node" is used in this report to refer to an external temperature. These data are specified in BLOCK 7 of the input data.

An alternative way to produce a temperature to be used as an external temperature is by using a node with very large heat capacity, so that its temperature is unaffected by heat flow between it and the system. The node may be assigned an initial temperature, which will remain constant; or may have a specified heat-generation rate g_n (GT), which can be a tabulated function of time; or may decrease or increase exponentially with time (see Section II.C.8). For a heat-flux boundary condition, a node with very small heat capacity could be assigned a constant or variable heat-generation rate and could be thermally connected to the actual surface node.

b. External Thermal Connections

Heat flow between the system and the external environment takes place through external thermal connections. Each thermal connection between a surface node N_s (NODS) and a boundary node N_b (NODSB) must have two specified dimensional factors

d_l (DLONG) and d_r (DRAD), which are used to calculate the surface area $A_{s,b}$ (AREAS) of the connection, as follows:

$$A_{s,b} = \alpha\, d_l\, d_r^3\, S_d^2, \qquad \text{(II.C.11.2)}$$

where α, β, and S_d are defined as in Eq. (IIC 7.1). The factors d_l and d_r are completely independent of dimensional factors specified for the node or any of its internal connection areas. Otherwise, the same comments apply to the use of d_l and d_r as in the case of internal connection areas (see Section II.C.10a).

The overall surface conductance or heat-transfer coefficient for the external connector may include free or forced convection and radiation. A convection coefficient $h_{s,b}$ may be constant (HSURE), or a tabulated function of time or temperature (HSURT vs TVARH). A free convection exponent $P_{s,b}$ (POWER) may be specified. A radiation form factor $\mathcal{F}_{s,b}$ (RSURE) may also be specified (see the comments on $\mathcal{F}_{i,j}$ in Section II. C. 10a). This data is specified in Block 6 of the input data deck. The overall surface conductance is calculated as follows:

$$U_{s,b} = A_{s,b}\left[\bar{h}_{s,b}\left|\bar{T}_s - \bar{T}_b\right|^{P_{s,b}} + \sigma\, \mathcal{F}_{s,b}\left(\bar{T}_s + \bar{T}_b\right)\left(\bar{T}_s^2 + \bar{T}_b^2\right)\right] \qquad \text{(II.C.11.3)}$$

where $\bar{h}_{s,b}$ is an estimated average value for the time step, evaluated at $0.5\,(\bar{T}_s + \bar{T}_b)$ is temperature dependent, and $\bar{T}_s$ and $\bar{T}_b$ are estimated average temperatures of the surface and boundary nodes for the time step (see Section III. B).

A specified variation of surface temperature with time may be obtained by use of a very large convection coefficient $h_{s,b}$ connecting the surface node to a boundary node with the desired temperature variation. For a surface temperature which is constant or changes linearly or exponentially with time, the surface node could be connected with a large interface conductance $h_{i,j}$ (HINT) to a regular node with large heat capacity and with a constant temperature or constant or exponentially changing heat generation.

A specified variation of surface heat flux with time may be obtained by use of a very small convection coefficient $h_{s,b}$ connecting the surface to a boundary node with a very high temperature T_b, and by tabulating $h_{s,b}$ vs time so that the product $h_{s,b}\, T_b$ is equal to the desired surface flux per unit area. An alternative method is to use a thin surface node with a specified internal heat generation rate, to provide the desired heat input at the surface.

To be absolutely correct, surface nodes should be either zero-volume or very thin nodes, or have their representative nodal points on the external surface. This is especially important in the cases of free convection and radiation heat transport, but less important when a surface heat flux is specified.

Where a large number of surface nodes are connected to the same external temperature, and the heat-transfer coefficients $h_{s,b}$ (HSURE) are not tabulated, the

number of external connections can be reduced by replacing them with connections to a zero-volume node that is connected to the boundary node with a very large convection coefficient.

Heat entering or leaving the system by mass flow is considered separately in Section II.C.10b).

12. Special Functions of System Temperatures

a. General

It is often useful or necessary to include in the system nodes whose temperatures are some function of other temperatures in the system, without disturbing the system itself. This may be done passively, to obtain information only, or the temperatures obtained may be used to control values of temperature-dependent variables at various places in the system (see Section II. C.13). Temperatures may be obtained which are equal to any linear combination of other temperatures in the system, including weighted averages, sums and differences. Time derivatives and time integrals of temperatures may be obtained. The variables controlled by these temperatures may be any tabulated functions of temperature. In combination with use of sinusoidal variation of boundary node temperatures and exponential variation of heat generation rates, a wide variety of functions and methods of control may be used. Several of the most useful measuring methods are described below.

The conductance of any thermal connections required between the measurement nodes and the other system nodes should be made small enough so that negligible amounts of heat flow occur. The heat capacities of the measurement nodes can be adjusted so that their time constants, or time lags in reacting to changes in system temperatures, may range from very small to very large times. The measurement nodes can be of any material present in the system or of a material with density and heat capacity values of 1.0, to simplify calculations. Areas of connections can be given values of 1.0, and HINT or HSURE can be used, to obtain any desired total conductance values.

b. Linear Combinations of Temperature

A general linear combination of temperatures may be expressed as follows:

$$T_s = \sum_{n=1}^{N} a_n T_n + a_c T_c, \qquad \text{(II.C.12.1)}$$

where T_s is the temperature of the measuring node, the T_n are node temperatures, T_c is any constant temperature, a_c is either 1 or 0, and a_1 is the largest of the a_n values.

The measuring node, which may be a small special node or a zero-volume node, is connected to nodes 1 through N with total conductances:

$$h_n = h_1 a_n / a_1 \tag{II.C.12.2}$$

and if a_c is 1, to a boundary node with temperature T_c with total conductance:

$$h_c = h_1 / a_1 \tag{II.C.12.3}$$

and if h_0 is not zero, to a boundary node with zero temperature with total conductance:

$$h_0 = h_1 \left(1 - a_c - \sum_{n=1}^{N} a_n\right) \Big/ a_1 . \tag{II.C.12.4}$$

The time constant of the measurement node is given by

$$t_s = C_s a_1 / h_1 \tag{II.C.12.5}$$

where C_s is the total heat capacity of the measurement node. The values of h_1 and C_s are adjusted so that the heat flow from the system into the measurement node is small, and so t_s is within the allowed time lag for the measurement.

For a simple arithmetic average, $a_1 = a_2 = 1/2$, $a_c = 0$, $h_1 = h_2$, $h_c = h_0 = 0$. For a simple temperature difference, $a_1 = 1/2$, $a_2 = -1/2$, $a_c = 0$, $h_2 = -h_1$, $h_c = 0$, and $h_0 = h_1$. For a constant added to a temperature, $a_1 = 1$, $a_c = 1$, $h_c = h_1$, and $h_0 = -h_1$.

c. Rates of Temperature Change

A measurement node may be used to obtain a temperature proportional to the rate of change of temperature in another node:

$$T_s \cong b\left(\frac{dT_1}{dt}\right) \tag{II.C.12.6}$$

This is done by connecting a node n, of heat capacity, C_n, to node 1 with a conductance h_n. If $t_n = C_n / h_n$ is sufficiently small, both nodes will change at about the same rate, given by

$$\frac{dT_n}{dt} = \frac{h_n}{C_n} (T_1 - T_n) \cong \frac{dT_1}{dt} \tag{II.C.12.7}$$

so that

$$T_1 - T_n \cong \frac{C_n}{h_n}\left(\frac{dT_1}{dt}\right) \tag{II.C.12.8}$$

A temperature proportional to this difference is obtained by use of a zero-volume node, connected to node 1, node n, and a boundary node with zero temperature, as described in the preceding section.

$$T_s = a\,(T_1 - T_n) = \frac{aC_n}{h_n}\left(\frac{dT_1}{dt}\right) = b\left(\frac{dT_1}{dt}\right) \qquad \text{(II. C.12.9)}$$

where $a = b\,h_n/C_n$. As before, h_n is small compared with the conductances of other connections of node n, and the conductance between the zero volume node and node n is small compared with h_n. The temperature T_s will be approximately proportional to the time derivative of T_1, smoothed over a time period, and with a time lag error, approximately equal to t_n. This type of measurement is subject to the same instabilities as would occur in analogous mechanical or electrical devices for rate measurements, plus the additional time step truncation error resulting from the use of discrete time steps.

d. Time Integrals of Temperature

The simplest way to obtain a time integral of a function of temperature is to add a node of unit volume, density and specific heat to the system, with an internal heat generation rate G(T) tabulated vs temperature, and controlled by the temperature of the node being measured. The temperature of the measuring node is given by:

$$T_s = T_{s,0} + \int_0^t G(T_1)\,dt, \qquad \text{(II.C.12.10)}$$

where $T_{s,0}$ is the initial temperature of the measuring node, and $G(T_1)$ may be expressed in forms such as aT and $a(T - T_0)$. The temperature T_s may then be used, in turn, to control the value of some other temperature-dependent tabulated function in the system, thus obtaining integral control.

13. Properties Dependent on Remote Temperatures

Any of the properties that may be tabulated functions of temperature can be evaluated at the temperature of nodes other than the nodes at which the properties apply. Such properties include heat capacities, thermal conductivities, heats of reaction, log collision frequencies, activation energies, heat-generation rates, mass-flow rates, and convection coefficients. The identification numbers of the node at which the property applies N_1 (NODP1), and the node whose temperature controls the value of the property N_2 (NODP2), and a property-identification number N_p (NPROP, 1 through 8 for the properties listed above) are specified for each such remote property dependence in BLOCK 12 of the input data. In mathematical form,

$$P_{N_p}\left(\underline{r}_{N_1}\right) = P_{N_p}\left[T\left(\underline{r}_{N_2}\right)\right], \qquad \text{(II.C.13.1)}$$

where $\underline{r}_{N_i}$ is the generalized spatial coordinate of node N_i.

It should be noted that the controlling node, N_2, may be connected in such a way as to have a temperature equal to a weighted average, difference, derivative, or integral of other temperatures in the system. This allows a wide variety of problems to be solved that involve feedback, automatic control, or determination of the required value of some property to obtain a given temperature distribution or variation with time.

14. Criteria for Ending the Problem

a. Miscellaneous Limits

A number of different criteria may be used to end a problem. Any or all may be used. A maximum problem time t_{max} (TIMAX) may be specified. If not specified, t_{max} will be ignored. If t_{max} is negative, the problem will end after the first time step. A maximum temperature T_{max} (TMAX) may be specified. This is most useful in problems involving runaway chemical decomposition. If not specified or if equal to T_{min}, T_{max} will be 10^{12}. A minimum temperature T_{min} (TMIN) may be specified. If not specified, or if equal to or greater than T_{max}, T_{min} will be -10^{12}. The limits T_{max} and T_{min} can be made to apply individually to any node in the system for any specified periods of time. This is done by connecting a zero-volume node to the controlling node, and also to a boundary node with a time-dependent heat transfer coefficient. The connection conductances and boundary node temperature are chosen so that the temperature of the zero-volume node reaches a specified value of T_{min} or T_{max} when the controlling node reaches the desired limiting temperature.

A maximum number of time steps M_{cyc} (MCYC) may be specified. If not specified, M_{cyc} will be ignored. A maximum number of seconds of machine time, M_{sec} (MSEC), may be specified. If not specified, M_{sec} will be ignored. If M_{cyc} or M_{sec} is negative, the problem will end after the first time step.

A monitor control card may specify a maximum amount of machine time. The machine operator may interrupt or end the problems by turning on Sense Switch 1. If the number on the Data End card at the end of the data deck is -2, this will end the problem; otherwise it will be interrupted and can be restarted and continued from restart tape 5A, by means of the ONMON or ONMONS routine.

A "*CHECK" TRUMP control card (columns 1-6) may precede the Problem Name card, to cause the problem to end on completion of the first time step. This option is convenient for checking input data for errors.

b. Steady State

If no other criteria end the problem, and boundary conditions or heat-generation rates do not vary with time, steady state will usually be reached eventually. The following requirements must be satisfied before the program will end the problem because of steady state:

input variable KSPEC must be nonnegative;

the upper time step limit DELTMX must be equal to either DELTO or 10^{12}, whichever is smaller; or, if DELTMX is less than either, the current time step DELT must be equal to DELTMX;

the maximum temperature change DTMAX must have been less than 0.1% of TVARY for two consecutive time steps;

at least ten time steps must have been completed;

no nodes can have been reclassified as special nodes for three consecutive time steps;

and no time steps can have been repeated for three consecutive time steps.

If the user requires only the steady-state solution, the specified average temperature change ΔT_{av} (TVARY) may be made quite large. However, if important heat production, absorption, or transport processes in the system are strongly temperature dependent, ΔT_{av} should be small enough to limit changes in the magnitudes of these processes to small percentages. Otherwise, an oscillatory solution may result, preventing the steady-state criteria from being satisfied. If Δt_{max} (DELTO) is specified, it should be no larger than 10% of the approximate time constant of the system.

15. Control of Method of Calculation

The types of difference equations used for thermal and mass-flow heat transport can be controlled to some extent by the user. Stable difference equations, not subject to a limiting time step, are used for all special nodes, which includes all zero-volume nodes, nodes with external thermal connections, nodes for which the user specifies a nonzero value of KS in data BLOCK 4, and regular nodes reclassified as special nodes during the calculation. No regular nodes can be reclassified as special nodes during the calculation if the user specifies a negative value of KSPEC in data BLOCK 1. If KSPEC is zero or not specified, each regular node will be reclassified as a special node when the time step required to obtain a maximum temperature change to TVARY approaches the stability limit of the node. If KSPEC is positive, all regular nodes will be changed to special nodes before the first time step. If KSPEC is 2, the interpolation factor FOR, which determines the amount of interpolation between initial and final temperature driving forces in the heat-transport equations, will be fixed at 1.0, so that simple backward difference equations will be used. If KSPEC is 3, the interpolation factor will be fixed at 0.5, so that central difference equations will be used. The options with nonzero KSPEC are less accurate and efficient than the option obtained when KSPEC is zero, but allow a more direct comparison with results obtained by other computer programs using simple forward, backward, or central difference equations.

In addition, the user may specify upper and lower limits for the size of the time step, SMALL and DELTO in data BLOCK 1. These may be equal, resulting in use of a fixed time step. The program will ignore the value of SMALL, however, if it is

greater than the stability limit determined for the regular nodes. In most problems, the most efficient calculation is obtained by not specifying KS, KSPEC, SMALL or DELTO.

16. Control of Accuracy

a. Principle Types of Errors

Accuracy is affected by six principle types of errors:

modeling errors, arising from use of inaccurate material and reactant properties, inaccurate initial and boundary conditions, other approximations used in modeling the real system, and interpolation errors in evaluating tabulated functions;

spatial truncation errors, arising from the subdivision of the system into discrete volume elements, or nodes, for which average values of spatially-dependent variables must be estimated, and for which inaccuracies in volumes, areas, and distances may arise;

time truncation errors, arising from the use, in the transient calculation, of discrete time steps for which average values of time-dependent variables must be estimated;

temperature truncation errors, arising from the discrete temperature changes that occur in each node in each time step, for which average values of temperature-dependent variables must be estimated;

convergence errors, arising from the use of an iterative method of solving the heat-transport difference equations for connected special nodes, with arbitrary convergence criteria;

and arithmetic truncation errors, arising from accumulation of round-off errors and from the loss of significant figures that results when numerical values of widely differing magnitudes are added to, or numerical values of similar magnitude are subtracted from, each other.

b. Modeling Errors

Modeling errors can usually be estimated or determined by comparing the results of using different models of the same system. Available material and reactant properties, and assumptions made concerning boundary conditions, are seldom accurate to more than two or three significant figures. There is no justification for attempting to reduce spatial and time-step truncation errors much below modeling errors, except where their effect tends to be accumulative.

c. Spatial Truncation Errors

Spatial truncation errors are controlled by subdividing the system just finely enough so that nonlinear temperature distributions are fitted within required accuracy by

linear interpolation between the nodal points of the nodes, and variation of temperature-dependent properties over the volume of each node is within required limits of the average values determined for the nodal point. This may be difficult to estimate in advance but is simple to determine after a calculation has been made. Often a simplified version of the problem may be calculated to help in estimating the required degree of subdivision in different parts of the system.

In most problems, reasonable accuracy and efficient use of machine time are obtained by subdividing the system into nodes whose time constants (total capacity divided by total conductance of all thermal and mass flow connections) are about 1% of the total time range in which large temperature changes occur at the location of the node in the system. This will vary widely from points near boundaries and interfaces where temperature discontinuities are initially present, to points distant from any discontinuity.

d. Time Truncation Errors

Time truncation errors are usually controlled indirectly, by specification of a desired maximum temperature change for each time step, TVARY. Each time step is adjusted from the previous value, within fixed limits, to try to maintain the maximum temperature change at an average of TVARY, and no more than twice TVARY.

In addition, the time step is adjusted to limit changes in time- and temperature-dependent functions to an average of 1%, and no more than 2% over each time step. The time step is also adjusted to limit the number of iterations required in the heat balance to an average of 40, and no more than 80. Results of time steps for which temperature changes, changes in tabulated functions, or the number of iterations exceed these limits are discarded and the size of the time step halved. Time-dependent quantities are evaluated at an intermediate time in each time step, obtained by adding to the initial time a fraction of the time step between 0.57 and 1.0 (see Section III.I.3).

The user may also specify upper and lower limits, DELTO and SMALL, on the size of the time step. The upper limit may be necessary to prevent a narrow peak in a tabulated function of time from being completely skipped, or to avoid slow convergence of the iteration procedure when approaching steady state, especially in problems involving highly temperature-dependent parameters, or thermal radiation transport. DELTO should be specified as about 0.1 to 1% of the total time in which important temperature changes will occur, if that is known. The lower limit may be necessary to allow time-dependent functions to be fit more closely than would be allowed with the lower limit selected by the program, which is usually 2/3 of 1% of the smallest stability limit of any regular node in the system. The lower limit may also be used to avoid very small time steps when a tabulated parameter undergoes large percentage changes that need not be accurately followed. DELTO and SMALL may be given the same value, to force use of a constant time step, for comparison with another program or for convenience in obtaining output data at exact times.

Instabilities can result from use of too large a time step when the estimated average values of driving forces for the time step are very inaccurate. This is automatically

prevented from occurring in TRUMP in the case of heat transport, by use of stable implicit equations for thermal and mass-flow connections that would otherwise cause instabilities. However, the effect of mass flow on reactant concentrations is not stabilized, and use of a time step larger than the residence time (node mass divided by flow rate) in any node can result in inaccurate values of reactant concentrations. This must be controlled by specifying a value of DELTO smaller than the residence time for any nodes containing reactants.

e. Temperature Truncation Errors

Temperature truncation errors are controlled by specification of TVARY, which controls the size of the time step within the limits of SMALL and DELTO. Temperature-dependent quantities are evaluated at estimated average temperatures for each time step and are obtained by adding to the initial temperature the rate of temperature change estimated from the previous time step, multiplied by a fraction of the time step between 0.57 and 1.0. In most problems, reasonable accuracy and efficient use of machine time are obtained by specifying a value of TVARY of about 0.1% to 1% of the greatest temperature change expected to take place in the system, or such that no more than 1% change occurs in temperature-dependent quantities in the temperature interval TVARY. For problems involving appreciable heat flow by thermal radiation, TVARY should be no more than 5 to 10°C, or an oscillatory solution may result. Average cumulative errors in the heat balance and calculated temperatures will amount to about 1% of TVARY, with individual errors no more than 10%, compared with a calculation of the same node system with very small time steps.

f. Convergence Errors

Convergence errors in the iterative solution for connected special nodes are controlled by the specified value of TVARY. The iteration is stopped when the change in total heat content, divided by the total heat capacity, of all the connected special nodes, from one step of the iteration to the next, is less than $5(10^{-5})$ of TVARY, and when no finite-volume special node has a change in calculated temperature, from one step of the iteration to the next, greater than 10^{-4} TVARY. A limit of 80 iteration steps is also imposed, which, if reached, causes the results to be rejected and the time step to be halved and repeated. TVARY should be no less than about 0.02% of the expected maximum total temperature change in the system. Otherwise, numerical truncation errors might prevent the convergence criterion from being satisfied for any size of time step.

The iteration procedure may converge very slowly, or fail to converge, without obtaining an accurate heat balance or accurate temperature, if the system is too finely subdivided, relative to the required accuracy in space and time. In each iteration, a node is only affected by directly connected nodes, so that the number of iterations required for two nodes to properly interact is proportional to the number of connections in series between them. Because of this method of iteration, convergence may also be slow, or fail, if a single thermal connection between two nodes has very high conductance compared with the conductances of other connections of the nodes, causing

large numerical truncation errors to occur. This can occur, for example, when two very thin layers of high-conductivity material are in contact between poorly conducting material. They may oscillate, or equilibrate with each other, then remain at constant temperature, regardless of the temperature changes in the poor conductors. This can be avoided by lumping any such pair of nodes into a single node. In every problem, the output data list of connection conductances (column headed TRAN) should be checked to see if such connections exist. The output data list of time constants of nodes (column headed SLIM) should also be checked to make sure that unnecessarily small time constants are not being used. Difficulties in convergence of the iteration procedure for special nodes and preservation of the heat balance may result if large groups of connected nodes have time constants much smaller than the time steps used in the calculations (by factors of 10 or 100 or more). Convergence can be improved by use of coarser zoning, or by specifying a much smaller value of TVARY, or by specifying a value of DELTO no larger than 10 to 100 times the average stability limit of the group of nodes, and no larger than 10% of the time constant of the whole system. Zero-volume nodes should be used only where absolutely necessary, as for obtaining special functions of system temperatures (see Section II.C.12) or for determining accurate surface temperatures when an appreciable thermal gradient is expected at the surface.

g. Arithmetic Truncation Errors

Numerical truncation errors arising from accumulated round-off errors are generally not important, in comparison with modeling errors and space, time, and temperature truncation errors. In general, their effect is in one or two units in the last significant figure of any quantity for each time step. After 1000 time steps, the accumulated error would appear in the seventh significant figure on the CDC-6600.

Truncation errors arising from algebraic addition of terms which are nearly equal but have opposite signs, or which have widely differing orders of magnitude, can be quite serious. Net heat-flow values and average flux rates for nodes, especially zero-volume nodes, can be affected by this type of error. These errors can be found by comparing the net heat-flow values for a node with the net heat flow across all thermal connection of the node. These may represent large percentage errors, but very small absolute errors, and have no effect on calculations of temperature or heat content.

Convergence of the iterative scheme for special nodes can be affected by this type of error, if any node has conductances of widely differing magnitude. This is why extremely high conductances should be avoided, where possible, by lumping nodes together, or by modeling very thin layers or gas gaps as interface conductances.

17. Selection of Output Data

a. Numerical Output

The user has several options controlling the amount and form of numerical output data. It may be written on a specified output tape (see Section II.E.4), which is normally

3, but may be 100 to obtain output on the dd80 instead of the radiation printer. The dd80 output is most convenient for making copies and for inclusion in reports.

All input data—and several derived quantities such as node volumes, connection areas and table slopes, and comments concerning special input options or input errors—are written out. The results of the first, second, and last time steps are always written out. In addition, data are written out at problem time intervals of TIMEP and time-step intervals of IPRINT. The type of data written out is controlled by the input value of KDATA. The following are included in each printout:

problem name (NAME) and number (NPROB);
initial clock time (CLOCKB) and machine letter and date (CLOCKA);
printout number (NPRINT);
time-step number (KCYC);
time-step lower-limit counter (MF) and upper-limit counter (MSS);
problem-end sentinel (KWIT);
time-step upper limit (DELTMX) and lower limit (SMALL);
temperature-change control (TVARY);
maximum temperature change (DTEMP);
time-step control (DTMAXS);
number of special node heat-balance iterations (NUTS);
total time (SUMTIM) and last time step (DELT);
net heat flow into the system (FLUX), resulting average temperature change (TEMPER), and average rates of heat flow (FX) and temperature change (TX);
average system temperature (TEMPAD);
total system heat capacity (CAPS) and heat content (HEAT);
total system heat-generation rate (GS), amount of heat generated (GENS), and resulting average temperature change (TEMPLE);
total number of special node iterations (NUTSUM);
average node iterations per time step (NUTAVG) and the maximum in any time step (NUTX);
and time-step interpolation factor (FOR).

The temperature table includes only node numbers and temperature, if KDATA is negative. If KDATA is nonnegative, the following data are printed out for each node: temperature (T), temperature change in the time step (DT), estimated rate of temperature change in the next time step (DDT), heat generation rate ($\bar{g}$, e.g., cal/sec), total heat content (W, e.g., cal), change in heat content from the beginning of the problem (H), and net heat transported into the node by internal and external thermal connections (F, e.g., cal). If phase change or chemical reaction, or both, are included in the problem, the following data are printed out for just those nodes with nonzero concentrations: temperature (T), weight fractions (A and B), changes in the time step (DA and DB), and estimated rates of concentration change in the next time step (DDA and DDB).

The following data are printed out for each boundary node: temperature (TEMPB), net heat flow into the system from the boundary node (HEAT FLOW, e.g., cal), and average rate of heat flow (AVG RATE, e.g., cal/sec) during the problem.

A node (NUM) may be specified in data BLOCK 1, for which T, DDT, A, B, G, and SUMTIM will be printed out at every time step.

Additional data are printed out only on the first and last time steps, and for other time steps if KDATA is positive. The following data are printed out for each material: the number of nodes (NODMS), density (DENS), total volume (VOLMS), total mass (HEFTMS), total heat capacity (CAPMS, e.g., cal/°C), total heat content (WMS, e.g., cal), and average temperature (TMS).

The following data are printed out for each node: material number (NODMAT), type (NTYPE), radius (RADIUS, DRAD*SCALE, BLOCK 4), volume (VOLS), mass (HEFT), heat capacity (CAP, e.g., cal/°C), thermal conductivity (CON, e.g., cal/sec-cm-°C), and time constant (SLIM, limiting stable time step usable with a forward difference equation).

If chemical reaction is included in the problem, the following data are also printed out for each node containing a reactant: mass (HEFT), heats of reaction (QA and QB, e.g., cal), log collision frequencies (ZA and ZB), and activation energies (EA and EB, e.g., cal/g mole-°K).

If mass-flow is included in the problem, the following data are printed out for each node having a mass-flow connection: net flow rate in (FLINT, e.g., g/sec), net flow rate out (FLOUT, e.g., g/sec), net flow in (FLIPS, e.g., grams), net flow out (FLOPS, e.g., grams) node mass (HEFTS), average residence time (DFLOT = HEFT/FLINT), and average flow rates in and out between TAU and SUMTIM. For each mass-flow connection, the following data are printed out: flow (FLOWN, e.g., g/sec), net flow (FLAPS, e.g., grams), average flow rate between TAU and SUMTIM, source node weight factor (DELF1), and sink node weight factor (DELF2). The latter two are proportional to input values DELF2 and DELF1, respectively, and are used to estimate the interface enthalpy.

The following data are printed out for each internal thermal connection: area (AREA), interfacial conductance (HINT, e.g., cal/sec-cm^2-°C), interfacial radiation form factor (RINT), overall conductance (TRAN, e.g., cal/sec-°C), net heat flow since the beginning of the problem (FI, e.g., cal) into the first node from the second node, and the average rate of heat flow during the problem.

The following data are printed out for each external thermal connection: area (AREAS), heat-transfer coefficient (HSURE, e.g., cal/sec-cm^2-°C), free convection exponent (POWER), radiation form factor (RSURE), overall conductance (TRANS, e.g., cal/sec-°C), net heat flow since the beginning of the problem (FS, e.g., cal) into the surface node from the boundary node, and average rate of heat flow since the beginning of the problem. The sum of the latter two values for all external connections is also given.

During the problem, statements are printed out whenever a regular node is changed to a special node, whenever the iteration procedure for special nodes fails to

converge in the allowed maximum number of iterations, and whenever a time step has to be repeated. In the latter case, the following data are printed out: the time step number (KCYC), the size of the unsuccessful time step (DELT), the problem time at the beginning of the time step (SUMTIM), the factor indicating iteration-convergence failure (DTMAXS = 80.0 × TVARY) or too large a change in a tabulated function (DTMAXS = % change × TVARY), and the maximum temperature change ΔT_{max} (DTEMP).

After the last time step, the following additional data are written out: the problem-end sentinel (KWIT) and a table of values of its meanings, and the elapsed machine time, in seconds, since the beginning of the problem.

For each problem, the following data are printed out on-line: the problem name and initial clock time, the total, average, and maximum number of iterations (NUTSUM, NUTAVG, and NUTX), the problem number (NPROB), number of time steps used (KCYC), final problem time (SUMTIM) and problem-end indicator (KWIT), elapsed machine time—in seconds—since the beginning of the problem, and the total heat content of the system (HEAT) and net flow into the system (FLUX).

In addition, statements are printed out on-line and on the regular output tape that are related to restart from tape, interruption by Sense Switch 1, occurrence of various input errors, use of data from a preceding problem when data decks are stacked, failure of convergence of the iteration procedure used for special node heat balance calculations, repetition of time steps that are rejected, and reclassification of nodes as special nodes.

b. Graphical Output

(1) General—Graphical output is produced by the TRUMP program by use of the dd80 library subroutines and plotting system. Input data controlling the graphical output are specified in data BLOCK 11. If an "*OUTPUT1ØØ" TRUMP control card has been read in, all numerical and graphical output will be produced by the dd80 system. Each plot produced will be accompanied by a table containing the numerical coordinates of all the points plotted. Both plots and tables are identified with the problem name (NAME), problem number (NPROB), initial clock time and date, plot number (NPLOT), and problem time (SUMTIM).

Plots are obtained by specifying a nonzero value of JPIC, or one or more values of NODEP, or both, in BLOCK 11. Several options are available as to the types of plots, scales, interpolation, and other options, as described below.

(2) Temperature vs Time—Plots of temperature vs time will be made for the nodes listed in the array NODEP, with points for times corresponding to the first and last time steps and every JPIC'th time step in between, and at problem time intervals of TIMEP, up to a maximum number depending on the value of the parameter M1 in the version of TRUMP in use.

(3) Temperature vs Node Location—If JPIC is preceded by a minus sign, plots of temperature vs node location (node number, if KD = 1, or DRAD*SCALE, if KD = 2 or 3) will be made on the first, last, and every JPIC'th time step, and at problem time intervals of TIMEP, for all the nodes in the system, and a final plot of maximum and minimum node temperature vs node location will be made after the last time step.

(4) Node Location vs Time—If 100 is added to LOGTIM, a plot of node location vs time will be made for the nodes in array NODEP, for about 20 to 40 isotherms between the minimum and maximum temperatures. The list of nodes in array NODEP must be in proper sequence of node location, for the plot to be meaningful.

(5) Plotting Options—Scales on the plots may be linear or logarithmic, with the maximum number of cycles in the latter case specified by LOGR, LOGTEM, and LOGTIM for the node location, temperature, and time scales, respectively. Separate plots for both linear and logarithmic scales may be obtained by preceding the input values of LOGR, LOGTEM, or LOGTIM, or all three, by minus signs. On the temperature vs node location plot, straight-line segments will be interpolated between the plotted points, in the order the nodes are listed in BLOCK 4, only if 100 is added to LOGR. All plots will have solid grid lines, unless 100 is added to LOGTEM, in which case the scales will be marked with tic marks.

(6) Linear Transformation of Scales—Node locations, temperatures, and times may be altered before plotting, by a linear transformation of the form

$$S' = F_1 + F_2 S \qquad \text{(II.C.17.1)}$$

for the purpose of conversion to a different unit system, normalizing, or referring to a constant base value, or all three options. The input values corresponding to F_1 and F_2 are FRAD1 and FRAD2 for node location, FTEMP1 and FTEMP2 for temperature, and FTIME1 and FTIME2 for time. If not specified, the latter of each pair of factors will be set equal to 1.0.

c. Punched Output

A set of punched cards containing the final values of node temperatures, reactant concentrations, and constant heat-generation rates, in the same format as the input BLOCK 9 cards, may be obtained when a problem is ended or interrupted. This is accomplished by specifying a nonzero value of NPUNCH in BLOCK 1 of the input data. The new BLOCK 9 may be used to replace the BLOCK 9 in the original data deck, as an alternative way to restart an interrupted problem or to specify the initial conditions for one or more new problems.

D. PREPARATION OF INPUT DATA

1. Card Input Formats

All input data for TRUMP are in Hollerith (A), integer (I), or floating point (E) formats.

Hollerith data may consist of any characters in the character set recognized by the computer and available on standard keypunch machines.

Integer data may consist of numbers only, preceded by a sign, and right-adjusted in the specified data fields. Only TRUMP variables that are read in with an integer format have names beginning with any of the letters I, J, K, L, M, or N.

Floating point data may include numbers, preceded by a sign, and must include a decimal point if the field is not entirely blank. The exponent to the base 10 may be included (right-adjusted in the specified data field), consisting of the letter E or a sign, or both, followed by numbers. Some variables used in floating point form in the program are read in with a Hollerith format, converted to floating point if a decimal point is included, or assigned some other value if no decimal point is included.

Field-free input may be used at LLL for integer and floating point variables. A comma following a number is interpreted as a field terminator. If commas are used, they must follow all numbers that are not right-adjusted in the field width specified in the format statement, including the last number in the input record. Commas may not be used after numbers that are right-adjusted or fields that have X or A specifications. For example, if the input format is (A5, 415, 5X, 5E10.3), the input record could be as follows:

"ABCDEb242,10745b27,1Ø,8bbbbb1.ØE-4,1.23456E-3bØ.2,,b1.ØE-8,"

beginning in column 1, where the letter b indicates a blank space. Note that the integer "10745" and the floating point number "1.23456E-3" each fills its data field, so cannot be preceded by a blank or followed by a comma. The Hollerith data "ABCDE" and the blank field specified by "5X" in the format also must each use the entire data field and cannot be followed by a comma. The fourth E data field is blank. The number in the fifth E field does not use 10 columns, so must be followed by a comma.

In this report, the symbol "Ø" is used for the number zero in descriptions of FORTRAN names and data values, where confusion with the letter "O" might otherwise result. The symbol "*" between FORTRAN names, numerical values, or combinations of these, indicates multiplication, unless otherwise noted.

2. Data Deck Organization

a. General

The complete card deck submitted for a computer run consists of the following: (1) one or more I. D. and control cards required by the Monitor system, (2) "*B" cards with operator instructions, (3) "*T" cards specifying tapes and tape units to be used,

(4) either a program deck (FORTRAN or binary) or a control card for loading the program from a restart tape, (5) and TRUMP control cards and data decks. Requirements of the Monitor system are described in manuals available in the Computer Information Center, LLL. Use of TRUMP control cards will be described in Section II.E. The requirements for data decks are described immediately below.

b. Data Decks

Each data deck must consist of a Problem Name card, any number of Block Number cards with their accompanying Block Item input cards, and a "end" line ~~Data End card~~. Any number of data decks can be stacked together, using the data carryover controls on the Problem Name cards, and the block modification controls on the Block Number cards, to eliminate the need for describing any input data block or block item more than once, when used for more than one problem. Data decks and data blocks may be read in from various input units specified by special control cards described in Section II.E.4 below. The final data deck ~~should~~ may be followed by ~~either~~ a "*SPLIT" ~~or~~ card. ~~"*OFFMON" TRUMP control card.~~ The final problem data deck need not be ended with a "END EO" line.

c. Problem Name Card

The first card of each data deck must be a Problem Name and Data Carryover card. The symbol "*" must be in column 1, any desired problem identification and description in columns 2 through 71 and 73 through 80, and the data carryover control, K, in column 72. If K is 2, 3, or 4, all data from the preceding problem will be saved, and only those data blocks and block items required to change data or add new data need be included in the data deck. The initial temperatures and reactant concentrations will be the same as either the initial (K = 2) or final (K = 3 or 4) values from the preceding problem, unless modified in a new BLOCK 9A or 9B. The initial time will be the same as either the initial (K = 2 or 3) or final (K = 4) time from the preceding problem, unless modified in a new BLOCK 1. SMALL should be specified in any problem for which K is 3 or 4, if not specified in the original BLOCK 1. The entire contents of the card (NAME) will appear on all output, along with a sequential problem number, the machine letter, and the time and date at the beginning of the run. Additional cards containing comments without "*" in column 1 may precede the name card and will appear on the on-line printer and the printout.

d. Block Number Cards

There are presently 12 allowed input data blocks, with block numbers from 1 through 12, as described in Table 2. Each block used must begin with a Block Number card and, except for BLOCKs 1 and 11, must end with a blank card. The specification of data within each data block is described in Section II.D.3 below.

The Block Number card must have the word "BLOCK" in columns 1 through 5, and the block number, from Ø1 through 12, in columns 6 and 7. A block type indicator (MOD) may be punched in column 8. If MOD is not A or B, the data read in for the block

Table 2. TRUMP input data blocks.

BLOCK	Description	Optional subroutines required
1[a,b]	Problem controls, limits, constants	GEN[c]
2[a]	Material properties	PRIME[d]
3	Reactant properties	CHEM
4[a]	Node descriptions	
5[2]	Internal thermal connections	
6	External thermal connections	SURE
7[e]	External temperatures (boundary nodes)	SURE
8	Variable heat-generation rates	GEN
9	Initial values of temperature, reactant concentrations, heat-generation rates	GEN[f]
10	Mass-flow connections	FLOW
11	Graphical output controls, nodes	PLOT
12	Nodes with properties dependent on remote temperature	

[a]Required data for all problems.

[b]Must be read in before any other data blocks.

[c]Only if GONE is nonzero.

[d]Only if "*MAST" option is used.

[e]Required if BLOCK 6 is used.

[f]Only if one or more GG values are nonzero.

replaces any data previously read in for the same block. If MOD is A, data previously read in are saved and the new data are added. If MOD is B, data previously read in are also saved, and the new data either modify individual items previously described with the same reference numbers or, if none can be found, are added to the old data. These options do not apply to BLOCKs 1 and 11.

Any desired block description may be in columns 9 through 80, which will appear on the printout. Additional cards containing comments may proceed or follow any data block, as long as columns 6 and 7 are blank, and will also appear on the printout.

An option is available for using the Block Number card to specify the logical input unit from which the block items are to be read (see Section II.E.4).

Data BLOCK 1 must follow the Problem Name card, unless the data in BLOCK 1 is being carried over from the preceding problem. Data BLOCK 1 may be used more than once in the data deck, to change the values of parameters used in initializing data in the following data blocks, such as SCALE, KD, TONE, ALONE, BONE, GONE, FONE, HONE, RONE and PONE. Thus a data block may be subdivided into sections requiring different values of one or more of the listed parameters, and placed in the data deck with a new BLOCK 1 preceding each section. All other data blocks may be placed in any order, and any number of type A or B blocks can be used for each data

block, except BLOCK 11. If more items are described for a particular block than the allowed limit (see Section II.E.2), a diagnostic statement is printed out, the excess items each successively stored in the last available memory location, and KWIT = 11. If any table length exceeds the allowed limit (see Section II.E.2), the data will be incorrectly stored, and KWIT = 12. In either case, the problem will end after completion of the input and initialization phases.

Although a type B data block can serve the same purpose as a type A data block, when all the items have new reference numbers, the time spent searching the block items previously read in is wasteful.

In a type 2, 3, or 4 continuation problem, a block of data may be cleared from memory, without substituting new data, by including a data block consisting only of a Block Number card with column 8 blank, followed by a blank card. This applies to all blocks except BLOCK 1 and BLOCK 11. BLOCK 11 requires two blank cards.

If the same item is described more than once in a block, the effects will be additive in BLOCKs 5, 6, or 10, but only the last description will be used in BLOCKs 2, 3, 4, 7, 8, 9, or 12.

e. Data End Card

The last card of each data deck must be a Data End card, with the word "ENDED" in columns 1 through 5, and ~~either~~ "-1" ~~or "-2"~~ in columns 6 and 7. ~~The options are for the problem either to be interrupted, so it can be restarted later, or to be ended, respectively, if a change switch is turned on during the problem.~~ I is typed long gather.

The advantage of ~~ending the problem~~ is that ~~the additional~~ output data produced on the final time step, including temperature vs time plots, if specified, are obtained. If NPUNCH in data BLOCK 1 is 1, the final temperatures and reactant concentrations will be punched on cards in the format of data BLOCK 9, and can be placed in the data deck to restart the problem (also see Section II.E.3 below).

3. Block Item Descriptions

Input data that may be specified in each data block are summarized in this section of this report. Formats, definitions, options, and suggested uses are described for all input data specified in BLOCKs 1 and 11, and for one block item in each of the other blocks. A block item is described on a card specifying the identification number or numbers of the material, reactant, node, boundary node, or combination of these involved, and other data, and any additional cards required for tables of temperature- or time-dependent variables associated with the block item.

The maximum table length and maximum number of block items allowed in each block in various versions of TRUMP are given in Table 1. The data blocks presently allowed in TRUMP are listed in Table 2. Short definitions of input data variables are included in the TRUMP Glossary, Section IV.G., and locations of the input data fields in the data deck are summarized in Fig. 1. Only the actual values of the variables are punched on the cards, not the variable names themselves.

*	Problem name card. Column 1 must be "*". Column 72 controls data carryover.										K
BLOCK	1	Controls, limits, constants. Must follow problem name card, if not carried over with K = 2 or 3.									
IPRINT	NUM	KDATA	KSPEC	MCYC	MSEC	NPUNCH (NUBLOG)	NDOT	IRITE (TOMMY ~~ITAPE~~)	TIMEP	SCALE	(SIGMA)
KD	KT	DELTO		SMALL		TVARY		TAU	TIMAX	TMIN	TMAX
TONE		ALONE		BONE		GONE		FONE	HONE	RONE	PONE
BLOCK	2 MOD	Material names, numbers, reactant references, thermal properties.									
AMAT	MAT	KA	KB	LTABC	LTABK	DENS		CAPT	CONT	TMELT	HMELT
CAPT(1)		TVARC(1)		CAPT(2)		TVARC(2)		(Etc., to LTABC, omit if CAPT constant.)			
CONT(1)		TVARK(1)		CONT(2)		TVARK(2)		(Etc., to LTABK, omit if CONT constant.)			
*MAST	NREAD	NLIST	NEXT	MALL	MUNCH	XD		XC	XK	XT	XTP
MATP(1)	MATS(1)	(Follows *MAST card if MALL is zero. End list with a blank card.)									
Repeat AMAT card for each material. End block with a blank card. See manual for use of optional *MAST cards.											
BLOCK	3 MOD	Reactant names, numbers, reaction properties. (Optional.)									
AKEM	KEM	LTABQ	LTABZ	LTABE		QT		ZT	ET	CORD	
QT(1)		TVARQ(1)		QT(2)		TVARQ(2)		(Etc., to LTABQ, omit if QT constant.)			
ZT(1)		TVARZ(1)		ZT(2)		TVARZ(2)		(Etc., to LTABZ, omit if ZT constant.)			
ET(1)		TVARE(1)		ET(2)		TVARE(2)		(Etc., to LTABE, omit if ET constant.)			
Repeat for each reactant, end block with a blank card.											
BLOCK	4 MOD	Node numbers, material references, types, volumes. Initial values TONE, ALONE, BONE, GONE will be assigned to each node.									
NODE	NSEQ	NADD	NODMAT	KS		DLONG		DWIDE	DRAD		
Repeat for each node or sequence, end block with a blank card.											
BLOCK	5 MOD	Internal thermal connection node numbers, connection lengths, interface areas, interface conductances. (Optional.)									
NOD1	NOD2	NSEQ	NAD1	NAD2	NZ	DEL1		DEL2	DLONG	DRAD	HINT	RINT
Repeat for each internal thermal connection or sequence, end block with a blank card.											

Fig. 1. TRUMP data input format summary.

BLOCK	6 MOD	Surface thermal connection node numbers, areas, surface conductances. (Optional.)							
NODS	NODSB	NSEQ	NADS	NADSB	LTABH	DLONG	DRAD	HSURE(Blank = HONE)	RSURE(Blank = RONE) POWER(Blank = PONE)
HSURT(1)		TVARH(1)		HSURT(2)		TVARH(2)	(Etc., to LTABH,	omit if HSURE constant.)	
Repeat for each surface thermal connection or sequence, end block with a blank card.									
BLOCK	7 MOD	Boundary node numbers, temperatures. (Optional, unless Block 6 is used.)							
NODB	LTABT			TEMPB(1)	TIMEB(1)	TEMPB(2)	TIMEB(2)	TEMPB(3)	TIMEB(3)
TEMPB(4)		TIMEB(4)		...	...	(Etc., to LTABT,	omit if LTABT is less than 4.)		
Repeat for each boundary node, end block with a blank card.									
BLOCK	8 MOD	Variable heat generating node numbers, heat generation rates. (Optional.)							
NODG	NSEQ	NADG	LTABG	QT(1)	TVARG(1)	QT(2)	TVARG(2)	QT(3)	TVARG(3)
QT(4)		TVARG(4)		...	...	(Etc., to LTABG,	omit if LTABG is less than 4.)		
Repeat for each variable heat generating node or sequence, end block with a blank card.									
BLOCK	9 MOD	Initial value node numbers, initial temperatures, reactant concentrations, heat generation rates. (Optional.)							
NOTE	NSEQ	NADD		TT (Blank = TONE)	AA (Blank = ALONE)	BB (Blank = BONE)	GG (Blank = GONE)		
Repeat for each initial value node or sequence, end block with a blank card.									
BLOCK	1 ∅ MOD	Mass flow connection node numbers, flow rates, flow path lengths. (Optional.)							
NODF1	NODF2	NSEQ	NADF1	NADF2	LTABFL	FLOWT (Blank=FONE)	DELF1	DELF2	
FLOWT(1)		TVARFL(1)		FLOWT(2)		TVARFL(2)	(Etc., to LTABFL,	omit if FLOWT constant.)	
Repeat for each mass flow connection or sequence, end block with a blank card.									
BLOCK	11	CRT plot controls, scale factors, nodes for temperature vs time plots. (Optional.)							
JPIC	LOGR	LOGTEM	LOGTIM	FRAD1		FRAD2(Blank = 1.0)	FTEMP1	FTEMP2 (Blank = 1.0)	FTIME1 FTIME2 (Blank = 1.0)
NODEP(1)	NODEP(2)	NODEP(3)	...						
BLOCK	12 MOD	Remote temperature dependence controlled node numbers, controlling node numbers, property numbers. (Optional.)							
NODP1	NODP2	NPROP	NSEQ	NADP1	NADP2	NADPR			
Repeat for each remote temperature dependence node or sequence, end block with a blank card.									
ENDED	-1	No more data for this problem. May be followed by additional data decks. With -2 in col. 6-7, sense switch 1 will end problem.							
Blocks 1, 2, and 4 are required, unless carried over with K = 2 or 3. Blocks may be placed in any order, except as noted.									

Fig. 1. TRUMP data input format summary (continued).

Section II.C. should be referred to for a detailed discussion of the development of the calculational model and selection of input data. Section III should be referred to for a detailed discussion of the methods of calculation used in the program.

a. BLOCK 1. Problem Controls, Limits, and Constants (Required).

Card 1. Format (10I5, 2E10.3). General Controls.

IPRINT Number of time steps between data output, in addition to output on first, second, and last time steps, and output controlled by TIMEP. IPRINT is not used if negative, zero, or unspecified.

NUM Identification number of a node for which temperature, rate of temperature change, phase or reactant concentrations, heat-generation rate, and time will be written out after each time step. NUM will not be used if zero or unspecified.

KDATA Controls options on output data, normal amount (0), minimum (-1), or maximum (1). See Section II.C.17a.

KSPEC Node classification and difference-equation control. Normally zero or unspecified. If zero, regular nodes will be reclassified as special nodes only when necessary to maintain the stability limit above the time step, and the interpolation factor in the heat-balance difference equations may vary in the range 0.57 to 1.0. If negative, no nodes may be reclassified, and the steady-state criteria will not be used to end the problem. If positive, all nodes will be reclassified as special nodes before the first time step. If 2, the interpolation factor will be fixed at 1.0, so that the backward difference method will be used. If 3, the interpolation factor will be fixed at 0.5, so that the central difference method will be used. Node classifications may be made individually in BLOCK 4 with KS. All nodes listed in BLOCK 6 are classified as special nodes. DELTO and SMALL must be specified when KSPEC is positive.

MCYC Maximum allowed number of time steps. MCYC will not be used if zero or unspecified. If negative, problem will end after the first time step.

MSEC Maximum allowed machine time, in seconds. MSEC will not be used if zero or unspecified. If negative, problem will end after the first time step.

NPUNCH Causes ~~a deck of punched cards~~ file AFILILØ9 [TT....ØX] in the format of BLOCK 9 to be produced when the problem is interrupted ~~by Sense Switch 1~~ or ends normally, if nonzero. The data are the final values of T, A, B and G. The new BLOCK 9 may be inserted in the input ~~deck~~ file, which may then be resubmitted to continue the problem.

NDOT Causes all time derivatives to be maintained at zero during the problem, if nonzero. Not normally used.

IRITE Number of time steps between data output on ~~unit ITAPE~~ file AFILPP (PØ....ØX) of problem time and table of node numbers and node temperatures, in addition to output on first, second, and last time steps. Not used if unspecified, zero, or negative.

~~ITAPE~~	~~Output unit on which output data will be written as described under IRITE above.~~ Will be set to 6 if unspecified or blank, if IRITE is positive. ~~Unit~~ must also be specified by a *T card, or by *MOUNT and *FINIS ~~control~~ cards.
TIMEP	Problem time interval between data output, in addition to output on first, second, and last time steps, and output controlled by IPRINT and JPIC. TIMEP is ignored if negative, zero, or unspecified. Output will be written at exact multiples of TIMEP, if possible, by adjusting the time steps in the range from SMALL to DELTO. The adjustment is also limited to a range from 2/3 to 3/2 of the time step that would otherwise be used.
SCALE	Scale factor. Set to 1.0 if negative, zero, or unspecified. Will be applied to all geometric input data in BLOCKs 4, 5, and 6 read in following this BLOCK 1 and preceding any other BLOCK 1 with a different scale factor. Lengths will be multiplied by SCALE, areas by $SCALE^2$, volumes by $SCALE^3$.*

Card 2. Format (2I5, 7E10.3). Symmetry, Units, and Limits.

KD	Symmetry type indicator: 1 for nonsymmetric, 2 for axisymmetric, 3 for centrisymmetric. Input values of DRAD in BLOCKS 4, 5, and 6 read in after BLOCK 1 will be replaced with DRAD, 2π DRAD, and 4π $DRAD^2$, respectively. KD is set to 1 if unspecified or zero (see Section II.C.4).
KT	Temperature scale indicator: 1 for Centigrade, 2 for Kelvin, 3 for Fahrenheit, 4 for Rankin. KT is set to 1 if unspecified or zero. (See Section II.C.2.)
DELTO	Maximum allowed time step. May be used with SMALL to limit range of time step. Not usually needed. DELTO is set to 10^{12} if unspecified or not in the range from 10^{-10} to 10^{12}. DELTO must be specified if KSPEC is positive.
SMALL	Minimum allowed time step. May be used with DELTO to limit range of time step. Not usually needed. SMALL is set to 10^{-12} if less than 10^{-12}. If SMALL is unspecified, the program sets SMALL to 2/3 of 1% of the smallest time constant of any regular node in the system, if at least 1/4 of the nodes are regular nodes. SMALL should be specified in continuation problems of type 3 or 4, if not specified in the original BLOCK 1.
TVARY	Desired maximum temperature change in each time step. TVARY is set to 5.0 if unspecified or zero. Controls size of the time step between limits of SMALL and DELTO. Steady state cannot end the

*Superscripts used here mean "squared" and "cubed."

problem until the maximum temperature change is less than 0.001 TVARY for two successive time steps. The convergence criteria for the iterative calculation for special nodes are a change in the weighted average temperature change of all connected special nodes of less than $5(10^{-5})$*TVARY, and in the temperature change of any finite-volume special node of less than $5(10^{-4})$*TVARY.

TAU — Initial problem time. Will be set to zero if unspecified.

TIMAX — Maximum allowable problem time. TIMAX will not be used if zero or unspecified. If it is negative, problem will end after first time step.

TMIN — Minimum allowable problem temperature. Will be set to -10^{12} if equal to or larger than TMAX. HMELT(N) in BLOCK 2 will be set to zero for any material for which TMELT(N) is less than TMIN.

TMAX — Maximum allowable problem temperature set to 10^{12} if equal to or less than TMIN. HMELT(N) in BLOCK 2 will be set to zero for any material for which TMELT(N) is greater than TMAX.

Card 3. Format (8E10.3). System Constants.

(Apply only to data in data blocks read in after BLOCK 1.)

TONE — Initial temperature of all nodes for which no TT is specified in BLOCK 9.

ALONE — Initial weight fraction of the first chemical reactant in all nodes for which no AA is specified in BLOCK 9.

BONE — Initial weight fraction of the second chemical reactant in all nodes for which no BB is specified in BLOCK 9.

GONE — Constant heat generation rate in all nodes for which no GG is given in BLOCK 9, or no table of GT vs TVARG is specified in BLOCK 8. Typical units: cal/sec-cm^3.

FONE — Constant mass-flow rate for all mass-flow connections for which no FLOWN is specified in BLOCK 10.

HONE — Heat-transfer coefficient for all external connections for which no HSURE or table of HSURT vs TVARH is specified in BLOCK 6.

RONE — Radiation form factor for all external connections for which no RSURE is specified in BLOCK 6.

PONE — Free convection exponent for all external connections for which no POWER is specified in BLOCK 6.

b. BLOCK 2. Material Properties.

~~Normal Input Option:~~

Card 1. Format (A5, 5I5, 5E10.3). Material Description.

AMAT	Material name. Column 1 must not be blank. If characters "*MAST" appear here, this card will be interpreted as a control card for the data-transfer option described below. The "*MAST" control card may follow any number of normal BLOCK 2 input items.
MAT	Material identification number.
KA, KB	Identification numbers of any chemical reactants contained in material. KA will not be used if a nonzero value of HMELT(N) is specified. Initial concentrations of reactants must be specified in BLOCK 1 (ALONE, BONE), or in BLOCK 9 (AA, BB).
LTABC	Number of points listed on heat-capacity table card or cards following this card, positive if vs temperature, negative if vs time, zero if only one value is specified.
LTABK	Number of points listed on thermal-conductivity table card or cards (following this card and any heat-capacity table cards), positive if vs temperature, negative if vs time, zero if only one value is specified.
DENS	Density. Typical units g/cm^3.
CAPT	Heat capacity, if constant. Typical units: cal/g-°C.
CONT	Thermal conductivity, if constant. Typical units: cal/sec-cm-°C.
TMELT	Temperature at which latent heat HMELT is absorbed or released.
HMELT	Latent heat effect at TMELT. Difference between heat contents of high-temperature phase and low-temperature phase at TMELT. Typical units: cal/g. HMELT will be set to zero if TMELT is not between TMIN and TMAX. See AA, BLOCK 9.

Card 2A, etc. Format (8E10.3). Heat Capacity Table.

(Omit if heat capacity is constant.)

CAPT(1)	Heat capacity. Column 1 must be blank on every table card.
TVARC(1)	Temperature of time corresponding to CAPT(1).
(Etc.)	

Card 3A, etc. Format (8E10.3). Thermal Conductivity Table.

(Omit if thermal conductivity is constant.)

CONT(1)	Thermal conductivity. Column 1 must be blank on every table card.
TVARK(1)	Temperature or time corresponding to CONT(1).
(Etc.)	

Data Transfer Option:

Card 1. Format (A5, 5I5, 5E10.3). Data-Transfer Control Card.

(May follow any number of normal BLOCK 2 input items.)

AMAT — Control symbols "*MAST" cause program to interpret this card as a control card for the data-transfer option described here.

NREAD — Logical I/O unit number assigned to the tape or other file containing the master materials data list or its substitute. NREAD may not be blank or zero. Unit 6 is suggested, for uniformity among users at LLL. (Assignment must have been made by a *T, *ASSIGN, or *MOUNT control card.) NREAD may be given the value NB, the logical I/O unit from which this control card is read in. In that case, the master data list or its substitute must follow this card, if MALL is 1, or the material selection list described below, if MALL is zero. If NREAD is either NB, 2 ~~NR~~ (unit from which the Block Number card was read in), or 2 ~~NRS~~ (the normal Monitor input unit, 2 at LRL), the master materials data list or its substitute must end with a card record with the symbol $ in column 1, which will be recognized as an end-of-file mark by the program. If NREAD is NB, 2 ~~NR~~, or 2 ~~NRS~~, or preceded by a minus sign, it will not be rewound before and after reading.

NLIST — Logical I/O unit number assigned to the tape or other file into which the secondary materials data list is to be written. Unit 7 is suggested for uniformity among users at LLL. (Assignment must have been made by a *T, *ASSIGN, or *MOUNT control card.) If NLIST is blank or zero, it will be given the value 4, which (at LLL) is a scratch tape or file and cannot be saved by the user at the end of the job. If NLIST is either 3 ~~MW~~, the normal ~~Monitor~~ TSO output unit, ~~3 at LLL),~~ ~~M (the TRUMP data output unit, either MW or the unit selected by the *OUTPUT control card), or 100 (dd80 CRT output),~~ or is preceded by a minus sign, it will not be rewound before or after being written upon. Otherwise, it will be rewound. Always precede NLIST with a minus sign if it is not going to be read in the same job. If NEST, described below, is not specified, and if NLIST is not ~~MW, M, or 100,~~ 3, the value of NB will be changed to NLIST after NLIST has been written upon, end-filed, and rewound, and material descriptions will be read in from NLIST, as in the normal input option described above. Also, the file on NLIST may be saved and used as normal input in other jobs by using the Block Number card "BLOCKØ2A*INPUTNN," where NN is the logical I/O unit containing the file.

NEST — Logical I/O unit number to be used as the input unit for the normal input-option material descriptions after the transfer of data from NREAD to NLIST is complete. Need not be specified if same as

NLIST, unless NLIST is ~~MW, M, or 100~~ 3 and the input unit from which the next data is to be read in is not NB.

MALL — Data selection control. If zero, data will be transferred from unit NREAD to unit NLIST only for those materials whose master identification numbers (MATP) are listed in the following material selection list. If 1, all data on unit NREAD will be transferred to unit NLIST.

MUNCH — Data unit conversion control. If blank or less than 2, no unit conversion will be done. Otherwise, units will be converted to the system m-kg-sec-joule-°K (2), ft-lb-hr-Btu-°F (3), in.-lb-sec-Btu-°F (4), m-ton-day-Gcal-°C (5), or converted by the factors XD, XC, XK, XT, XTP specified on this card (6). If the data on unit NREAD are not in the cm-g-sec-cal-°C system, and conversion is required, MUNCH must be 6, and the conversion factors must be specified.

XD — Unit conversion factor for density, DENS = XD*DENS.

XC — Unit conversion factor for heat capacity, CAPT = XC*CAPT.

XK — Unit conversion factor for thermal conductivity, CONT = XK*CONT.

XT, XTP — Unit conversion factors for temperatures, T (TMELT, TVARC or TVARK) = XT*T+XTP. (Latent heat is converted by HMELT = XC*XT*HMELT.)

<u>Card 2A, etc. Format (2I5). Material Selection List.</u>

(Omit if MALL is positive.)

(This list must be followed by a blank card.)

MATP(1) — Identification number of a material on the master materials data list on unit NREAD, for which data are to be transferred to the secondary materials data list on unit NLIST.

MATS(1) — Identification number to be substituted for MATP(N) on the secondary materials data list on unit NLIST. This is the number which will become MAT if unit NLIST is later used as the input unit for BLOCK 2 items, with the normal input option, and which is referred to by NODMAT in Data BLOCK 4. If MATS(1) is blank or zero, MATP on same card will be used.

(Etc.)

c. BLOCK 3. Chemical Reactant Properties.

Card 1. Format (A5, 4I5, 5X, 6E10.3). Chemical Reactant Description.

AKEM	Chemical reactant name.
KEM	Chemical reactant identification number.
LTABQ	Number of tabulated heat of reaction values, negative if vs time, positive if vs temperature, zero if no table.
LTABZ	Number of tabulated log collision frequency values, negative if vs time, positive if vs temperature, zero if no table.
LTABE	Number of tabulated activation energy values, negative if vs time, positive if vs temperature, zero if no table.
QT	Heat of reaction, if constant. Typical units: cal/g.
ZT	Log collision frequency, if constant. Typical units: $\log_e$ (sec^{-1}).
ET	Activation energy, if constant. Must be in one of the following sets of units: cal/g mole-°C, Btu/lb mole-°F.
CORD	Reaction order. CORD is set to 1.0 if not specified. Must specify 0.0 to get zero order.

Card 2A, etc. Format (8E10.3). Heat of Reaction Table.

(Omit if heat of reaction is constant.)

QT(1)	Heat of reaction.
TVARQ(1)	Temperature or time corresponding to QT(1).
(Etc.)	

Card 3A, etc. Format (8E10.3). Log Collision Frequency Table.

(Omit if log collision frequency is constant.)

ZT(1)	Log collision frequency.
TVARZ(1)	Temperature or time corresponding to ZT(1).
(Etc.)	

Card 4A, etc. Format (8E10.3). Activation Energy Table.

(Omit if activation energy is constant.)

ET(1)	Activation energy
TVARE(1)	Temperature or time corresponding to ET(1).
(Etc.)	

d. BLOCK 4. Node Descriptions.

Card 1. Format (5I5, 5X, 3E10.3).

NODE	Node identification number.
NSEQ	Number of additional identical nodes, or, if DRAD is preceded by a minus sign, nodes with DRAD values incremented by the difference between DRAD for this node and DRAD for the immediately preceding node.
NADD	Increment between successive values of NODE in the sequence of NSEQ + 1 nodes generated when NSEQ is used.
NODMAT	Identification number of the material of which the node is a part.
KS	Node type indicator. Node will be a special node if KS is nonzero. Only used when KSPEC in BLOCK 1 is negative. Not needed for zero-volume and surface nodes.
DLONG, DWIDE, DRAD	Geometric factors whose product with $SCALE^3$ is equal to the node volume, if KD is 1. For KD values of 2 or 3, the input value of DRAD is replaced with 2π DRAD, or 4π $DRAD^2$, respectively, before calculation of the volume. The program substitutes 10^{-24} if the calculated volume is zero.

e. BLOCK 5. Internal Thermal Connections.

Card 1. Format (2I5, 3I3, I1, 6E10.3).

NOD1, NOD2	Identification numbers of the connected nodes.
NSEQ	Number of additional identical connections, or if DRAD is preceded by a minus sign, connections with DRAD values incremented by the difference between DRAD for this connection and DRAD for the immediately preceding connection.
NAD1, NAD2	Increments between successive values of NOD1 and NOD2, respectively, in the sequence of NSEQ + 1 connections generated when NSEQ is used.
NZ	Indicates NAD1 and NAD2 are each to be multiplied by 10^{NZ} before use in generating a sequence of connections.
DEL1, DEL2	Lengths, when multiplied by SCALE, of the heat conduction paths from the nodal points in NOD1 and NOD2 to the connected interface. Should not both be zero unless either HINT or RINT is specified.
DLONG, DRAD	Geometric factors whose product with $SCALE^2$ is the area of the connected interface if KD is 1. For KD values of 2 or 3, the input value of DRAD is replaced with 2π DRAD or 4π $DRAD^2$, respectively, before calculating the area.
HINT	Heat-transfer coefficient for conduction and convection across the space between the connected surfaces, including such effects as of surface films, a gas gap, or contact resistances (see Section II.C.10a). If HINT is not specified, it will be set to 10^{12} if RINT is not specified, or to 10^{-24} if RINT is specified. Typical units: cal/sec-cm^2-°C.
RINT	Thermal radiation form factor for the connected surfaces of the nodes (see Section II.C.10a). RINT will be set to zero if the data field (columns 71 through 80) contains no decimal point. If RINT is negative, the absolute value will be used as a free convection exponent [Eq. (II.C.10.4)].

f. BLOCK 6. External Thermal Connections.

Card 1. Format (6I5, 5E10.3).

NODS	Surface node identification number. NODS should be either a zero-volume node or a node with the nodal point on its external surface.
NODSB	External temperature (boundary node) identification number.
NSEQ	Number of additional identical connections, or, if DRAD is preceded by a minus sign, connections with DRAD values incremented by the difference between DRAD for the connection and DRAD for the immediately preceding connection.
NADS, NADSB	Increments between successive values of NODS and NODSB respectively, in the sequence of NSEQ + 1 connections generated when NSEQ is used.
LTABH	Number of tabulated convection coefficient values, negative if vs time, positive if vs temperature, zero if no table.
DLONG, DRAD	Geometric factors whose product with $SCALE^2$ is the area of the external surface of NODS if KD is 1. For KD values of 2 or 3, the input value of DRAD is replaced by 2π DRAD or 4π $DRAD^2$, respectively, before calculating the area.
HSURE	Convection coefficient, if constant (see Section II.C.11b). HSURE is set to HONE (BLOCK 1) if not specified. Typical units: cal/sec-cm^2-°C.
RSURE	Thermal radiation form factor (see Section II.C.11b). RSURE is set to RONE (BLOCK 1) if not specified.
POWER	Free-convection exponent (see Section II.C.11b). POWER is set to PONE (BLOCK 1) if not specified, or if the data field (columns 71 through 80) contains no decimal point.

Card 2A, etc. Format (8E10.3). Convection Coefficient Table

(Omit if convection coefficient is constant.)

HSURT(1)	Convection coefficient.
TVARH(1)	Temperature or time corresponding to HSURT(1).
(Etc.)	

g. BLOCK 7. External Temperature (Boundary Nodes).

Card 1. Format (2I5, 10X, 6E10.3).

NODB	External temperature (boundary node) identification number.
LTABT	Number of tabulated external temperature values. To obtain sinusoidal variation of external temperature with time, LTABT must be 100 (see Section II.C.11a). Zero if no table.
TEMPB(1)	External temperature. (Average value of sinusoidal variation, if LTABT is 100.)
TIMEB(1)	Time corresponding to TEMPB(1). (Period of sinusoidal variation, if LTABT is 100.)
TEMPB(2)	External temperature. (Amplitude of sinusoidal variation, if LTABT is 100.)
TIMEB(2)	Time corresponding to TEMPB(2). (Phase advance time of sinusoidal variation, if LTABT is 100.)
(Etc.)	[Up to 3 pairs of values on this card, and 4 pairs on each additional card required, if LTABT is not zero. Use format (8E10.3) for additional cards.]

h. BLOCK 8. Variable Heat-Generation Rates.

Card 1. Format (4I5, 6E10.3).

NODG	Identification number of node.
NSEQ	Number of additional nodes with identical variation of volumetric heat-generation rate as NODG.
NADG	Increment between successive values of NODG in the sequence of NSEQ + 1 nodes generated when NSEQ is used.
LTABG	Number of tabulated values of heat-generation rate, negative if vs time, positive if vs temperature. To obtain exponential decay of heat-generation rate with time, LTABG must be between -1 and 1, or unspecified, and GT(1) and TVARG(1) specified.
GT(1)	Volumetric heat-generation rate at TVARG(1), or at zero time (SUMTIM = 0), if exponential decay is specified. Typical units: cal/sec-cm^3. See AA and BB, BLOCK 9, for an additional method to produce an initial pulse of heat at the beginning of a problem.
TVARG(1)	Temperature or time corresponding to GT(1), or half-life of exponential decay, if the latter is specified.
(Etc.)	[Up to 3 pairs of values on this card, and 4 pairs on each additional card required. Use format (8E10.3) for additional cards.]

i. BLOCK 9. Initial Conditions and Constant Heat Generation Rates.

Card 1. Format (3I5, 5X, 4E10.3).

NOTE	Node identification number.
NSEQ	Number of additional nodes with identical initial conditions.
NADD	Increment between successive node numbers in sequence of NSEQ + 1 nodes generated when NSEQ is used.
TT	Initial temperature. TT is set to TONE (BLOCK 1) if not specified.
AA	Initial weight fraction of first chemical reactant. AA is set to ALONE (BLOCK 1) if not specified. AA may be used to specify the initial concentration of low-temperature phase in nodes of materials with specified values of TMELT and HMELT, if columns 67 through 71 of the Problem Name Card are "A=BL9." Otherwise, the concentration will be 1.0 for initial temperature at or below TMELT and 0.0 for initial temperature above TMELT. If a negative value of AA, or of BB below, is specified, and the node contains the corresponding reactant, the concentration will be set to zero during the first time step, and the heat of reaction will be converted to sensible heat. No further reaction will then take place. The resulting temperature change will be accurate only if the change in specific heat of the material is small over the temperature interval involved. The temperature will increase if the heat of reaction is negative, and decrease if the heat of reaction is positive.
BB	Initial weight fraction of second chemical reactant. BB is set to BONE (BLOCK 1) if not specified. BB may be negative (see AA above).
GG	Heat-generation rate. GG is set to GONE (BLOCK 1) if not specified. Has no effect if GT vs TVARG is specified for this node in BLOCK 8. Typical units: cal/sec-cm^3.

j. BLOCK 10. Mass-Flow Connections.

Card 1. Format (6I5, 3E10.3).

NODF1	Identification number of the source node.
NODF2	Identification number of the sink node. May not be a zero-volume node.
NSEQ	Number of additional identical mass-flow connections.
NADF1, NADF2	Increments between successive values of NODF1 and NODF2, respectively, in the sequence of NSEQ + 1 mass-flow connections generated when NSEQ is used.
LTABFL	Number of tabulated values of mass-flow rate, negative if vs time, positive if vs temperature, zero if no table.
FLOWN	Mass-flow rate, if constant. Typical units: g/sec.
DELF1, DELF2	Lengths of mass flow paths from nodal points of NODF1 and NODF2, respectively, to the connected surfaces. DELF2 may not be zero.

Card 2A, etc. Format (8E10.3). Mass-Flow Rate Table.

(Omit if mass-flow rate is constant.)

FLOWT(1) Mass-flow rate.

TVARFL(1) Temperature or time corresponding to FLOWT.

(Etc.)

k. BLOCK 11. Graphical Output Controls and Nodes.

Card 1. Format (4I5, 6E10.3). Controls.

JPIC — Number of time steps between generation of data for graphical output, in addition to output generation for first, second, and last time steps, and output controlled by TIMEP. Precede with minus sign to obtain temperature vs node location graphs, in addition to temperature vs time graphs. No graphical output will be obtained if JPIC is unspecified or zero and no values of NODEP are specified (see Section II.C.17b).

LOGR, LOGTEM, LOGTIM — Maximum number of logarithmic cycles on node location, temperature, and time scales, respectively, of graphs. If unspecified or zero, linear scales will be used. If preceded with minus sign, graphs with both linear and log scales will be obtained. To obtain interpolation between points on temperature vs node location plots (in order nodes are described in BLOCK 4), add 100 to LOGR. To obtain tic marks on axes, rather than grid lines, add 100 to LOGTEM. To obtain a node location vs time graph for 20 to 40 isotherms, add 100 to LOGTIM. To replace node location scale with node location divided by 2.Ø*SUMTIM**Ø.5, add 1000 to LOGR.

FRAD1, FRAD2, FTEMP1, FTEMP2, FTIME1, FTIME2 — Constants used to convert node locations, temperatures and times, respectively, from problem values to scale values on graphs, by use of the linear transformation $x' = f_1 + f_2 x$. FRAD2, FTEMP2 and FTIME2 will each be set to 1.0 if unspecified or zero.

Card 2. Format (16I5). Nodes for Graphical Output.

NODEP(1) — Identification number of node for which temperature vs time graph, and node location vs time graph (see LOGTIM), will be obtained. All nodes will be plotted on temperature vs node location graph (see JPIC).

Etc. — (Up to maximum number of CRT plot nodes allowed.)

1. BLOCK 12. Nodes with Properties Dependent on Remote Temperatures.

Card 1. Format (7I5).

NODP1, NODP2	Identification numbers of the node to which the property applies and the node on whose temperature the property depends, respectively. I.e. P(NODP1) = P[T(NODP2)].
NPROP	Property indicator, 1 for heat capacity, 2 for thermal conductivity, 3 for heat of reaction, 4 for log collision frequency, 5 for activation energy, 6 for variable heat-generation rate, 7 for mass-flow rate, and 8 for surface convection coefficient.
NSEQ	Number of additional nodes with properties dependent on remote temperatures, with node and property numbers determined by NADP1, NADP2, and NADPR.
NADP1, NADP2, NADPR	Increments between successive values of NODP1, NODP2, and NPROP, respectively, in the sequence of NSEQ + 1 nodes with properties dependent on remote temperatures, generated when NSEQ is used.

E. USE OF COMPUTERS

1. General

To use TRUMP at LLL, the user must have an assigned box number for card and printout retrieval, assigned tape vault slots for tape storage, and one or more account numbers to be charged with the use of computer time. These arrangements are made at LLL through department computation coordinators. The computation department number code assigned to TRUMP is 367.

A number of procedures are available for submission of card decks for running under various LLL computer systems, and the department computation coordinator should be consulted to determine the best procedure for the particular needs of the user.

TRUMP requires a minimum of system control cards, but several options are available for use of system control cards and special TRUMP control cards for added convenience and flexibility of operation. These include interruption and restart capabilities, optional use of various combinations and sequences of input and output units, and handling of disk files under the OCTOPUS/6600 system. The various TRUMP control cards discussed below are also summarized in Section IV.G.2.

2. Choice of Version and Form of TRUMP Program

The master FORTRAN listing of the TRUMP program is available on computer cards and on magnetic tape. All versions are compiled from the master listing by making temporary changes in PARAMETER statements in the main program, to choose the sets of array sizes and various options desired. Separate main program decks are kept for the most widely used versions for each LLL computer system. The parameter specifications and options included in the most widely used versions are described in Section IV. Binary object decks for these versions are also available. Users at LLL may use restart tapes produced by the OFFMON routine, by means of the Monitor control card "* LOAD (1, 5, ONMONS)" or "* LOAD (1, 5, ONMON)." The ONMONS routine is used to load the master restart tape, which is then unloaded. When the problem is interrupted by sense switch 1, or an "*OFFMON" TRUMP control card is read in, a new restart tape is made on the user's own tape. The ONMON routine may then be used to load the user's own tape in later runs. The required Monitor control cards and operator instruction cards are described in various Monitor manuals and the Livermore Time Sharing System manual.[17] At present, the TRUMP/6600/STD restart tapes are CE408 and CE409, and the TRUMP/6600/DS restart tapes are CE411 and CE412. These tapes are always loaded on unit 5A.

3. Interruption and Restart Procedures

TRUMP problems may be restarted after ending normally, or after being interrupted by Sense Switch 1, if the input value of NPUNCH is 1 (data BLOCK 1, Card 1). The final values of temperature, concentrations, and constant heat-generation rate for each node will be punched in data BLOCK 9 format, preceded by cards containing the

user's I. D., the problem name (NAME), the clock time and date for the problem, and the problem time (SUMTIM) when the interruption occurred. The new BLOCK 9 can be placed in the original data deck, TAU in BLOCK 1 given the value of SUMTIM corresponding to the new BLOCK 9, any other desired changes made, and the data deck submitted for another run.

Under the Monitor system, a restart tape will be made by the OFFMON routine whenever the operator turns on Sense Switch 1, or the program reads in an "*OFFMON" (columns 1 through 7) card while searching for the next Problem Name card in the data deck. A "*T VAULT = 5A = VAULT" Monitor control card must precede the program deck, or a "*MOUNTØ5=VAULT" TRUMP control card must precede the "*OFFMON" card, where "VAULT" is a tape-vault number assigned to the user. The restart tape contains not only the TRUMP program, but also all library routines used by the program and all data stored in core memory at the time OFFMON is called. The restart tape can be used until such time that the resident system routines are changed.

When restarting from a tape made by OFFMON (because of a problem interruption) the user has the option of either continuing the interrupted problem until it ends normally, or ending the problem on the next time step. This may be done by preceding all data decks and other control cards submitted with the new job with a "*DISCONTINUE" card (columns 1 through 12). Type 2, 3, or 4 continuation problems may be submitted, using data from the previous problem saved on the restart tape.

If desired, the user can cause all data decks following an interrupted problem to be saved on a tape, which can be used later as an input tape, as described below, so that the data decks need not be resubmitted. This will be done if a "*DECKSNN" TRUMP control card (columns 1 through 8, where NN is a two-digit tape unit number, e.g., Ø7) has been previously read in. A "*T" card must be included in the Monitor control cards, or a "*MOUNT" TRUMP control card included in the input deck to specify the tape unit and tape-vault number of the tape to be used (see Section II.E.4b below).

An option is available for Sense Switch 1 to either interrupt a problem, or cause it to be ended, before OFFMON is called (see Section II.D.2e).

4. Control of Input and Output Unit Assignments

a. Normal I/O Assignments

The normal input units used by TRUMP are tape 2 for the Batch Monitor system, and a disk file for the OCTOPUS controllee version. The normal output units are logical unit 3 for numerical output to be printed on the radiation printer, logical unit 100 for graphical output to be plotted on the dd80 plotter, and logical unit 14 for punched output to be punched on the IBM-1401. Output resulting from PRINT statements goes to the on-line printer under Batch Monitor systems, and to the user's teletype when using the OCTOPUS controllee version. On the OCTOPUS system, these four types of output initially go to disk files assigned by the system, and are transferred to the

appropriate output unit at the end of the job. In addition, the restart type produced by the OFFMON routine and reloaded into core by the ONMON or ONMONS is always assigned to logical I/O unit 5.

The normal input and output units will always be used when a TRUMP run is started from a source or object program deck, or when restarted with ONMON or ONMONS from a tape made by the OFFMON routine, if the first data card is "*DISCONTINUE."

For convenience and flexibility of operation, TRUMP users may change input and output unit assignments as desired among the available input and output devices on the various machines, including the OCTOPUS disk files. In addition to the Monitor "*T" and "*ASSIGN" control cards that can be used with Batch Monitor systems, as described in various user's manuals, there are a number of TRUMP control cards that may be placed before and after any data deck in the input deck.

b. Changes in System I/O Assignments

If a special tape is to be used as an input or output tape, the operator or system must be informed by either a "*T" Monitor Control card or the TRUMP control card "*MOUNTNN=VAULT." This is punched in columns 1 through 14, where NN is a logical I/O unit number, and VAULT is a tape-vault slot number. This TRUMP control card may be used with the OCTOPUS controllee version, and may be used more than once with any version, to assign tapes to several I/O units or to assign several tapes sequentially to the same unit.

When a special tape is no longer needed for a job, the TRUMP control card "*FINISNN=VAULT" should be used. This is punched in columns 1 through 14, where NN is a logical I/O unit number. This card must always precede a "*MOUNT" card assigning a new tape to a previously used logical I/O unit number.

Assignment of logical I/O units to tapes, disk files, and other hardware types may also be made with the TRUMP control card:

"*ASSIGNK, NN, MM, HHHHHHHHHH, LLLLLL"

punched in columns 1 through 32, where K is 2, 3, or 4, specifying the number of the following arguments to be used in a "CALL ASSIGN" statement, NN is a logical I/O unit number, MM specifies a hardware type, HHHHHHHHHH is a right-adjusted tape-vault slot number or disk-file name, and LLLLLL is a right-adjusted integer specifying a disk-file length, or a nonpositive number indicating a special operation on a disk file. This allows the user control of input, HSP, PRINT, PUNCH, and CRT files, which may be reassigned as desired within the restrictions and limitation of the system, as described in Monitor and OCTOPUS manuals.

In the OCTOPUS controllee version of TRUMP, logical I/O unit 2 is assigned to a disk file whose name is specified in the teletype "XEQ" line: "TRUMP FILENAME/TV," where FILENAME is the name chosen by the user to identify the input data file.

All special system assignments must be remade when restarting with ONMON or ONMONS.

c. Changes in Program I/O Assignments

The logical input unit specified in "READ INPUT TAPE" statements may be changed by use of the TRUMP control card "*INPUTNR." This is punched in columns 1 through 8, where NR is the desired input unit number. If NR is not Ø2, it must have been previously given a system assignment with a "*T," "*MOUNTNR=VAULT," or "*ASSIGN" control card. The latter must be used if the data are in a disk file. The new input unit specification will remain in effect until another "*INPUT" card is read in or, when restarting with ONMON or ONMONS, the first card read in (which is always from unit 2) is "*DISCONTINUE." In the latter case, 2 continues to be the input unit. Input tapes made as a result of use of a "*DECKS" TRUMP control card, as described in Section II.E.3 above, always end with a "*INPUTØ2" card, so that additional control cards or data decks will be read from unit 2. Special input tapes made by the user should also end with a "*INPUTØ2" card, for the same reason. When an input tape is no longer needed, a "*FINISNR=VAULT" control card should be read in.

The logical input unit from which block items are read in may be changed from the normal unit, or the unit NR specified in the last "*INPUTNN" control card, by punching "*INPUTNB" in columns 9 through 16 of the Block Number card, where NB is the desired logical input unit number. Block items include all cards to be read in following a Block Number card and preceding the next Block Number card or Data End card, including the blank card required at the end of all blocks except 1 and 11. More than one set of block items may be read in from the same input unit NB, the only requirement being that they are arranged in the same order as the corresponding Block Number cards on unit NR. Special block item tapes must have been previously given system assignments with a "*T" card, "*MOUNTNB=VAULT" card, or "*ASSIGN" card. The latter must be used if the data are in a disk file.

The special block-item input-unit option is particularly useful for reading in a standard list of material properties for BLOCK 2 (a list of data block items to be used in a large number of separate problems, or lists of data block items produced for TRUMP by other computer programs).

F. EVALUATION OF CALCULATION RESULTS

1. Diagnosis of Input Errors

All input values, values substituted by the program for blank data fields, and values of a number of derived quantities, such as node volumes, connection areas, and material summary data, are written out at the beginning of each problem. These should be carefully checked against the intended calculational model.

If the problem ended or dumped before all the input data were read in, the cause may have been a missing Problem Name card, a missing Block Number card, a missing blank card at the end of a data block, a missing Data End card, an incorrect table length, an illegal character punched, data punched in the wrong columns, a card out of place, or a floating point exponent which is too large. These errors usually cause an illegal character, indefinite operator, overflow, or insufficient-data diagnostic.

If all the input data were read in but the problem ended before the first time step, the value of KWIT on the printout may be 5, 6, 9, 11, or 12 and indicates the type of error which occurred, as listed in Table 3. Diagnostic statements are written out for each occurrence of most input errors detected by the program, and the value of KWIT may indicate only the last type of error detected.

If the problem ends after the first time step, KWIT may be 1, 2, 3, 4, 6, 7, 8 or 10, because of one of the causes listed in Table 3 or because of an incorrect floating-point exponent in the input data, resulting in an overflow or indefinite-operator error, causing a system dump. The input values of TIMAX, TMIN, TMAX, MCYC and MSEC must be carefully considered to allow the problem to run long enough to obtain the desired results.

2. Analysis of Calculation Results

All numerical and graphical output data obtained for each problem should be carefully checked for accuracy, consistency, and credibility. Accuracy of the heat balance can be checked for the system, each material, each node, each boundary node, each internal connection, and each external connection. Spatial and time accuracy can be determined to some degree by the smoothness of plots of temperature vs position and time. Consistency and credibility can be determined by comparing the results with those obtained for similar systems or with those expected on the basis of experience, sound judgment, and familiarity with the physical laws governing the system.

Errors in the heat balance can result from use of too large a value of TVARY, relative to the maximum total temperature change in the system (see Section II.C.16); use of zoning that is too fine, which results in time constants (SLIM) that are very small, relative to the time steps used in the calculation, and usually results in slowness or failure of the iteration procedure; and use of heat capacities that are strongly temperature dependent, combined with time steps too large to allow the heat-capacity curves to be closely followed.

Physically unrealistic solutions can also result from evaluation of temperature-dependent quantities at temperatures outside the range given in input tables. This can

Table 3. Values of the problem-end sentinel KWIT.

KWIT	Cause of problem ending
1[a]	The problem time, SUMTIM, reached TIMAX.
2[a]	A temperature exceeded TMAX+Ø.ØØ1*TVARY.
3[a]	A temperature was less than TMIN-Ø.ØØ1*TVARY.
4[a]	Steady-state criteria were satisfied.
5[b]	An unspecified material, reactant, node, or boundary node was referred to.
6[c]	Sense Switch 1 was turned on when a "-2" was punched in columns 6 through 7 of the Data End card, a "*CHECK" card preceded the Problem Name card, or, when an interrupted problem was restarted, a "*DISCONTINUE" card followed the "*LOAD" Monitor control card.
7[a]	The number of time steps reached MCYC.
8[a]	The number of seconds of machine time used since reading in the Problem Name card reached MSEC.
9[b]	No material list (BLOCK 2) or node list (BLOCK 4) was included in the input data or carried over from the preceding problem.
10[b]	The iterative scheme for temperature changes in connected special nodes failed to converge in 80 iterations by use of a time step equal to or less than 2 * SMALL.
11[d]	The number of items in a data block exceeded the maximum size.
12[d]	The length of a table for a time- or temperature-dependent input quantity exceeded the maximum size.

[a]Limits specified in data BLOCK 1, see Section II.C.14.

[b]Diagnostic statements are written out for each occurrence.

[c]See Section II.E.3.

[d]See Table 2.

be avoided by use of a temperature range in the table that includes all possible calculated temperatures, or by beginning and ending the table with zero slopes. The program substitutes 10^{-24} for zero or negative values of thermal conductivity, and 10^{-36} for zero or negative values of total node heat capacity, but surface heat-transfer coefficients and other tabulated quantities may be negative.

3. Requirements for Additional Calculations

Additional calculations may be required to correct errors by (1) improving accuracy by using a more finely subdivided system, (2) employing different methods of controlling the calculations, (3) determining the effects of uncertainties in input values by trying a range of values, or (4) determining the effects of simplifications in the model by trying other models. When a large number of similar calculations are to be made for the same or similar systems, careful choice of calculational controls and degree of subdivision into nodes will result in the minimum use of machine time for

the required accuracy. Also, improvements such as in optimum organization of input data and use of tapes may greatly reduce the effort required for additional calculations.

4. Generalization of Results

Many problems can be generalized by expressing the results in dimensionless or normalized form, so as to be applicable to problems other than the specific problem calculated or to simplify comparison with the results of other similar calculations. Input data may be specified in dimensionless or normalized form (or optional linear transformations applied to plotted variables) to produce graphs in dimensionless or normalized form.

The parameters temperature or temperature difference T, position x, time t, density ρ, specific heat c, thermal conductivity k, a basic dimension L, a basic temperature or temperature difference T_0, volumetric heat-generation rate G, surface heat flux per unit area F, mass-flow rate M, heat-transfer coefficient h, and total quantity of heat W may be combined into the following dimensionless ratios, and into any combinations of them:

$$T/T_0,\ x/L,\ kt/\rho cL^2,\ Gt/\rho cT_0,\ FL/kT_0,\ Mc/kL,\ hL/k,\ \text{and}\ W/\rho cL^3T_0\ .$$

Graphs are usually made by using two of the first four groups, or combinations thereof with other groups, as ordinate and abcissa. By selecting suitable groups, and scales, useful analytic approximations can often be found for large, small, or intermediate ranges of values of the groups.

G. SAMPLE PROBLEMS

(Warning: Recent versions produce results differing in last significant digits!)

1. Heat Regenerator

A heat regenerator consists of a rectangular fluid duct and a rectangular solid, each 21 cm long, with a cross section of 0.2 cm × 1.0 cm connected on the 1.0 cm side and insulated on the other three sides. At time zero the system is at 0°C, and fluid begins flowing at 0.4 g/sec or 2.0 cm/sec entering at Z = 0 at 100°C. Densities of the fluid and solid are 1.0 and 10.0 g/cm^3, and specific heats are 1.0 and 0.05 cal/g-°C, respectively.

It is assumed that no heat conduction takes place in the Z direction, that the temperature is uniform in each material in any x-y plane, and that the overall heat-transfer coefficient between the fluid and solid is 0.02 cal/sec-cm^2-°C.

The analytical solution for this problem is discussed in Carslaw and Jaeger.[4] The temperature in the solid at the fluid inlet, Z = 0, is

$$T_s(o,t) = 100\,[1 - \exp(-0.2t)], \tag{II.G.1.1}$$

and the fluid temperature on the upstream side of the moving interface between hot and cold fluid is

$$(T_f)_{t=Z/2} = 100 \exp(-Z_i/20). \tag{II.G.1.2}$$

The initial rate of temperature rise in the solid is

$$\left(\frac{\partial T_s}{\partial t}\right)_{t=Z/2} = 20\, e^{-Z/20}. \tag{II.G.1.3}$$

Temperature profiles in the fluid and solid are calculated for the first 20.0 sec of flow.

The input data for this problem are shown in Fig. 2. The fluid nodes for the TRUMP calculation consist of a thin (10^{-4} cm) fluid-inlet node, No. 2, at Z = 0, and 21 fluid nodes, each 1 cm in length, Nos. 6, 16, . . . , 196, 206 in the direction of flow, and a zero-volume node, No. 199, between Nos. 196 and 206. The solid nodes consist of a thin (10^{-4} cm) solid node, No. 1, at Z = 0, and 21 solid nodes, each 1 cm in length, Nos. 5, 15, . . . , 195, 205 in the direction of flow, and a zero-volume node, No. 200, between Nos. 195 and 205. Node 2 is used to maintain the inlet fluid temperature, and node 1 is used to obtain a temperature to compare with Eq. (II.G.1.1). Nodes 199 and 200 are used to obtain temperatures at Z = 20.0 to compare with Eqs. (II.G.1.2) and (II.G.1.3).

A boundary node, No. 1001, with a temperature of 100°C is connected to node 2 with an area of 0.2 × 1.0 cm and a heat-transfer coefficient of 10^3 cal/sec-cm-°C.

```
* HEAT REGENERATOR.  SAMPLE PROBLEM FOR TRUMP REPORT.   2/27/68.                **
NOTE.    THIS PROBLEM DOES NOT USE GEN OR CHEM SUBROUTINES.
BLOCK 1  PROBLEM LIMITS AND CONTROLS.
  150    6                          1
                                 1.0                  20.
                                              0.4
BLOCK11 CRT PLOT CONTROLS, SCALE COEFFICIENTS, AND TIME PLOT NODES.
    5  100
    1    6   45   46   95   96  145  146  199  200
BLOCK 2  MATERIAL NAMES, NUMBERS, AND THERMAL PROPERTIES.
FLUID    1                       1.0        1.0        0.0
SOLID    2                       10.0       0.05       0.0

NOTE     NODE NUMBERS CORRESPOND APPROX TO AXIAL POSITION IN MM.
BLOCK 4 NODES.
    2                1            1.0   E-4 1.0        0.2
    6   20   10      1            1.0       1.0        0.2
  199                1
    1                2            1.0   E-4 1.0        0.2
    5   20   10      2            1.0       1.0        0.2
  200                2

NOTE:    NODE 199 CONNECTED TO 196 AND 206 TO GET TEMP AT Z = 20.0.
NOTE:    NODE 200 CONNECTED TO 195 AND 205 TO GET TEMP AT Z = 20.0.
BLOCK 5  INTERNAL CONNECTION NODE NUMBERS, AREAS, AND INTERFACE CONDUCTANCES.
    2    1                                  1.0   E-4 1.0        0.02
    6    5 20 10 10                         1.0       1.0        0.02
  199  196                                  1.0       1.0        0.65  E-8
  199  206                                  1.0       1.0        0.35  E-8
  200  195                                  1.0       1.0        0.65  E-8
  200  205                                  1.0       1.0        0.35  E-8

BLOCK 6  EXTERNAL CONNECTION NODE NUMBERS, AREAS, AND SURFACE CONDUCTANCES.
    2 1001                       1.0        0.2        1.0    E3

BLOCK 7  BOUNDARY NODE TEMPERATURES.
 1001                  100.0

NOTE:    NODAL POINT OF NODE 2 IS AT Z = 0.
NOTE:    OTHER NODAL POINTS OFFSET DOWNSTREAM FOR ADDED MASS FLOW STABILITY.
BLOCK10  MASS FLOW CONNECTION NODE NUMBERS, FLOW RATES, PATH LENGTHS.
    2    6                                  1.0   E-4 0.65
    6   16   19   10   10                   0.35      0.65

ENDED-1  END OF DATA DECK.  SSI WILL INTERRUPT PROBLEM.
COL 5    10   15   20   25   30   35   40   45   50   55   60   65   70   75   80
```

Fig. 2. Input data for heat-regenerator problem.

These data are specified in BLOCKs 6 and 7. Node 2 is connected to node 1 with an area of 10^{-4} cm × 1.0 cm, and with the pairs Nos. 5 and 6 through Nos. 195 and 196 by areas of 1.0 cm × 1.0 cm, all with the heat-transfer coefficient 0.2 cal/sec-cm^2-°C. Nodes 199 and 200 are each connected to two other nodes so as to obtain linearly interpolated temperatures at Z = 20.0.

The mass-flow connections are between nodes 2 and 6 and the pairs Nos. 6 and 16 through Nos. 196 and 206, with the nodal points in nodes 6 through 206 assumed to be 0.65 cm from their upstream interfaces. This assumption results in a smoother temperature profile, and smoother time variation of the node temperatures, than would result with the nodal points at the node centers. The flow rate is specified in BLOCK 1 by making FONE 0.4. The connections are described in BLOCK 10.

The duration of the calculation is fixed by making TIMAX 20.0. Accuracy is controlled by making TVARY 1.0. Numerical and graphical output intervals are controlled by making IPRINT 150 and JPIC 5.

Nodes 1, 6, 45, 46, 95, 96, 145, 146, 199, and 200 are specified for the temerature vs time plot. In BLOCK 1, the symmetry indicator KD and the temperature indicator KT are not specified, which is equivalent to making both of them 1. NUM is made 6 to obtain printouts of the temperature of node 6 at every time step. NPUNCH is made 1 to get a BLOCK 9 punched-card deck of final temperatures when the problem is interrupted or ended.

Some of the output data for the problem are shown in Fig. 3 for the data-input phase, the first time step, and the last time step.

The temperature vs time curves resulting from the calculation are shown in Fig. 4, along with analytic solutions for fixed positions in the fluid and the solid, and the upstream side of the moving interface between hot and cold fluid. The time at which fluid node temperatures reach 50% of the correct interface temperature corresponds very closely to the time of arrival of the interface. The fluid node temperatures overshoot by about 4%, oscillate about 1% below the correct values, then approach the exact solution very closely. The solid node temperature at Z = 0 is exact, and, at other axial positions, the temperatures begin rising too soon but become asymptotic to the exact solutions by the time the liquid front has moved one node thickness past the point in the solid.

Much better accuracy, with little or no oscillation of the fluid temperatures, would result from using about 100 nodes in both the fluid and the solid, with the nodal points located even closer to the downstream interfaces.

A total of 461 time steps is required to reach a problem time of 20.0 sec, using an average time step of 0.043 sec. Only the thin surface nodes are reclassified as special nodes during the calculation, as other fluid and solid nodes have stability limits of 0.476 sec and 5.0 sec, respectively. Machine time averages 0.8 msec per time step per calculational item (see Section IV.F) on the CDC/3600, for a total of 54 sec.

```
TRUMP    INPUT DATA              PROGRAM DATE  15.774  N04/05                                  DATA    CK    1
* HEAT REGENERATOR.  SAMPLE PROBLEM FOR TRUMP REPORT.   2/27/68.                  **      20.141  N04/10
==========================================================================================================

NO DATA CARRYOVER FROM PRECEDING PROBLEM.
==========================================================================================================
                                THIS PROBLEM DOES NOT USE GEN OR CHEM SUBROUTINES.
DATA BLOCK  1                   PROBLEM LIMITS AND CONTROLS.

IFRINT    NUM KDATA KSPEC  MCYC  MSEC NPUNCH NDOT IRITE ITAPE   TIMEP        SCALE
  150       6     0     0     0     0     1    0     0     0  0.           1.00000E+00

   KD    KT    DELTO         SMALL         TVARY         TAU          TIMAX         TMIN          TMAX
    1     1  1.00000E+12  1.00000E-12  1.00000E+00  0.           2.00000E+01  -1.00000E+12  1.00000E+12

   KD KSYM    GEOM          SIGMA         TBASE
    1    1  1.00000E+00  1.35500E-12  2.73150E+02

   TONE        ALONE        BONE         GONE         FONE         HONE         RONE         PONE
  0.           0.           0.           0.           4.0000E-01   0.           0.           .
==========================================================================================================
DATA BLOCK 11                   CRT PLOT CONTROLS, SCALE COEFFICIENTS, AND TIME PLOT NODES.

  JPIC  LOGR LOGTEM LOGTIM    FRAD1        FRAD2        FTEMP1       FTEMP2       FTIME1       FTIME2
     5     0     0     0     0.           1.0000E+00   0.           1.0000E+00   0.           1.0000E+00

NODES FOR TEMPERATURE VS TIME CRT PLOTS.  MAX = 12
INDEX    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16
 NODE    1    6   45   46   95   96  145  146  199  200
==========================================================================================================
DATA BLOCK  2                   MATERIAL NAMES, NUMBERS, AND THERMAL PROPERTIES.
```

Fig. 3. Heat regenerator. Sample problem for TRUMP report.

NAME	MATL	INDEX	KA	KB	LTABC	LTABK	DENSITY	CAPACITY	CONDUCTIVITY	TMELT	HMELT
FLUID	1	1	0	0	0	0	1.0000E+00	1.0000E+00	0.	0.	0.
SOLID	2	2	0	0	0	0	1.0000E+01	5.0000E-02	0.	0.	0.

NODE NUMBERS CORRESPOND APPROX TO AXIAL POSITION IN MM.

DATA BLOCK 4 NODES.

NODE	INDEX	MATL	NTYPE	DLONG	DWIDE	DRAD	VOLUME
2	1	1	0	1.00000E-04	1.00000E+00	2.00000E-01	2.00000E-01
6	2	1	0	1.00000E+00	1.00000E+00	2.00000E-01	2.00000E-01
16	3	1	0	1.00000E+00	1.00000E+00	2.00000E-01	2.00000E-01
26	4	1	0	1.00000E+00	1.00000E+00	2.00000E-01	2.00000E-01
36	5	1	0	1.00000E+00	1.00000E+00	2.00000E-01	2.00000E-01
46	6	1	0	1.00000E+00	1.00000E+00	2.00000E-01	2.00000E-01
56	7	1	0	1.00000E+00	1.00000E+00	2.00000E-01	2.00000E-01
66	8	1	0	1.00000E+00	1.00000E+00	2.00000E-01	2.00000E-01
76	9	1	0	1.00000E+00	1.00000E+00	2.00000E-01	2.00000E-01
86	10	1	0	1.00000E+00	1.00000E+00	2.00000E-01	2.00000E-01
96	11	1	0	1.00000E+00	1.00000E+00	2.00000E-01	2.00000E-01
106	12	1	0	1.00000E+00	1.00000E+00	2.00000E-01	2.00000E-01
116	13	1	0	1.00000E+00	1.00000E+00	2.00000E-01	2.00000E-01
126	14	1	0	1.00000E+00	1.00000E+00	2.00000E-01	2.00000E-01
136	15	1	0	1.00000E+00	1.00000E+00	2.00000E-01	2.00000E-01
146	16	1	0	1.00000E+00	1.00000E+00	2.00000E-01	2.00000E-01
156	17	1	0	1.00000E+00	1.00000E+00	2.00000E-01	2.00000E-01
166	18	1	0	1.00000E+00	1.00000E+00	2.00000E-01	2.00000E-01
176	19	1	0	1.00000E+00	1.00000E+00	2.00000E-01	2.00000E-01
186	20	1	0	1.00000E+00	1.00000E+00	2.00000E-01	2.00000E-01
196	21	1	0	1.00000E+00	1.00000E+00	2.00000E-01	2.00000E-01
206	22	1	0	1.00000E+00	1.00000E+00	2.00000E-01	2.00000E-01
199	23	1	2	0.	0.	0.	1.00000E-24
1	24	2	0	1.00000E-04	1.00000E+00	2.00000E-01	2.00000E-01
5	25	2	0	1.00000E+00	1.00000E+00	2.00000E-01	2.00000E-01
15	26	2	0	1.00000E+00	1.00000E+00	2.00000E-01	2.00000E-01
25	27	2	0	1.00000E+00	1.00000E+00	2.00000E-01	2.00000E-01
35	28	2	0	1.00000E+00	1.00000E+00	2.00000E-01	2.00000E-01
45	29	2	0	1.00000E+00	1.00000E+00	2.00000E-01	2.00000E-01
55	30	2	0	1.00000E+00	1.00000E+00	2.00000E-01	2.00000E-01
65	31	2	0	1.00000E+00	1.00000E+00	2.00000E-01	2.00000E-01
75	32	2	0	1.00000E+00	1.00000E+00	2.00000E-01	2.00000E-01
85	33	2	0	1.00000E+00	1.00000E+00	2.00000E-01	2.00000E-01

Fig. 3. Heat regenerator. Sample problem for TRUMP report (continued).

95	34	2	0	1.00000E+00	1.00000E+00	2.00000E-01	2.00000E-01
105	35	2	0	1.00000E+00	1.00000E+00	2.00000E-01	2.00000E-01
115	36	2	0	1.00000E+00	1.00000E+00	2.00000E-01	2.00000E-01
125	37	2	0	1.00000E+00	1.00000E+00	2.00000E-01	2.00000E-01
135	38	2	0	1.00000E+00	1.00000E+00	2.00000E-01	2.00000E-01
145	39	2	0	1.00000E+00	1.00000E+00	2.00000E-01	2.00000E-01
155	40	2	0	1.00000E+00	1.00000E+00	2.00000E-01	2.00000E-01
165	41	2	0	1.00000E+00	1.00000E+00	2.00000E-01	2.00000E-01
175	42	2	0	1.00000E+00	1.00000E+00	2.00000E-01	2.00000E-01
185	43	2	0	1.00000E+00	1.00000E+00	2.00000E-01	2.00000E-01
195	44	2	0	1.00000E+00	1.00000E+00	2.00000E-01	2.00000E-01
205	45	2	0	1.00000E+00	1.00000E+00	2.00000E-01	2.00000E-01
200	46	2	2	0.	0.	0.	1.00000E-24

NODE 199 CONNECTED TO 196 AND 206 TO GET TEMP AT Z = 20.0.
NODE 200 CONNECTED TO 195 AND 205 TO GET TEMP AT Z = 20.0.

DATA BLCCK 5 INTERNAL CONNECTION NODE NUMBERS, AREAS, AND INTERFACE CONDUCTANCES.

NOD1	NOD2	INDEX	DEL1	DEL2	DLONG	DRAD	HINT	RINT	AREA
2	1	1	0.	0.	1.0000E-04	1.0000E+00	2.0000E-02	0.	1.0000E-04
6	5	2	0.	0.	1.0000E+00	1.0000E+00	2.0000E-02	0.	1.0000E+00
16	15	3	0.	0.	1.0000E+00	1.0000E+00	2.0000E-02	0.	1.0000E+00
26	25	4	0.	0.	1.0000E+00	1.0000E+00	2.0000E-02	0.	1.0000E+00
36	35	5	0.	0.	1.0000E+00	1.0000E+00	2.0000E-02	0.	1.0000E+00
46	45	6	0.	0.	1.0000E+00	1.0000E+00	2.0000E-02	0.	1.0000E+00
56	55	7	0.	0.	1.0000E+00	1.0000E+00	2.0000E-02	0.	1.0000E+00
66	65	8	0.	0.	1.0000E+00	1.0000E+00	2.0000E-02	0.	1.0000E+00
76	75	9	0.	0.	1.0000E+00	1.0000E+00	2.0000E-02	0.	1.0000E+00
86	85	10	0.	0.	1.0000E+00	1.0000E+00	2.0000E-02	0.	1.0000E+00
96	95	11	0.	0.	1.0000E+00	1.0000E+00	2.0000E-02	0.	1.0000E+00
106	105	12	0.	0.	1.0000E+00	1.0000E+00	2.0000E-02	0.	1.0000E+00
116	115	13	0.	0.	1.0000E+00	1.0000E+00	2.0000E-02	0.	1.0000E+00
126	125	14	0.	0.	1.0000E+00	1.0000E+00	2.0000E-02	0.	1.0000E+00
136	135	15	0.	0.	1.0000E+00	1.0000E+00	2.0000E-02	0.	1.0000E+00
146	145	16	0.	0.	1.0000E+00	1.0000E+00	2.0000E-02	0.	1.0000E+00
156	155	17	0.	0.	1.0000E+00	1.0000E+00	2.0000E-02	0.	1.0000E+00
166	165	18	0.	0.	1.0000E+00	1.0000E+00	2.0000E-02	0.	1.0000E+00
176	175	19	0.	0.	1.0000E+00	1.0000E+00	2.0000E-02	0.	1.0000E+00
186	185	20	0.	0.	1.0000E+00	1.0000E+00	2.0000E-02	0.	1.0000E+00
196	195	21	0.	0.	1.0000E+00	1.0000E+00	2.0000E-02	0.	1.0000E+00
206	205	22	0.	0.	1.0000E+00	1.0000E+00	2.0000E-02	0.	1.0000E+00
199	196	23	0.	0.	1.0000E+00	1.0000E+00	5.5000E-09	0.	1.0000E+00

Fig. 3. Heat regenerator. Sample problem for TRUMP report (continued).

199	206	24	0.	0.	1.0000E+00	1.0000E+00	3.5000E-09	0.	1.0000E+00
200	195	25	0.	0.	1.0000E+00	1.0000E+00	3.5000E-09	0.	1.0000E+00
200	205	26	0.	0.	1.0000E+00	1.0000E+00	3.5000E-09	0.	1.0000E+00

DATA BLOCK 6 EXTERNAL CONNECTION NODE NUMBERS, AREAS, AND SURFACE CONDUCTANCES.

NODS	NODSB	INDEX	LTABF	POWER	DLONG	DRAD	HSURE	RSURE	AREAS
2	1001	1	0	0.	1.0000E+00	2.0000E-01	1.0000E+03	0.	2.0000E-01

DATA BLOCK 7 BOUNDARY NODE TEMPERATURES.

NODB	INDEX	LTABT	TEMPB	SLOPE	TIMEB
1001	1	0	1.000000E+02		0.

NODAL POINT OF NODE 2 IS AT Z = 0.
OTHER NODAL POINTS OFFSET DOWNSTREAM FOR ADDED MASS FLOW STABILITY.

DATA BLOCK 10 MASS FLOW CONNECTION NODE NUMBERS, FLOW RATES, PATH LENGTHS.

NODF1	NODF2	INDEX	LTABFL	FLOWT	SLOFL	TVARFL	DELF1	DELF2
2	6	1	0	4.000000E-01			1.000000E-04	6.500000E-01
6	16	2	0	4.000000E-01			3.500000E-01	6.500000E-01
16	26	3	0	4.000000E-01			3.500000E-01	6.500000E-01
26	36	4	0	4.000000E-01			3.500000E-01	6.500000E-01
36	46	5	0	4.000000E-01			3.500000E-01	6.500000E-01
46	56	6	0	4.000000E-01			3.500000E-01	6.500000E-01
56	66	7	0	4.000000E-01			3.500000E-01	6.500000E-01
66	76	8	0	4.000000E-01			3.500000E-01	6.500000E-01
76	86	9	0	4.000000E-01			3.500000E-01	6.500000E-01
86	96	10	0	4.000000E-01			3.500000E-01	6.500000E-01
96	106	11	0	4.000000E-01			3.500000E-01	6.500000E-01
106	116	12	0	4.000000E-01			3.500000E-01	6.500000E-01
116	126	13	0	4.000000E-01			3.500000E-01	6.500000E-01
126	136	14	0	4.000000E-01			3.500000E-01	6.500000E-01
136	146	15	0	4.000000E-01			3.500000E-01	6.500000E-01
146	156	16	0	4.000000E-01			3.500000E-01	6.500000E-01
156	166	17	0	4.000000E-01			3.500000E-01	6.500000E-01
166	176	18	0	4.000000E-01			3.500000E-01	6.500000E-01
176	186	19	0	4.000000E-01			3.500000E-01	6.500000E-01
186	196	20	0	4.000000E-01			3.500000E-01	6.500000E-01
196	206	21	0	4.000000E-01			3.500000E-01	6.500000E-01

Fig. 3. Heat regenerator. Sample problem for TRUMP report (continued).

```
==========================================================================================
DATA ENDEC -1                    END OF DATA DECK.  SSI WILL INTERRUPT PROBLEM.
==========================================================================================
==========================================================================================
MACHINE TIME (NO4/10) =      3  SECONDS.  CPU,I/O,SYS PERCENTS =   32/67/0

SUMMARY OF INPUT DATA
BLOCK NUMBER          2     3     4     5     6     7     8     9    10    11    12
TIMES READ IN         1     0   001   001   001     1   000   000   001   001   000

ITEM I   E          MAT   KEM  NODE  NOD1  NODS  NODB  NODG  NOTE NODF1 NODEP NODP1
MAXIMUM SIZE         40     5   480  1200   125    20   100   480    80    12   480
UNMODIFIED SIZE       0     0     0     0     0     0     0     0     0     0     0
FINAL SIZE            2     0    46    26     1     1     0     0    21    10     0

TABLES             CAPT  CONT    QT    ZT    ET HSURT TEMPB    GT FLOWT TOTAL
                      0     0     0     0     0     0     0     0     0     0

                MAXIMUM ALLOWED TABLE LENGTH IS  12.

ARRAY STORAGE REQUIREMENTS.
     NWREQD = (8 + 7*M9)*M2 + 28*M4 + 12*M5 + 3*M9      =     31556
     NWCHEM = 4*M2 + (6 + 9*M9)*M3 + 10*M4 (M3 NOT 0)   =      5530
     NWGEN8 = 2*M4 + (3 + 3*M9)*M8          (M8 NOT 0)  =      4860
     NWFLOW = 4*M4 + (9 + 3*M9)*M10        (M10 NOT 0)  =      5520
     NWSURE = (12 + 3*M9)*M6 + (5+3*M9)*M7 (M6 NOT 0)   =      6820
     NWPLOT = 4*M4 + 3*M11 + (1 + M11)*M1 (M11 NOT 0)   =      4556
     NWBL12 = 8*M4 + 5*M12                 (M12 NOT 0)  =      6240

     TOTAL ADJUSTABLE ARRAY STORAGE                     =     65082

OTHER TOTALS     NOSPEC NOGEN NORAD NORADS NMELT NREACT
                      2     0     0     0     0     0

MATERIAL SUMMARY

  NAME   MATL NODES   DENSITY      CAPACITY      TOT VOL       TOT MASS      TOT CAP       TOT HEAT
FLUID      1    23  1.00000E+00  1.00000E+00  4.20002E+00  4.20002E+00  4.20002E+00  0.
SOLID      2    23  1.00000E+01  5.00000E-02  4.20002E+00  4.20002E+01  2.10001E+00  0.

SYSTEM TOTAL    46                            8.40004E+00  4.62002E+01  6.30003E+00  0.
```

Fig. 3. Heat regenerator. Sample problem for TRUMP report (continued).

TRUMP TPUT DATA DATA CK 1

* HEAT REGENERATOR. SAMPLE PROBLEM FOR TRUMP REPORT. 2/27/68. ** 20.141 NC 4/10

PRINTOUT	CYCLE	MF	MSS	KWIT	DELTMX	SMALL	TVARY	DTEMP	DTMAXS	NUTS
1	0	1	0	0	3.17460E-01	3.17460E-03	1.00000E+00	9.99990E-04	0.	0

TOTAL TIME	TIME STEP	HEAT FLOW	TEMP FROM FLUX	FLUX RATE	TEMP RATE
1.00000E-12	1.00000E-12	1.99998E-08	3.17456E-09	1.99998E+04	3.17456E+03

AVG TEMP	HEAT CAPACITY	HEAT CONTENT	GEN RATE	HEAT GEN	TEMP FROM GEN
3.17456E-09	6.30003E+00	1.99998E-08	0.	0.	0.

NODE	TEMP	DT	DDT	GEN RATE	W	H	F
2	9.9999E-04	9.9999E-04	9.9998E+08	0.	2.0000E-08	2.0000E-08	2.0000E-08
6	0.	0.	0.	0.	0.	0.	0.
16	0.	0.	0.	0.	0.	0.	0.
26	0.	0.	0.	0.	0.	0.	0.
36	0.	0.	0.	0.	0.	0.	0.
46	0.	0.	0.	0.	0.	0.	0.
56	0.	0.	0.	0.	0.	0.	0.
66	0.	0.	0.	0.	0.	0.	0.
76	0.	0.	0.	0.	0.	0.	0.
86	0.	0.	0.	0.	0.	0.	0.
96	0.	0.	0.	0.	0.	0.	0.
106	0.	0.	0.	0.	0.	0.	0.
116	0.	0.	0.	0.	0.	0.	0.
126	0.	0.	0.	0.	0.	0.	0.
136	0.	0.	0.	0.	0.	0.	0.
146	0.	0.	0.	0.	0.	0.	0.
156	0.	0.	0.	0.	0.	0.	0.
166	0.	0.	0.	0.	0.	0.	0.
176	0.	0.	0.	0.	0.	0.	0.
186	0.	0.	0.	0.	0.	0.	0.
196	0.	0.	0.	0.	0.	0.	0.
206	0.	0.	0.	0.	0.	0.	0.
199	0.	0.	0.	0.	0.	0.	0.
1	2.0000E-16	2.0000E-16	2.0000E-04	0.	2.0000E-21	2.0000E-21	2.0000E-21
5	0.	0.	0.	0.	0.	0.	0.
15	0.	0.	0.	0.	0.	0.	0.

Fig. 3. Heat regenerator. Sample problem for TRUMP report (continued).

25	0.	0.	0.	0.	0.	0.	0.
35	0.	0.	0.	0.	0.	0.	0.
45	0.	0.	0.	0.	0.	0.	0.
55	0.	0.	0.	0.	0.	0.	0.
65	0.	0.	0.	0.	0.	0.	0.
75	0.	0.	0.	0.	0.	0.	0.
85	0.	0.	0.	0.	0.	0.	0.
95	0.	0.	0.	0.	0.	0.	0.
105	0.	0.	0.	0.	0.	0.	0.
115	0.	0.	0.	0.	0.	0.	0.
125	0.	0.	0.	0.	0.	0.	0.
135	0.	0.	0.	0.	0.	0.	0.
145	0.	0.	0.	0.	0.	0.	0.
155	0.	0.	0.	0.	0.	0.	0.
165	0.	0.	0.	0.	0.	0.	0.
175	0.	0.	0.	0.	0.	0.	0.
185	0.	0.	0.	0.	0.	0.	0.
195	0.	0.	0.	0.	0.	0.	0.

NODE	TEMP	DT	DDT	GEN RATE	W	H	F
205	0.	0.	0.	0.	0.	0.	0.
200	0.	0.	0.	0.	0.	0.	0.

MATERIAL DATA

NAME	MATL	TOT CAP	TOT HEAT	AVG TEMP	TMELT	HMELT
FLUID	1	4.20002E+00	1.99998E-08	4.76183E-09	0.	0.
SOLID	2	2.10001E+00	1.99998E-21	9.52367E-22	0.	0.

NODE DATA. CHECK TOTAL CONDUCTANCES (ZIP) AND TIME CONSTANTS (SLIM).
LARGE DIFFERENCES BETWEEN NODES MAY BE DUE TO POOR ZONING, AND MAY PRODUCE POOR RESULTS.

NODE	MATL	NTYPE	RADIUS	VOLUME	MASS	CAPACITY	CONDUCTIVITY	ZIP	SLIM
2	1	1	2.0000E-01	2.0000E-05	2.0000E-05	2.0000E-05	1.0000E-24	2.0000E+02	1.0000E-07
6	1	0	2.0000E-01	2.0000E-01	2.0000E-01	2.0000E-01	1.0000E-24	4.2000E-01	4.7619E-01
16	1	0	2.0000E-01	2.0000E-01	2.0000E-01	2.0000E-01	1.0000E-24	4.2000E-01	4.7619E-01
26	1	0	2.0000E-01	2.0000E-01	2.0000E-01	2.0000E-01	1.0000E-24	4.2000E-01	4.7619E-01
36	1	0	2.0000E-01	2.0000E-01	2.0000E-01	2.0000E-01	1.0000E-24	4.2000E-01	4.7619E-01
46	1	0	2.0000E-01	2.0000E-01	2.0000E-01	2.0000E-01	1.0000E-24	4.2000E-01	4.7619E-01
56	1	0	2.0000E-01	2.0000E-01	2.0000E-01	2.0000E-01	1.0000E-24	4.2000E-01	4.7619E-01
66	1	0	2.0000E-01	2.0000E-01	2.0000E-01	2.0000E-01	1.0000E-24	4.2000E-01	4.7619E-01
76	1	0	2.0000E-01	2.0000E-01	2.0000E-01	2.0000E-01	1.0000E-24	4.2000E-01	4.7619E-01

Fig. 3. Heat regenerator. Sample problem for TRUMP report (continued).

86	1	0	2.0000E-01	2.0000E-01	2.0000E-01	2.0000E-01	1.0000E-24	4.2000E-01	4.7619E-01
96	1	0	2.0000E-01	2.0000E-01	2.0000E-01	2.0000E-01	1.0000E-24	4.2000E-01	4.7619E-01
106	1	0	2.0000E-01	2.0000E-01	2.0000E-01	2.0000E-01	1.0000E-24	4.2000E-01	4.7619E-01
116	1	0	2.0000E-01	2.0000E-01	2.0000E-01	2.0000E-01	1.0000E-24	4.2000E-01	4.7619E-01
126	1	0	2.0000E-01	2.0000E-01	2.0000E-01	2.0000E-01	1.0000E-24	4.2000E-01	4.7619E-01
136	1	0	2.0000E-01	2.0000E-01	2.0000E-01	2.0000E-01	1.0000E-24	4.2000E-01	4.7619E-01
146	1	0	2.0000E-01	2.0000E-01	2.0000E-01	2.0000E-01	1.0000E-24	4.2000E-01	4.7619E-01
156	1	0	2.0000E-01	2.0000E-01	2.0000E-01	2.0000E-01	1.0000E-24	4.2000E-01	4.7619E-01
166	1	0	2.0000E-01	2.0000E-01	2.0000E-01	2.0000E-01	1.0000E-24	4.2000E-01	4.7619E-01
176	1	0	2.0000E-01	2.0000E-01	2.0000E-01	2.0000E-01	1.0000E-24	4.2000E-01	4.7619E-01
186	1	0	2.0000E-01	2.0000E-01	2.0000E-01	2.0000E-01	1.0000E-24	4.2000E-01	4.7619E-01
196	1	0	2.0000E-01	2.0000E-01	2.0000E-01	2.0000E-01	1.0000E-24	4.2000E-01	4.7619E-01
206	1	0	2.0000E-01	2.0000E-01	2.0000E-01	2.0000E-01	1.0000E-24	4.2000E-01	4.7619E-01
199	1	2	0.	1.0000E-24	1.0000E-24	1.0000E-24	1.0000E-24	1.0000E-18	1.0000E-16
1	2	0	2.0000E-01	2.0000E-05	2.0000E-04	1.0000E-05	1.0000E-24	2.0000E-06	5.0000E+00
5	2	0	2.0000E-01	2.0000E-01	2.0000E+00	1.0000E-01	1.0000E-24	2.0000E-02	5.0000E+00
15	2	0	2.0000E-01	2.0000E-01	2.0000E+00	1.0000E-01	1.0000E-24	2.0000E-02	5.0000E+00
25	2	0	2.0000E-01	2.0000E-01	2.0000E+00	1.0000E-01	1.0000E-24	2.0000E-02	5.0000E+00
35	2	0	2.0000E-01	2.0000E-01	2.0000E+00	1.0000E-01	1.0000E-24	2.0000E-02	5.0000E+00
45	2	0	2.0000E-01	2.0000E-01	2.0000E+00	1.0000E-01	1.0000E-24	2.0000E-02	5.0000E+00
55	2	0	2.0000E-01	2.0000E-01	2.0000E+00	1.0000E-01	1.0000E-24	2.0000E-02	5.0000E+00
65	2	0	2.0000E-01	2.0000E-01	2.0000E+00	1.0000E-01	1.0000E-24	2.0000E-02	5.0000E+00
75	2	0	2.0000E-01	2.0000E-01	2.0000E+00	1.0000E-01	1.0000E-24	2.0000E-02	5.0000E+00
85	2	0	2.0000E-01	2.0000E-01	2.0000E+00	1.0000E-01	1.0000E-24	2.0000E-02	5.0000E+00
95	2	0	2.0000E-01	2.0000E-01	2.0000E+00	1.0000E-01	1.0000E-24	2.0000E-02	5.0000E+00
105	2	0	2.0000E-01	2.0000E-01	2.0000E+00	1.0000E-01	1.0000E-24	2.0000E-02	5.0000E+00
115	2	0	2.0000E-01	2.0000E-01	2.0000E+00	1.0000E-01	1.0000E-24	2.0000E-02	5.0000E+00
125	2	0	2.0000E-01	2.0000E-01	2.0000E+00	1.0000E-01	1.0000E-24	2.0000E-02	5.0000E+00
135	2	0	2.0000E-01	2.0000E-01	2.0000E+00	1.0000E-01	1.0000E-24	2.0000E-02	5.0000E+00
145	2	0	2.0000E-01	2.0000E-01	2.0000E+00	1.0000E-01	1.0000E-24	2.0000E-02	5.0000E+00
155	2	0	2.0000E-01	2.0000E-01	2.0000E+00	1.0000E-01	1.0000E-24	2.0000E-02	5.0000E+00
165	2	0	2.0000E-01	2.0000E-01	2.0000E+00	1.0000E-01	1.0000E-24	2.0000E-02	5.0000E+00
175	2	0	2.0000E-01	2.0000E-01	2.0000E+00	1.0000E-01	1.0000E-24	2.0000E-02	5.0000E+00
185	2	0	2.0000E-01	2.0000E-01	2.0000E+00	1.0000E-01	1.0000E-24	2.0000E-02	5.0000E+00
195	2	0	2.0000E-01	2.0000E-01	2.0000E+00	1.0000E-01	1.0000E-24	2.0000E-02	5.0000E+00
205	2	0	2.0000E-01	2.0000E-01	2.0000E+00	1.0000E-01	1.0000E-24	2.0000E-02	5.0000E+00
200	2	2	0.	1.0000E-24	1.0000E-23	5.0000E-25	1.0000E-24	1.0000E-18	5.0000E-17

Fig. 3. Heat regenerator. Sample problem for TRUMP report (continued).

INTERNAL CONNECTION DATA. CHECK CONDUCTANCES (TRAN).
LARGE DIFFERENCES BETWEEN CONNECTIONS MAY BE DUE TO POOR ZONING, AND MAY PRODUCE POOR RESULTS.

NOD1	NOD2	AREA	FINT	RINT	TRAN	HEAT FLOW	AVG RATE
2	1	1.0000E-04	2.0000E-02	0.	2.0000E-06	-2.0000E-21	-2.0000E-09
6	5	1.0000E+00	2.0000E-02	0.	2.0000E-02	0.	0.
16	15	1.0000E+00	2.0000E-02	0.	2.0000E-02	0.	0.
26	25	1.0000E+00	2.0000E-02	0.	2.0000E-02	0.	0.
36	35	1.0000E+00	2.0000E-02	0.	2.0000E-02	0.	0.
46	45	1.0000E+00	2.0000E-02	0.	2.0000E-02	0.	0.
56	55	1.0000E+00	2.0000E-02	0.	2.0000E-02	0.	0.
66	65	1.0000E+00	2.0000E-02	0.	2.0000E-02	0.	0.
76	75	1.0000E+00	2.0000E-02	0.	2.0000E-02	0.	0.
86	85	1.0000E+00	2.0000E-02	0.	2.0000E-02	0.	0.
96	95	1.0000E+00	2.0000E-02	0.	2.0000E-02	0.	0.
106	105	1.0000E+00	2.0000E-02	0.	2.0000E-02	0.	0.
116	115	1.0000E+00	2.0000E-02	0.	2.0000E-02	0.	0.
126	125	1.0000E+00	2.0000E-02	0.	2.0000E-02	0.	0.
136	135	1.0000E+00	2.0000E-02	0.	2.0000E-02	0.	0.
146	145	1.0000E+00	2.0000E-02	0.	2.0000E-02	0.	0.
156	155	1.0000E+00	2.0000E-02	0.	2.0000E-02	0.	0.
166	165	1.0000E+00	2.0000E-02	0.	2.0000E-02	0.	0.
176	175	1.0000E+00	2.0000E-02	0.	2.0000E-02	0.	0.
186	185	1.0000E+00	2.0000E-02	0.	2.0000E-02	0.	0.
196	195	1.0000E+00	2.0000E-02	0.	2.0000E-02	0.	0.
206	205	1.0000E+00	2.0000E-02	0.	2.0000E-02	0.	0.
199	196	1.0000E+00	6.5000E-09	0.	6.5000E-09	0.	0.
199	206	1.0000E+00	3.5000E-09	0.	3.5000E-09	0.	0.
200	195	1.0000E+00	6.5000E-09	0.	6.5000E-09	0.	0.
200	205	1.0000E+00	3.5000E-09	0.	3.5000E-09	0.	0.

MASS FLOW DATA

SOURCE	SINK	FLOW RATE	NET FLOW	AVG RATE	SOURCE WT	SINK WT
2	6	4.00000E-01	4.00000E-13	4.00000E-01	9.99846E-01	1.53822E-04
6	16	4.00000E-01	4.00000E-13	4.00000E-01	6.50000E-01	3.50000E-01
16	26	4.00000E-01	4.00000E-13	4.00000E-01	6.50000E-01	3.50000E-01
26	36	4.00000E-01	4.00000E-13	4.00000E-01	6.50000E-01	3.50000E-01
36	46	4.00000E-01	4.00000E-13	4.00000E-01	6.50000E-01	3.50000E-01
46	56	4.00000E-01	4.00000E-13	4.00000E-01	6.50000E-01	3.50000E-01
56	66	4.00000E-01	4.00000E-13	4.00000E-01	6.50000E-01	3.50000E-01

Fig. 3. Heat regenerator. Sample problem for TRUMP report (continued).

66	76	4.00000E-01	4.00000E-13	4.00000E-01	6.50000E-01	3.50000E-01
76	86	4.00000E-01	4.00000E-13	4.00000E-01	6.50000E-01	3.50000E-01
86	96	4.00000E-01	4.00000E-13	4.00000E-01	6.50000E-01	3.50000E-01
96	106	4.00000E-01	4.00000E-13	4.00000E-01	6.50000E-01	3.50000E-01
106	116	4.00000E-01	4.00000E-13	4.00000E-01	6.50000E-01	3.50000E-01
116	126	4.00000E-01	4.00000E-13	4.00000E-01	6.50000E-01	3.50000E-01
126	136	4.00000E-01	4.00000E-13	4.00000E-01	6.50000E-01	3.50000E-01
136	146	4.00000E-01	4.00000E-13	4.00000E-01	6.50000E-01	3.50000E-01
146	156	4.00000E-01	4.00000E-13	4.00000E-01	6.50000E-01	3.50000E-01
156	166	4.00000E-01	4.00000E-13	4.00000E-01	6.50000E-01	3.50000E-01
166	176	4.00000E-01	4.00000E-13	4.00000E-01	6.50000E-01	3.50000E-01
176	186	4.00000E-01	4.00000E-13	4.00000E-01	6.50000E-01	3.50000E-01
186	196	4.00000E-01	4.00000E-13	4.00000E-01	6.50000E-01	3.50000E-01
196	206	4.00000E-01	4.00000E-13	4.00000E-01	6.50000E-01	3.50000E-01

NODE	RATE IN	RATE OUT	NET FLOW IN	NET FLOW OUT	AVG FLOW IN	AVG FLOW OUT	RESIDENCE TIME
2	0.	4.00000E-01	0.	4.00000E-13	0.	4.00000E-01	5.00000E-05
6	4.00000E-01	4.00000E-01	4.00000E-13	4.00000E-13	4.00000E-01	4.00000E-01	5.00000E-01
16	4.00000E-01	4.00000E-01	4.00000E-13	4.00000E-13	4.00000E-01	4.00000E-01	5.00000E-01
26	4.00000E-01	4.00000E-01	4.00000E-13	4.00000E-13	4.00000E-01	4.00000E-01	5.00000E-01
36	4.00000E-01	4.00000E-01	4.00000E-13	4.00000E-13	4.00000E-01	4.00000E-01	5.00000E-01
46	4.00000E-01	4.00000E-01	4.00000E-13	4.00000E-13	4.00000E-01	4.00000E-01	5.00000E-01
56	4.00000E-01	4.00000E-01	4.00000E-13	4.00000E-13	4.00000E-01	4.00000E-01	5.00000E-01
66	4.00000E-01	4.00000E-01	4.00000E-13	4.00000E-13	4.00000E-01	4.00000E-01	5.00000E-01
76	4.00000E-01	4.00000E-01	4.00000E-13	4.00000E-13	4.00000E-01	4.00000E-01	5.00000E-01
86	4.00000E-01	4.00000E-01	4.00000E-13	4.00000E-13	4.00000E-01	4.00000E-01	5.00000E-01
96	4.00000E-01	4.00000E-01	4.00000E-13	4.00000E-13	4.00000E-01	4.00000E-01	5.00000E-01
106	4.00000E-01	4.00000E-01	4.00000E-13	4.00000E-13	4.00000E-01	4.00000E-01	5.00000E-01
116	4.00000E-01	4.00000E-01	4.00000E-13	4.00000E-13	4.00000E-01	4.00000E-01	5.00000E-01
126	4.00000E-01	4.00000E-01	4.00000E-13	4.00000E-13	4.00000E-01	4.00000E-01	5.00000E-01
136	4.00000E-01	4.00000E-01	4.00000E-13	4.00000E-13	4.00000E-01	4.00000E-01	5.00000E-01
146	4.00000E-01	4.00000E-01	4.00000E-13	4.00000E-13	4.00000E-01	4.00000E-01	5.00000E-01
156	4.00000E-01	4.00000E-01	4.00000E-13	4.00000E-13	4.00000E-01	4.00000E-01	5.00000E-01
166	4.00000E-01	4.00000E-01	4.00000E-13	4.00000E-13	4.00000E-01	4.00000E-01	5.00000E-01
176	4.00000E-01	4.00000E-01	4.00000E-13	4.00000E-13	4.00000E-01	4.00000E-01	5.00000E-01
186	4.00000E-01	4.00000E-01	4.00000E-13	4.00000E-13	4.00000E-01	4.00000E-01	5.00000E-01
196	4.00000E-01	4.00000E-01	4.00000E-13	4.00000E-13	4.00000E-01	4.00000E-01	5.00000E-01
206	4.00000E-01	0.	4.00000E-13	0.	4.00000E-01	0.	5.00000E-01

BOUNDARY NODE DATA

NODB	TEMPB	HEAT FLOW	AVG RATE
1001	1.0000E+02	2.0000E-08	2.0000E+04

Fig. 3. Heat regenerator. Sample problem for TRUMP report (continued).

SYSTEM TOTAL 2.0000E-08 2.0000E+04

EXTERNAL CONNECTION DATA

NODS	NODSB	AREAS	HSURE	POWER	RSURE	TRANS	HEAT FLOW	AVG RATE
2	1001	2.0000E-01	1.0000E+03	0.	0.	2.0000E+02	2.0000E-08	2.0000E+04

KCYC = 0, ITERATIONS... TOTAL = 0, AVERAGE = 0, MAXIMUM = 0. FOR = 0.570 .

* HEAT REGENERATOR. SAMPLE PROBLEM FOR TRUMP REPORT. 2/27/68. ** 20.141 N04/10

PRINTOUT	CYCLE	MF	MSS	KWIT	DELTMX	SMALL	TVARY	DTEMP	DTMAXS	NUTS
6	461	3	0	1	3.17460E-01	3.17460E-03	1.00000E+00	2.27034E-01	0.	0

TOTAL TIME	TIME STEP	HEAT FLOW	TEMP FROM FLUX	FLUX RATE	TEMP RATE
2.00000E+01	5.16402E-02	1.13220E-06	1.79713E-07	5.66099E-08	8.98565E-09

AVG TEMP	HEAT CAPACITY	HEAT CONTENT	GEN RATE	HEAT GEN	TEMP FROM GEN
8.99629E+01	6.30003E+00	5.66769E+02	0.	0.	0.

NODE	TEMP	DT	DDT	GEN RATE	W	H	F
2	1.0000E+02	-6.9351E-10	-1.3386E-08	0.	2.0000E-03	2.0000E-03	-9.8045E-04
6	9.9937E+01	6.4987E-04	1.2543E-02	0.	1.9987E+01	1.9987E+01	-9.7869E+00
16	9.9804E+01	1.9693E-03	3.8011E-02	0.	1.9961E+01	1.9961E+01	-9.7208E+00
26	9.9627E+01	3.6408E-03	7.0272E-02	0.	1.9925E+01	1.9925E+01	-9.6441E+00
36	9.9400E+01	5.7188E-03	1.1038E-01	0.	1.9880E+01	1.9880E+01	-9.5558E+00
46	9.9116E+01	8.2655E-03	1.5954E-01	0.	1.9823E+01	1.9823E+01	-9.4552E+00
56	9.8766E+01	1.1299E-02	2.1810E-01	0.	1.9753E+01	1.9753E+01	-9.3415E+00
66	9.8341E+01	1.4812E-02	2.8589E-01	0.	1.9668E+01	1.9668E+01	-9.2140E+00
76	9.7837E+01	1.8875E-02	3.6432E-01	0.	1.9567E+01	1.9567E+01	-9.0722E+00
86	9.7240E+01	2.3682E-02	4.5711E-01	0.	1.9448E+01	1.9448E+01	-8.9153E+00
96	9.6544E+01	2.9158E-02	5.6278E-01	0.	1.9309E+01	1.9309E+01	-8.7432E+00
106	9.5734E+01	3.5166E-02	6.7875E-01	0.	1.9147E+01	1.9147E+01	-8.5549E+00
116	9.4819E+01	4.1575E-02	8.0247E-01	0.	1.8964E+01	1.8964E+01	-8.3517E+00
126	9.3762E+01	4.8811E-02	9.4212E-01	0.	1.8752E+01	1.8752E+01	-8.1304E+00
136	9.2606E+01	5.7446E-02	1.1088E+00	0.	1.8521E+01	1.8521E+01	-7.8969E+00
146	9.1228E+01	6.6858E-02	1.2905E+00	0.	1.8246E+01	1.8246E+01	-7.6407E+00
156	8.9833E+01	7.5790E-02	1.4629E+00	0.	1.7967E+01	1.7967E+01	-7.3824E+00
166	8.8054E+01	8.5715E-02	1.6544E+00	0.	1.7611E+01	1.7611E+01	-7.0847E+00
176	8.6542E+01	9.3594E-02	1.8065E+00	0.	1.7308E+01	1.7308E+01	-6.8193E+00

Fig. 3. Heat regenerator. Sample problem for TRUMP report (continued).

186	8.4155E+01	1.0856E-01	2.0954E+00	0.	1.6831E+01	1.6831E+01	-6.4546E+00
196	8.2811E+01	1.2055E-01	2.3267E+00	0.	1.6562E+01	1.6562E+01	-6.2362E+00
206	7.9078E+01	1.3825E-01	2.6684E+00	0.	1.5816E+01	1.5816E+01	-5.7184E+00
199	8.1504E+01	1.2674E-01	2.4463E+00	0.	8.1504E-23	8.1504E-23	2.4431E-22
1	9.8158E+01	1.9106E-02	3.6877E-01	0.	9.8158E-04	9.8158E-04	9.8158E-04
5	9.7869E+01	2.1577E-02	4.1646E-01	0.	9.7869E+00	9.7869E+00	9.7869E+00
15	9.7208E+01	2.7064E-02	5.2238E-01	0.	9.7208E+00	9.7208E+00	9.7208E+00
25	9.6441E+01	3.3213E-02	6.4106E-01	0.	9.6441E+00	9.6441E+00	9.6441E+00
35	9.5558E+01	4.0038E-02	7.7280E-01	0.	9.5558E+00	9.5558E+00	9.5558E+00
45	9.4552E+01	4.7541E-02	9.1762E-01	0.	9.4552E+00	9.4552E+00	9.4552E+00
55	9.3415E+01	5.5720E-02	1.0755E+00	0.	9.3415E+00	9.3415E+00	9.3415E+00
65	9.2140E+01	6.4559E-02	1.2461E+00	0.	9.2140E+00	9.2140E+00	9.2140E+00
75	9.0722E+01	7.4053E-02	1.4293E+00	0.	9.0722E+00	9.0722E+00	9.0722E+00
85	8.9153E+01	8.4145E-02	1.6241E+00	0.	8.9153E+00	8.9153E+00	8.9153E+00
95	8.7432E+01	9.4789E-02	1.8296E+00	0.	8.7432E+00	8.7432E+00	8.7432E+00
105	8.5549E+01	1.0592E-01	2.0444E+00	0.	8.5549E+00	8.5549E+00	8.5549E+00
115	8.3517E+01	1.1751E-01	2.2682E+00	0.	8.3517E+00	8.3517E+00	8.3517E+00
125	8.1304E+01	1.2949E-01	2.4994E+00	0.	8.1304E+00	8.1304E+00	8.1304E+00
135	7.8969E+01	1.4172E-01	2.7353E+00	0.	7.8969E+00	7.8969E+00	7.8969E+00
145	7.6407E+01	1.5398E-01	2.9720E+00	0.	7.6407E+00	7.6407E+00	7.6407E+00
155	7.3824E+01	1.6628E-01	3.2094E+00	0.	7.3824E+00	7.3824E+00	7.3824E+00
165	7.0847E+01	1.7868E-01	3.4487E+00	0.	7.0847E+00	7.0847E+00	7.0847E+00
175	6.8193E+01	1.9051E-01	3.6772E+00	0.	6.8193E+00	6.8193E+00	6.8193E+00
185	6.4546E+01	2.0350E-01	3.9279E+00	0.	6.4546E+00	6.4546E+00	6.4546E+00
195	6.2362E+01	2.1214E-01	4.0946E+00	0.	6.2362E+00	6.2362E+00	6.2362E+00
NODE	TEMP	DT	DDT	GEN RATE	W	H	F
205	5.7184E+01	2.2703E-01	4.3821E+00	0.	5.7184E+00	5.7184E+00	5.7184E+00
200	6.0550E+01	2.1735E-01	4.1952E+00	0.	3.0275E-23	3.0275E-23	2.0603E-22

MATERIAL DATA

NAME	MATL	TOT CAP	TOT HEAT	AVG TEMP	TMELT	HMELT
FLUID	1	4.20002E+00	3.93049E+02	9.35825E+01	0.	0.
SOLID	2	2.10001E+00	1.73720E+02	8.27235E+01	0.	0.

INTERNAL CONNECTION DATA. CHECK CONDUCTANCES (TRAN).
LARGE DIFFERENCES BETWEEN CONNECTIONS MAY BE DUE TO POOR ZONING, AND MAY PRODUCE POOR RESULTS.

NOD1	NOD2	AREA	HINT	RINT	TRAN	HEAT FLOW	AVG RATE
2	1	1.0000E-04	2.0000E-02	0.	2.0000E-06	-9.8158E-04	-4.9079E-05
6	5	1.0000E+00	2.0000E-02	0.	2.0000E-02	-9.7869E+00	-4.8934E-01
16	15	1.0000E+00	2.0000E-02	0.	2.0000E-02	-9.7208E+00	-4.8604E-01

Fig. 3. Heat regenerator. Sample problem for TRUMP report (continued).

26	25	1.0000E+00	2.0000E-02	0.	2.0000E-02	-9.6441E+00	-4.8223E-01
36	35	1.0000E+00	2.0000E-02	0.	2.0000E-02	-9.5558E+00	-4.7779E-01
46	45	1.0000E+00	2.0000E-02	0.	2.0000E-02	-9.4552E+00	-4.7276E-01
56	55	1.0000E+00	2.0000E-02	0.	2.0000E-02	-9.3415E+00	-4.6707E-01
66	65	1.0000E+00	2.0000E-02	0.	2.0000E-02	-9.2140E+00	-4.6070E-01
76	75	1.0000E+00	2.0000E-02	0.	2.0000E-02	-9.0722E+00	-4.5361E-01
86	85	1.0000E+00	2.0000E-02	0.	2.0000E-02	-8.9153E+00	-4.4576E-01
96	95	1.0000E+00	2.0000E-02	0.	2.0000E-02	-8.7432E+00	-4.3716E-01
106	105	1.0000E+00	2.0000E-02	0.	2.0000E-02	-8.5549E+00	-4.2775E-01
116	115	1.0000E+00	2.0000E-02	0.	2.0000E-02	-8.3517E+00	-4.1758E-01
126	125	1.0000E+00	2.0000E-02	0.	2.0000E-02	-8.1304E+00	-4.0652E-01
136	135	1.0000E+00	2.0000E-02	0.	2.0000E-02	-7.8969E+00	-3.9484E-01
146	145	1.0000E+00	2.0000E-02	0.	2.0000E-02	-7.6407E+00	-3.8203E-01
156	155	1.0000E+00	2.0000E-02	0.	2.0000E-02	-7.3824E+00	-3.6912E-01
166	165	1.0000E+00	2.0000E-02	0.	2.0000E-02	-7.0847E+00	-3.5423E-01
176	175	1.0000E+00	2.0000E-02	0.	2.0000E-02	-6.8193E+00	-3.4096E-01
186	185	1.0000E+00	2.0000E-02	0.	2.0000E-02	-6.4546E+00	-3.2273E-01
196	195	1.0000E+00	2.0000E-02	0.	2.0000E-02	-5.2362E+00	-3.1181E-01
206	205	1.0000E+00	2.0000E-02	0.	2.0000E-02	-5.7184E+00	-2.8592E-01
199	196	1.0000E+00	6.5000E-09	0.	6.5000E-09	1.8792E-07	9.3961E-09
199	206	1.0000E+00	3.5000E-09	0.	3.5000E-09	-1.8792E-07	-9.3961E-09
200	195	1.0000E+00	6.5000E-09	0.	5.5000E-09	1.2982E-07	6.4911E-09
200	205	1.0000E+00	3.5000E-09	0.	3.5000E-09	-1.2982E-07	-6.4911E-09

BOUNDARY NODE DATA

NODB	TEMPB	HEAT FLOW	AVG RATE
1001	1.0000E+02	1.1322E-06	5.6609E-08

SYSTEM TOTAL 1.1322E-06 5.6609E-08

EXTERNAL CONNECTION DATA

NODS	NODSB	AREAS	HSURE	POWER	RSURE	TRANS	HEAT FLOW	AVG RATE
2	1001	2.0000E-01	1.0000E+03	0.	0.	2.0000E+02	1.1322E-06	5.6609E-08

KCYC = 461, ITERATIONS... TOTAL = 0, AVERAGE = 0, MAXIMUM = 0. FOR = 0.570 .
ENDED PRCB 1 KCYC = 461 KWIT = 1 SUMTIM = 2.00000E+01 20.141 N04/10
KWIT...1=TIMAX, 2=IMAX, 3=TMIN, 4=S.S., 5=DATA, 6=CARD OR KEY, 7=MCYC, 8=MSEC, 9=BLK 2 OR 4,
10=CONVERGENCE FAILURE, 11=PROBLEM SIZE LIMIT, 12=TABLE LENGTH OR ORDER.

MACHINE TIME (N04/10) = 40 SECONDS. CPU,I/O,SYS PERCENTS = 48:51:0

Fig. 3. Heat regenerator. Sample problem for TRUMP report (continued).

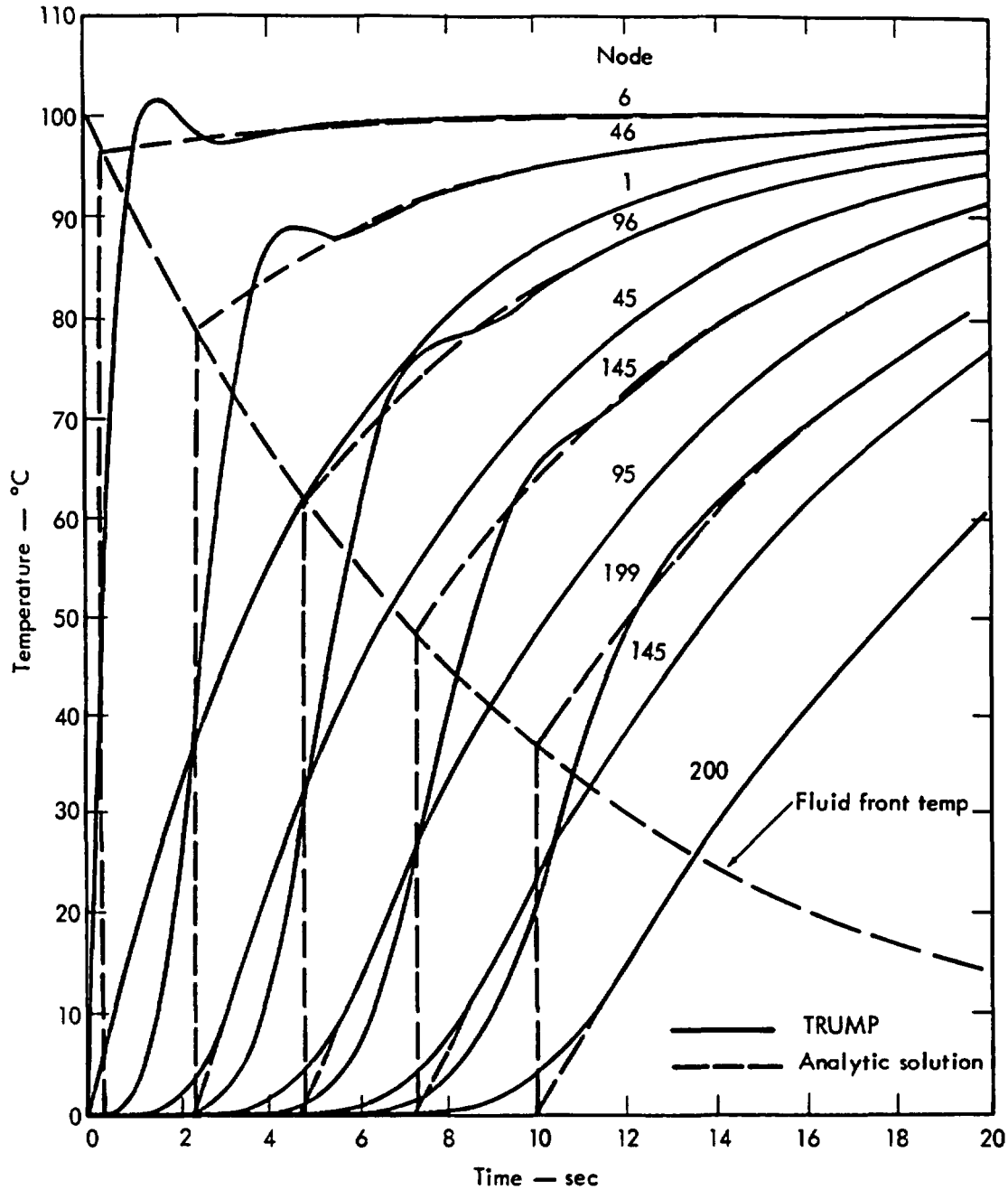

Fig. 4. Plot of temperature vs time for heat regenerator. Sample problem for TRUMP report.

2. Slab Melt

A semiinfinite slab ($-\infty < X < \infty$, $-\infty < Y < \infty$, $0 \leq Z < \infty$), has thermal conductivity 1.0, density 1.0, heat capacity 1.0, a melting point of 50.0°C, and a latent heat of melting of 25.0 cal/g. The initial temperature is 100.0°C, and, for times $t > 0$, the surface $Z = 0$ is maintained at 0°C. The temperature distribution in the slab, and the motion of the solid-liquid interface, Z_{int}, are calculated for times up to 200 sec. For the values used here, the solution to the problem[4] is as follows:

$$T = 112.4 \text{ erf } (Z/2\sqrt{t}), \quad 0 \leq T \leq 50°C \tag{II.G.2.1}$$

$$T = 9.9 + 90.1 \text{ erf } (Z/2\sqrt{t}), \quad 50°C \leq T \leq 100°C \tag{II.G.2.2}$$

$$Z_{int} = 0.835 \sqrt{t}, \quad T = 50°C. \tag{II.G.2.3}$$

The input data for this problem are shown in Fig. 5. A boundary node, No. 2001, is given a constant temperature of 0°C and is connected to a zero-volume node, No. -1, with a heat-transfer coefficient of 1.0×10^8 and unit area. The slab consists of 20 nodes 1 cm thick, Nos. 1 to 20, 10 nodes 2 cm thick, Nos. 21 to 30, 5 nodes 4 cm thick, Nos. 31 to 35, and 5 nodes 8 cm thick, Nos. 36 to 40, all with unit area perpendicular to the Z direction, for a total thickness of 100 cm in the Z direction. These thicknesses are chosen to limit time-truncation errors to about 0.5 sec near Z=0 and to 1% or less of the time at which the temperature changes most rapidly at other locations, and to include sufficient total thickness so that no appreciable temperature change will take place at the maximum value of Z, 100 cm, at a time of 200 sec. The nodal points of all nodes are at the node centers. By making TONE 100.0 in BLOCK 1, all nodes are given initial temperatures of 100.0°C except No. -1, which is given an initial temperature of 0.0°C in BLOCK 9. The latter is unnecessary but allows the calculations to begin with larger time steps, saving a small amount of machine time.

Production of output data is controlled by making IPRINT 120 and JPIC 15. Nodes 1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 14, and 16 are specified for the temperature vs time plot, on which the temperature scale is converted to the decrease below 100°C by using 100.0 for FTEMP1 and -1.0 for FTEMP2. To obtain logarithmic scales for temperature and time with 3 and 4 cycles, respectively, and an isotherm plot of node location vs time, LOGTEM is 3 and LOGTIM is 104. The node locations on the latter plot are obtained by using -0.5 for FRAD1 (FRAD2 is set to 1.0 by the program, if zero or blank), to convert node numbers to the Z coordinates of the node centers. To obtain a logarithmic scale with three cycles on the node location scale, and interpolation between points on temperature vs node location plots (not made in this problem, but obtainable by preceding JPIC with a minus sign), LOGR is 103.

For temperature accuracy, TVARY is 1.0°C. TIMAX is 200.0 sec, to end the problem at that time.

Some of the numerical output data for this problem are shown in Fig. 6 for the data input phase, the first time step, and the last time step. The calculated curves of

```
* SLAB MELT.  SAMPLE PROBLEM FOR TRUMP REPORT.   2/27/68.                    **
NOTE.    H(M) = 25 CAL/G, T(M) = 50.0, TIME 0-200 SEC.
NOTE.    THIS PROBLEM DOES NOT USE GEN, CHEM, OR FLOW SUBROUTINES.
BLOCK 1
  120
                                  1.0                  200.0
 100.0
BLOCK11  CRT PLOT CONTROLS, SCALE COEFFICIENTS, AND TIME PLOT NODES.
   15  103    3  104 -0.5                  100.0     -1.0
    1    2    3    4    5    6    7    8   10   12   14   16
BLOCK 2  MATERIAL NAMES, NUMBERS, CHEMICAL COMPONENTS, AND THERMAL PROPERTIES.
SMELT   50                        1.0       1.0       1.0       50.0      25.0

BLOCK 4   NODE NUMBERS, MATERIALS, TYPES, AND DIMENSIONS.
   -1              50
    1   19    1    50            1.0       1.0       1.0
   21    9    1    50            2.0       1.0       1.0
   31    4    1    50            4.0       1.0       1.0
   36    4    1    50            8.0       1.0       1.0

BLOCK 5  INTERNAL CONNECTION NODE NUMBERS AND DESCRIPTIONS.
   -1    1                       0.5       1.0       1.0
    1    2 18   1   1  0.5       0.5       1.0       1.0
   20   21            0.5       1.0       1.0       1.0
   21   22  8   1   1  1.0       1.0       1.0       1.0
   30   31            1.0       2.0       1.0       1.0
   31   32  3   1   1  2.0       2.0       1.0       1.0
   35   36            2.0       4.0       1.0       1.0
   36   37  3   1   1  4.0       4.0       1.0       1.0

BLOCK 6  EXTERNAL CONNECTION NODE NUMBERS AND DESCRIPTIONS.
   -1 2001                       1.0       1.0       1.0    E8

BLOCK 7  BOUNDARY NODE TEMPERATURES.
 2001                 0.0

BLOCK 9   INITIAL CONDITIONS.
   -1          1      0.0

ENDED-1  LAST CARD OF DATA DECK.  MORE DATA DECKS MAY FOLLOW.
COL 5   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80
```

Fig. 5. Input data for slab-melt problem.

```
TRUMP OUTPUT DATA              PROGRAM DATE  15.774  N04/05                                  DATA DECK    2

* SLAB MELT.  SAMPLE PROBLEM FOR TRUMP REPORT.   2/27/68.                                    20.161  N04/10
==========================================================================================================

NO DATA CARRYOVER FROM PRECEDING PROBLEM.
==========================================================================================================
                                H(M) = 25 CAL/G, T(M) = 50.0, TIME 0-200 SEC.
                                THIS PROBLEM DOES NOT USE GEN, CHEM, OR FLOW SUBROUTINES.
DATA BLOCK  1

IPRINT   NUM KDATA KSPEC  MCYC  MSEC NPUNCH NDOT IRITE ITAPE   TIMEP        SCALE
  120      0     0     0     0     0     0    0     0     0  0.           1.00000E+00

  KD   KT    DELTO        SMALL        TVARY        TAU          TIMAX        TMIN         TMAX
   1    1  1.00000E+12  1.00000E-12  1.00000E+00  0.           2.00000E+02 -1.00000E+12  1.00000E+12

  KO KSYM    GEOM         SIGMA        TBASE
   1    1  1.00000E+00  1.35500E-12  2.73150E+02

  TONE        ALONE        BONE         GONE         FONE         HONE         RONE         PONE
 1.0000E+02  0.           0.           0.           0.           0.           0.           0.
==========================================================================================================
DATA BLOCK 11                   CRT PLOT CONTROLS, SCALE COEFFICIENTS, AND TIME PLOT NODES.

 JPIC  LCGR LOGTEM LOGTIM    FRAD1        FRAD2        FTEMP1       FTEMP2       FTIME1       FTIME2
  15     3     3     4     -5.0000E-01  1.0000E+00  1.0000E+02 -1.0000E+00  0.           1.0000E+00

NODES FOR TEMPERATURE VS TIME CRT PLOTS.  MAX = 12
INDEX    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16
 NODE    1    2    3    4    5    6    7    8   10   12   14   16
```

Fig. 6. Slab melt. Sample problem for TRUMP report.

DATA BLCCK 2 MATERIAL NAMES, NUMBERS, CHEMICAL COMPONENTS, AND THERMAL PROPERTIES.

NAME	MATL	INDEX	KA	KB	LTABC	LTABK	DENSITY	CAPACITY	CONDUCTIVITY	TMELT	HMELT
SMELT	50	1	0	0	0	0	1.0000E+00	1.0000E+00	1.0000E+00	5.0000E+01	2.500[illegible]E+01

DATA BLCCK 4 NODE NUMBERS, MATERIALS, TYPES, AND DIMENSIONS.

NCDE	INDEX	MATL	NTYPE	DLONG	DWIDE	DRAD	VOLUME
-1	1	50	2	0.	0.	0.	1.00000E-24
1	2	50	0	1.00000E+00	1.00000E+00	1.00000E+00	1.00000E+00
2	3	50	0	1.00000E+00	1.00000E+00	1.00000E+00	1.00000E+00
3	4	50	0	1.00000E+00	1.00000E+00	1.00000E+00	1.00000E+00
4	5	50	0	1.00000E+00	1.00000E+00	1.00000E+00	1.00000E+00
5	6	50	0	1.00000E+00	1.00000E+00	1.00000E+00	1.00000E+00
6	7	50	0	1.00000E+00	1.00000E+00	1.00000E+00	1.00000E+00
7	8	50	0	1.00000E+00	1.00000E+00	1.00000E+00	1.00000E+00
8	9	50	0	1.00000E+00	1.00000E+00	1.00000E+00	1.00000E+00
9	10	50	0	1.00000E+00	1.00000E+00	1.00000E+00	1.00000E+00
10	11	50	0	1.00000E+00	1.00000E+00	1.00000E+00	1.00000E+00
11	12	50	0	1.00000E+00	1.00000E+00	1.00000E+00	1.00000E+00
12	13	50	0	1.00000E+00	1.00000E+00	1.00000E+00	1.00000E+00
13	14	50	0	1.00000E+00	1.00000E+00	1.00000E+00	1.00000E+00
14	15	50	0	1.00000E+00	1.00000E+00	1.00000E+00	1.00000E+00
15	16	50	0	1.00000E+00	1.00000E+00	1.00000E+00	1.00000E+00
16	17	50	0	1.00000E+00	1.00000E+00	1.00000E+00	1.00000E+00
17	18	50	0	1.00000E+00	1.00000E+00	1.00000E+00	1.00000E+00
18	19	50	0	1.00000E+00	1.00000E+00	1.00000E+00	1.00000E+00
19	20	50	0	1.00000E+00	1.00000E+00	1.00000E+00	1.00000E+00
20	21	50	0	1.00000E+00	1.00000E+00	1.00000E+00	1.00000E+00
21	22	50	0	2.00000E+00	1.00000E+00	1.00000E+00	2.00000E+00
22	23	50	0	2.00000E+00	1.00000E+00	1.00000E+00	2.00000E+00
23	24	50	0	2.00000E+00	1.00000E+00	1.00000E+00	2.00000E+00
24	25	50	0	2.00000E+00	1.00000E+00	1.00000E+00	2.00000E+00
25	26	50	0	2.00000E+00	1.00000E+00	1.00000E+00	2.00000E+00
26	27	50	0	2.00000E+00	1.00000E+00	1.00000E+00	2.00000E+00
27	28	50	0	2.00000E+00	1.00000E+00	1.00000E+00	2.00000E+00
28	29	50	0	2.00000E+00	1.00000E+00	1.00000E+00	2.00000E+00
29	30	50	0	2.00000E+00	1.00000E+00	1.00000E+00	2.00000E+00
30	31	50	0	2.00000E+00	1.00000E+00	1.00000E+00	2.00000E+00
31	32	50	0	4.00000E+00	1.00000E+00	1.00000E+00	4.00000E+00

Fig. 6. Slab melt. Sample problem for TRUMP report (continued).

32	33	50	0	4.00000E+00	1.00000E+00	1.00000E+00	4.00000E+00
33	34	50	0	4.00000E+00	1.00000E+00	1.00000E+00	4.00000E+00
34	35	50	0	4.00000E+00	1.00000E+00	1.00000E+00	4.00000E+00
35	36	50	0	4.00000E+00	1.00000E+00	1.00000E+00	4.00000E+00
36	37	50	0	8.00000E+00	1.00000E+00	1.00000E+00	8.00000E+00
37	38	50	0	8.00000E+00	1.00000E+00	1.00000E+00	8.00000E+00
38	39	50	0	8.00000E+00	1.00000E+00	1.00000E+00	8.00000E+00
39	40	50	0	8.00000E+00	1.00000E+00	1.00000E+00	8.00000E+00
40	41	50	0	8.00000E+00	1.00000E+00	1.00000E+00	8.00000E+00

DATA BLOCK 5 INTERNAL CONNECTION NODE NUMBERS AND DESCRIPTIONS.

NOD1	NOD2	INDEX	DEL1	DEL2	DLONG	DRAD	HINT	RINT	AREA
-1	1	1	0.	5.0000E-01	1.0000E+00	1.0000E+00	1.0000E+12	0.	1.0000E+00
1	2	2	5.0000E-01	5.0000E-01	1.0000E+00	1.0000E+00	1.0000E+12	0.	1.0000E+00
2	3	3	5.0000E-01	5.0000E-01	1.0000E+00	1.0000E+00	1.0000E+12	0.	1.0000E+00
3	4	4	5.0000E-01	5.0000E-01	1.0000E+00	1.0000E+00	1.0000E+12	0.	1.0000E+00
4	5	5	5.0000E-01	5.0000E-01	1.0000E+00	1.0000E+00	1.0000E+12	0.	1.0000E+00
5	6	6	5.0000E-01	5.0000E-01	1.0000E+00	1.0000E+00	1.0000E+12	0.	1.0000E+00
6	7	7	5.0000E-01	5.0000E-01	1.0000E+00	1.0000E+00	1.0000E+12	0.	1.0000E+00
7	8	8	5.0000E-01	5.0000E-01	1.0000E+00	1.0000E+00	1.0000E+12	0.	1.0000E+00
8	9	9	5.0000E-01	5.0000E-01	1.0000E+00	1.0000E+00	1.0000E+12	0.	1.0000E+00
9	10	10	5.0000E-01	5.0000E-01	1.0000E+00	1.0000E+00	1.0000E+12	0.	1.0000E+00
10	11	11	5.0000E-01	5.0000E-01	1.0000E+00	1.0000E+00	1.0000E+12	0.	1.0000E+00
11	12	12	5.0000E-01	5.0000E-01	1.0000E+00	1.0000E+00	1.0000E+12	0.	1.0000E+00
12	13	13	5.0000E-01	5.0000E-01	1.0000E+00	1.0000E+00	1.0000E+12	0.	1.0000E+00
13	14	14	5.0000E-01	5.0000E-01	1.0000E+00	1.0000E+00	1.0000E+12	0.	1.0000E+00
14	15	15	5.0000E-01	5.0000E-01	1.0000E+00	1.0000E+00	1.0000E+12	0.	1.0000E+00
15	16	16	5.0000E-01	5.0000E-01	1.0000E+00	1.0000E+00	1.0000E+12	0.	1.0000E+00
16	17	17	5.0000E-01	5.0000E-01	1.0000E+00	1.0000E+00	1.0000E+12	0.	1.0000E+00
17	18	18	5.0000E-01	5.0000E-01	1.0000E+00	1.0000E+00	1.0000E+12	0.	1.0000E+00
18	19	19	5.0000E-01	5.0000E-01	1.0000E+00	1.0000E+00	1.0000E+12	0.	1.0000E+00
19	20	20	5.0000E-01	5.0000E-01	1.0000E+00	1.0000E+00	1.0000E+12	0.	1.0000E+00
20	21	21	5.0000E-01	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+12	0.	1.0000E+00
21	22	22	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+12	0.	1.0000E+00
22	23	23	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+12	0.	1.0000E+00
23	24	24	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+12	0.	1.0000E+00
24	25	25	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+12	0.	1.0000E+00
25	26	26	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+12	0.	1.0000E+00
26	27	27	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+12	0.	1.0000E+00
27	28	28	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+12	0.	1.0000E+00

Fig. 6. Slab melt. Sample problem for TRUMP report (continued).

28	29	29	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+12	0.	1.0000E+00
29	30	30	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+12	0.	1.0000E+00
30	31	31	1.0000E+00	2.0000E+00	1.0000E+00	1.0000E+00	1.0000E+12	0.	1.0000E+00
31	32	32	2.0000E+00	2.0000E+00	1.0000E+00	1.0000E+00	1.0000E+12	0.	1.0000E+00
32	33	33	2.0000E+00	2.0000E+00	1.0000E+00	1.0000E+00	1.0000E+12	0.	1.0000E+00
33	34	34	2.0000E+00	2.0000E+00	1.0000E+00	1.0000E+00	1.0000E+12	0.	1.0000E+00
34	35	35	2.0000E+00	2.0000E+00	1.0000E+00	1.0000E+00	1.0000E+12	0.	1.0000E+00
35	36	36	2.0000E+00	4.0000E+00	1.0000E+00	1.0000E+00	1.0000E+12	0.	1.0000E+00
36	37	37	4.0000E+00	4.0000E+00	1.0000E+00	1.0000E+00	1.0000E+12	0.	1.0000E+00
37	38	38	4.0000E+00	4.0000E+00	1.0000E+00	1.0000E+00	1.0000E+12	0.	1.0000E+00
38	39	39	4.0000E+00	4.0000E+00	1.0000E+00	1.0000E+00	1.0000E+12	0.	1.0000E+00
39	40	40	4.0000E+00	4.0000E+00	1.0000E+00	1.0000E+00	1.0000E+12	0.	1.0000E+00

DATA BLOCK 6 EXTERNAL CONNECTION NODE NUMBERS AND DESCRIPTIONS.

NODS	NODSB	INDEX	LTABH	POWER	DLONG	DRAD	HSURE	RSURE	AREAS
-1	2001	1	0	0.	1.0000E+00	1.0000E+00	1.0000E+08	0.	1.0000E+00

DATA BLOCK 7 BOUNDARY NODE TEMPERATURES.

NODB	INDEX	LTABT	TEMPB	SLOPE	TIMEB
2001	1	0	0.		0.

DATA BLOCK 9 INITIAL CONDITIONS.

NOTE	INDEX	TT	AA	BB	GG
-1	1	0.	0.	0.	0.

DATA ENDED -1 LAST CARD OF DATA DECK. MORE DATA DECKS MAY FOLLOW.

MACHINE TIME (N04/10) = 2 SECONDS. CPU,I/O,SYS PERCENTS = 37:62:0

SUMMARY OF INPUT DATA

BLOCK NUMBER	2	3	4	5	6	7	8	9	10	11	12
TIMES READ IN	1	0	001	001	001	1	000	001	000	001	000

Fig. 6. Slab melt. Sample problem for TRUMP report (continued).

```
ITEM NAME            MAT   KEM  NODE  NOD1  NODS  NODB  NODG  NOTE NODF1 NODEP NODP1
MAXIMUM SIZE          40     5   480  1200   125    20   100   480    80    12   480
UNMODIFIED SIZE        0     0     0     0     0     0     0     0     0     0     0
FINAL SIZE             1     0    41    40     1     1     0     1     0    12     0

TABLES               CAPT  CONT    QT    ZT    ET HSURT TEMPB    GT FLOWT TOTAL
                        0     0     0     0     0     0     0     0     0     0

                 MAXIMUM ALLOWED TABLE LENGTH IS  12.

ARRAY STORAGE REQUIREMENTS.
     NWREQD = (8 + 7*M9)*M2 + 28*M4 + 12*M5 + 3*M9     =     31556
     NWCHEM = 4*M2 + (6 + 9*M9)*M3 + 10*M4 (M3 NOT 0)  =      5530
     NWGEN8 = 2*M4 + (3 + 3*M9)*M8          (M8 NOT 0)  =      4860
     NWFLOW = 4*M4 + (9 + 3*M9)*M10        (M10 NOT 0)  =      5520
     NWSURE = (12 + 3*M9)*M6 + (5+3*M9)*M7 (M6 NOT 0)   =      6820
     NWPLCT = 4*M4 + 3*M11 + (1 + M11)*M1 (M11 NOT 0)   =      4556
     NWEL12 = 8*M4 + 5*M12                 (M12 NOT 0)  =      6240

     TOTAL ADJUSTABLE ARRAY STORAGE                     =     65082

OTHER TOTALS     NOSPEC NOGEN NORAD NORADS NMELT NREACT
                      1     0     0     0     1     0


MATERIAL SUMMARY

  NAME   MATL NODES   DENSITY      CAPACITY     TOT VOL      TOT MASS     TOT CAP      TOT HEAT
SMELT     50    41  1.00000E+00  1.00000E+00  1.00000E+02  1.00000E+02  1.00000E+02  1.00000E+04

SYSTEM TOTAL    41                             1.00000E+02  1.00000E+02  1.00000E+02  1.00000E+04
* SLAB MELT.  SAMPLE PROBLEM FOR TRUMP REPORT.   2/27/68.                              20.161  N04/10
==========================================================================================================

PRINTOUT CYCLE  MF   MSS  KWIT   DELTMX       SMALL        TVARY        DTEMP        DTMAXS      NUTS
     1     0     0     0     0  2.22222E-01  2.22222E-03  1.00000E+00  2.00000E-10  0.             0

   TOTAL TIME     TIME STEP      HEAT FLOW    TEMP FROM FLUX   FLUX RATE      TEMP RATE
 1.00000E-12    1.00000E-12   -2.00000E-10   -2.00000E-12   -2.00000E+02   -2.00000E+00

   AVG TEMP     HEAT CAPACITY   HEAT CONTENT    GEN RATE       HEAT GEN     TEMP FROM GEN
 1.00000E+02    1.00000E+02    1.25000E+04    0.             0.             0.
```

Fig. 6. Slab melt. Sample problem for TRUMP report (continued).

NODE	TEMP	DT	DDT	GEN RATE	W	H	F
-1	2.0000E-06	2.0000E-06	0.	0.	2.0000E-30	2.0000E-30	8.2718E-25
1	1.0000E+02	-2.0000E-10	-2.0000E+02	0.	1.2500E+02	-2.0000E-10	-2.0000E-10
2	1.0000E+02	0.	0.	0.	1.2500E+02	0.	0.
3	1.0000E+02	0.	0.	0.	1.2500E+02	0.	0.
4	1.0000E+02	0.	0.	0.	1.2500E+02	0.	0.
5	1.0000E+02	0.	0.	0.	1.2500E+02	0.	0.
6	1.0000E+02	0.	0.	0.	1.2500E+02	0.	0.
7	1.0000E+02	0.	0.	0.	1.2500E+02	0.	0.
8	1.0000E+02	0.	0.	0.	1.2500E+02	0.	0.
9	1.0000E+02	0.	0.	0.	1.2500E+02	0.	0.
10	1.0000E+02	0.	0.	0.	1.2500E+02	0.	0.
11	1.0000E+02	0.	0.	0.	1.2500E+02	0.	0.
12	1.0000E+02	0.	0.	0.	1.2500E+02	0.	0.
13	1.0000E+02	0.	0.	0.	1.2500E+02	0.	0.
14	1.0000E+02	0.	0.	0.	1.2500E+02	0.	0.
15	1.0000E+02	0.	0.	0.	1.2500E+02	0.	0.
16	1.0000E+02	0.	0.	0.	1.2500E+02	0.	0.
17	1.0000E+02	0.	0.	0.	1.2500E+02	0.	0.
18	1.0000E+02	0.	0.	0.	1.2500E+02	0.	0.
19	1.0000E+02	0.	0.	0.	1.2500E+02	0.	0.
20	1.0000E+02	0.	0.	0.	1.2500E+02	0.	0.
21	1.0000E+02	0.	0.	0.	2.5000E+02	0.	0.
22	1.0000E+02	0.	0.	0.	2.5000E+02	0.	0.
23	1.0000E+02	0.	0.	0.	2.5000E+02	0.	0.
24	1.0000E+02	0.	0.	0.	2.5000E+02	0.	0.
25	1.0000E+02	0.	0.	0.	2.5000E+02	0.	0.
26	1.0000E+02	0.	0.	0.	2.5000E+02	0.	0.
27	1.0000E+02	0.	0.	0.	2.5000E+02	0.	0.
28	1.0000E+02	0.	0.	0.	2.5000E+02	0.	0.
29	1.0000E+02	0.	0.	0.	2.5000E+02	0.	0.
30	1.0000E+02	0.	0.	0.	2.5000E+02	0.	0.
31	1.0000E+02	0.	0.	0.	5.0000E+02	0.	0.
32	1.0000E+02	0.	0.	0.	5.0000E+02	0.	0.
33	1.0000E+02	0.	0.	0.	5.0000E+02	0.	0.
34	1.0000E+02	0.	0.	0.	5.0000E+02	0.	0.
35	1.0000E+02	0.	0.	0.	5.0000E+02	0.	0.
36	1.0000E+02	0.	0.	0.	1.0000E+03	0.	0.
37	1.0000E+02	0.	0.	0.	1.0000E+03	0.	0.
38	1.0000E+02	0.	0.	0.	1.0000E+03	0.	0.

Fig. 6. Slab melt. Sample problem for TRUMP report (continued).

39	1.0000E+02	0.	0.	0.	1.0000E+03	0.	0.
40	1.0000E+02	0.	0.	0.	1.0000E+03	0.	0.

NODE	TEMP	W/F A	DA	DDA	W/F B	DB	DDB
-1	2.0000E-06	1.0000E+00	0.	0.	0.	0.	0.

MATERIAL DATA

NAME	MATL	TOT CAP	TOT HEAT	AVG TEMP	TMELT	HMELT
SMELT	50	1.00000E+02	1.25000E+04	1.00000E+02	5.00000E+01	2.50000E+01

NODE DATA. CHECK TOTAL CONDUCTANCES (ZIP) AND TIME CONSTANTS (SLIM).
LARGE DIFFERENCES BETWEEN NODES MAY BE DUE TO POOR ZONING, AND MAY PRODUCE POOR RESULTS.

NODE	MATL	NTYPE	RADIUS	VOLUME	MASS	CAPACITY	CONDUCTIVITY	ZIP	SLIM
-1	50	2	0.	1.0000E-24	1.0000E-24	1.0000E-24	1.0000E+00	1.0000E+08	1.0000E-24
1	50	0	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	3.0000E+00	3.3333E-01
2	50	0	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	2.0000E+00	5.0000E-01
3	50	0	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	2.0000E+00	5.0000E-01
4	50	0	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	2.0000E+00	5.0000E-01
5	50	0	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	2.0000E+00	5.0000E-01
6	50	0	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	2.0000E+00	5.0000E-01
7	50	0	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	2.0000E+00	5.0000E-01
8	50	0	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	2.0000E+00	5.0000E-01
9	50	0	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	2.0000E+00	5.0000E-01
10	50	0	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	2.0000E+00	5.0000E-01
11	50	0	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	2.0000E+00	5.0000E-01
12	50	0	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	2.0000E+00	5.0000E-01
13	50	0	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	2.0000E+00	5.0000E-01
14	50	0	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	2.0000E+00	5.0000E-01
15	50	0	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	2.0000E+00	5.0000E-01
16	50	0	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	2.0000E+00	5.0000E-01
17	50	0	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	2.0000E+00	5.0000E-01
18	50	0	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	2.0000E+00	5.0000E-01
19	50	0	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	2.0000E+00	5.0000E-01
20	50	0	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	1.6667E+00	6.0000E-01
21	50	0	1.0000E+00	2.0000E+00	2.0000E+00	2.0000E+00	1.0000E+00	1.1667E+00	1.7143E+00
22	50	0	1.0000E+00	2.0000E+00	2.0000E+00	2.0000E+00	1.0000E+00	1.0000E+00	2.0000E+00
23	50	0	1.0000E+00	2.0000E+00	2.0000E+00	2.0000E+00	1.0000E+00	1.0000E+00	2.0000E+00
24	50	0	1.0000E+00	2.0000E+00	2.0000E+00	2.0000E+00	1.0000E+00	1.0000E+00	2.0000E+00
25	50	0	1.0000E+00	2.0000E+00	2.0000E+00	2.0000E+00	1.0000E+00	1.0000E+00	2.0000E+00

Fig. 6. Slab melt. Sample problem for TRUMP report (continued).

26	50	0	1.0000E+00	2.0000E+00	2.0000E+00	2.0000E+00	1.0000E+00	1.0000E+00	2.0000E+00
27	50	0	1.0000E+00	2.0000E+00	2.0000E+00	2.0000E+00	1.0000E+00	1.0000E+00	2.0000E+00
28	50	0	1.0000E+00	2.0000E+00	2.0000E+00	2.0000E+00	1.0000E+00	1.0000E+00	2.0000E+00
29	50	0	1.0000E+00	2.0000E+00	2.0000E+00	2.0000E+00	1.0000E+00	1.0000E+00	2.0000E+00
30	50	0	1.0000E+00	2.0000E+00	2.0000E+00	2.0000E+00	1.0000E+00	8.3333E-01	2.4000E+00
31	50	0	1.0000E+00	4.0000E+00	4.0000E+00	4.0000E+00	1.0000E+00	5.8333E-01	6.8571E+00
32	50	0	1.0000E+00	4.0000E+00	4.0000E+00	4.0000E+00	1.0000E+00	5.0000E-01	8.0000E+00
33	50	0	1.0000E+00	4.0000E+00	4.0000E+00	4.0000E+00	1.0000E+00	5.0000E-01	8.0000E+00
34	50	0	1.0000E+00	4.0000E+00	4.0000E+00	4.0000E+00	1.0000E+00	5.0000E-01	8.0000E+00
35	50	0	1.0000E+00	4.0000E+00	4.0000E+00	4.0000E+00	1.0000E+00	4.1667E-01	9.6000E+00
36	50	0	1.0000E+00	8.0000E+00	8.0000E+00	8.0000E+00	1.0000E+00	2.9167E-01	2.7429E+01
37	50	0	1.0000E+00	8.0000E+00	8.0000E+00	8.0000E+00	1.0000E+00	2.5000E-01	3.2000E+01
38	50	0	1.0000E+00	8.0000E+00	8.0000E+00	8.0000E+00	1.0000E+00	2.5000E-01	3.2000E+01
39	50	0	1.0000E+00	8.0000E+00	8.0000E+00	8.0000E+00	1.0000E+00	2.5000E-01	3.2000E+01
40	50	0	1.0000E+00	8.0000E+00	8.0000E+00	8.0000E+00	1.0000E+00	1.2500E-01	6.4000E+01

INTERNAL CONNECTION DATA. CHECK CONDUCTANCES (TRAN).
LARGE DIFFERENCES BETWEEN CONNECTIONS MAY BE DUE TO POOR ZONING, AND MAY PRODUCE POOR RESULTS.

NOD1	NOD2	AREA	HINT	RINT	TRAN	HEAT FLOW	AVG RATE
-1	1	1.0000E+00	1.0000E+12	0.	2.0000E+00	2.0000E-10	2.0000E+02
1	2	1.0000E+00	1.0000E+12	0.	1.0000E+00	0.	0.
2	3	1.0000E+00	1.0000E+12	0.	1.0000E+00	0.	0.
3	4	1.0000E+00	1.0000E+12	0.	1.0000E+00	0.	0.
4	5	1.0000E+00	1.0000E+12	0.	1.0000E+00	0.	0.
5	6	1.0000E+00	1.0000E+12	0.	1.0000E+00	0.	0.
6	7	1.0000E+00	1.0000E+12	0.	1.0000E+00	0.	0.
7	8	1.0000E+00	1.0000E+12	0.	1.0000E+00	0.	0.
8	9	1.0000E+00	1.0000E+12	0.	1.0000E+00	0.	0.
9	10	1.0000E+00	1.0000E+12	0.	1.0000E+00	0.	0.
10	11	1.0000E+00	1.0000E+12	0.	1.0000E+00	0.	0.
11	12	1.0000E+00	1.0000E+12	0.	1.0000E+00	0.	0.
12	13	1.0000E+00	1.0000E+12	0.	1.0000E+00	0.	0.
13	14	1.0000E+00	1.0000E+12	0.	1.0000E+00	0.	0.
14	15	1.0000E+00	1.0000E+12	0.	1.0000E+00	0.	0.
15	16	1.0000E+00	1.0000E+12	0.	1.0000E+00	0.	0.
16	17	1.0000E+00	1.0000E+12	0.	1.0000E+00	0.	0.
17	18	1.0000E+00	1.0000E+12	0.	1.0000E+00	0.	0.
18	19	1.0000E+00	1.0000E+12	0.	1.0000E+00	0.	0.
19	20	1.0000E+00	1.0000E+12	0.	1.0000E+00	0.	0.
20	21	1.0000E+00	1.0000E+12	0.	6.6667E-01	0.	0.
21	22	1.0000E+00	1.0000E+12	0.	5.0000E-01	0.	0.
22	23	1.0000E+00	1.0000E+12	0.	5.0000E-01	0.	0.

Fig. 6. Slab melt. Sample problem for TRUMP report (continued).

23	24	1.0000E+00	1.0000E+12	0.	5.0000E-01	0.	0.
24	25	1.0000E+00	1.0000E+12	0.	5.0000E-01	0.	0.
25	26	1.0000E+00	1.0000E+12	0.	5.0000E-01	0.	0.
26	27	1.0000E+00	1.0000E+12	0.	5.0000E-01	0.	0.
27	28	1.0000E+00	1.0000E+12	0.	5.0000E-01	0.	0.
28	29	1.0000E+00	1.0000E+12	0.	5.0000E-01	0.	0.
29	30	1.0000E+00	1.0000E+12	0.	5.0000E-01	0.	0.
30	31	1.0000E+00	1.0000E+12	0.	3.3333E-01	0.	0.
31	32	1.0000E+00	1.0000E+12	0.	2.5000E-01	0.	0.
32	33	1.0000E+00	1.0000E+12	0.	2.5000E-01	0.	0.
33	34	1.0000E+00	1.0000E+12	0.	2.5000E-01	0.	0.
34	35	1.0000E+00	1.0000E+12	0.	2.5000E-01	0.	0.
35	36	1.0000E+00	1.0000E+12	0.	1.6667E-01	0.	0.
36	37	1.0000E+00	1.0000E+12	0.	1.2500E-01	0.	0.
37	38	1.0000E+00	1.0000E+12	0.	1.2500E-01	0.	0.
38	39	1.0000E+00	1.0000E+12	0.	1.2500E-01	0.	0.
39	40	1.0000E+00	1.0000E+12	0.	1.2500E-01	0.	0.

BOUNDARY NODE DATA

NODB	TEMPB	HEAT FLOW	AVG RATE
2001	0.	-2.0000E-10	-2.0000E+02
SYSTEM TOTAL		-2.0000E-10	-2.0000E+02

EXTERNAL CONNECTION DATA

NODS	NODSB	AREAS	HSURE	POWER	RSURE	TRANS	HEAT FLOW	AVG RATE
-1	2001	1.0000E+00	1.0000E+08	0.	0.	1.0000E+08	-2.0000E-10	-2.0000E+02

KCYC = 0, ITERATIONS... TOTAL = 0, AVERAGE = 0, MAXIMUM = 0. FOR = 0.570 .

* SLAB MELT. SAMPLE PROBLEM FOR TRUMP REPORT. 2/27/68. 20.161 N04/10

PRINTOUT	CYCLE	MF	MSS	KWIT	DELTMX	SMALL	IVARY	DTEMP	DTMAXS	NUIS
7	491	0	4	1	4.57143E+00	1.6000E-02	1.00000E+00	4.69325E-02	6.84504E-01	2

TOTAL TIME	TIME STEP	HEAT FLOW	TEMP FROM FLUX	FLUX RATE	TEMP RATE
2.00000E+02	4.36641E-01	-1.78814E+03	-1.78814E+01	-8.94071E+00	-8.94071E-02

Fig. 6. Slab melt. Sample problem for TRUMP report (continued).

AVG TEMP	HEAT CAPACITY	HEAT CONTENT	GEN RATE	HEAT GEN	TEMP FROM GEN
8.50820E+01	1.00000E+02	1.07121E+04	0.	0.	0.

NODE	TEMP	DT	DDT	GEN RATE	W	H	F
-1	4.4552E-08	-3.7296E-11	-8.5209E-11	0.	4.4552E-32	4.4552E-32	1.0090E-33
1	2.2276E+00	-1.8627E-03	-4.2556E-03	0.	2.2276E+00	-1.2277E+02	-1.2277E+02
2	6.6786E+00	-5.4299E-03	-1.2405E-02	0.	6.6786E+00	-1.1832E+02	-1.1832E+02
3	1.1117E+01	-8.5372E-03	-1.9505E-02	0.	1.1117E+01	-1.1388E+02	-1.1388E+02
4	1.5537E+01	-1.0931E-02	-2.4974E-02	0.	1.5537E+01	-1.0946E+02	-1.0947E+02
5	1.9932E+01	-1.2430E-02	-2.8399E-02	0.	1.9932E+01	-1.0507E+02	-1.0507E+02
6	2.4299E+01	-1.2941E-02	-2.9566E-02	0.	2.4299E+01	-1.0070E+02	-1.0071E+02
7	2.8637E+01	-1.2458E-02	-2.8462E-02	0.	2.8637E+01	-9.6363E+01	-9.6377E+01
8	3.2946E+01	-1.1072E-02	-2.5296E-02	0.	3.2946E+01	-9.2054E+01	-9.2071E+01
9	3.7231E+01	-8.8230E-03	-2.0158E-02	0.	3.7231E+01	-8.7769E+01	-8.7790E+01
10	4.1496E+01	-5.9714E-03	-1.3643E-02	0.	4.1496E+01	-8.3504E+01	-8.3527E+01
11	4.5748E+01	-2.0205E-03	-4.6161E-03	0.	4.5748E+01	-7.9252E+01	-7.9276E+01
12	5.0000E+01	6.7502E-14	1.5422E-13	0.	5.3896E+01	-7.1104E+01	-7.1125E+01
13	5.2668E+01	-5.6340E-03	-1.2872E-02	0.	7.7668E+01	-4.7332E+01	-4.7351E+01
14	5.5329E+01	-1.3250E-02	-3.0273E-02	0.	8.0329E+01	-4.4671E+01	-4.4687E+01
15	5.7959E+01	-1.9866E-02	-4.5388E-02	0.	8.2959E+01	-4.2041E+01	-4.2054E+01
16	6.0544E+01	-2.6015E-02	-5.9435E-02	0.	8.5544E+01	-3.9456E+01	-3.9468E+01
17	6.3070E+01	-3.1456E-02	-7.1867E-02	0.	8.8070E+01	-3.6930E+01	-3.6941E+01
18	6.5524E+01	-3.6155E-02	-8.2601E-02	0.	9.0524E+01	-3.4476E+01	-3.4485E+01
19	6.7896E+01	-3.9995E-02	-9.1377E-02	0.	9.2896E+01	-3.2104E+01	-3.2112E+01
20	7.0177E+01	-4.2935E-02	-9.8093E-02	0.	9.5177E+01	-2.9823E+01	-2.9828E+01
21	7.3451E+01	-4.6009E-02	-1.0511E-01	0.	1.9690E+02	-5.3099E+01	-5.3100E+01
22	7.7396E+01	-4.6933E-02	-1.0723E-01	0.	2.0479E+02	-4.5208E+01	-4.5209E+01
23	8.0912E+01	-4.5702E-02	-1.0442E-01	0.	2.1182E+02	-3.8176E+01	-3.8176E+01
24	8.4010E+01	-4.3254E-02	-9.8822E-02	0.	2.1802E+02	-3.1979E+01	-3.1979E+01
25	8.6712E+01	-4.0193E-02	-9.1828E-02	0.	2.2342E+02	-2.6575E+01	-2.6575E+01
26	8.9046E+01	-3.6836E-02	-8.4157E-02	0.	2.2809E+02	-2.1907E+01	-2.1907E+01
27	9.1043E+01	-3.3337E-02	-7.6165E-02	0.	2.3209E+02	-1.7914E+01	-1.7914E+01
28	9.2734E+01	-2.9791E-02	-6.8063E-02	0.	2.3547E+02	-1.4531E+01	-1.4531E+01
29	9.4153E+01	-2.6274E-02	-6.0029E-02	0.	2.3831E+02	-1.1695E+01	-1.1695E+01
30	9.5330E+01	-2.2867E-02	-5.2244E-02	0.	2.4066E+02	-9.3394E+00	-9.3394E+00
31	9.6782E+01	-1.7985E-02	-4.1090E-02	0.	4.8713E+02	-1.2871E+01	-1.2871E+01
32	9.8062E+01	-1.2781E-02	-2.9199E-02	0.	4.9225E+02	-7.7531E+00	-7.7531E+00
33	9.8872E+01	-8.6738E-03	-1.9817E-02	0.	4.9549E+02	-4.5130E+00	-4.5130E+00
34	9.9363E+01	-5.6294E-03	-1.2861E-02	0.	4.9745E+02	-2.5485E+00	-2.5485E+00
35	9.9647E+01	-3.5209E-03	-8.0440E-03	0.	4.9859E+02	-1.4128E+00	-1.4128E+00
36	9.9878E+01	-1.4722E-03	-3.3635E-03	0.	9.9902E+02	-9.7583E-01	-9.7583E-01

Fig. 6. Slab melt. Sample problem for TRUMP report (continued).

37	9.9969E+01	-4.5152E-04	-1.0316E-03	0.	9.9975E+02	-2.4935E-01	-2.4935E-01
38	9.9993E+01	-1.2289E-04	-2.8076E-04	0.	9.9994E+02	-5.7842E-02	-5.7842E-02
39	9.9998E+01	-3.0227E-05	-6.9059E-05	0.	9.9999E+02	-1.2325E-02	-1.2325E-02
40	1.0000E+02	-7.9588E-06	-1.8183E-05	0.	1.0000E+03	-2.8143E-03	-2.8143E-03

NODE	TEMP	W/F A	DA	DDA	W/F B	DB	DDB
-1	4.4552E-08	1.0000E+00	0.	0.	0.	0.	0.
1	2.2276E+00	1.0000E+00	0.	0.	0.	0.	0.
2	6.6786E+00	1.0000E+00	0.	0.	0.	0.	0.
3	1.1117E+01	1.0000E+00	0.	0.	0.	0.	0.
4	1.5537E+01	1.0000E+00	0.	0.	0.	0.	0.
5	1.9932E+01	1.0000E+00	0.	0.	0.	0.	0.
6	2.4299E+01	1.0000E+00	0.	0.	0.	0.	0.
7	2.8637E+01	1.0000E+00	0.	0.	0.	0.	0.
8	3.2946E+01	1.0000E+00	0.	0.	0.	0.	0.
9	3.7231E+01	1.0000E+00	0.	0.	0.	0.	0.
10	4.1496E+01	1.0000E+00	0.	0.	0.	0.	0.
11	4.5748E+01	1.0000E+00	0.	0.	0.	0.	0.
12	5.0000E+01	8.4416E-01	2.7380E-02	6.2555E-02	0.	0.	0.

MATERIAL DATA

NAME	MATL	TOT CAP	TOT HEAT	AVG TEMP	TMELT	HMELT
SMELT	50	1.00000E+02	1.07121E+04	8.50820E+01	5.00000E+01	2.50000E+01

INTERNAL CONNECTION DATA. CHECK CONDUCTANCES (TRAN).
LARGE DIFFERENCES BETWEEN CONNECTIONS MAY BE DUE TO POOR ZONING, AND MAY PRODUCE POOR RESULTS.

NOD1	NOD2	AREA	HINT	RINT	TRAN	HEAT FLOW	AVG RATE
-1	1	1.0000E+00	1.0000E+12	0.	2.0000E+00	1.7881E+03	8.9407E+00
1	2	1.0000E+00	1.0000E+12	0.	1.0000E+00	1.6654E+03	8.3268E+00
2	3	1.0000E+00	1.0000E+12	0.	1.0000E+00	1.5470E+03	7.7352E+00
3	4	1.0000E+00	1.0000E+12	0.	1.0000E+00	1.4332E+03	7.1658E+00
4	5	1.0000E+00	1.0000E+12	0.	1.0000E+00	1.3237E+03	6.6185E+00
5	6	1.0000E+00	1.0000E+12	0.	1.0000E+00	1.2186E+03	6.0931E+00
6	7	1.0000E+00	1.0000E+12	0.	1.0000E+00	1.1179E+03	5.5896E+00
7	8	1.0000E+00	1.0000E+12	0.	1.0000E+00	1.0215E+03	5.1077E+00
8	9	1.0000E+00	1.0000E+12	0.	1.0000E+00	9.2947E+02	4.6473E+00
9	10	1.0000E+00	1.0000E+12	0.	1.0000E+00	8.4168E+02	4.2084E+00
10	11	1.0000E+00	1.0000E+12	0.	1.0000E+00	7.5815E+02	3.7907E+00
11	12	1.0000E+00	1.0000E+12	0.	1.0000E+00	6.7887E+02	3.3944E+00

Fig. 6. Slab melt. Sample problem for TRUMP report (continued).

12	13	1.0000E+00	1.0000E+12	0.	1.0000E+00	6.0775E+02	3.0387E+00
13	14	1.0000E+00	1.0000E+12	0.	1.0000E+00	5.6040E+02	2.8020E+00
14	15	1.0000E+00	1.0000E+12	0.	1.0000E+00	5.1571E+02	2.5786E+00
15	16	1.0000E+00	1.0000E+12	0.	1.0000E+00	4.7366E+02	2.3683E+00
16	17	1.0000E+00	1.0000E+12	0.	1.0000E+00	4.3419E+02	2.1709E+00
17	18	1.0000E+00	1.0000E+12	0.	1.0000E+00	3.9725E+02	1.9862E+00
18	19	1.0000E+00	1.0000E+12	0.	1.0000E+00	3.6276E+02	1.8138E+00
19	20	1.0000E+00	1.0000E+12	0.	1.0000E+00	3.3065E+02	1.6533E+00
20	21	1.0000E+00	1.0000E+12	0.	6.6667E-01	3.0082E+02	1.5041E+00
21	22	1.0000E+00	1.0000E+12	0.	5.0000E-01	2.4772E+02	1.2386E+00
22	23	1.0000E+00	1.0000E+12	0.	5.0000E-01	2.0251E+02	1.0126E+00
23	24	1.0000E+00	1.0000E+12	0.	5.0000E-01	1.6434E+02	8.2169E-01
24	25	1.0000E+00	1.0000E+12	0.	5.0000E-01	1.3236E+02	6.6179E-01
25	26	1.0000E+00	1.0000E+12	0.	5.0000E-01	1.0578E+02	5.2891E-01
26	27	1.0000E+00	1.0000E+12	0.	5.0000E-01	8.3876E+01	4.1938E-01
27	28	1.0000E+00	1.0000E+12	0.	5.0000E-01	5.5962E+01	3.2981E-01
28	29	1.0000E+00	1.0000E+12	0.	5.0000E-01	5.1430E+01	2.5715E-01
29	30	1.0000E+00	1.0000E+12	0.	5.0000E-01	3.9736E+01	1.9868E-01
30	31	1.0000E+00	1.0000E+12	0.	3.3333E-01	3.0396E+01	1.5198E-01
31	32	1.0000E+00	1.0000E+12	0.	2.5000E-01	1.7526E+01	8.7628E-02
32	33	1.0000E+00	1.0000E+12	0.	2.5000E-01	9.7725E+00	4.8862E-02
33	34	1.0000E+00	1.0000E+12	0.	2.5000E-01	5.2595E+00	2.6297E-02
34	35	1.0000E+00	1.0000E+12	0.	2.5000E-01	2.7110E+00	1.3555E-02
35	36	1.0000E+00	1.0000E+12	0.	1.6667E-01	1.2982E+00	6.4908E-03
36	37	1.0000E+00	1.0000E+12	0.	1.2500E-01	3.2233E-01	1.6116E-03
37	38	1.0000E+00	1.0000E+12	0.	1.2500E-01	7.2981E-02	3.6491E-04
38	39	1.0000E+00	1.0000E+12	0.	1.2500E-01	1.5139E-02	7.5695E-05
39	40	1.0000E+00	1.0000E+12	0.	1.2500E-01	2.8143E-03	1.4071E-05

BOUNDARY NODE DATA

NODB	TEMPB	HEAT FLOW	AVG RATE
2001	0.	-1.7881E+03	-8.9407E+00
SYSTEM TOTAL		-1.7881E+03	-8.9407E+00

EXTERNAL CONNECTION DATA

NODS	NODSB	AREAS	HSURE	POWER	RSURE	TRANS	HEAT FLOW	AVG RATE
-1	2001	1.0000E+00	1.0000E+08	0.	0.	1.0000E+08	-1.7881E+03	-8.9407E+00

Fig. 6. Slab melt. Sample problem for TRUMP report (continued).

```
KCYC =    491, ITERATIONS...  TOTAL =  1769, AVERAGE =      3, MAXIMUM =     26.  FOR = 0.570 .
ENDED PROB    2     KCYC =   491     KWIT =  1     SUMTIM =  2.00000E+02                20.161  N04/10
KWIT...1=TIMAX, 2=TMAX, 3=TMIN, 4=S.S., 5=DATA, 6=CARD OR KEY, 7=MCYC, 8=MSEC, 9=BLK 2 OR 4,
     10=CONVERGENCE FAILURE, 11=PROBLEM SIZE LIMIT, 12=TABLE LENGTH OR ORDER.
```

```
MACHINE TIME (N04/10) =     28  SECONDS.  CPU,I/O,SYS PERCENTS =   64:35:0
```

Fig. 6. Slab melt. Sample problem for TRUMP report (continued).

position vs time for temperature-drop isotherms spaced 5°C apart from 5°C to 95°C, and the analytic solutions for the phase interface temperature (50°C) and temperature drops of 5°C and 95°C, are shown in Fig. 7. Agreement of isotherm positions is within 1/4 of a node thickness for the entire range shown. The discontinuities in slope are caused by the use of nodes of finite thickness. The phase interface must actually move discontinuously from one node to the next.

A total of 493 time steps is required to reach a problem time of 200.0 sec for an average time step of 0.41 sec. The required machine time is 61 sec, or 1.0 msec per time step per calculational item (see Section IV.F). A total of 1621 iterations is necessary, or an average of 3.3 per time step. The most used in a single time step are 25. Nodes 1 through 20 are reclassified as special nodes after about 300 time steps. Thereafter, the initial estimates of temperature changes are inaccurate because of the discontinuous rates of change, so that several iterations are required for convergence.

3. Nitrocellulose Cylinder

An experimental apparatus consists of a long hollow cylinder of nitrocellulose with inner and outer radii 2.0 and 3.0 cm, containing a concentric solid cylindrical heater core. The space between them contains air, and the outer radius of the nitrocellulose is exposed to air. For a constant heat-generation rate of 0.4 cal/sec-cm^3 in the heater core, two cases are calculated: (1) the steady-state radial temperature distribution, assuming no chemical reaction takes place in the nitrocellulose, and (2) the time of ignition, assuming the nitrocellulose undergoes a first-order decomposition reaction and the rate constant has an Arrhenius temperature dependence:

$$K = \exp\left(48.3 - 50000/1.987\ T\right) \sec^{-1}, \qquad \text{(II.G.3.1)}$$

where T is in °K. The heat of reaction is 800 cal/g.

The system is initially at 25°C, and the air outside the nitrocellulose is maintained at 25°C. It is assumed that the system is long enough in the axial direction that axial heat flow is negligible. Heat may flow across the air gap by convection and radiation. The surface emissivities for the heater and nitrocellulose are 0.8 and 0.9, so that the radiation new factor for this geometry is given by

$$\mathcal{F} = \frac{1}{\frac{1}{\epsilon_1} + \frac{r_1}{r_2}\left(\frac{1}{\epsilon_2} - 1\right)} = 0.809. \qquad \text{(II.G.3.2)}$$

To model convective transport across the air gap, the thermal conductivity specified for air is 0.0005 cal/sec-cm-°C, or about 6 times its normal value for the expected temperature range, corresponding to film thicknesses on each surface of about 0.08 cm. Inaccuracy in this estimate primarily affects the calculated heater-core temperature, and not the nitrocellulose inner-surface temperature, except indirectly by changing the effective heat-transfer coefficient for radiation, which accounts for about half the heat

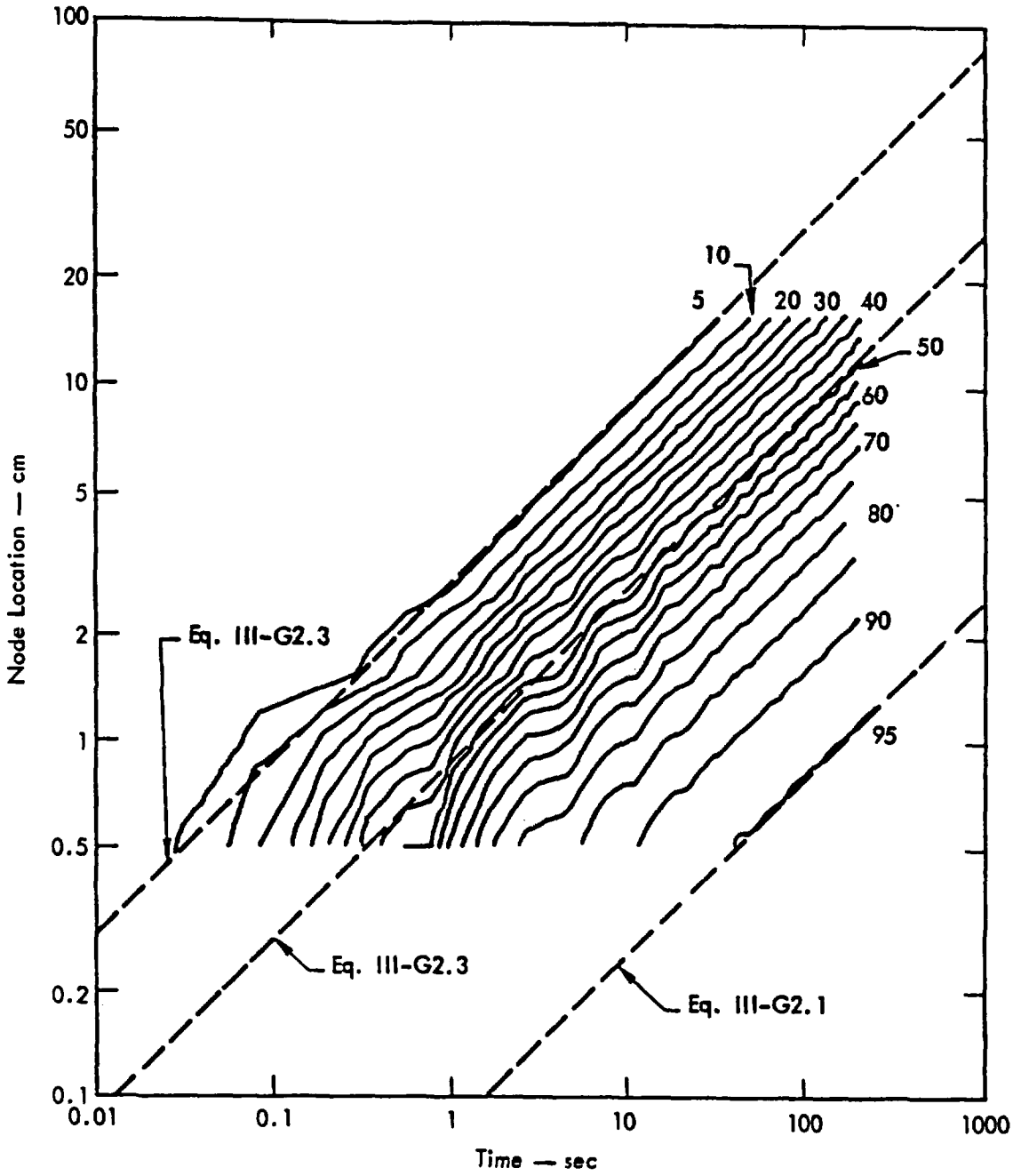

Fig. 7. Plot of node location vs time for temperature drops. Slab-melt TRUMP calculations and analytic solutions.

flow across the air gap. However, the change is in the direction which compensates for any error in the convective-transport coefficient. In a real problem, a range of values could be used to determine the effects on the solution.

At the outer surface of the nitrocellulose, it is assumed that heat transport takes place by free convection, with a convection coefficient given by

$$h = 0.0002\,(T_s - T_{air})^{0.333} \qquad \text{(II.G.3.3)}$$

where T_s is the surface temperature of the nitrocellulose. The same type of relationship could have been used for the inner air gap, but different methods were used for purposes of illustration.

The heater core has a density of 2.5 g/cm^3, a specific heat of 0.6 cal/g-°C, and a thermal conductivity of 0.01 cal/sec-cm-°C. Only the thermal conductivity affects the steady-state solution, and that effect is small, since most of the thermal resistance, and therefore the largest temperature drops, are across the air gap and in the nitrocellulose. In the transient calculation, the heat capacity is important only at short times, when the rate of heat flow across the air gap is small compared with the rate of heat production in the core.

The air has a density of 0.001 g/cm^3 and a heat capacity of 0.3 cal/g-°C, neither of which has any significant effect on the calculations. The nitrocellulose has a density of 2.0 g/cm^3, a heat capacity of 0.25 cal/g-°C, and a thermal conductivity of 0.0005 cal/sec-cm-°C. Only the conductivity is important in the steady-state calculation, but all three values are important in the transient solution involving chemical reaction.

The input data for the steady-state problem are shown in Fig. 8. For simplicity, the zoning is somewhat coarse for this type of problem. Finer zoning would be used in an actual problem, perhaps by a factor of 3 to 5. Zero-volume nodes 2, 4, and 10 are used to model the surface temperatures at the outer radius of the heater core, and the inner and outer radii of the nitrocellulose cylinder, respectively. The heater core is modeled by node 1, the air gap by nodes 31, 32, and 33, bounded by the radii 1.0, 1.8, 1.9, and 2.0, and the nitrocellulose by nodes 5, 6, 7, 8, and 9, bounded by the radii 2.0, 2.1, 2.2, 2.4, 2.7, and 3.0. The nodes are described in BLOCK 4 of the input data, each with an axial length of 1.0 for DLONG, the radial thickness for DWIDE, and the arithmetic mean radius for DRAD, which results in the correct volume when the symmetry factor KD is 2. The thermal connections are described in BLOCK 5 of the input data, each with the heat flow paths DEL1 and DEL2 measured from the arithmetic mean radius of each node to the node interfaces. More accuracy would be obtained if the node interfaces were located at log mean radii (geometric mean in a spherical system) between the nodal points, but the improvement is not significant unless the radial thickness of a node exceeds 20% of its average radius. Finer zoning usually makes this correction unnecessary—except possibly in the first two or three zones at the center of a solid cylinder (or sphere), if the thermal conductance there significantly affects the calculation. Radiation transport is specified between nodes 2 and 4 with a thermal

```
 NITROCELLULOSE CYLINDER.  SAMPLE PROBLEM FOR TRUMP REPORT.  2/27/68.          **
NOTE.    NO REACTION.  FIND STEADY STATE TEMPERATURE DISTRIBUTION.
NOTE.    THIS PROBLEM DOES NOT USE CHEM OR FLOW SUBROUTINES.
BLOCK 1  TVARY 20.0 FOR FAST S.S. CALC.  DELTO 1000.0 FOR CONVERGENCE.
 9999    4    1
    2      1000.0                20.0
 25.0
BLOCK11  GET FINAL TEMP VS RADIUS PLOT. NO TEMP VS TIME PLOT.
-9999  100

BLOCK 2 MATERIALS ARE HEATER CORE, AIR, AND NON-REACTIVE NITROCELLULOSE.
CORE     1                      2.5          0.6       0.01
AIR      2                      1.0    E-3 0.3         5.0    E-4
NC-NR    3                      2.0         0.25       5.0    E-4

BLOCK 4   USE ARITHMETIC MEAN AVERAGE RADII TO GET CORRECT VOLUME.
    1              1             1.0        1.0        0.5
    2              1                                    1.0
   31              2             1.0        0.8        1.40
   32              2     1       1.0        0.1        1.85
   33              2     1       1.0        0.1        1.95
    4              3                                    2.0
    5              3             1.0        0.1        2.05
    6              3             1.0        0.1        2.15
    7              3             1.0        0.2        2.30
    8              3             1.0        0.3        2.55
    9              3             1.0        0.3        2.85
   10              3                                    3.0

BLOCK 5  USE ACTUAL NODE INTERFACES FOR INTERFACE RADII.
    1    2           0.5        0.0        1.0         1.0
    2   31           0.0        0.4        1.0         1.0
   31   32           0.4        0.05       1.0         1.8
   32   33           0.05       0.05       1.0         1.90
   33    4           0.05       0.0        1.0         2.0
    2    4          0.0        0.0        1.0         1.0                    0.809
    4    5           0.0        0.05        1.0        2.0
    5    6           0.05       0.05        1.0        2.10
    6    7           0.05       0.1        1.0         2.2
    7    8           0.1        0.15       1.0         2.4
    8    9           0.15       0.15       1.0         2.7
    9   10           0.15       0.0        1.0         3.0

BLOCK 6 OUTER SURFACE CONNECTED TO 25-DEG EXTERNAL TEMP BY H = 0.0002
   10  101                       1.0        3.0        2.0    E-4          0.333

BLOCK 7  EXTERNAL TEMP IS 25.0 DEGREES C.  BOUNDARY NODE 101.
  101                25.0

BLOCK 9 NODE 1 HAS HEAT GENERATION RATE 0.4 CAL/SEC-CC.
    1                                                     0.4

ENDED-1 END OF TRUMP NITROCELLULOSE STEADY STATE PROBLEM.
COL 5   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80
```

Fig. 8. Input data for nitrocellulose cylinder problem. (Steady-state calculation.)

connection for which DEL1 and DEL2 are zero, the area specified is at the inner radius, and RINT is 0.809, calculated by Eq. (II.G.3.2).

The convective heat-transport boundary condition, Eq. (II.G.3.3), is obtained by connecting node 10 to boundary node 101 in BLOCK 6, specifying 0.0002 for HSURE and 0.333 for POWER. Boundary node 101 is assigned a constant temperature of 25°C in BLOCK 7. The initial conditions are obtained by assigning TONE in BLOCK 1 a value of 25°C, which will be assigned by the program to all the nodes. A heat-generation rate of 0.4 cal/sec-cm^3 is assigned to node 1 in BLOCK 9. The material properties are assigned in BLOCK 2.

For the steady-state calculation to obtain a fast solution, TVARY is 20.0. DELTO is 1000.0, to prevent oscillation caused by the radiation transport connection when approaching steady state. The same result could be obtained by using a smaller value of TVARY, such as 5.0, which is more certain. A proper value of DELTO can only be chosen after the calculation has been attempted, and the size of time steps for which the iteration procedure failed to converge noted. To limit numerical output, IPRINT is 9999, so that printouts are obtained only for the first, second, and last time steps. KDATA is 1 to obtain all optional summary output data on these time steps. NUM is 4, so that a printout of the temperature of node 4 will be obtained at the end every time step, for later comparison with the reaction calculation.

A final plot of temperature vs radius is obtained by assigning a value of -9999 to JPIC. LOGR is 100, to obtain interpolation between points on the plot.

Numerical output data for the steady-state calculation are shown in Fig. 9. Because of the large value used for TVARY, significant errors (up to 10%) occur in the heat balance, particularly in node 1; but these are transient errors, having no effect on the final steady-state temperature distribution or the final rates of heat flow. The steady-state temperature distribution is shown in Fig. 10.

The input data for the reaction and ignition calculation is shown in Fig. 11. Maximum use is made of input data already read in from the preceding steady-state problem, by assigning a value of 2 to the data carryover control parameter in column 72 of the Problem Name card. A new BLOCK 1 is included to make SMALL 0.002 sec, TVARY 2.0°C, and TMAX 400.0°C, with other input values the same as before. SMALL is specified to limit the number of time steps used to reach the arbitrary ignition temperature of 400°C, rather than letting TVARY continue to control the size of the time steps after reaction begins. A new BLOCK 11 is included, in which JPIC is -5, LOGTIM is -4, and all the nodes in the system are specified for the temperature vs time plots. Temperature vs radius plots are obtained every 5 time steps, and both linear and 4-cycle log scales are obtained for the time scales on the temperature vs time plots. LOGR is 100, as before, to obtain interpolation on the temperature vs radius plots. A BLOCK 2, type B, is included to change the description of nitrocellulose to specify that it contains reactant 1, which is described in BLOCK 3. A BLOCK 9, type B, is included to assign an initial reactant concentration of 1.0 to nodes 4 through 10.

```
TRUMP OUTPUT DATA              PROGRAM DATE  15.774  N04/05                                  DATA DECK    3

* NITROCELLULOSE CYLINDER.  SAMPLE PROBLEM FOR TRUMP REPORT.  2/27/68.          **        20.172  N04/10
==========================================================================================================

NO DATA CARRYOVER FROM PRECEDING PROBLEM.
==========================================================================================================
                                NO REACTION.  FIND STEADY STATE TEMPERATURE DISTRIBUTION.
                                THIS PROBLEM DOES NOT USE CHEM OR FLOW SUBROUTINES.
DATA BLOCK  1                   TVARY 20.0 FOR FAST S.S. CALC.  DELTC 1000.0 FOR CONVERGENCE.

IPRINT    NUM KDATA KSPEC  MCYC  MSEC NPUNCH NDOT IRITE ITAPE   TIMEP        SCALE
 9999      4     1     0     0     0     0    0     0     0  0.           1.00000E+00

  KC   KT    DELTO        SMALL        TVARY        TAU          TIMAX        TMIN         TMAX
   2    1  1.00000E+03  1.00000E-12  2.00000E+01  0.           0.           -1.00000E+12  1.00000E+12

  KC KSYM    GEOM         SIGMA        TBASE
   2    1  6.28319E+00  1.35500E-12  2.73150E+02

  TONE        ALONE       BONE        GONE        FONE        HONE        RONE        PONE
 2.5000E+01  0.          0.          0.          0.          0.          0.          0.
==========================================================================================================
DATA BLOCK 11                   GET FINAL TEMP VS RADIUS PLOT. NO TEMP VS TIME PLOT.

WILL MAKE TEMP VS RADIUS PLOTS

  JPIC  LOGR LOGTEM LOGTIM    FRAD1       FRAD2       FTEMP1      FTEMP2      FTIME1      FTIME2
  9999    0     0     0      0.          1.0000E+00  0.          1.0000E+00  0.          1.0000E+00
==========================================================================================================
DATA BLOCK  2                   MATERIALS ARE HEATER CORE, AIR, AND NON-REACTIVE NITROCELLULOSE.

NAME MATL INDEX  KA  KB LTABC LTABK   DENSITY      CAPACITY    CONDUCTIVITY   TMELT        HMELT
CORE    1    1   0   0    0     0    2.5000E+00   6.0000E-01  1.0000E-02     0.           0.
AIR     2    2   0   0    0     0    1.0000E-03   3.0000E-01  5.0000E-04     0.           0.
NC-NR   3    3   0   0    0     0    2.0000E+00   2.5000E-01  5.0000E-04     0.           0.
==========================================================================================================
```

Fig. 9. Nitrocellulose cylinder. Sample problem for TRUMP report. (Steady-state calculation.)

DATA BLOCK 4 USE ARITHMETIC MEAN AVERAGE RADII TO GET CORRECT VOLUME.

NODE	INDEX	MATL	NTYPE	DLONG	DWIDE	DRAD	VOLUME
1	1	1	0	1.00000E+00	1.00000E+00	5.00000E-01	3.14159E+00
2	2	1	2	0.	0.	1.00000E+00	1.00000E-24
31	3	2	0	1.00000E+00	8.00000E-01	1.40000E+00	7.03717E+00
32	4	2	3	1.00000E+00	1.00000E-01	1.85000E+00	1.16239E+00
33	5	2	3	1.00000E+00	1.00000E-01	1.95000E+00	1.22522E+00
4	6	3	2	0.	0.	2.00000E+00	1.00000E-24
5	7	3	0	1.00000E+00	1.00000E-01	2.05000E+00	1.28805E+00
6	8	3	0	1.00000E+00	1.00000E-01	2.15000E+00	1.35088E+00
7	9	3	0	1.00000E+00	2.00000E-01	2.30000E+00	2.89027E+00
8	10	3	0	1.00000E+00	3.00000E-01	2.55000E+00	4.80664E+00
9	11	3	0	1.00000E+00	3.00000E-01	2.85000E+00	5.37212E+00
10	12	3	2	0.	0.	3.00000E+00	1.00000E-24

DATA BLOCK 5 USE ACTUAL NODE INTERFACES FOR INTERFACE RADII.

NOD1	NOD2	INDEX	DEL1	DEL2	DLONG	DRAD	HINT	RINT	AREA
1	2	1	5.0000E-01	0.	1.0000E+00	1.0000E+00	1.0000E+12	0.	6.2832E+00
2	31	2	0.	4.0000E-01	1.0000E+00	1.0000E+00	1.0000E+12	0.	6.2832E+00
31	32	3	4.0000E-01	5.0000E-02	1.0000E+00	1.8000E+00	1.0000E+12	0.	1.1310E+01
32	33	4	5.0000E-02	5.0000E-02	1.0000E+00	1.9000E+00	1.0000E+12	0.	1.1938E+01
33	4	5	5.0000E-02	0.	1.0000E+00	2.0000E+00	1.0000E+12	0.	1.2566E+01
2	4	6	0.	0.	1.0000E+00	1.0000E+00	1.0000E-24	8.0900E-1	6.2832E+00
4	5	7	0.	5.0000E-02	1.0000E+00	2.0000E+00	1.0000E+12	0.	1.2566E+01
5	6	8	5.0000E-02	5.0000E-02	1.0000E+00	2.1000E+00	1.0000E+12	0.	1.3195E+01
6	7	9	5.0000E-02	1.0000E-01	1.0000E+00	2.2000E+00	1.0000E+12	0.	1.3823E+01
7	8	10	1.0000E-01	1.5000E-01	1.0000E+00	2.4000E+00	1.0000E+12	0.	1.5080E+01
8	9	11	1.5000E-01	1.5000E-01	1.0000E+00	2.7000E+00	1.0000E+12	0.	1.6965E+01
9	10	12	1.5000E-01	0.	1.0000E+00	3.0000E+00	1.0000E+12	0.	1.8850E+01

DATA BLOCK 6 OUTER SURFACE CONNECTED TO 25-DEG EXTERNAL TEMP BY H = 0.0002

NODS	NODSB	INDEX	LTABP	POWER	DLONG	DRAD	HSURE	RSURE	AREAS
10	101	1	0	3.3300E-01	1.0000E+00	3.0000E+00	2.0000E-04	0.	1.8850E+01

DATA BLOCK 7 EXTERNAL TEMP IS 25.0 DEGREES C. BOUNDARY NODE 101.

NODB	INDEX	LTABT	TEMPB	SLOPE	TIMEB
101	1	0	2.5000000E+01		0.

Fig. 9. Nitrocellulose cylinder. Sample problem for TRUMP report. (Steady-state calculation.) (continued)

```
DATA BLCCK  9                    NODE 1 HAS HEAT GENERATION RATE 0.4 CAL/SEC-CC.

  NCTE INCEX          TT                AA                BB                GG
    1     1       2.500000E+C1    0.                0.                4.000000E-01
==========================================================================================
DATA ENCEC -1                    END OF TRUMP NITRCCELLLLOSE STEADY STATE PRCBLEM.
==========================================================================================
==========================================================================================
MACHINE TIME (N04/10) =      2  SECONDS.  CPU,I/O,SYS PERCENTS =   36


SUMMARY CF INPUT CATA
BLCCK NUMBER          2     3     4     5     6     7     8     9    10    11    12
TIMES REAC IN         1     0   C01   001   C01     1   000   001   000   001   000
ITEM NAME           MAT   KEM  NODE  NOD1  NODS  NCDB  NODG  NCTE NODF1 NODEP NODP1
MAXIMUM SIZE         40     5   480  12CC   125    20   100   480    80    12   480
UNMCDIFIEC SIZE       0     0     0     0     0     0     0     0     0     0     0
FINAL SIZE            3     C    12    12     1     1     0     1     0     0     0

TABLES              CAPT  CONT    QT    ZT    ET HSLRT TEMPB    GT FLCWT TCTAL
                       0     0     C     0     0     0     0     0     0     0

                MAXIMUM ALLOWED TABLE LENGTH IS  12.

ARRAY STCRAGE REQUIREMENTS.
     NWRECD = (8 + 7*M9)*M2 + 28*M4 + 12*M5 + 3*M9     =     31556
     NWCHEM = 4*M2 + (6 + 9*M9)*M3 + 10*M4 (M3 NOT 0) =      5530
     NWGEN8 = 2*M4 + (3 + 3*M9)*M8          (M8 NOT C) =      4860
     NWFLCW = 4*M4 + (9 + 3*M9)*M1C        (M10 NOT C) =      5520
     NWSURE = (12 + 3*M9)*M6 + (5+3*M9)*M7 (M6 NOT C) =       6820
     NWPLCT = 4*M4 + 3*M11 + (1 + M11)*M1 (M11 NOT C) =       4556
     NWBL12 = 8*M4 + 5*M12                (M12 NOT 0) =       6240

     TCTAL ADJUSTABLE ARRAY STORAGE                     =     65082

CTHER TCTALS     NOSPEC NOGEN NORAD NORADS NMELT NREACT
                      5     1     1     0     C     C
```

Fig. 9. Nitrocellulose cylinder. Sample problem for TRUMP report. (Steady-state calculation.) (continued)

MATERIAL SUMMARY

NAME	MATL	NODES	DENSITY	CAPACITY	TOT VOL	TOT MASS	TOT CAP	TOT HEAT
CORE	1	2	2.50000E+00	6.00000E-01	3.14159E+00	7.85398E+00	4.71239E+00	1.17810E+02
AIR	2	3	1.00000E-03	3.00000E-01	9.42478E+00	9.42478E-03	2.82743E-03	7.06858E-02
NC-NR	3	7	2.00000E+00	2.50000E-01	1.57080E+01	3.14159E+01	7.85398E+00	1.96350E+02
SYSTEM TOTAL		12			2.82743E+01	3.92793E+01	1.25692E+01	3.14230E+02

* NITROCELLULOSE CYLINDER. SAMPLE PROBLEM FOR TRUMP REPORT. 2/27/68. ** 20.172 N04/10

PRINTOUT	CYCLE	MF	MSS	KWIT	DELTMX	SMALL	TVARY	DTEMP	DTMAXS	NUTS
1	0	0	1	0	6.89231E-02	6.89231E-04	2.00000E+01	2.66667E-13	0.	0

TOTAL TIME	TIME STEP	HEAT FLOW	TEMP FROM FLUX	FLUX RATE	TEMP RATE
1.00000E-12	1.00000E-12	2.86986E-42	2.28325E-43	2.86986E-30	2.28325E-31

AVG TEMP	HEAT CAPACITY	HEAT CONTENT	GEN RATE	HEAT GEN	TEMP FROM GEN
2.50000E+01	1.25692E+01	3.14230E+02	1.25664E+00	1.25664E-12	9.99775E-14

NODE	TEMP	DT	DDT	GEN RATE	W	H	F
1	2.5000E+01	2.6667E-13	2.6667E-01	1.2566E+00	1.1781E+02	1.2566E-12	-2.1427E-27
2	2.5000E+01	2.4962E-13	0.	0.	3.7500E-23	3.7442E-37	3.7430E-37
31	2.5000E+01	9.2863E-25	9.2863E-13	0.	5.2779E-02	1.9605E-27	1.9605E-27
32	2.5000E+01	0.	0.	0.	8.7179E-03	0.	0.
33	2.5000E+01	0.	0.	0.	9.1892E-03	0.	0.
4	2.5000E+01	0.	0.	0.	1.2500E-23	0.	1.8227E-28
5	2.5000E+01	0.	0.	0.	1.5101E+01	0.	0.
6	2.5000E+01	0.	0.	0.	1.5886E+01	0.	0.
7	2.5000E+01	0.	0.	0.	3.6128E+01	0.	0.
8	2.5000E+01	0.	0.	0.	6.0083E+01	0.	0.
9	2.5000E+01	0.	0.	0.	6.7152E+01	0.	0.
10	2.5000E+01	0.	0.	0.	1.2500E-23	0.	0.

MATERIAL DATA

NAME	MATL	TOT CAP	TOT HEAT	AVG TEMP	TMELT	HMELT
CORE	1	4.71239E+00	1.17810E+02	2.50000E+01	0.	0.
AIR	2	2.82743E-03	7.06858E-02	2.50000E+01	0.	0.
NC-NR	3	7.85398E+00	1.96350E+02	2.50000E+01	0.	0.

Fig. 9. Nitrocellulose cylinder. Sample problem for TRUMP report. (Steady-state calculation.) (continued)

NODE DATA. CHECK TOTAL CONDUCTANCES (ZIP) AND TIME CONSTANTS (SLIM).
LARGE DIFFERENCES BETWEEN NODES MAY BE DUE TO POOR ZONING, AND MAY PRODUCE POOR RESULTS.

NODE	MATL	NTYPE	RADIUS	VOLUME	MASS	CAPACITY	CONDUCTIVITY	ZIP	SLIM
1	1	0	5.0000E-01	3.1416E+00	7.8540E+00	4.7124E+00	1.0000E-02	1.2566E-01	3.7500E+01
2	1	2	1.0000E+00	1.0000E-24	2.5000E-24	1.5000E-24	1.0000E-02	1.3425E-01	1.1173E-23
31	2	0	1.4000E+00	7.0372E+00	7.0372E-03	2.1112E-03	5.0000E-04	2.0420E-02	1.0338E-01
32	2	3	1.8500E+00	1.1624E+00	1.1624E-03	3.4872E-04	5.0000E-04	7.2257E-02	4.8261E-03
33	2	3	1.9500E+00	1.2252E+00	1.2252E-03	3.6757E-04	5.0000E-04	1.8535E-01	1.9831E-03
4	3	2	2.0000E+00	1.0000E-24	2.0000E-24	5.0000E-25	5.0000E-04	2.5206E-01	1.9837E-24
5	3	0	2.0500E+00	1.2881E+00	2.5761E+00	6.4403E-01	5.0000E-04	1.9164E-01	3.3607E+00
6	3	0	2.1500E+00	1.3509E+00	2.7018E+00	6.7544E-01	5.0000E-04	1.1205E-01	6.0280E+00
7	3	0	2.3000E+00	2.8903E+00	5.7805E+00	1.4451E+00	5.0000E-04	7.6236E-02	1.8956E+01
8	3	0	2.5500E+00	4.8066E+00	9.6133E+00	2.4033E+00	5.0000E-04	5.8434E-02	4.1129E+01
9	3	0	2.8500E+00	5.3721E+00	1.0744E+01	2.6861E+00	5.0000E-04	9.1106E-02	2.9483E+01
10	3	2	3.0000E+00	1.0000E-24	2.0000E-24	5.0000E-25	5.0000E-04	6.2832E-02	7.9577E-24

INTERNAL CONNECTION DATA. CHECK CONDUCTANCES (TRAN).
LARGE DIFFERENCES BETWEEN CONNECTIONS MAY BE DUE TO POOR ZONING, AND MAY PRODUCE POOR RESULTS.

NOD1	NOD2	AREA	HINT	RINT	TRAN	HEAT FLOW	AVG RATE
1	2	6.2832E+00	1.0000E+12	0.	1.2566E-01	-2.1427E-27	-2.1427E-15
2	31	6.2832E+00	1.0000E+12	0.	7.8540E-03	-1.9605E-27	-1.9605E-15
31	32	1.1310E+01	1.0000E+12	0.	1.2566E-02	0.	0.
32	33	1.1938E+01	1.0000E+12	0.	5.9690E-02	0.	0.
33	4	1.2566E+01	1.0000E+12	0.	1.2566E-01	0.	0.
2	4	6.2832E+00	1.0000E-24	8.0900E-01	7.3018E-04	-1.8227E-28	-1.8227E-16
4	5	1.2566E+01	1.0000E+12	0.	1.2566E-01	0.	0.
5	6	1.3195E+01	1.0000E+12	0.	6.5973E-02	0.	0.
6	7	1.3823E+01	1.0000E+12	0.	4.6077E-02	0.	0.
7	8	1.5080E+01	1.0000E+12	0.	3.0159E-02	0.	0.
8	9	1.6965E+01	1.0000E+12	0.	2.8274E-02	0.	0.
9	10	1.8850E+01	1.0000E+12	0.	6.2832E-02	0.	0.

BOUNDARY NODE DATA

NODB	TEMPB	HEAT FLOW	AVG RATE
101	2.5000E+01	0.	0.
SYSTEM TOTAL		0.	0.

Fig. 9. Nitrocellulose cylinder. Sample problem for TRUMP report. (Steady-state calculation.) (continued)

EXTERNAL CONNECTION DATA

NCCS	NCCSB	AREAS	HSURE	POWER	RSURE	TRANS	HEAT FLOW	AVG RATE
10	101	1.8850E+01	2.0000E-04	3.3300E-01	0.	3.8400E-11	0.	0.

KCYC = 0, ITERATIONS... TOTAL = 0, AVERAGE = 0, MAXIMUM = 0. FOR = 0.570 .
* NITROCELLULOSE CYLINDER. SAMPLE PROBLEM FOR TRUMP REPORT. 2/27/68. ** 20.172 N04/10

PRINTOUT	CYCLE	MF	MSS	KWIT	DELTMX	SMALL	TVARY	DTEMP	DTMAXS	NUTS
3	52	0	17	4	1.00000E+03	1.96552E-01	2.00000E+01	9.27702E-03	5.00000E-01	1

TOTAL TIME	TIME STEP	HEAT FLOW	TEMP FROM FLUX	FLUX RATE	TEMP RATE
1.64562E+04	1.00000E+03	-1.77430E+04	-1.41163E+03	-1.07819E+00	-8.57807E-02

AVG TEMP	HEAT CAPACITY	HEAT CONTENT	GEN RATE	HEAT GEN	TEMP FROM GEN
2.57226E+02	1.25692E+01	3.23313E+03	1.25664E+00	2.06795E+04	1.64525E+03

NODE	TEMP	DT	DDT	GEN RATE	W	H	F
1	3.9733E+02	8.5763E-03	4.7070E-06	1.2566E+00	1.8724E+03	1.7545E+03	-1.8925E+04
2	3.8733E+02	7.7048E-03	4.2287E-06	0.	5.8099E-22	5.4349E-22	3.1479E-01
31	3.2037E+02	8.7911E-03	4.8249E-06	0.	6.7634E-01	6.2357E-01	8.8779E-01
32	2.7852E+02	9.2270E-03	5.0642E-06	0.	9.7124E-02	8.8406E-02	1.4752E+00
33	2.6971E+02	9.0639E-03	4.9747E-06	0.	9.9136E-02	8.9947E-02	3.7969E+00
4	2.6552E+02	9.2770E-03	5.0916E-06	0.	1.3276E-22	1.2026E-22	4.7769E+00
5	2.5552E+02	9.0130E-03	4.9467E-06	0.	1.6456E+02	1.4846E+02	1.5191E+02
6	2.3648E+02	8.2898E-03	4.5498E-06	0.	1.5973E+02	1.4284E+02	1.4445E+02
7	2.0920E+02	7.9067E-03	4.3396E-06	0.	3.0233E+02	2.6620E+02	2.6700E+02
8	1.6754E+02	6.1241E-03	3.3612E-06	0.	4.0265E+02	3.4256E+02	3.4306E+02
9	1.2309E+02	3.9503E-03	2.1681E-06	0.	3.3064E+02	2.6349E+02	2.637[illegible]E+02
10	1.0309E+02	3.5585E-03	1.9531E-06	0.	5.1547E-23	3.9047E-23	4.2243E-01

MATERIAL DATA

NAME	MATL	TOT CAP	TOT HEAT	AVG TEMP	TMELT	HMELT
CORE	1	4.71239E+00	1.87235E+03	3.97325E+02	0.	0.
AIR	2	2.82743E-03	8.72605E-01	3.08621E+02	0.	0.
NC-NR	3	7.85398E+00	1.3559CE+03	1.73148E+02	0.	0.

Fig. 9. Nitrocellulose cylinder. Sample problem for TRUMP report. (Steady-state calculation.) (continued)

NODE DATA. CHECK TOTAL CONDUCTANCES (ZIP) AND TIME CONSTANTS (SLIM).
LARGE DIFFERENCES BETWEEN NODES MAY BE DUE TO POOR ZONING, AND MAY PRODUCE POOR RESULTS.

NODE	MATL	NTYPE	RADIUS	VOLUME	MASS	CAPACITY	CONDUCTIVITY	ZIP	SLIM
1	1	4	5.0000E-01	3.1416E+00	7.8540E+00	4.7124E+00	1.0000E-02	1.2566E-01	3.7500E+01
2	1	2	1.0000E+00	1.0000E-24	2.5000E-24	1.5000E-24	1.0000E-02	1.3952E-01	1.0751E-23
31	2	4	1.4000E+00	7.0372E+00	7.0372E-03	2.1112E-03	5.0000E-04	2.0420E-02	1.0338E-01
32	2	3	1.8500E+00	1.1624E+00	1.1624E-03	3.4872E-04	5.0000E-04	7.2257E-02	4.8261E-03
33	2	3	1.9500E+00	1.2252E+00	1.2252E-03	3.6757E-04	5.0000E-04	1.8535E-01	1.9831E-03
4	3	2	2.0000E+00	1.0000E-24	2.0000E-24	5.0000E-25	5.0000E-04	2.5733E-01	1.9431E-24
5	3	4	2.0500E+00	1.2881E+00	2.5761E+00	6.4403E-01	5.0000E-04	1.9164E-01	3.3607E+00
6	3	4	2.1500E+00	1.3509E+00	2.7018E+00	6.7544E-01	5.0000E-04	1.1205E-01	6.0280E+00
7	3	4	2.3000E+00	2.8903E+00	5.7805E+00	1.4451E+00	5.0000E-04	7.6236E-02	1.8956E+01
8	3	4	2.5500E+00	4.8066E+00	9.6133E+00	2.4033E+00	5.0000E-04	5.8434E-02	4.1129E+01
9	3	4	2.8500E+00	5.3721E+00	1.0744E+01	2.6861E+00	5.0000E-04	9.1106E-02	2.9483E+01
10	3	2	3.0000E+00	1.0000E-24	2.0000E-24	5.0000E-25	5.0000E-04	7.8922E-02	6.3353E-24

INTERNAL CONNECTION DATA. CHECK CONDUCTANCES (TRAN).
LARGE DIFFERENCES BETWEEN CONNECTIONS MAY BE DUE TO POOR ZONING, AND MAY PRODUCE POOR RESULTS.

NOD1	NOD2	AREA	HINT	RINT	TRAN	HEAT FLOW	AVG RATE
1	2	6.2832E+00	1.0000E+12	0.	1.2566E-01	-1.8925E+04	-1.1500E+00
2	31	6.2832E+00	1.0000E+12	0.	7.8540E-03	-8.3751E+03	-5.0893E-01
31	32	1.1310E+01	1.0000E+12	0.	1.2566E-02	-8.3742E+03	-5.0888E-01
32	33	1.1938E+01	1.0000E+12	0.	5.9690E-02	-8.3728E+03	-5.0879E-01
33	4	1.2566E+01	1.0000E+12	0.	1.2566E-01	-8.3690E+03	-5.0856E-01
2	4	6.2832E+00	1.0000E-24	8.0900E-01	5.9994E-03	-1.0549E+04	-6.4106E-01
4	5	1.2566E+01	1.0000E+12	0.	1.2566E-01	-1.8914E+04	-1.1493E+00
5	6	1.3195E+01	1.0000E+12	0.	6.5973E-02	-1.8762E+04	-1.1401E+00
6	7	1.3823E+01	1.0000E+12	0.	4.6077E-02	-1.8617E+04	-1.1313E+00
7	8	1.5080E+01	1.0000E+12	0.	3.0159E-02	-1.8350E+04	-1.1151E+00
8	9	1.6965E+01	1.0000E+12	0.	2.8274E-02	-1.8007E+04	-1.0942E+00
9	10	1.8850E+01	1.0000E+12	0.	6.2832E-02	-1.7743E+04	-1.0782E+00

BOUNDARY NODE DATA

NODB	TEMPB	HEAT FLOW	AVG RATE
101	2.5000E+01	-1.7743E+04	-1.0782E+00

SYSTEM TOTAL	-1.7743E+04	-1.0782E+00

Fig. 9. Nitrocellulose cylinder. Sample problem for TRUMP report. (Steady-state calculation.) (continued)

```
EXTERNAL CONNECTION DATA

  NCDS NCDSB     AREAS        HSURE        POWER        RSURE        TRANS       HEAT FLOW    AVG RATE
    10   101  1.8850E+01  2.0000E-04  3.3300E-01  0.           1.6091E-02 -1.7743E+04 -1.0782E+00
==========================================================================================================

KCYC =     52, ITERATIONS...  TOTAL =  1076, AVERAGE =    20, MAXIMUM =    53.  FOR = 0.646 .
ENDED PRCB    2     KCYC =    52     KWIT =  4     SUMTIM =  1.64562E+04                20.172  N04/10
KWIT...1=TIMAX, 2=TMAX, 3=TMIN, 4=S.S., 5=DATA, 6=CARD OR KEY, 7=MCYC, 8=MSEC, 9=BLK 2 OR 4,
    10=CONVERGENCE FAILURE, 11=PROBLEM SIZE LIMIT, 12=TABLE LENGTH OR ORDER.


MACHINE TIME (N04/10) =     9  SECONDS.  CPU,I/O,SYS PERCENTS =   51:48: 0
```

Fig. 9. Nitrocellulose cylinder. Sample problem for TRUMP report. (Steady-state calculation.) (continued)

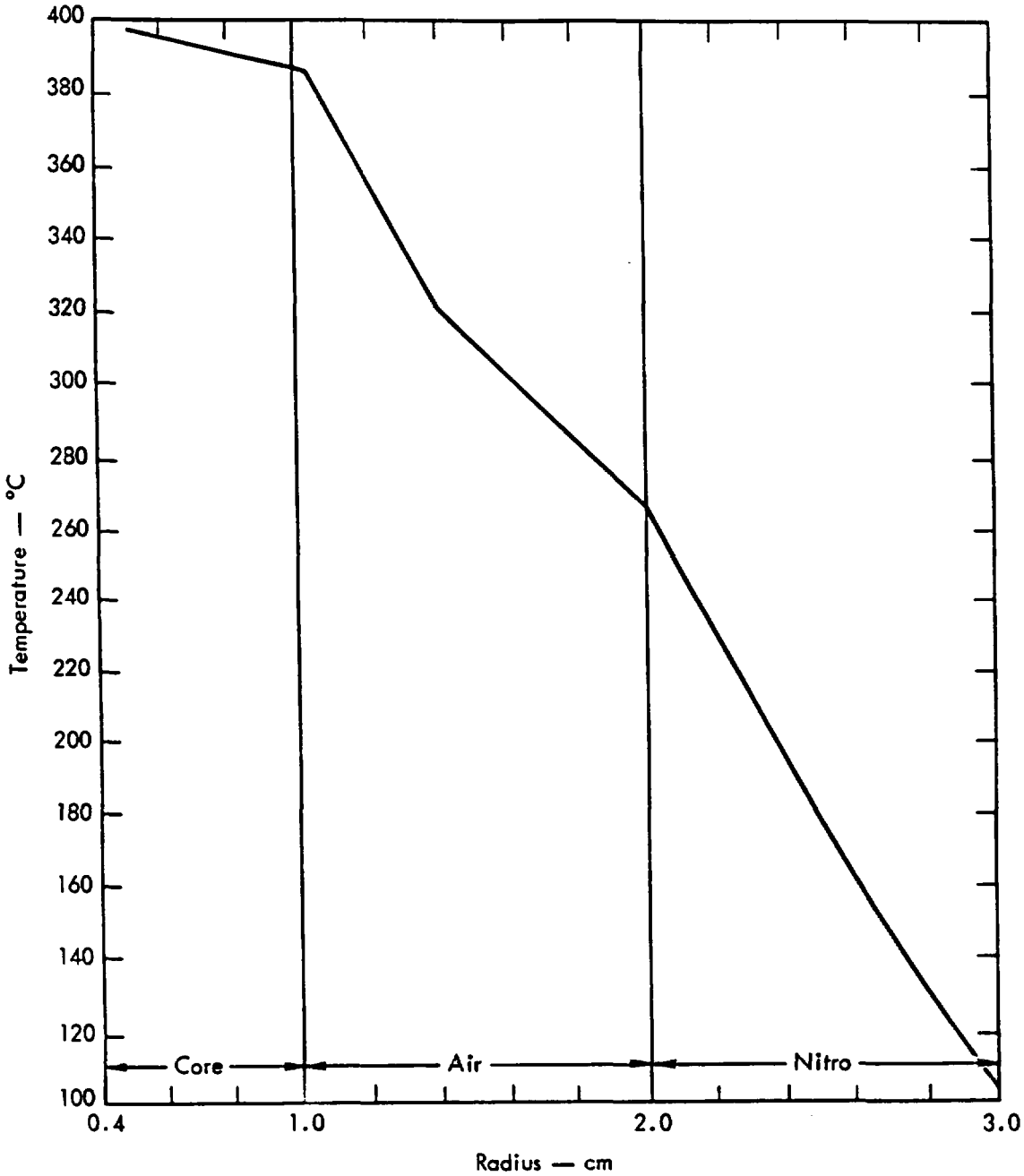

Fig. 10. Plot of temperature vs radius for nitrocellulose cylinder. (Steady-state calculations.)

```
* NITROCELLULOSE CYLINDER.  SAMPLE PROBLEM FOR TRUMP REPORT.  2/27/68.(2)      **
NOTE.    REACTICN AND IGNITION TAKE PLACE.  FIND IGNITION TEMPERATURE AND TIME.
NOTE.    UPPER TEMPERATURE LIMIT IS 400.0 TC END PROBLEM AT IGNITICN.
NOTE.    THIS PROBLEM DCES NOT USE FLOW SUBROUTINE.
BLOCK 1  TVARY 2.0 FOR ACCURACY.  SMALL 0.002 FOR ACCURATE IGNITICN TIME.
 9999    4    1
    2                C.002     2.0                                        4C0.C
 25.0
BLOCK11 GET INTERPOLATED T VS R PLOTS, T VS TIME PLOTS WITH LIN AND LOG SCALES.
   -5  100        -4
    1    2   31   32   33    4    5    6    7    8    9   10
BLOCK 26 MODIFY PROPERTIES CF NITRCCELLULOSE TO INCLUDE CHEM REACTION.
NITRO    3    1                2.0       0.25      5.0   E-4

BLOCK 3  CHEMICAL COMPONENT NAMES, NUMBERS, AND REACTION PROPERTIES.
N.C.     1                     800.      48.3      5.0    E4

BLOCK 98  NODE 1 HAS C.4 HEAT GEN, NODES 4 TO 10 HAVE 1.0 CONC A.
    4    6    1                1.0

ENDED-1.  LAST CARD OF DATA DECK.
COL 5   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80
```

Fig. 11. Input data for nitrocellulose cylinder problem. (Reaction calculation utilizes data from steady-state calculation.)

Numerical output data for the reaction and ignition calculation are shown in Fig. 12. In this problem, the heat balance is very precise.

The calculated temperature vs radius curves for the last 112 sec before ignition are shown superimposed in Fig. 13. The beginning and acceleration of reaction in the nitrocellulose are shown by the numerical printouts of the temperature, rate of temperature change, and concentration in node 4 preceding the printout for the final time step. The beginning of appreciable reaction first causes an increase in the rate of temperature change of node 4 at about 1580 sec. By 1830 sec, the rate of increase has doubled. This indicates that the rate of heat addition due to chemical reaction equals that previously due to heat transport, and that turning off the heater probably would not stop the reaction; the "point of no return" has been reached. The reaction rate, temperature, and rate of temperature change in node 4 increase rapidly from 1820 to 1878.07 sec. At that time, the temperature in node 5 equals that in node 4, which is losing heat to the adjacent air. By 1878.10 sec, the temperature in node 5 exceeds TMAX, which is 400°C, and the problem ends. Other versions of this problem, using different zoning and different values of TVARY and SMALL, resulted in ignition times ranging from 1878 to 1955 sec, the latter value with coarser zoning near the inner surface of nitrocellulose.

Great care must be taken in calculating thermal ignition caused by heat input through a surface, so as not to distort the results by the effects of coarse zoning or unrealistic boundary conditions. Zoning should be fine enough that the calculated temperature profile in the reacting material does not have large discontinuities in slope. The region near the surface in which reaction rates become large should be subdivided into several zones. The boundary condition at the surface should be as physically realistic as possible. For example, use of a fixed surface temperature instead of a large heat-transfer coefficient connecting the surface to an external temperature may prevent ignition from occurring, particularly when the material is too coarsely zoned near the surface. In this problem, the surface is only weakly connected thermally to the air in the gap and to the heater-core surface, so coarse zoning is a less serious problem.

The steady-state calculation requires 46 time steps, with 1815 iterations, to reach steady state, at a total time of 15,955 sec. The average time step is 347 sec, and the average number of iterations per time step is 39.5. Machine time is about 14 sec on the CDC/3600, for the time step phase, or 8 msec per time step per calculational item (see Section IV.F).

The ignition and reaction calculation requires 181 time steps, with 2386 iterations, to reach ignition at a total time of 1878 sec. The average time step is 1.0 sec, and the average number of iterations per time step is 13. Machine time is 55 sec on the CDC/3600 for the time step phase, or 8 msec per time step per calculational items (see Section IV.F). These times are larger than normal, because of the high frequency of CRT plot output and the large number of iterations per time step, and because the numerical output goes to a special tape, using nonbuffered output routines.

```
TRUMP OUTPUT DATA              PROGRAM DATE  15.774  NC4/05                                  DATA DECK    4

* NITRCCELLULOSE CYLINDER.  SAMPLE PROBLEM FOR TRUMP REPORT.  2/27/68 (2)   **        20.177  NC4/1
=================================================================================================

WILL USE DATA AND INITIAL CONDITIONS FROM PRECEDING PROBLEM WITH NAME, DATE ---
* NITRCCELLULOSE CYLINDER.  SAMPLE PROBLEM FOR TRUMP REPORT.  2/27/68.        **        20.172  N 4/1C

SUMMARY CF INPUT DATA
BLCCK NUMBER          2     3     4     5     6     7     8     9    10    11    12
ITEM NAME           MAT   KEM  NODE  NOD1  NODS  NODB  NODG  NCTE NODF1 NODEP NODP1
INITIAL SIZE          3     C    12    12     1     1     0     1     0     0     0
=================================================================================================
                                REACTION AND IGNITION TAKE PLACE.  FIND IGNITION TEMPERATURE AND TIME.
                                UPPER TEMPERATURE LIMIT IS 400.0 TC END PROBLEM AT IGNITION.
                                THIS PROBLEM DOES NOT USE FLCW SUBRCUTINE.
DATA BLCCK  1                   TVARY 2.C FOR ACCURACY.  SMALL 0.002 FOR ACCURATE IGNITION TIME.

IPRINT  NUM KCATA KSPEC  MCYC  MSEC NPUNCH NDOT IRITE ITAPE   TIMEP        SCALE
 9999     4     1     C     C     C     C    0     0     0   0.          1.00000E+00

  KC   KT    CELTO        SMALL        TVARY        TAU          TIMAX        TMIN         TMAX
   2    1  1.00000E+12  2.0000CE-03  2.0CCCOE+00  C.           0.           -2.0000CE-03  4.000C2E+02

  KC KSYM    GEOM         SIGMA        TBASE
   2    1  6.28319E+00  1.35500E-12  2.73150E+02

  TCNE        ALONE       BONE        GONC        FONE        HONE        RCNE        PONE
 2.5000E+01  0.          0.          0.          0.          0.          0.          0.
=================================================================================================
DATA BLCCK 11                   GET INTERPOLATED T VS R PLOTS, T VS TIME PLOTS WITH LIN AND LOG SCALES.

WILL MAKE TEMP VS RADIUS PLOTS

 JFIC  LCGR LOCTEM LOGTIM    FRAD1       FRAD2       FTEMP1      FTEMP2      FTIME1      FTIME2
    5     0     0     4     0.          1.0000E+00  0.          1.0000E+00  0.          1.0000E+00
```

Fig. 12. Nitrocellulose cylinder. Sample problem for TRUMP report. (Reaction and ignition calculation.)

NODES FOR TEMPERATURE VS TIME CRT PLOTS. MAX = 12

INDEX	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
NODE	1	2	31	32	33	4	5	6	7	8	9	10				

DATA BLOCK 2B MODIFY PROPERTIES OF NITROCELLULOSE TO INCLUDE CHEM REACTION.
DATA MODIFIES OR ADDS TO DATA ALREADY READ IN,

NAME	MATL	INDEX	KA	KB	LTABC	LTABK	DENSITY	CAPACITY	CONDUCTIVITY	TMELT	HMELT
NITRO	3	3	1	0	0	0	2.0000E+00	2.5000E-01	5.0000E-04	0.	0.

DATA BLOCK 3 CHEMICAL COMPONENT NAMES, NUMBERS, AND REACTION PROPERTIES.

NAME	REACT	INDEX	LTABQ	LTABZ	LTABE	QT	ZT	ET	ORDER
N.C.	1	1	0	0	0	8.00000E+02	4.83000E+01	5.00000E+04	1.00000E+00

DATA BLOCK 9B NODE 1 HAS 0.4 HEAT GEN, NODES 4 TO 10 HAVE 1.0 CONC A.
DATA MODIFIES OR ADDS TO DATA ALREADY READ IN,

NOTE	INDEX	TT	AA	BB	GG
4	6	2.500000E+01	1.000000E+00	0.	0.
5	7	2.500000E+01	1.000000E+00	0.	0.
6	8	2.500000E+01	1.000000E+00	0.	0.
7	9	2.500000E+01	1.000000E+00	0.	0.
8	10	2.500000E+01	1.000000E+00	0.	0.
9	11	2.500000E+01	1.000000E+00	0.	0.
10	12	2.500000E+01	1.000000E+00	0.	0.

DATA ENDEC -1. LAST CARD OF DATA DECK.

MACHINE TIME (NO4/10) = 1 SECONDS. CPU,I/O,SYS PERCENTS = 33:66:0

SUMMARY OF INPUT DATA

BLOCK NUMBER	2	3	4	5	6	7	8	9	10	11	12
TIMES READ IN	001	1	000	0	0	0	0	1	0	1	0
ITEM NAME	MAT	KEM	NODE	NOD1	NODS	NODB	NODG	NOTE	NODF1	NODEP	NODP1
MAXIMUM SIZE	40	5	480	1200	125	20	100	480	80	12	480

Fig. 12. Nitrocellulose cylinder. Sample problem for TRUMP report. (Reaction and ignition calculation.) (continued)

```
UNMODIFIED SIZE      2     0    12    12     1     1     0     1     0     0     0
FINAL SIZE           3     1    12    12     1     1     0    12     0    12     0

TABLES             CAPT  CONT    QT    ZT    ET HSLRT TEMPB    GT FLOWT TOTAL
                      0     0     0     0     0     0     0     0     0     0

                MAXIMUM ALLOWED TABLE LENGTH IS  12.

ARRAY STORAGE REQUIREMENTS.
     NWREQD = (8 + 7*M9)*M2 + 28*M4 + 12*M5 + 3*M9      =     31556
     NWCHEM = 4*M2 + (6 + 9*M9)*M3 + 10*M4 (M3 NOT 0)   =      5530
     NWGEN8 = 2*M4 + (3 + 3*M9)*M8         (M8 NOT 0)   =      4860
     NWFLOW = 4*M4 + (9 + 3*M9)*M10       (M10 NOT 0)   =      5520
     NWSURE = (12 + 3*M9)*M6 + (5+3*M9)*M7 (M6 NOT 0)   =      6820
     NWPLCT = 4*M4 + 3*M11 + (1 + M11)*M1 (M11 NOT 0)   =      4556
     NWBL12 = 8*M4 + 5*M12                (M12 NOT 0)   =      6240

     TOTAL ADJUSTABLE ARRAY STORAGE                     =     65082

OTHER TOTALS     NOSPEC NOGEN NORAD NORADS NMELT NREACT
                      5     1     1     0     3     7

MATERIAL SUMMARY

  NAME   MATL NODES   DENSITY      CAPACITY     TOT VOL      TOT MASS     TOT CAP      TOT HEAT
CORE       1     2  2.50000E+00  6.00000E-01  3.14159E+00  7.85398E+00  4.71239E+00  1.17810E+02
AIR        2     3  1.00000E-03  3.00000E-01  9.42478E+00  9.42478E-03  2.82743E-03  7.06858E-02
NITRO      3     7  2.00000E+00  2.50000E-01  1.57080E+01  3.14159E+01  7.85398E+00  1.96350E+02

SYSTEM TOTAL     12                            2.82743E+01  3.92793E+01  1.25692E+01  3.14230E+02
* NITROCELLULOSE CYLINDER.  SAMPLE PROBLEM FOR TRUMP REPORT.  2/27/68.(2)      **      20.177  N04/10
==========================================================================================

PRINTOUT CYCLE  MF   MSS  KWIT   DELTMX       SMALL        TVARY        DTEMP        DTMAXS      NUTS
    1     0     0     1     0  6.89231E-02  2.00000E-03  2.00000E+00  2.66667E-13  0.              0

  TOTAL TIME     TIME STEP     HEAT FLOW    TEMP FROM FLUX   FLUX RATE     TEMP RATE
 1.00000E-12   1.00000E-12   9.65630E-42   7.68251E-43   9.65630E-30   7.68251E-31

  AVG TEMP     HEAT CAPACITY   HEAT CONTENT    GEN RATE      HEAT GEN     TEMP FROM GEN
 2.50000E+01   1.25692E+01   3.14230E+02   1.25664E+00   1.25664E-12   9.99775E-14
==========================================================================================
```

Fig. 12. Nitrocellulose cylinder. Sample problem for TRUMP report. (Reaction and ignition calculation.) (continued)

NODE	TEMP	DT	DDT	GEN RATE	W	H	F
1	2.5000E+01	2.6667E-13	2.6667E-01	1.2566E+00	1.1781E+02	1.2566E-12	-2.1427E-27
2	2.5000E+01	2.4962E-13	0.	0.	3.7500E-23	3.7442E-37	3.7454E-37
31	2.5000E+01	9.2863E-25	9.2863E-13	0.	5.2779E-02	1.9605E-27	1.96[illegible]5E-27
32	2.5000E+01	0.	0.	0.	8.7179E-03	0.	0.
33	2.5000E+01	0.	0.	0.	9.1892E-03	0.	4.2118E-38
4	2.5000E+01	3.3517E-25	0.	0.	1.2500E-23	1.6758E-49	1.8227E-28
5	2.5000E+01	6.7228E-25	6.7228E-13	0.	1.6101E+01	4.3297E-25	-4.2363E-38
6	2.5000E+01	6.7228E-25	6.7228E-13	0.	1.5886E+01	4.5409E-25	0.
7	2.5000E+01	6.7228E-25	6.7228E-13	0.	3.6128E+01	9.7154E-25	0.
8	2.5000E+01	6.7228E-25	6.7228E-13	0.	6.0083E+01	1.6157E-24	0.
9	2.5000E+01	6.7228E-25	6.7228E-13	0.	6.7152E+01	1.8058E-24	-2.5816E-47
10	2.5000E+01	6.7228E-25	0.	0.	1.2500E-23	3.3614E-49	-5.4699E-53

NODE	TEMP	W/F A	DA	DDA	W/F B	DB	DDB
4	2.5000E+01	1.0000E+00	-2.1009E-28	0.	0.	0.	0.
5	2.5000E+01	1.0000E+00	-2.1009E-28	-2.1009E-16	0.	0.	0.
6	2.5000E+01	1.0000E+00	-2.1009E-28	-2.1009E-16	0.	0.	0.
7	2.5000E+01	1.0000E+00	-2.1009E-28	-2.1009E-16	0.	0.	0.
8	2.5000E+01	1.0000E+00	-2.1009E-28	-2.1009E-16	0.	0.	0.
9	2.5000E+01	1.0000E+00	-2.1009E-28	-2.1009E-16	0.	0.	0.
10	2.5000E+01	1.0000E+00	-2.1009E-28	0.	0.	0.	0.

MATERIAL DATA

NAME	MATL	TOT CAP	TOT HEAT	AVG TEMP	TMELT	HMELT
CORE	1	4.71239E+00	1.1781CE+02	2.50000E+01	0.	0.
AIR	2	2.82743E-03	7.06858E-02	2.50000E+01	0.	0.
NITRO	3	7.85398E+00	1.96350E+02	2.50000E+01	0.	0.

NODE DATA. CHECK TOTAL CONDUCTANCES (ZIP) AND TIME CONSTANTS (SLIM).
LARGE DIFFERENCES BETWEEN NODES MAY BE DUE TO POOR ZONING, AND MAY PRODUCE POOR RESULTS.

NODE	MATL	NTYPE	RADIUS	VOLUME	MASS	CAPACITY	CONDUCTIVITY	ZIP	SLIM
1	1	0	5.0000E-01	3.1416E+00	7.8540E+00	4.7124E+00	1.0000E-02	1.2566E-[illegible]1	3.750[illegible]E+01
2	1	2	1.0000E+00	1.0000E-24	2.5000E-24	1.5000E-24	1.0000E-02	1.3425E-01	1.1173E-23
31	2	0	1.4000E+00	7.0372E+00	7.0372E-03	2.1112E-03	5.0000E-04	2.0420E-02	1.0338E-01
32	2	3	1.8500E+00	1.1624E+00	1.1624E-03	3.4872E-04	5.0000E-04	7.2257E-02	4.8261E-03
33	2	3	1.9500E+00	1.2252E+00	1.2252E-03	3.6757E-04	5.0000E-04	1.8535E-01	1.9831E-03
4	3	2	2.0000E+00	1.0000E-24	2.0000E-24	5.0000E-25	5.0000E-04	2.5206E-01	1.9837E-24
5	3	0	2.0500E+00	1.2881E+00	2.5761E+00	6.4403E-01	5.0000E-04	1.9164E-01	3.3607E+00

Fig. 12. Nitrocellulose cylinder. Sample problem for TRUMP report. (Reaction and ignition calculation.) (continued)

6	3	0	2.1500E+00	1.3509E+00	2.7018E+00	6.7544E-01	5.0000E-04	1.1205E-01	6.0280E+00
7	3	0	2.3000E+00	2.8903E+00	5.7805E+00	1.4451E+00	5.0000E-04	7.6236E-02	1.8956E+01
8	3	0	2.5500E+00	4.8066E+00	9.6133E+00	2.4033E+00	5.0000E-04	5.8434E-02	4.1129E+01
9	3	0	2.8500E+00	5.3721E+00	1.0744E+01	2.6861E+00	5.0000E-04	9.1106E-02	2.8483E+01
10	3	2	3.0000E+00	1.0000E-24	2.0000E-24	5.0000E-25	5.0000E-04	6.2832E-02	7.9577E-24

REACTANT DATA

NODE	A	B	MASS	QA	ZA	EA	QB	ZB	EB
4	1	0	2.0000E-24	8.0000E+02	4.8300E+01	5.0000E+04	0.	0.	0.
5	1	0	2.5761E+00	8.0000E+02	4.8300E+01	5.0000E+04	0.	0.	0.
6	1	0	2.7018E+00	8.0000E+02	4.8300E+01	5.0000E+04	0.	0.	0.
7	1	0	5.7805E+00	8.0000E+02	4.8300E+01	5.0000E+04	0.	0.	0.
8	1	0	9.6133E+00	8.0000E+02	4.8300E+01	5.0000E+04	0.	0.	0.
9	1	0	1.0744E+01	8.0000E+02	4.8300E+01	5.0000E+04	0.	0.	0.
10	1	0	2.0000E-24	8.0000E+02	4.8300E+01	5.0000E+04	0.	0.	0.

INTERNAL CONNECTION DATA. CHECK CONDUCTANCES (TRAN).
LARGE DIFFERENCES BETWEEN CONNECTIONS MAY BE DUE TO POOR ZONING, AND MAY PRODUCE POOR RESULTS.

NOD1	NOD2	AREA	HINT	RINT	TRAN	HEAT FLOW	AVG RATE
1	2	6.2832E+00	1.0000E+12	0.	1.2566E-01	-2.1427E-27	-2.1427E-15
2	31	6.2832E+00	1.0000E+12	0.	7.8540E-03	-1.9605E-27	-1.9605E-15
31	32	1.1310E+01	1.0000E+12	0.	1.2566E-02	0.	0.
32	33	1.1938E+01	1.0000E+12	0.	5.9690E-02	0.	0.
33	4	1.2566E+01	1.0000E+12	0.	1.2566E-01	4.2118E-38	4.2118E-26
2	4	6.2832E+00	1.0000E-24	8.0900E-01	7.3018E-04	-1.8227E-28	-1.8227E-16
4	5	1.2566E+01	1.0000E+12	0.	1.2566E-01	4.2363E-38	4.2363E-26
5	6	1.3195E+01	1.0000E+12	0.	5.5973E-02	0.	0.
6	7	1.3823E+01	1.0000E+12	0.	4.6077E-02	0.	0.
7	8	1.5080E+01	1.0000E+12	0.	3.0159E-02	0.	0.
8	9	1.6965E+01	1.0000E+12	0.	2.8274E-02	0.	0.
9	10	1.8850E+01	1.0000E+12	0.	6.2832E-02	-2.5816E-47	-2.5816E-35

BOUNDARY NODE DATA

NODB	TEMPB	HEAT FLOW	AVG RATE
101	2.5000E+01	-2.5816E-47	-2.5816E-35
SYSTEM TOTAL		-2.5816E-47	-2.5816E-35

Fig. 12. Nitrocellulose cylinder. Sample problem for TRUMP report. (Reaction and ignition calculation.) (continued)

EXTERNAL CONNECTION DATA

NCDS	NCDSB	AREAS	HSURE	POWER	RSURE	TRANS	HEAT FLOW	AVG RATE
10	101	1.8850E+01	2.0000E-04	3.3300E-01	0.	3.8400E-11	-2.5816E-47	-2.5816E-35

KCYC = 0, ITERATIONS... TOTAL = 0, AVERAGE = 0, MAXIMUM = 0. FOR = 0.570 .

* NITROCELLULOSE CYLINDER. SAMPLE PROBLEM FOR TRUMP REPORT. 2/27/68.(2) ** 20.177 N04/10

PRINTOUT	CYCLE	MF	MSS	KWIT	DELTMX	SMALL	TVARY	DTEMP	DTMAXS	NUTS
3	182	7	5	2	2.50000E+01	2.00000E-03	2.00000E+00	2.97724E+03	9.00000E-01	18

TOTAL TIME	TIME STEP	HEAT FLOW	TEMP FROM FLUX	FLUX RATE	TEMP RATE
1.87671E+03	2.00000E-03	-4.34138E+02	-3.45398E+01	-2.31329E-01	-1.84044E-02

AVG TEMP	HEAT CAPACITY	HEAT CONTENT	GEN RATE	HEAT GEN	TEMP FROM GEN
3.42480E+02	1.25692E+01	4.30470E+03	1.25664E+00	2.35835E+03	1.87629E+02

NODE	TEMP	DT	DDT	GEN RATE	W	H	F
1	2.9509E+02	2.8660E-03	2.8298E+00	1.2566E+00	1.3906E+03	1.2728E+03	-1.0856E+03
2	3.6406E+02	7.1313E+01	7.0413E+04	0.	5.4609E-22	5.0859E-22	5.1512E-03
31	2.5239E+02	1.2524E+00	1.2366E+03	0.	5.3284E-01	4.8006E-01	4.8929E-01
32	3.3381E+02	8.7480E+01	8.6376E+04	0.	1.1641E-01	1.0769E-01	1.5382E-01
33	7.3751E+02	4.7592E+02	4.6991E+05	0.	2.7108E-01	2.6189E-01	3.7338E-01
4	1.9892E+03	1.6887E+03	1.6674E+06	0.	9.9458E-22	9.8208E-22	1.2456E-01
5	3.3168E+03	2.9772E+03	2.9397E+06	0.	2.1361E+03	2.1200E+03	5.9169E+01
6	1.6490E+02	4.0661E-01	4.0147E+02	0.	1.1138E+02	9.4496E+01	8.9289E+01
7	1.3553E+02	6.2852E-04	6.2058E-01	0.	1.9586E+02	1.5973E+02	1.5921E+02
8	1.0567E+02	1.5456E-04	1.5261E-01	0.	2.5396E+02	1.9388E+02	1.9386E+02
9	8.0384E+01	7.5406E-05	7.4454E-02	0.	2.1592E+02	1.4876E+02	1.4876E+02
10	7.0616E+01	-1.3699E-04	-1.3526E-25	0.	3.5308E-23	2.2808E-23	1.1094E-04

NODE	TEMP	W/F A	DA	DDA	W/F B	DB	DDB
6	1.6490E+02	9.9758E-01	-2.0289E-07	-2.0033E-04	0.	0.	0.
7	1.3553E+02	9.9989E-01	-3.4414E-09	-3.3979E-06	0.	0.	0.
8	1.0567E+02	1.0000E+00	-2.6848E-11	-2.6509E-08	0.	0.	0.
9	8.0384E+01	1.0000E+00	-2.3203E-13	-2.2910E-10	0.	0.	0.
10	7.0616E+01	1.0000E+00	-3.0711E-14	-3.0323E-11	0.	0.	0.

Fig. 12. Nitrocellulose cylinder. Sample problem for TRUMP report. (Reaction and ignition calculation.) (continued)

MATERIAL DATA

NAME	MATL	TOT CAP	TOT HEAT	AVG TEMP	TMELT	HMELT
CORE	1	4.71239E+00	1.39058E+03	2.95090E+02	0.	0.
AIR	2	2.82743E-03	9.20331E-01	3.25500E+02	0.	0.
NITRO	3	7.85398E+00	2.91320E+03	3.70920E+02	0.	0.

NODE DATA. CHECK TOTAL CONDUCTANCES (ZIP) AND TIME CONSTANTS (SLIM).
LARGE DIFFERENCES BETWEEN NODES MAY BE DUE TO POOR ZONING, AND MAY PRODUCE POOR RESULTS.

NODE	MATL	NTYPE	RADIUS	VOLUME	MASS	CAPACITY	CONDUCTIVITY	ZIP	SLIM
1	1	0	5.0000E-01	3.1416E+00	7.8540E+00	4.7124E+00	1.0000E-02	1.2566E-01	3.7500E+01
2	1	2	1.0000E+00	1.0000E-24	2.5000E-24	1.5000E-24	1.0000E-02	1.3939E-01	1.0761E-23
31	2	4	1.4000E+00	7.0372E+00	7.0372E-03	2.1112E-03	5.0000E-04	2.0420E-02	1.0333E-01
32	2	3	1.8500E+00	1.1624E+00	1.1624E-03	3.4872E-04	5.0000E-04	7.2257E-02	4.8261E-03
33	2	3	1.9500E+00	1.2252E+00	1.2252E-03	3.6757E-04	5.0000E-04	1.8535E-01	1.9831E-03
4	3	2	2.0000E+00	1.0000E-24	2.0000E-24	5.0000E-25	5.0000E-04	2.5720E-01	1.944 E-24
5	3	4	2.0500E+00	1.2881E+00	2.5761E+00	6.4403E-01	5.0000E-04	1.9164E-01	3.3637E+00
6	3	4	2.1500E+00	1.3509E+00	2.7018E+00	6.7544E-01	5.0000E-04	1.1205E-01	6.028[illegible]E+00
7	3	4	2.3000E+00	2.8903E+00	5.7805E+00	1.4451E+00	5.0000E-04	7.6236E-02	1.8956E+01
8	3	0	2.5500E+00	4.8066E+00	9.6133E+00	2.4033E+00	5.0000E-04	5.8434E-02	4.1129E+01
9	3	4	2.8500E+00	5.3721E+00	1.0744E+01	2.6861E+00	5.0000E-04	9.1106E-02	2.9483E+01
10	3	2	3.0000E+00	1.0000E-24	2.0000E-24	5.0000E-25	5.0000E-04	7.6285E-02	6.5544E-24

INTERNAL CONNECTION DATA. CHECK CONDUCTANCES (TRAN).
LARGE DIFFERENCES BETWEEN CONNECTIONS MAY BE DUE TO POOR ZONING, AND MAY PRODUCE POOR RESULTS.

NOD1	NOD2	AREA	FINT	RINT	TRAN	HEAT FLOW	AVG RATE
1	2	6.2832E+00	1.0000E+12	0.	1.2566E-01	-1.0856E+03	-5.7845E-01
2	31	6.2832E+00	1.0000E+12	0.	7.8540E-03	-7.0169E+02	-3.7389E-01
31	32	1.1310E+01	1.0000E+12	0.	1.2566E-02	-7.0120E+02	-3.7363E-01
32	33	1.1938E+01	1.0000E+12	0.	5.9690E-02	-7.0105E+02	-3.7355E-01
33	4	1.2566E+01	1.0000E+12	0.	1.2566E-01	-7.0067E+02	-3.7335E-01
2	4	6.2832E+00	1.0000E-24	8.0900E-01	5.8688E-03	-3.8388E+02	-2.0455E-01
4	5	1.2566E+01	1.0000E+12	0.	1.2566E-01	-1.0844E+03	-5.7784E-01
5	6	1.3195E+01	1.0000E+12	0.	6.5973E-02	-1.0253E+03	-5.4631E-01
6	7	1.3823E+01	1.0000E+12	0.	4.6077E-02	-9.3597E+02	-4.9873E-01
7	8	1.5080E+01	1.0000E+12	0.	3.0159E-02	-7.7677E+02	-4.1390E-01
8	9	1.6965E+01	1.0000E+12	0.	2.8274E-02	-5.8290E+02	-3.1060E-01
9	10	1.8850E+01	1.0000E+12	0.	6.2832E-02	-4.3414E+02	-2.3133E-01

Fig. 12. Nitrocellulose cylinder. Sample problem for TRUMP report. (Reaction and ignition calculation.) (continued)

```
BOUNDARY NODE DATA

 NODB     TEMPB          HEAT FLOW     AVG RATE
  101   2.5000E+01  -4.3414E+02  -2.3133E-01

SYSTEM TOTAL          -4.3414E+02  -2.3133E-01
==========================================================================================
EXTERNAL CONNECTION DATA

  NODS NODSB    AREAS        HSURE        POWER        RSURE        TRANS       HEAT FLOW    AVG RATE
    10   101  1.8850E+01  2.0000E-04  3.3300E-01  0.           1.3453E-02 -4.3414E+02 -2.3133E-01
==========================================================================================

KCYC =    182, ITERATIONS...  TOTAL =  2731, AVERAGE =     15, MAXIMUM =     33.  FOR = 0.664 .
ENDED PROB    4     KCYC =   182     KWIT =  2     SUMTIM =  1.87671E+03                20.177  N04/10
KWIT...1=TIMAX, 2=TMAX, 3=TMIN, 4=S.S., 5=DATA, 6=CARD OR KEY, 7=MCYC, 8=MSEC, 9=BLK 2 OR 4,
     10=CONVERGENCE FAILURE, 11=PROBLEM SIZE LIMIT, 12=TABLE LENGTH OR ORDER.


MACHINE TIME (N04/10) =     36  SECONDS.  CPU,I/O,SYS PERCENTS =   62:37:0
```

Fig. 12. Nitrocellulose cylinder. Sample problem for TRUMP report. (Reaction and ignition calculation.) (continued)

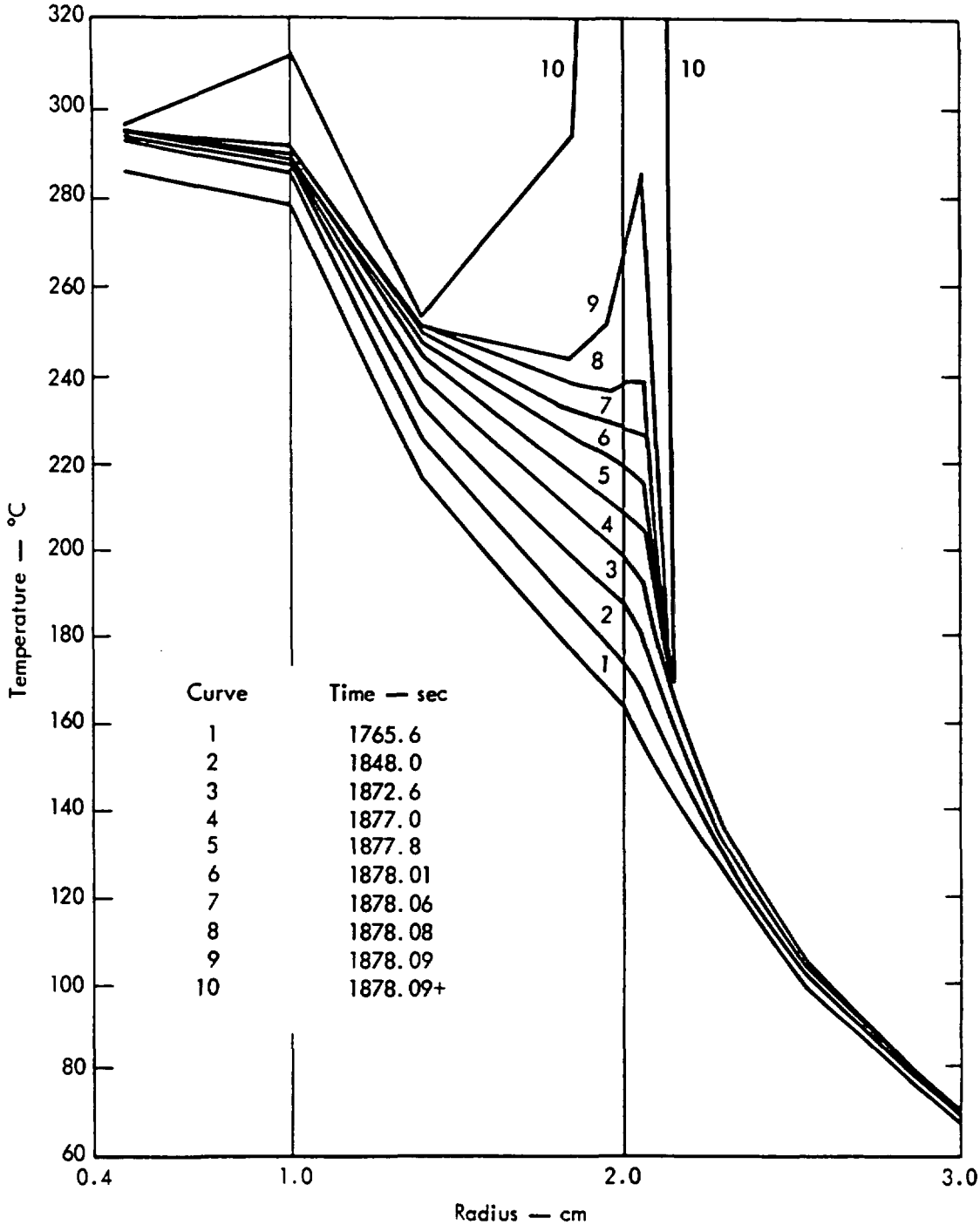

Fig. 13. Plot of temperature vs radius for nitrocellulose cylinder. (Reaction calculation, times near ignition.)

III. Methods of Solution

A. GENERAL SEQUENCE OF CALCULATIONS

1. Data Input and Initialization

A number of quantities are initialized at the time input data are read in. Many other quantities, including those whose value may remain constant throughout the calculations, are calculated after all input data have been read in. Other quantities, which may change or may represent cumulative summations of incremental values for each time step, are assigned initial values. With few exceptions, no calculations are unnecessarily repeated every time step. This is accomplished by use of many branch points, at which various counters and indicators are tested. The exceptions are made to preserve certain physical parameters in their original form for clarity in output data tables.

For the first time step, total time t (SUMTIM) is set equal to the input value t_0 (TAU), the time step Δt (DELT) is 10^{-12}, and the interpolation factor f (FOR) is 1.0; and estimated rates of temperature change $\dot{T}_n$ (DDT), and concentration changes $\dot{a}_n$ (DDA) and $\dot{b}_n$ (DDB), are zero. If the input value of KSPEC is positive, all nodes in the system are classified as special nodes. Several special procedures are used for the initial time step (for which the time-step counter KCYC is zero) and the next time step, to start the calculation off in a smooth and stable manner. These will be described below.

2. Calculations Made at Each Time Step

At the beginning of each time step, the measure of maximum temperature change, DTMAX, is reset to zero. The incremental changes in temperature, ΔT_n (DT), and concentration, Δa_n (DA) and Δb_n (DB), are reset to zero. Initial heat content W_n (W) and estimated average heat capacity $\bar{C}_n$ (CAP), if variable, and thermal conductivity $\bar{k}_n$ (CON), if variable, are calculated for each node, in subroutine THERM.

In subroutine GEN, the estimated average internal heat generation rate $\bar{g}_n$ (G), if variable, and the resulting regular node temperature change, $\Delta T_{n,G,Reg}$, are calculated.

In subroutine CHEM, the estimated average heats of reaction $Q_{a,n}$ (QA) and $Q_{b,n}$ (QB), log collision frequencies $Z_{a,n}$ (ZA) and $Z_{b,n}$ (ZB), and activation energies $E_{a,n}$ (EA) and $E_{b,n}$ (EB) are recalculated if variable, for nodes containing reactants a and b; and the resulting concentration changes $\Delta a_{n,R}$ (DA) and $\Delta b_{n,R}$ (DB) and regular node temperature changes $\Delta T_{n,R,Reg}$ (DT) are calculated.

In subroutine FINK, the estimated average conductances of internal thermal connections $U_{n,k}$ (TRAN) and their contributions to the total conductance of each node Z_n and Z_k (ZIP) are calculated, and the resulting regular node temperature changes $\Delta T_{n,I,Reg}$ (DT) are calculated.

In subroutine FLOW, the estimated average mass-flow rates of mass-flow connections $\overline{F}_{n,k}$ (FLOWN) and their contributions to the total conductance of each node Z_n (ZIP) are calculated, and the resulting regular node temperature changes $\Delta T_{n,F,Reg}$ and reactant concentrations $\Delta a_{n,F}$ and $\Delta b_{n,F}$ are calculated.

In subroutine SURE, the estimated average conductances of external thermal connections $U_{n,b}$ (TRANS) and their contributions to the total conductance of each node Z_n (ZIP) are calculated, and the resulting regular node temperature changes $\Delta T_{n,E,Reg}$ are calculated.

In subroutine THERM1, the amount of phase change $\Delta a_{n,P}$ (DA) and resulting latent heat effect on regular node temperature $\Delta T_{n,P}$ (DT) are estimated.

In subroutine SPECK, the temperature changes in special nodes and in regular nodes connected to special nodes are recalculated, with an iteration scheme for interconnected special nodes, based on interpolated average temperatures during the time step for internal thermal and mass-flow connections and external thermal connections.

In subroutine THERM2, the final calculation of the amount of phase change $\Delta a_{n,P}$ (DA) and latent heat effect $\Delta T_{n,P}$ (DT), consistent with the results of the calculations in subroutine SPECK, are calculated.

In subroutine TALLY, the stability limits τ_n (SLIM) of all nodes are calculated, and the resulting maximum stable time step for regular nodes, $\Delta t_{s,Reg}$ (DELTMX) are calculated. Regular nodes are reclassified as special nodes as required, if allowed by an input option, to keep $\Delta t_{s,Reg}$ larger than the present time step Δt (DELT). The minimum allowable time step is calculated. The maximum temperature change for the time step is found, and the new time step, total time, interpolation factor, and time derivative correction factor are calculated. If the time step need not be repeated, the new temperatures and concentrations are found by adding the calculated changes to the values for the beginning of the time step. Estimated rates of temperature and concentration change are then calculated. The criteria for producing data printouts and for ending the problem are tested.

The sequence of calculations is repeated for each additional time step, until the problem is ended. Parts of the program not needed for a specific problem are skipped; and wherever possible, calculations not involving time- or temperature-dependent quantities are skipped after the first time step.

B. CALCULATION OF PROPERTIES, HEAT AND MASS BALANCE DATA

1. Evaluation of Tabulated Properties

A number of input properties may be tabulated functions of time or temperature. These include material specific heats and enthalpies (CAPT and WT vs TVARC) and thermal conductivities (CONT vs TVARK), chemical reactant heats of reaction (QT vs TVARQ), log collision frequencies (ZT vs TVARZ), activation energies (ET vs TVARE), heat source strengths (GT vs TVARG), surface conductances (HSURT vs TVARH), and mass-flow rates (FLOWT vs TVARFL). External temperatures may be tabulated functions of time (TEMPB vs TIMEB).

At each time t (SUMTIM) in the calculation, average values of all tabulated properties must be estimated for use during the next time step Δt (DELT). These values are calculated from the tables by the point-slope method of interpolation, with extrapolation beyond the range of the table. Time-dependent parameters are calculated for an average time:

$$\bar{t} = t + f\,\Delta t\,, \qquad \text{(III.B.1.1)}$$

except for external temperatures, evaluated at $t + \Delta t$. Temperature-dependent parameters are calculated for estimated average node temperatures $\bar{T}_n$ and average external temperatures $\bar{T}_b$:

$$\bar{T}_n = T_n + f\,\Delta t\,\dot{T}_n \qquad \text{(III.B.1.2)}$$

$$\bar{T}_b = T_b + f\,(T_b' - T_b)\,. \qquad \text{(III.B.1.3)}$$

T_n is a node temperature at time t, $\dot{T}_n$ is an estimated average rate of temperature change (see Section III.I.3), T_b and T_b' are external temperatures evaluated at times t and $t + \Delta t$, and f is an interpolation factor between 0.57 and 1.0 (see Section III.I.4).

Accuracy in evaluation of tabulated properties depends on how well the curve obtained by joining the tabulated points with straight line segments fits the actual function. Errors due to use of finite time steps are controlled by limiting the percentage change in any tabulated property over any time step to 2%. If necessary, the time step is halved and repeated, subject to a lower limit, SMALL (see Section II.C.16).

2. Node Heat and Mass Balance

a. Thermal Properties

Several quantities required for the temperature and concentration change calculations, and additional quantities related to the heat and mass balances, are calculated

for each node in the system. The node volume V_n (VOL) and node mass M_n (HEFT) remain constant throughout the calculation:

$$V_n = \alpha\, d_l d_w d_r^3 S_d^3 \qquad \text{(III.B.2.1)}$$

$$M_n = \rho_m V_n , \qquad \text{(III.B.2.2)}$$

where α and β depend on system symmetry (see Section II. C. 4); d_l, d_w, and d_r are input values for each node (see Section II. C. 7), and ρ_m is the constant input value of the density of the node material.

The average total heat capacity of each node $\overline{C}_n$ (CAP) is calculated as follows:

$$\overline{C}_n = \overline{c}_n M_n , \qquad \text{(III.B.2.3)}$$

where $\overline{c}_n$ is the specific heat of the node material at the average time and estimated average temperature of node n for the time step [Eq. (III. B.1.2)].

The initial heat content W_n (W) of each node is calculated at the beginning of each time step as follows, if the heat capacity of the node material is tabulated vs temperature:

$$\overline{W}_n = W_0 + (\overline{C}_n + C_0)(T_n - T_0)/2 \qquad \text{(III.B.2.4a)}$$

$$W_n = \left[\overline{W}_n + \Delta H_m (1 - a_n)\right] M_n \qquad \text{(III.B.2.4b)}$$

where T_n is the node temperature at the beginning of the time step, C_0 is the tabulated specific heat of the node material at the nearest table temperature T_0 below T_n, (or the first temperature in the table, if higher than T_n), $\overline{W}_n$ is the sensible heat content per unit mass of the node material at temperature T_n (equal to the area under the curve of tabulated specific heat values between zero temperature and T_n), ΔH_m is the latent heat of phase change per unit mass of the node material, and a_n is the weight fraction of the low-temperature phase in the node at the beginning of the time step.

If the heat capacity of the node material is not tabulated vs temperature, W_n is calculated as follows:

$$W_n = \overline{C}_n T_n + \Delta H_m (1 - a_n) M_n \qquad \text{(III.B.2.5)}$$

At the end of each time step, the final total heat contents of each node, W_n' (W), are calculated as follows:

$$W_n' = W_n + \overline{C}_n \Delta T_n - \Delta H_m \Delta a_n M_n \qquad \text{(III.B.2.6)}$$

The initial values W_n are used only in the mass-flow heat-transport calculations (see Section III.F.2), and the final values W_n' are calculated only to provide additional heat balance information.

b. Heat Balance Data

The net change in heat content of each node since the beginning of the calculation, $H_{n,net}$ (H), is calculated at the end of each time step as follows:

$$H_{n,net} = \sum_{\Delta t} (\bar{C}_n \Delta T_n - \Delta H_m \Delta a_n M_n), \qquad \text{(III.B.2.7)}$$

where the summation is over all the time steps up to the current time.

The net amount of heat transported into each node by internal and external thermal connections since the beginning of the calculation, $H_{n,flux}$ (F), is calculated at the end of each time step as follows:

$$H_{n,flux} = \sum_{\Delta t} \left(\sum_k \Delta H_{n,k} + \sum_b \Delta H_{n,b} \right), \qquad \text{(III.B.2.8)}$$

where the k summation is over all the internal thermal connections of node n, and the b summation is over all the external thermal connections of node n. The heat-flow increments $\Delta H_{n,k}$ and $\Delta H_{n,b}$ are calculated for each time step with Eqs. (III.F.1.1) and (III.G.1.1).

3. Material Heat Balance

Several quantities related to the overall heat balance of each material are calculated. These include total volume V_{mat} (VOLMS) and mass M_{mat} (HEFTMS), which remain constant throughout the calculation:

$$V_{mat} = \sum V_{n,mat} \qquad \text{(III.B.3.1)}$$

$$M_{mat} = \rho_m V_{mat}, \qquad \text{(III.B.3.2)}$$

where the summation is over all the nodes of the material.

Other data are calculated at the end of each time step. These include the total heat capacity $\bar{C}_{mat}$ (CAPMS), the total heat content W_{mat} (WMS), and average temperature (assuming constant heat capacity and no latent heat effects) $\bar{T}_{mat}$ (TMS) of each material, calculated as follows:

$$\bar{C}_{mat} = \sum_n \bar{C}_n \qquad \text{(III.B.3.3)}$$

$$W_{mat} = \sum_n W'_n \qquad \text{(III.B.3.4)}$$

$$\bar{T}_{mat} = \sum_n \bar{C}_n \; T_n / \bar{C}_{mat}. \qquad \text{(III.B.3.5)}$$

4. System Heat Balance

Several quantities related to the overall heat balance of the system are calculated. These include total volume V_{sys} (VOLS) and mass M_{sys} (HEFTS), which remain constant throughout the calculations:

$$V_{sys} = \sum_n V_n \qquad \text{(III.B.4.1)}$$

$$M_{sys} = \sum_n \rho_n V_n, \qquad \text{(III.B.4.2)}$$

where the summations are over all the nodes in the system.

At the end of each time step, the system heat capacity $\bar{C}_{sys}$ (CAPS), heat content W_{sys} (HEAT), and average temperature (assuming constant heat capacity and no latent heat effects) $\bar{T}_{sys}$ (TEMPAD) are calculated as follows:

$$\bar{C}_{sys} = \sum_n \bar{C}_n \qquad \text{(III.B.4.3)}$$

$$W_{sys} = \sum_n W'_n \qquad \text{(III.B.4.4)}$$

$$\bar{T}_{sys} = \sum_n \bar{C}_n \; T_n / \bar{C}_{sys}. \qquad \text{(III.B.4.5)}$$

Also, the net heat flow into the system $H_{sys,flux}$ (FLUX), the resulting net temperature change (assuming constant heat capacity and no latent heat effects) $\Delta T_{sys,flux}$ (TEMPER), the average rate of heat flow into the system $\dot{\bar{H}}_{sys,flux}$ (FX), and the resulting average rate of temperature change $\dot{T}_{sys,flux}$ (TX), are calculated as follows:

$$H_{sys,flux} = \sum_n H_{n,flux} \qquad \text{(III.B.4.6)}$$

$$\Delta T_{sys,flux} = H_{sys,flux} / \bar{C}_{sys} \qquad \text{(III.B.4.7)}$$

$$\dot{\bar{H}}_{sys,flux} = H_{sys,flux} / (t - t_0) \qquad \text{(III.B.4.8)}$$

$$\dot{\bar{T}}_{sys,flux} = \Delta T_{sys,flux}/(t-t_0) \ , \qquad \text{(III.B.4.9)}$$

where t is the total time SUMTIM and t_0 is the initial time TAU.

The effects of internal heat generation on the system are summarized by the net rate of heat generation in the system G_{sys} (GS), the total amount of heat generated $H_{sys,gen}$ (GENS), and the resulting net temperature change (assuming constant heat capacity and no latent heat effects) $\Delta T_{sys,gen}$ (TEMPLE), calculated at each time step as follows:

$$G_{sys} = \sum_n G_n \qquad \text{(III.B.4.10)}$$

$$H_{sys,gen} = \sum_s G_{sys} \Delta t_s \qquad \text{(III.B.4.11)}$$

$$\Delta T_{sys,gen} = H_{sys,gen}/\bar{C}_{sys} \ , \qquad \text{(III.B.4.12)}$$

where the s summation is over all the time steps up to the current time.

C. PHASE CHANGE

In each node n of a material m with a specified phase-change or transition temperature T_m between T_{min} (TMIN) and T_{max} (TMAX), and nonzero latent heat effect ΔH_m, the initial concentration of low-temperature phase a_n is calculated as follows:

$$a_n = \begin{cases} 1.0, & T_n \leq T_m \\ 0.0, & T > T_m \end{cases} \qquad \text{(III.C.1.1)}$$

(the value specified for a_n in input BLOCK 9 will be used instead, if columns 67-71 of the Problem Name card are "A=BL9").

The calculation of changes in phase concentrations and the effects on temperature changes are done in two stages in each time step. In the first stage, an estimate is made of the amount of phase change that will occur in node n, as follows:

$$\Delta a_{n,P,est} = \dot{a}_n \Delta t + (T_m - T_n - \dot{T}_n \Delta t)\, \bar{c}_n/\Delta H_m , \qquad \text{(III.C.1.2)}$$

subject to the limits:

$$- a_n \leq \Delta a_{n,P,est} \leq 1 - a_n . \qquad \text{(III.C.1.3)}$$

The estimated temperature change in node n due to phase change is calculated as follows:

$$\Delta T_{n,P,est} = \Delta a_{n,P,est}\, \Delta H_m/\bar{c}_n . \qquad \text{(III.C.1.4)}$$

This calculation is done before use of the iterative procedure described in Section III.H, below.

The second stage of the phase-change calculation is done after the iterative procedure has been completed. A corrected amount of phase change is calculated as follows:

$$\Delta a_{n,P} = \Delta a_{n,P,est} + (T_m - T_n - \Delta T_n')\, c_n/\Delta H_m , \qquad \text{(III.C.1.5)}$$

subject to the limits of Eq. (III.C.1.3). $\Delta T_n'$ is the total temperature change for node n, exclusive of the final correction for phase change. The final temperature change in node n due to phase change is calculated as follows:

$$\Delta T_{n,P,Reg} = \Delta a_{n,P}\, \Delta H_m/\bar{c}_n . \qquad \text{(III.C.1.6)}$$

The result of this method of calculation is that the temperature of a node remains exactly at T_m while a_n is between 0.0 and 1.0, and the total amount of latent heat is accounted for exactly. The latent heat effect is ignored in zero-volume nodes. Also, the final correction does not affect the heat balance for the node, but converts sensible heat to latent heat, or vice-versa, to make the phase concentration consistent with the heat content and temperature of the node.

D. CHEMICAL REACTION

The changes in concentration of components a and b in node n during a time step Δt, due to chemical reaction, are calculated as follows:

$$\Delta a_{n,R} = -a_n^{p_{a,n}} \Delta t \exp[\bar{Z}_{a,n} - \bar{E}_{a,n}/R(\bar{T}_n + T_0)] \qquad \text{(III.D.1.1)}$$

$$\Delta b_{n,R} = -b_n^{p_{b,n}} \Delta t \exp[\bar{Z}_{b,n} - \bar{E}_{b,n}/R(\bar{T}_n + T_0)] \ , \qquad \text{(III.D.1.2)}$$

where a_n and b_n are concentrations (weight fractions) at the beginning of the time step, $p_{a,n}$ and $p_{b,n}$ are reaction orders, R is the gas constant, 1.987 cal/g mole - K, and the $\bar{Z}$ and $\bar{E}$ values are estimated averages for the time step (see Section III.B.1). T_0 is the constant added to convert temperatures to absolute temperatures, and $\bar{T}_n$ is an estimated average node temperature for the time step, calculated as follows:

$$\bar{T}_n = T_n + f \Delta t \dot{T}_n \ , \qquad \text{(III.D.1.3)}$$

where f is the time-step interpolation factor (see Section III.I.4) and $\dot{T}_n$ is an estimated rate of temperature change of node n (see Section III.I.3).

The temperature change in node n due to chemical reaction is initially calculated as follows:

$$\Delta T_{n,R,Reg} = -(\bar{Q}_{a,n} \Delta a_{n,R} + \bar{Q}_{b,n} \Delta b_{n,R})/\bar{C}_n, \qquad \text{(III.D.1.4)}$$

where the $\bar{Q}$ values and $\bar{C}_n$ are estimated averages for the time step (see Section III.B).

If node n is a special node, part of the heat produced by chemical reaction is, in effect, redistributed among other nodes in the system, and the resulting net temperature change in node n is given by

$$\Delta T_{n,R} = \Delta T_{n,R,Reg}/(1 + f\Delta t Z_n/\bar{C}_n), \qquad \text{(III.D.1.5)}$$

where Z is the total conductance of node n [Eq.(III.H.1.4)].

E. INTERNAL HEAT GENERATION

The temperature change of node n due to internal heat generation during a time step Δt is initially calculated as follows:

$$\Delta T_{n,G,Reg} = \bar{G}_n \, \Delta t / \bar{C}_n , \qquad \text{(III.E.1.1)}$$

where

$$\bar{G}_n = \bar{g}_n \, V_n \qquad \text{(III.E.1.2)}$$

and $\bar{g}_n$ is the estimated average volumetric heat-generation rate in the node during the time step (see Sections II. C.8 and III. B). If node n is a special node, part of the heat generated is redistributed, as a result of the iteration procedure, among other nodes in the system, and the resulting net temperature change is as follows:

$$\Delta T_{n,G} = \Delta T_{n,G,Reg} / [1 + f \Delta t Z_n / \bar{C}_n] , \qquad \text{(III.E.1.3)}$$

where f is the time-step interpolation factor (see Section III. I. 4) and Z_n is the total conductance of node n [Eq.(III.H.1.4)].

F. INTERNAL HEAT AND REACTANT TRANSPORT

1. Internal Thermal Connections

The heat flow into node n from node k during a time step Δt, due to a thermal connection, $\Delta H_{n,k,I}$, is calculated as follows:

$$\Delta H_{n,k,I} = U_{n,k} (\overline{T}_k - \overline{T}_n) \Delta t = -\Delta H_{k,n,I} \qquad \text{(III.F.1.1)}$$

where $U_{n,k}$ is an average conductance, calculated as described in Sections II. C.10a and III. B, and $\overline{T}_k$ and $\overline{T}_n$ are average temperatures for the time step. All nodes are first treated as regular nodes. An iterative scheme is later used to find the solution when special nodes are involved, to correspond to the following average temperature values:

Nodes n and k both regular:

$$\overline{T}_n = T_n \qquad \text{(III.F.1.2)}$$

$$\overline{T}_k = T_k \qquad \text{(III.F.1.3)}$$

Node n regular, node k special:

$$\overline{T}_n = T_n + f \Delta T_{n,Reg} \qquad \text{(III.F.1.4)}$$

$$\overline{T}_k = T_k + f \Delta T_k \qquad \text{(III.F.1.5)}$$

Nodes n and k both special:

$$\overline{T}_n = T_n + f \Delta T_n \qquad \text{(III.F.1.6)}$$

$$\overline{T}_k = T_k + f \Delta T_k , \qquad \text{(III.F.1.7)}$$

where T_n and T_k are temperatures at the beginning of the time step; f is an interpolation factor between 0.57 and 1.0 (see Section III. I.4); $\Delta T_{n,Reg}$ is the net temperature change in regular node n due to all causes, excluding the final corrections made for connections to special nodes (see Section III. H.4) and for phase change (see Section III. C); and ΔT_n and ΔT_k are net temperature changes in special nodes n and k due to all causes, excluding the final correction made for phase change (see Section III. C).

The total heat flow across each internal thermal connection $H_{n,k}$ (FI), and the average rate $\dot{H}_{n,k}$ (FX), are calculated at each time step Δt_s, as follows:

$$H_{n,k,I} = \sum_s U_{n,k} (\overline{T}_k - \overline{T}_n) \Delta t_s \qquad \text{(III.F.1.8)}$$

$$\dot{H}_{n,k,I} = H_{n,k,I} \Big/ \sum_s \Delta t_s \qquad \text{(III.F.1.9)}$$

The rate for the time step, $U_{n,k}(\bar{T}_k - \bar{T}_n)$ is also calculated.

The net temperature change in node n due to internal thermal connections, $\Delta T_{n,I}$, is initially calculated as follows:

$$\Delta T_{n,I,Reg} = \frac{\Delta t}{\bar{C}_n} \sum_k U_{n,k}(T_k - T_n) \qquad \text{(III.F.1.10)}$$

For systems which include special nodes, the final results, after applying the iteration scheme are as follows:

Regular nodes:

$$\Delta T_{n,I} = \frac{\Delta t}{\bar{C}_n} \left[\sum_k^{all} U_{n,k}(T_k - T_n) + f \sum_k^{Spec} U_{n,k}(\Delta T_k - \Delta T_{n,Reg}) \right] \qquad \text{(III.F.1.11)}$$

Special nodes:

$$\Delta T_{n,I} = \frac{\Delta t}{\bar{C}_n} \left[\sum_k^{all} U_{n,k}(T_k - T_n) + f \sum_k^{Reg} U_{n,k}(\Delta T_{k,Reg} - \Delta T_n) + f \sum_k^{Spec} U_{n,k}(\Delta T_k - \Delta T_n) \right] \qquad \text{(III.F.1.12)}$$

2. Mass Flow Connections

a. Heat Transport

The heat flow into node n from node k during a time step Δt, due to mass flow, $\Delta H_{n,k,F}$, is calculated as follows:

$$\Delta H_{n,k,F} = \bar{F}_{n,k}(\bar{w}_{n,k} - \bar{w}_n)\,\Delta t, \qquad \text{(III.F.2.1)}$$

where $\bar{F}_{n,k}$ is an average mass-flow rate, calculated as described in Section III. B; $\bar{w}_{n,k}$ is an average specific enthalpy at the interface between nodes n and k; and w_n is the average specific enthalpy of node n. The $\bar{w}_n$ term is included to insure an effective mass balance for each node, so that each specified flow stream is implicitly accompanied by an opposite flow stream at the average node temperature.

Just as for internal thermal connections, all nodes are first treated as regular nodes, and an iterative scheme is later used to solve the implicit equations that result when special nodes are present. The average specific enthalpies used for the various cases are as follows:

Nodes n and k both regular:

$$\bar{w}_n = w_n - f\,\Delta t\,\dot{a}_n\,\Delta H_n = w_{n,\,Reg} \tag{III.F.2.2}$$

$$\bar{w}_k = w_k - f\,\Delta t\,\dot{a}_k\,\Delta H_k = w_{k,\,Reg} \tag{III.F.2.3}$$

$$\bar{w}_{n,\,k} = (d_k\,w_{n,\,Reg} + d_n\,w_{k,\,Reg})/(d_n + d_k) = w_{n,\,k,\,Reg}, \tag{III.F.2.4}$$

where w_n and w_k are specific enthalpies, including latent heat, at the beginning of the time step; f is the interpolation factor (see Section III. I.4); $\dot{a}_n$ and $\dot{a}_k$ are estimated rates of phase change; ΔH_n and ΔH_k are latent heats of phase change; and d_n and d_k are relative distances from the nodal points of nodes n and k to their interface. The estimated phase change is included here to keep the iteration scheme simple but could cause instability where phase change and mass flow interact strongly. Other cases are as follows:

Node n regular, node k special:

$$\bar{w}_n = w_{n,\,Reg} + f\,\bar{C}_n\,\Delta T_{n,\,Reg} \tag{III.F.2.5}$$

$$\bar{w}_{n,\,k} = w_{n,\,k,\,Reg} + \begin{cases} f\,\bar{C}_n\,\Delta T_{n,\,Reg}, & \text{n upstream} \\ f\,\bar{C}_k\,\Delta T_k, & \text{k upstream} \end{cases}, \tag{III.F.2.6}$$

where $\bar{C}_n$ and $\bar{C}_k$ are estimated average specific heats, and ΔT_n, ΔT_k, $\Delta T_{n,\,Reg}$ and $\Delta T_{k,\,Reg}$ are defined in Section III. F.1, above.

Node n special, node k regular:

$$\bar{w}_n = w_{n,\,Reg} + f\,\bar{C}_n\,\Delta T_n \tag{III.F.2.7}$$

$$\bar{w}_{n,\,k} = w_{n,\,k,\,Reg} + \begin{cases} f\,\bar{C}_n\,\Delta T_n, & \text{n upstream} \\ f\,\bar{C}_k\,\Delta T_{k,\,Reg}, & \text{k upstream} \end{cases} \tag{III.F.2.8}$$

Node n and k both special:

$$\bar{w}_n = w_{n,\,Reg} + f\,\bar{C}_n\,\Delta T_n \tag{III.F.2.9}$$

$$\bar{w}_{n,\,k} = w_{n,\,k,\,Reg} + \begin{cases} f\,\bar{C}_n\,\Delta T_n, & \text{n upstream} \\ f\,\bar{C}_k\,\Delta T_k, & \text{k upstream} \end{cases} \tag{III.F.2.10}$$

It was found that the estimated change in interface specific enthalpy must be weighted in favor of the upstream node in order to make the application of the iterative scheme to the mass-flow equations stable. For simplicity, the change in the temperature of the upstream node was substituted for an interpolated change in interface temperature.

Net amounts of mass flow across each mass flow connection, $M_{n,k}$ (FLAPS), and average values $\dot{M}_{n,k}$ (FX), are calculated as follows:

$$M_{n,k} = \sum_s \bar{F}_{n,k} \Delta t_s \qquad \text{(III.F.2.11)}$$

$$\dot{M}_{n,k} = \bar{M}_{n,k} \Big/ \sum_s \Delta t_s \qquad \text{(III.F.2.12)}$$

For each node, net rates of mass flow in, $\bar{F}_{n,in}$ (FLINT), and out, $\bar{F}_{n,out}$ (FLOUT); net amounts of mass flow in, $M_{n,in}$ (FLIPS), and out, $M_{n,out}$ (FLOPS); and average flow rate in, $\dot{M}_{n,in}$ (FX1), and out, $\dot{M}_{n,out}$ (FX2), are calculated as follows:

$$\bar{F}_{n,in} = \sum_k^{in} \bar{F}_{n,k} \qquad \text{(III.F.2.13)}$$

$$\bar{F}_{n,out} = \sum_k^{out} \bar{F}_{n,k} \qquad \text{(III.F.2.14)}$$

$$M_{n,in} = \sum_s \bar{F}_{n,in} \Delta t_s \qquad \text{(III.F.2.15)}$$

$$M_{n,out} = \sum_s \bar{F}_{n,out} \Delta t_s \qquad \text{(III.F.2.16)}$$

$$\dot{M}_{n,in} = M_{n,in} \Big/ \sum_s \Delta t_s \qquad \text{(III.F.2.17)}$$

$$\dot{M}_{n,out} = M_{n,out} \Big/ \sum_s \Delta t_s \,. \qquad \text{(III.F.2.18)}$$

The net temperature change in each node due to mass-flow connections, $\Delta T_{n,f}$, is initially calculated as follows:

$$\Delta T_{n,F,Reg} = \frac{\Delta t}{\bar{C}_n} \sum_k \bar{F}_{n,k} (w_{n,k,Reg} - w_{n,Reg}). \qquad \text{(III.F.2.19)}$$

For systems that include special nodes, the final results, after applying the iteration scheme, are as follows:

$$\Delta T_{n,F} = \frac{\Delta t}{\bar{C}_n} \sum_k \bar{F}_{n,k} (\bar{w}_{n,k} - \bar{w}_n), \qquad \text{(III.F.2.20)}$$

where $\bar{w}_{n,k}$ and $\bar{w}_n$ are given for the various cases of flow direction and node classification by Eqs. (III.F.2.2) to (III.F.2.10).

b. Reactant Transport

The changes in reactant concentrations a_n and b_n in node n due to mass flow are calculated as follows:

$$\Delta a_{n,F} = \sum_k \bar{F}_{n,k} (\bar{a}_{n,k} - \bar{a}_n) \Delta t/\rho_n V_n \qquad \text{(III.F.2.21)}$$

$$\Delta b_{n,F} = \sum_k \bar{F}_{n,k} (\bar{b}_{n,k} - \bar{b}_n) \Delta t/\rho_n V_n , \qquad \text{(III.F.2.22)}$$

where the summation is over all mass-flow connections of node n; $\bar{F}_{n,k}$ is an average mass-flow rate for the time step, evaluated as described in Section III.B; $\bar{a}_n$ and $\bar{b}_n$ are estimated average concentrations in node n; and $\bar{a}_{k,n}$ and $\bar{b}_{k,n}$ are estimated average interface concentrations, calculated as follows:

$$\bar{a}_n = a_n + f \Delta t \dot{a}_n \qquad \text{(III.F.2.23)}$$

$$\bar{b}_n = b_n + f \Delta t \dot{b}_n \qquad \text{(III.F.2.24)}$$

$$\bar{a}_{k,n} = (d_n \bar{a}_k + d_k \bar{a}_n)/(d_n + d_k) \qquad \text{(III.F.2.25)}$$

$$\bar{b}_{k,n} = (d_n \bar{b}_k + d_k \bar{b}_n)/(d_n + d_k), \qquad \text{(III.F.2.26)}$$

where d_n and d_k are relative distances from the nodal points of nodes n and k to the interface, and $\dot{a}_n$ and $\dot{b}_n$ are estimated rates of concentration change (see Section III.I.4).

These equations are not unconditionally stable. The maximum stable time step is approximately:

$$\Delta t_{max} = \frac{1}{2} \rho_n V_n \Big/ \sum_k \left| F_{n,k} \right| \text{(n downstream)}. \qquad \text{(III.F.2.27)}$$

An iterative procedure like that used for temperature changes could be used for concentration changes due to mass flow, to obtain unconditional stability, but has not been included in the program.

G. SURFACE HEAT TRANSPORT

The heat flow into node n from boundary node b during time step Δt, $\Delta H_{n,b,E}$, is calculated as follows:

$$\Delta H_{n,b,E} = U_{n,b} (\overline{T}_b - \overline{T}_n) \Delta t, \qquad \text{(III.G.1.1)}$$

where $U_{n,b}$ is an average conductance calculated as described in Sections II. C.10a and III. B, and $\overline{T}_b$ and $\overline{T}_n$ are average temperatures for the time step. All nodes are first treated as regular nodes, and the iterative scheme is later applied when special nodes are included in the system. All nodes with connections to boundary nodes are classified as special nodes.

$$\overline{T}_b = T_b + f\left(T_b' - T_b\right) \qquad \text{(III.G.1.2)}$$

$$\overline{T}_n = T_n \quad \text{(initially)} \qquad \text{(III.G.1.3)}$$

$$\overline{T}_n = T_n + f\,\Delta T_n \quad \text{(finally)}, \qquad \text{(III.G.1.4)}$$

where T_b and T_b' are the boundary node (external) temperatures at the beginning and end of the time step, found as described in Section III. B; T_n is the temperature of node n at the beginning of the time step; and ΔT_n is the net temperature change in node n due to all causes, excluding any final phase change corrections.

The total heat flow across each surface connection, $H_{n,b,E}$ (FS), and average rate $\dot{H}_{n,b,E}$ (FX), are calculated each time step Δt_s as follows:

$$H_{n,b,E} = \sum_s U_{n,b} (\overline{T}_b - \overline{T}_n) \Delta t_s \qquad \text{(III.G.1.5)}$$

$$\dot{H}_{n,b,E} = H_{n,b,E} \Big/ \sum_s \Delta t_s . \qquad \text{(III.G.1.6)}$$

The rate for the time step, $U_{n,b} (\overline{T}_b - \overline{T}_n)$, is also calculated.

The total heat flow into the system from each boundary node, H_b (FB) and average rate $\dot{H}_b$ (FX), are calculated each time step as follows:

$$H_b = \sum_n H_{n,b,E} \qquad \text{(III.G.1.7)}$$

$$\dot{H}_b = H_b \Big/ \sum_s \Delta t_s . \qquad \text{(III.G.1.8)}$$

The rate for the time step, $\sum_n U_{n,b} (\overline{T}_b - \overline{T}_n)$, is also calculated.

The net temperature change in node n due to surface connections, $\Delta T_{n,E,Reg}$, is initially calculated as follows:

$$\Delta T_{n,E,Reg} = \frac{\Delta t}{\overline{\overline{C}}_n} \sum_b U_{n,b} \left[T_b + f\left(T_b' - T_b\right) - T_n \right]. \qquad \text{(III.G.1.9)}$$

The final result, after applying the iteration scheme, is as follows:

$$\Delta T_{n,E} = \frac{\Delta t}{\overline{\overline{C}}_n} \sum_b U_{n,b} \left[T_b - T_n + f\left(T_b' - T_b - \Delta T_n\right) \right]. \qquad \text{(III.G.1.10)}$$

H. ITERATIVE SOLUTION OF IMPLICIT EQUATIONS

1. Overall Temperature Change Equations

The net temperature change is first estimated for all nodes, using the equations applicable to regular nodes for the first estimate of phase change (III.C.1.4), internal heat generation (III.E.1.1), chemical reaction (III.D.1.4), internal heat transport by thermal (III.F.1.11) and mass flow (III.F.2.19) connections, and surface heat transport (III.G.1.9). This can be summarized by the equation

$$\Delta T_{n,Reg} = \frac{\Delta t}{\bar{C}_n} \left[S_n + \sum_b U_{n,b} (\bar{T}_b - T_n) + \sum_k U_{n,k} (T_k - T_n) + \sum_k \bar{F}_{n,k} (\bar{w}_{n,k,Reg} - \bar{w}_{n,Reg}) \right], \qquad \text{(III.H.1.1)}$$

where S_n represents the net heat input into node n due to phase change, internal heat generation, and chemical reaction. The complete equation for special nodes is as follows:

$$\Delta T_n = \Delta T_{n,Reg} + \frac{f\Delta t}{\bar{C}_n} \left[- \sum_b U_{n,b} \Delta T_n + \sum_k^{Spec} U_{n,k} (\Delta T_k - \Delta T_n) + \sum_k^{Reg} U_{n,k} (\Delta T_{k,Reg} - \Delta T_n) + \sum_k^{Spec,up} \bar{F}_{n,k} (\bar{c}_k \Delta T_k - \bar{c}_n \Delta T_n) + \sum_k^{Reg,up} \bar{F}_{n,k} (\bar{c}_k \Delta T_{k,Reg} - \bar{c}_n \Delta T_n) \right], \qquad \text{(III.H.1.2)}$$

where the mass-flow summations are only over nodes upstream of node n.

Equation III.H.1.2 can be solved for ΔT_n as follows:

$$\Delta T_n = \left[\Delta T_{n,Reg} + \frac{f\Delta t}{\bar{C}_n} \left(\sum_k^{Spec} U_{n,k} \Delta T_k + \sum_k^{Reg} U_{n,k} \Delta T_{k,Reg} + \sum_k^{Spec,up} \bar{F}_{n,k} \bar{c}_k \Delta T_k + \sum_k^{Reg,up} \bar{F}_{n,k} \bar{c}_k \Delta T_{k,Reg} \right) \right] \Big/ \left(1 + \frac{f\Delta t}{\bar{C}_n} z_n \right), \qquad \text{(III.H.1.3)}$$

where

$$Z_n = \sum_b U_{n,b} + \sum_k^{all} U_{n,k} + \bar{c}_n \sum_k^{up} \bar{F}_{n,k} . \qquad \text{(III.H.1.4)}$$

The only unknowns on the right-hand side of Eq. (III.H.1.3) are the ΔT_k values, i.e., the temperature changes in other special nodes thermally connected to node n, or upstream of node n in a mass-flow connection. This is the set of equations that is solved iteratively, as will be described below.

The stability limit, τ_n (SLIM), or time constant of a node, is given by the expression

$$\tau_n = \bar{C}_n \cdot Z_n \tag{III.H.1.5}$$

and represents the largest stable time step that could be used for a regular node, and physically, the approximate time required for a node to react significantly to changes in the temperatures of nodes to which it is connected thermally or by mass flow.

The overall temperature-change equation for zero-volume nodes, $\Delta T_{n\ zv}$, is of particular interest, because of their usefulness for a number of special purposes (see Section II. C.12). Because zero-volume nodes may not be used as sink nodes in mass-flow connections and, when used as source nodes, the interface conditions are equal to those in the zero-volume node, all the terms involving mass flow drop out of Eqs. (III.H.1.3) and (III.H.1.4). The limiting form of Eq. (III.H.1.3), as V_n approaches zero, is as follows:

$$\Delta T_{n,zv} = \left\{ \frac{1}{f} \left[\sum_b U_{n,b} (T_b - T_n) + \sum_k^{all} U_{n,k} (T_k - T_n) \right] + \sum_b U_{n,b} \left(T_b' - T_b \right) + \sum_k^{Spec} U_{n,k} \Delta T_k + \sum_k^{Reg} U_{n,k} \Delta T_{k,Reg} \right\} \Big/ \left(\sum_b U_{n,b} + \sum_k^{all} U_{n,k} \right). \tag{III.H.1.6}$$

It can be shown by substitution that if T_n is in thermal equilibrium with the connected nodes at the beginning of the time step,

$$T_{n,equil} = \left(\sum_b U_{n,b} T_b + \sum_k^{all} U_{n,k} T_k \right) \Big/ \left(\sum_b U_{n,b} + \sum_k U_{n,k} \right), \tag{III.H.1.7}$$

then the first two summations in Eq. (III.H.1.6) drop out; and, except for the final correction to $\Delta T_{k,Reg}$ (see Section III. H.4), $\Delta T_{n,zv}$ will be just sufficient to keep the temperature of the zero-volume node in equilibrium with the connected nodes at the end of the time step. If the zero-volume node is not initially in thermal equilibrium, and the factor f is 1.0, the temperature change will also bring the zero-volume node into equilibrium at the end of the time step.

In the first time step of the calculation, f is 1.0, to bring all zero-volume nodes into thermal equilibrium with connected nodes. In later time steps, f varies between

0.57 and 1.0 (see Section III.I.3), and the temperature of zero-volume nodes will overshoot their equilibrium values at each time step, by an amount $\Delta T_{n,excess}$ given by the equation

$$\Delta T_{n,\,excess} = \left(\frac{1-f}{f}\right)\left(T_{n,\,equi} - T_n\right) + \sum_k U_{n,k}\left(\Delta T_{k,\,Reg} - \Delta T_{k,\,Reg,Corr}\right) \Big/ \left(\sum_b U_{n,b} + \sum_k U_{n,k}\right). \qquad \text{(III.H.1.8)}$$

The factor $(1 - f)/f$ varies from 0.0 to 0.75, so that any oscillation of T_n about the equilibrium value is damped. The terms involving corrections to regular node temperature changes interact in such a way that they are subject to the same damping effect, and tend to be much smaller than the first term. In approaching steady state, regular nodes are reclassified as special nodes, and the interpolation factor f approaches 1.0, so that $\Delta T_{n,\,excess}$ tends to zero. Zero-volume nodes generally maintain equilibrium values very closely with oscillations occurring only when strongly time- or temperature-dependent properties or boundary conditions are used, or when the iteration procedure converges too slowly (see Sections III.H.2 and III.H.3).

2. Iterative Procedure

The iterative scheme used in TRUMP was adapted and generalized from one discussed by Evans et al.,[22] which is unconditionally stable, provided coefficients in the equations are not strongly temperature or time dependent. Convergence is generally rapid, but the number of iterations necessary depends on the relative number and time constants of interconnected special nodes in the system; the relative values of conductances between such nodes; and the input value of TVARY, which is used in the convergence test.

The equation used for the iterative scheme is obtained from Eq. (III.H.1.2) by making the following substitutions, where the superscripts p and p+1 indicate the number of the iteration:

$$\Delta T_n \text{ (left side)} \rightarrow \Delta T_n^{(p+1)} \qquad \text{(III.H.2.1)}$$

$$\Delta T_n \text{ (right side)} \rightarrow (1 + g)\,\Delta T_n^{(p+1)} - g\,\Delta T_n^{(p)} \qquad \text{(III.H.2.2)}$$

$$\Delta T_k \text{ (right side)} \rightarrow \Delta T_k^{(p)}\,. \qquad \text{(III.H.2.3)}$$

The acceleration factor g must be zero or positive for convergence; and a value of 0.2 was finally chosen empirically, by minimizing the required total machine time for a large group of test problems of various types. Use of zero for g caused stable

oscillation rather than convergence in certain simple problems with symmetric heat flow, while use of a larger value of g generally resulted in slower convergence. On the first time step, and whenever special nodes are present, but not interconnected, g is given the value zero. The first iteration then gives the final correct temperature for such special nodes.

Solving $\Delta T_n^{(p+1)}$, after making the substitutions of Eqs. (III.H.2.1) to (III.H.2.3) and finally substituting Z_n, results in the following equation:

$$\Delta T_n^{(p+1)} = \left\{ \Delta T_{n,Reg} + \frac{f\Delta t}{\bar{C}_n} \left[\sum_k^{Spec} U_{n,k}\, \Delta T_k^{(p)} + \sum_k^{Reg} U_{n,k}\, \Delta T_{k,Reg} + \sum_k^{Spec,up} \bar{F}_{n,k}\, \bar{c}_k\, \Delta T_k^{(p)} + \sum_k^{Reg,up} \bar{F}_{n,k}\, \bar{c}_k\, \Delta T_{k,Reg} + g\, Z_n\, \Delta T_n^{(p)} \right] \right\} \Big/ \left[1 + \frac{f\Delta t}{\bar{C}_n} (1+g)\, Z_n \right] . \qquad \text{(III.H.2.4)}$$

For the first iteration (p = 0), the following values are used:

$$\Delta T_n^{(0)} = \Delta t\, \dot{T}_n \qquad \text{(III.H.2.5)}$$

$$\Delta T_k^{(0)} = \Delta t\, \dot{T}_k \qquad \text{(III.H.2.6)}$$

The difference between successive values of ΔT_n in the iteration is given by:

$$E_n^{(p+1)} = \Delta T_n^{(p+1)} - \Delta T_n^{(p)} \qquad \text{(III.H.2.7)}$$

$$E_n^{(p+1)} = \frac{f\Delta t}{\bar{C}_n} \left\{ \sum_k^{Spec} U_{n,k} \left[\Delta T_k^{(p)} - \Delta T_k^{(p-1)} \right] + \sum_k^{Spec,up} \bar{F}_{n,k}\, \bar{c}_k \left[\Delta T_k^{(p)} - \Delta T_k^{(p-1)} \right] + g\, Z_n \left[\Delta T_n^{(p)} - \Delta T_n^{(p-1)} \right] \right\} \Big/ \left[1 + \frac{f\Delta t}{\bar{C}_n} (1+g)\, Z_n \right] \qquad \text{(III.H.2.8)}$$

$$E_n^{(p+1)} = \frac{f\Delta t}{\bar{C}_n} \left[\sum_k^{Spec} U_{n,k}\, E_k^{(p)} + \sum_k^{Spec,up} \bar{F}_{n,k}\, \bar{c}_k\, E_k^{(p)} + g\, Z_n\, E_n^{(p)} \right] \Big/ \left[1 + \frac{f\Delta t}{\bar{C}_n} (1+g)\, Z_n \right] . \qquad \text{(III.H.2.9)}$$

The values of $\Delta T_n^{(1)}$ are calculated using Eqs. (III.H.2.4) to (III.H.2.6), and the values of $E_n^{(1)}$ are then calculated as follows:

$$E_n^{(1)} = \Delta T_n^{(1)} - \Delta t\, \dot{T}_n \quad . \tag{III.H.2.10}$$

Then $E_n^{(2)}$ is calculated with Eq. (III.H.2.9), and $\Delta T_n^{(2)}$ is found from the relation

$$\Delta T_n^{(p+1)} = \Delta T_n^{(p)} + \Delta E_n^{(p+1)} \tag{III.H.2.11}$$

This scheme is continued until the convergence criteria are satisfied, or until 80 iterations have been completed. In the latter case, a diagnostic statement is written out, the results of the time step are discarded, and a new time step, half as large, is used. If the time step has already reached its minimum value SMALL, another diagnostic statement is written out and the problem is ended.

3. Convergence Criterion

The net correction to heat content and the net heat capacity of all special nodes are calculated as follows:

$$\Delta H_{net}^{(p)} = \sum_{n}^{Spec} \bar{C}_n E_n^{(p)} \tag{III.H.3.1}$$

$$C_{net} = \sum_{n}^{Spec} \bar{C}_n \quad . \tag{III.H.3.2}$$

The iteration procedure is stopped after any iteration in which the following conditions are satisfied:

$$\Delta H_{net}^{(p)} < 10^{-5}\, C_{net}\, TVARY \tag{III.H.3.3}$$

$$E_{n,max}^{(p)} < 10^{-4}\, TVARY\ , \tag{III.H.3.4}$$

i.e., the net error in heat content is less than 10^{-5} of the heat required to change the temperature of all special nodes by the amount TVARY, and no special node (excluding zero-volume nodes) has a temperature error greater than 10^{-4} TVARY. The net result in most large problems is a net accumulative error, after several hundred time steps, of no more than 0.01 TVARY in the average system temperature and 0.1 TVARY in individual temperatures, and much smaller errors if any surface temperatures are fixed. A larger value of TVARY than would be used for transient problems may be specified, to reach steady state with a minimum of machine time. However, in systems with no external connections, this may result in a gradual drift of the final equilibrium temperature and, by requiring more iterations for convergence in each time step, may result in a larger cumulative truncation error.

4. Final Corrections for Regular Nodes

After the final temperature changes have been found for all special nodes by the iterative scheme described above, final corrections must be made to the temperature

changes in all regular nodes connected to special nodes. This is necessary to obtain a correct heat balance. The complete equation for regular nodes is as follows:

$$\Delta T_{n,Reg,Corr} = \Delta T_{n,Reg} + \frac{f\Delta t}{\bar{C}_n}\left[\sum_{k}^{Spec} U_{n,k}(\Delta T_k - \Delta T_{n,Reg}) - \sum_{k}^{Spec,Down} \bar{F}_{n,k}\,\bar{c}_n\,\Delta T_{n,Reg}\right]. \qquad \text{(III.H.4.1)}$$

Corrections are also made to the flux values calculated for internal thermal connections and surface connections, to correspond to Eqs. (III.F.1.1) and (III.G.1.1).

I. PREPARATION FOR NEXT TIME STEP

1. Reclassification of Nodes

The reclassification of nodes from regular to special depends on the input values of KSPEC, TVARY, and DELTO in BLOCK 1. If KSPEC is negative, no regular nodes can be changed to special nodes during the calculation, and the only special nodes will be zero-volume nodes, surface nodes, and nodes for which the input value of KS, in BLOCK 4, is nonzero. If KSPEC is positive, all regular nodes will be changed to special nodes before the first time step, and the value of the interpolation factor FOR can also be controlled (see Section III.I.3).

If KSPEC is zero at the end of each time step for which the maximum allowed time step DELTMX was used (see Section III.I.2), and is less than DELTO, the stability limits SLIM of all regular nodes are tested. Regular nodes for which SLIM is equal to or less than 1.8 DELTMX are reclassified as special nodes. Since DELTMX is 2/3 of the smallest stability limit of any regular nodes, all regular nodes with stability limits from 1.0 to 1.2 times the smallest value are reclassified as special nodes. This range was empirically chosen to minimize the required computation time for a large group of test problems.

The larger the value of TVARY, the sooner the size of the time step (see Section III.I.2) will become large, relative to the stability limits of most of the nodes in the system, and the sooner they will be reclassified as special nodes if KSPEC is zero.

2. Control of Time Step

The user may specify minimum and maximum time steps SMALL and DELTO and a desired average maximum temperature change per time step TVARY, and choose the method of calculation, by specifying that certain nodes be special nodes (with KS in input BLOCK 4) and by using KSPEC in input BLOCK 1 (see Section II.C.15). These, and a number of other factors, all affect the way in which the time step varies as the calculation proceeds.

The first time step (KCYC=0) is always 10^{-12}. This allows zero-volume nodes to reach equilibrium, and allows time derivatives to be established. At the end of the first time step, and each succeeding time step, the minimum stability limit, τ_n (SLIM), of any regular node in the system, Δt_{stab}, is found. This search is skipped after the first time step, if no change can occur in the stability limits and node classifications. The maximum allowed time step, DELTMX, is then set at 2/3 of Δt_{stab}, DELTO, or 10^{12}, whichever is smaller. However, DELTMX is not allowed to be less than 10^{-10}, so that no regular node should have a stability limit τ_n (SLIM) less than 10^{-10}, and DELTO must not be less than 10^{-10}. Use of 2/3 Δt_{stab} greatly reduces time-truncation error in coarsely zoned systems, compared with use of Δt_{stab}. The minimum time step, SMALL, is initially set equal to SMALT, the input value of SMALL. If SMALL is equal or greater than DELTMX, SMALL is reduced to slightly less than DELTMX so that the input value cannot force an unstable time step to be used. If no input value of SMALL is specified, if at least 1/4 of the nodes in the system are regular nodes, and if DELTMX is not equal to 10^{12}, SMALL is set equal to 1/100 of DELTMX.

Next, the largest temperature change for the time step, DTEMP, is found, exclusive of zero-volume nodes on the first two time steps. Another quantity, DTMAX, is the product of TVARY and either the largest percentage change that took place in any tabulated property, or 1/40 of the number of iterations required for convergence, whichever is larger. The ratio R_s is then calculated as follows:

$$R_s = \text{TVARY}/(\text{DTEMP, DTMAX})_{max}. \tag{III.I.2.1}$$

If R_s is less than 0.5, and Δt_s is greater than 1.01 SMALLS, the time step will be repeated. If there are no regular nodes in the system, R_s is reduced by a factor of 100 for the first time step, to insure that the calculation starts out smoothly. For values of R_s less than 1.0, a new ratio is calculated:

$$R_s' = R_s^2, \; R_s' \geq 0.5. \tag{III.I.2.2}$$

For values of R_s greater than 1.0, a new ratio is calculated:

$$R_s' = 0.5\,(1 + R_s), \quad R_s' \leq 2.0. \tag{III.I.2.3}$$

The new time step is then calculated as follows:

$$\Delta t_{s+1} = \Delta t_s R_s' \tag{III.I.2.4}$$

$$\Delta t_{s+1} \geq \text{SMALL} \tag{III.I.2.5}$$

$$\Delta t_{s+1} \leq \text{DELTMX}. \tag{III.I.2.6}$$

This method provides a continual gradual adjustment in the size of the time step to obtain a maximum temperature change of TVARY, or a maximum change in any tabulated property of 1%, and to prevent the number of iterations from averaging more than 40. The calculation of R_s' provides for a rapid decrease in Δt when these limits are exceeded, with a more gradual increase in Δt when the changes are smaller than these limits.

An additional adjustment in the size of the new time step is made if the input value of Δt_p (TIMEP) in BLOCK 1 is positive. This is done by multiplying the unadjusted new time step, $\Delta t_{s+1}'$ (DELT) by a factor as close as possible to 1.0, and in the range from 2/3 to 1.5, so that the remaining time to the next desired printout time is an integer multiple of the adjusted time step. Also, the adjusted time step must remain in the range from Δt_{min} (SMALL) to Δt_{max} (DELTMX). The remaining time to next desired printout time is first calculated as follows:

$$\Delta t_{rem}' = \left[\Delta t_p(1 + |t/\Delta t_p|) - t\right] \text{ modulo } \Delta t_p \tag{III.I.2.7}$$

where the vertical bars indicate that the integer value of the expression is to be used. An adjustment is made to insure that Δt_{rem} is at least 2/3 of the unadjusted new time step:

$$\Delta t_{rem} = \Delta t_{rem}' + \Delta t_p \left[1 - |(\Delta t_{rem}' - \Delta t_{s+1}'/1.5)/\Delta t_p|\right]. \tag{III.I.2.8}$$

The number of time steps remaining until the next printout time is then estimated as follows:

$$N_p = |\Delta t_{rem}/\Delta t_{s+1} + 0.5|,$$

and the adjusted new time step calculated as follows:

$$\Delta t_{s+1} = \Delta t_{rem}/N_p.$$

If Δt_{s+1} is less than SMALL, the next smaller positive value of N_p is used. If Δt_{s+1} is greater than DELTMX, the next larger value of N_p is used. The limit in Eq. (III.I.2.6) is then reapplied. The adjustment factor is always between the limits 2/3 and 1.5, and should average much closer to 1.0 if Δt_{s+1} is small compared with Δt_p. This method provides a continual minimum adjustment in the size of the time step, if possible, between the limits SMALL and DELTMX, to obtain the desired printout times.

If a maximum problem time t_{max} (TIMAX) has been specified, the new time step is also limited as follows, so that t will slightly exceed t_{max} on the last time step:

$$\Delta t_{s+1} \le t_{max} - t + 10^{-12}. \qquad \text{(III.I.2.9)}$$

The small increment (10^{-12}) insures against truncation error.

When KSPEC is not negative, the limit in Eq. (III.I.2.8) is seldom reached, since nodes are reclassified as special nodes when their stability limits are less than 1.8 DELTMX.

3. Estimation of Time Derivatives

Time derivatives of temperatures are used in estimating the average temperatures during the time step, to evaluate temperature-dependent tabulated properties, and are used to obtain the first estimate of temperature change in special nodes to begin the iterative heat-balance calculation. Time derivatives of phase concentrations are used to make the first estimate of the amount of phase change during the time step. Time derivatives of reactant concentrations are used to estimate the average concentrations in the nodes for the calculation of concentration changes due to mass flow.

The average time derivatives for the next time step are estimated at the end of each time step, from the rates of change calculated for the latter time step. The estimate makes use of the ratio of the maximum rates of change for the two preceding time steps, R_c, and the ratio of the extrapolated time interval to the preceding time interval between the midpoints of the time step, R_t:

$$R_c = \left(\frac{\Delta T_{max,\,s-1}}{\Delta t_{s-1}}\right) \Big/ \left(\frac{\Delta T_{max,\,s}}{\Delta t_s}\right) \qquad \text{(III.I.3.1)}$$

$$R_t = (\Delta t_{s+1} + \Delta t_s)/(\Delta t_s + \Delta t_{s-1}),\ 0.5 \le R_t \le 2.0. \qquad \text{(III.I.3.2)}$$

The limits on R_t result from the limits placed on changes in the size of the time step, i.e., no more than a factor of 2. If R_c is greater than 1.0, i.e., the maximum rate of temperature change is decreasing, the estimate is based on the assumption that all temperatures are approaching equilibrium exponentially, with the same time constant. In that case, the ratio between the slope at the center of the next time step, Δt_{s+1}, to the slope at the center of the time step just completed, Δt_s, is given by F_c, as follows:

$$F_c = R_c^{-R_t}, \quad 10^{-24} \le F_c \le 1. \tag{III.I.3.3}$$

If R_c is less than 1.0, i.e., if the maximum rate of temperature change is increasing, the estimate is based on the assumption that the temperature changing most rapidly is following a quadratic curve and that the ratio between successive slopes is the same for all nodes. In that case, F_c is given by

$$F_c = 1 + R_t(1 - R_c), \quad 1 \le F_c \le 3. \tag{III.I.3.4}$$

The upper limit on F_c results from the limits on the value of R_t. This is a more conservative estimate than assuming an exponential increase, which would result in a greater estimated change in slope. In no case are the signs of the derivatives changed.

The estimated time derivatives are then calculated as follows:

$$\dot{T}_n = F_c\,(\Delta T_{n,s}/\Delta t_s) \tag{III.I.3.5}$$

$$\dot{A}_n = F_c\,(\Delta A_{n,s}/\Delta t_s) \tag{III.I.3.6}$$

$$\dot{B}_n = F_c\,(\Delta B_{n,s}/\Delta t_s) \tag{III.I.3.7}$$

To safeguard against several possible sources of instability, the following precautions are taken: (a) F_c is fixed at 1.0 for the first two time steps; (b) F_c is fixed at 1.0 for two time steps after any time step has been rejected, or a regular node has been reclassified as a special node; (c) time derivatives for the initial time step ($\Delta t = 10^{-12}$) are zero; (d) time derivatives of zero-volume nodes are fixed at zero for the first two time steps; (e) time derivatives of other special nodes are calculated as follows for the first two time steps:

$$\dot{T}_{n,\,\mathrm{special}} = (\Delta T_{n,s}/\Delta t_s)/(1 + \Delta t_s\, Z_n/\overline{C}_n), \tag{III.I.3.8}$$

where Z_n is the total conductance of the node, including all thermal connections and including mass-flow connections for which the node is a sink—this produces a more accurate result for special nodes with stability limits small compared with Δt_s; (f) time derivatives are multiplied by 10^{-24} whenever they change sign.

Each of these safeguards has been found necessary for some particular type of problem.

The user may also cause all time derivatives to remain fixed at zero throughout the problem, by specifying a positive value of NDOT in data BLOCK 1. This will usually reduce accuracy and increase machine time, but may be done to investigate the effect of using time derivatives, or to compare calculational results with those of a program that does not use time derivatives.

4. Estimation of Interpolation Factor

The interpolation factor f (FOR) is used (a) to estimate average values of temperatures during the time step, for use in evaluating temperature-dependent tabulated properties; (b) to estimate average values of reactant concentrations for use in calculation of concentration changes due to mass flow; and (c) to interpolate between initial and final temperatures for the time step, to obtain the temperatures used to determine rates of heat flow for thermal and mass-flow connections involving special nodes. The values of f corresponding to use of forward, central, or backward difference equations are 0.0, 0.5, and 1.0, respectively, for the use listed in (c), above. In TRUMP, f is usually allowed to vary between 0.57 and 1.0, depending on the estimated ratio between successive rates of temperature change F_c (see Section III.I.3 preceding), as follows:

$$f = [0.57, (1.0, F_c)_{max} / (1.0 + F_c)]_{max}, \qquad \text{(III.I.4.1)}$$

where the maximum of each bracketed pair of values is used. It can be shown that the heat-transport difference equations [Eq. (III.H.1.2)] are only stable if f is 0.5 or more, for time steps larger than the time constant of any node [Eq. (III.H.1.5)]. In addition, small oscillations in the solution do not damp out unless f is greater than 0.5. The minimum value for f, 0.57, was empirically chosen to optimize the stability and accuracy of a large number of test problems.

The form of Eq. (III.I.4.1) was chosen to give f a value close to the interpolation factor required to obtain a correct average temperature when temperatures are approaching equilibrium values exponentially, i.e., near 0.5 when slopes do not change appreciably over a time step, and near 1.0 when large time steps are used near equilibrium. Thus the accuracy inherent in use of central-difference equations is obtained, without the undamped oscillations resulting from such equations when rapid changes occur in boundary conditions or temperature- or time-dependent properties, during the part of a problem in which temperature changes are rapid. The interpolation factor is gradually shifted toward 1.0 as equilibrium is approached, to maintain accuracy that is lost when central-difference equations are used. Approach to equilibrium is usually too rapid when forward- or central-difference equations are used. A much more accurate approach is obtained by the method used here. During the transient part of the problem, the value of f used need not be exact, as errors averaged over several time steps are usually much smaller than the error for a single time step, a particular characteristic of the conduction equation.

As an additional safeguard against possible sources of instability, the value of f is always fixed at 1.0 for the initial time step ($\Delta t = 10^{-12}$) and for any time step following a rejected time step. This insures that zero-volume nodes and other nodes with small stability limits reach equilibrium with nodes to which they are connected, without overshoot and resulting damped oscillation [see Eq. (III.H.1.8)], after any change in a boundary condition or a temperature- or time-dependent property.

The user may control the value of f used on other time steps by specifying in data BLOCK 1, a value of KSPEC of 2, to fix f at 1.0—or, if KSPEC is 3, to fix f at 0.5. In both cases, all nodes are reclassified as special nodes before the first time step, so that the resulting calculation method is either the "backward" or "central" method, respectively; this will generally reduce accuracy and increase machine time, but it may be done to investigate the effect on accuracy or to compare calculational results with those of a program in which one of those methods is used.

5. Testing Criteria for Ending Problem

Before beginning the next time step, the various input criteria and the steady-state criteria (see Section II.C.14) are tested. The limit on machine time (msec) is only tested on time steps for which the cycle counter (KCYC) is divisible by 2000/NODES, to reduce the machine time used to call the TICHEK routine. This interval was chosen so that the test is made approximately every 2 to 10 sec.

IV. The Program

A. GENERAL

The TRUMP program[12] consists of a main program and several subroutines. The language is mostly FORTRAN IV, with a few statements, primarily in optional parts of the program, specific for the computers and versions of FORTRAN in use at LLL. All versions in present use at LLL are combined in a single FORTRAN source deck. When compiling to obtain a specific version, the choices of inclusion or exclusion of the various optional parts of the program, and the dimensions assigned to variable arrays, are determined by the values of several parameters in the main program. When recompiling with new values of any of these parameters, only these subroutines directly affected by the change, as shown in Table 4, must be recompiled. Compilation of optional statements and subroutines are controlled by FORTRAN DIF statements, which test the values of various parameters and indicate the statement labels to which the compiler will skip during compilation. If the DIF statement or an equivalent is not available in a language to which TRUMP is being adapted, special versions may be formed by removing the indicated statements by hand.

The TRUMP program has been in use at LLL since July 1, 1966, and superseded the HARC and HEART THROB transport programs in use since 1961.[3*]

All parts of the program have been thoroughly checked for accuracy and efficiency. The results of a large number and variety of problems have been carefully analyzed and compared with analytic solutions, with the results of other computer programs, and with experimental data.

Information concerning system subroutines used in TRUMP, for which equivalent subroutines may not be available at another computer installation, is available from the Computer Information Center, LLL, Livermore.[9-17]

---

*See the Introduction to this report, for a partial listing of users of TRUMP.

Table 4. TRUMP subroutine characteristics.

Name	Input data blocks	Cliches used	Labeled common blocks used	Parameters determining compilation of	
				Entire subroutine	Some statements
MAIN	–	STORK, STORE	–	–	MV36ST, MV66DS, MV66ST, M_8, M_{12}
KITTY	–	STORK	GOBCOM (System)	KITTEN, MACHINE	–
HEART	–	STORK, STORE, STORM	–	–	MACHINE
THERM	2, 4, 12	STORK, STORE, STORM	ACHEMS	–	M_3, M_8, M_{12}
GEN	8	STORK, STORE, STORM	AGEN	M_8	M_{12}
CHEM	3	STORK, STORE, STORM	ACHEM, ACHEMS	M_3	M_{12}
FINK	5	STORK, STORE, STORM	AFINK	–	–
FLOW	10	STORK, STORE, STORM	AFLOW, AFLOWS, ACHEMS	M_{10}	M_3, M_{12}
SURE	6, 7	STORK, STORE, STORM	ASURE, ASURES	M_6, M_7	M_{12}
SPECK	–	STORK, STORE, STORM	ASPECK, AFLOWS, ACHEMS	–	M_6, M_7, M_{10}
TALLY	1, 9	STORK, STORE, STORM	ACHEMS	–	M_3, M_8
PLOT	11	STORK, STORE, STORM	APLOT	M_1, M_{11}	KPLOTR, KPLOTL, KPLOTT
PRIME	2(*Mast)	STORK, STORE	APRIM	KPRIME	MACHINE
PATCH	–	STORK	–	–	MACHINE
REFER	–	STORE	–	–	–
SEEK I	–	STORE	–	–	–
SEEK 2	–	STORE	–	–	–
TIMER	–	STORK	–	–	MACHINE

B. TRUMP SUBROUTINES

1. General

TRUMP consists of a main program and several required and optional subroutines, divided according to logical function, the sequence of calculations, and the various modes of heat production and transport. The subroutines required for all heat conduction problems are HEART, THERM, FINK, SPECK, TALLY for the input, initialization, and time-step phases of the calculation; PATCH, SEEK1, and SEEK2 for the input phase only; and REFER for the initialization phase only. The optional subroutines include GEN for problems with internal heat sources and sinks, CHEM for problems with chemical reaction, FLOW for problems with motion of conducting materials (or body forces in flow problems where pressure replaces temperature as the dependent variable), and SURE for problems with variable boundary conditions. Other optional subroutines include KITTY for use of TRUMP as a controllee under the Livermore Timesharing System,[17] PLOT for production of graphical output with the dd80 plotting system, PRIME for obtaining material property data from a master data file[19] and for transferring data from one file to another, and TIMER for obtaining the machine time used by each problem. The additional entry points THERM1 and THERM2 in THERM are for calculation of transitions involving latent heat, such as melting. The additional entry point TALLY1 in TALLY is for the option of punching a card deck in the format of input Data BLOCK 9, containing the final values of temperature, phase or reactant concentrations, and constant heat-generation rates for the problem.

Some of the characteristics of the TRUMP subroutines are summarized in Table 4 and in the "Glossary," Section IV.G.5.

2. Main Program

The main program contains all PARAMETER and CLICHE statements used in the program. The parameters control the compilation of various optional and machine-dependent parts of the program, and the dimensions of most variable arrays. CLICHE STORK contains the PARAMETER statements. CLICHE STORE contains the first part of blank common, used in all subroutines except KITTY, PATCH, and TIMER. CLICHE STORM contains the remainder of blank common, used in all subroutines except KITTY, PATCH, PRIME, REFER, SEEK1, SEEK2, and TIMER.

The main program is the normal initial entry point when a source or object deck is being loaded under control of the Monitor System. When used as a controllee executed under control of a remote teletype, KITTY is the initial entry point. Before HEART is called, the compilation date (entered by hand in the source deck) is written out. Normal exit (return of control to the Monitor System) takes place from the main program, after returning from HEART.

3. KITTY

Subroutine KITTY is the initial entry point when TRUMP is used as a controllee under the Livermore Time Sharing System. In KITTY, the OØDROP subroutine is called, to create a copy of the TRUMP controllee disk file, and transfer execution to the new file, named +TRUMP, so that the file TRUMP will be saved. A FROST call is made to obtain the name of the data input file, NFILE, from the "TRUMP NFILE/TV" teletype execute line. Logical input unit 2 is assigned to the disk file NFILE, and HEART is called.

4. HEART

Subroutine HEART is the logic control center of the TRUMP program. All calls to other TRUMP subroutines except PRIME, PATCH, SEEK1, SEEK2, REFER, and TIMER are made from HEART. Before any input data are read in, several flags are initially set, and calls are made to ERRORS to set underflows to zero. Calls are made to initially set the dd80 subroutines, and the machine clock is monitored. Any TRUMP control cards preceding the first Problem Name card are read in, and the appropriate variables are reset or initially set, and calls are made as necessary to control input and output unit specifications.

After the Problem Name card has been read in, various arrays are initially set depending on the value of the data carryover control in column 72. If data are being carried over from the preceding problem, TALLY is called to initially set node classifications and the initial conditions for the new problem.

Block Number cards are then read in, and the appropriate subroutines are called for reading in data from the various data blocks. After the Data End card is read in, the Problem Name and the integer value on the Data End card are stored in new locations, and subroutines TALLY and SPECK are called to initially set variables before the first time step.

At the beginning of each time step, the time-step counter is incremented. Calls are then made to the subroutines required for the time step calculations, and to subroutine PLOT for production of graphical output at intervals determined by the input data. After each time step is completed, various tests are made to determine whether the problem should be continued, ended, or interrupted and saved for later continuation. If the problem is ended, the input file is read again and TRUMP control cards and Problem Name cards are handled as described above. If the problem is to be interrupted, system subroutine OFFMON is called.

When restarting with system subroutine ONMON or ONMONS, from a file produced by OFFMON, the input file is read in search of a "*DISCO" TRUMP control card to determine whether or not a previously interrupted problem is to be continued or ended. If the latter, the last time step is completed before reading in data for the next problem.

The machine clock is monitored at the beginning of each problem, again after all data has been read in, and again when the problem is interrupted or ended.

All data read in from the input file are written out, and statements are written out concerning the Problem Name, clock time, date, machine designation, machine time used, data carryover, specification of input and output files, interruption and restart, and the end of each problem.

HEART calls TRUMP subroutines THERM, THERM1, THERM2, GEN, CHEM, FINK, FLOW, SURE, SPECK, TALLY, TALLY1, PLOT, and TIMER, and returns to MAIN or KITTY after reading in a "*SPLIT" TRUMP control card.

5. THERM

Subroutine THERM is entered to read in the data items in data BLOCKs 2 (material properties), 4 (node descriptions), and 12 (nodes with properties dependent on temperatures of other nodes). All input data and immediately derived data, such as table slopes, material enthalpies, and node classifications and volumes, are written out.

For the first time step, subroutine THERM is entered to initially set cross-reference tables that relate the node list to the material list and relate the remote temperature dependence list to the node list. The node conductivities, masses, heat capacities, and heat contents, and the total material volumes, masses, heat capacities, heat contents, and average temperatures, are calculated.

The phase concentrations in materials with transitions involving latent heat are initially set.

For each time step, THERM is entered to recalculate any of the quantities above that are variable. At intervals determined by input data, summary data are written out for each material, listing name, number, heat capacity, heat content, average temperature, transition temperature, and latent heat of transition. Summary data are written out for each node, listing number, material number, classification, location, volume, mass, heat capacity, conductivity, net conductance, and time constant.

If the problem includes nodes having transitions with latent heat, THERM is entered at each time step at entry THERM1 to estimate the amount of latent heat released or absorbed in each such node. THERM is entered at entry THERM2 after temperature changes are calculated in subroutine SPECK, for the final calculation or release or absorption of latent heat. THERM calls TRUMP subroutines PATCH, PRIME, SEEK1, SEEK2 and REFER and returns to HEART.

6. GEN

Subroutine GEN is entered to read in the data items in data BLOCK 8, the list of nodes with variable heat-generation rates. All input data and immediately derived data, such as table slopes, are written out.

For the first time step, GEN is entered to initially set the cross-reference table that relates the variable-heat-generation tables to the node list. At each time step, all variable heat-generation rates are recalculated, and the heat added to or removed from each node by variable or constant heat generation is calculated.

GEN calls TRUMP subroutines SEEK1 and REFER, and returns to HEART.

7. CHEM

Subroutine CHEM is entered to read in the data items in data BLOCK 3, the reactant list. All input data and immediately derived data, such as table slopes, are written out.

For the first time step, CHEM is entered to initially set the cross-reference table that relates the material property list to the reactant list, and to initially set the heats of reaction, log collision frequencies, and activation energies of all nodes of materials containing reactants.

For each time step, any of these quantities which are variable are recalculated, and the heat added to each reacting node is calculated. At intervals determined by input data, if any of the input quantities are variable, their values in each reacting node are written out.

CHEM calls TRUMP subroutines SEEK1 and REFER, and returns to HEART.

8. FINK

Subroutine FINK is entered to read in the data items in data BLOCK 5, the thermal connection list. All input data and immediately derived data, such as connection areas, are written out.

For the first time step, FINK is entered to initially set the cross-reference table that relates thermal connections to the node list, and to initially set the conductances of each connection.

At each time step, FINK is entered to recalculate the conductance of connections with thermal radiation or involving nodes with variable thermal conductivity, and to calculate the amount of heat flow across each connection, based on the temperature difference at the beginning of the time step.

At intervals determined by input data, summary data for each thermal connection are written out, including the node numbers, area, interface conductance, radiation view factor, overall conductance, cumulative total heat flow, and average rate of heat flow, across the connection.

9. FLOW

Subroutine FLOW is entered to read in the data items in data BLOCK 10, the mass-flow connection list. All input data and immediately derived data, such as table slopes, are written out.

For the first time step, FLOW is entered to initially set the cross-reference table that relates the mass-flow list to the node list, test for connections between different materials, calculate the geometric factors used to determine the interface conditions for each connection, reorder connections with negative mass-flow rates, and initially set variables used to summarize the mass-flow rates and cumulative flow for connections and nodes.

At each time step, FLOW is entered to recalculate variable mass-flow rates and to calculate the amounts of heat transferred by mass-flow across each connection, based on temperatures and enthalpies at the beginning of the time step and estimated

values of latent heat release or absorption during the time step. The changes in reactant concentration due to mass flow are also calculated, based on estimated average values of reactant concentrations during the time step.

At intervals determined by input data, summary data are written out for each mass-flow connection, including the node numbers, the flow rate, the total flow, the average flow rate, and the geometric factors used to determine the interface conditions. Summary data are written out for each node with one or more mass-flow connections, including the node number, the net flow rates in and out, the total amounts of flow in and out, the average flow rates in and out, and the average residence time of material in the node.

FLOW calls TRUMP subroutines PATCH, SEEK2 and REFER, and returns to HEART.

10. SURE

Subroutine SURE is entered to read in the data items in data BLOCK 6, the surface-connection list, and in data BLOCK 7, the external-temperature (boundary-node) list. All input data and immediately derived data, such as connection areas and table slopes, are written out.

For the first time step, SURE is entered to initially set the cross-reference tables that relates the surface-connection list to the node list and the boundary-node list. The initial values of several quantities are calculated, including the conductance of the surface connections.

At each time step, SURE is entered to calculate the cumulative heat flow across each surface connection, the new boundary node temperatures, the new surface-connection conductances, and the amounts of heat flow across each surface connection. These calculations are based on estimated average conductances during the time step, surface-node temperatures at the beginning of the time step, and boundary-node temperatures at the end of the time step.

At intervals determined by input data, summary data are written out for each surface connection, including the surface and boundary node numbers, the connection area, the surface conductance, the free-convection exponent, the radiative transfer form factor, the overall conductance, and the net heat flow and average heat-flow rate into the surface node from the boundary node. Summary data are written out for each boundary node, including the boundary node number, the external temperature, and the net heat flow and average heat-flow rate into the system from the boundary node. The total heat flow and average flow rate into the system for all boundary nodes are also written out.

SURE calls TRUMP subroutines PATCH, SEEK2, and REFER, and returns to HEART.

11. SPECK

Subroutine SPECK is entered after all input data have been read in, and, for the first time step, to initially set various quantities.

At each time step, SPECK is entered to make the final calculation of heat flow across surface connections, and across internal thermal and mass flow connections including nodes classified as special nodes. This calculation is based on interpolated values of special node temperatures during the time step, weighted from 0.57 to 1.00 toward the temperatures at the end of the time step.

When special nodes are connected to each other, an iterative calculation is required, which is done by an accelerated method and imposes strict convergence criteria on average and individual temperature changes. Diagnostic statements are written out if a specified maximum number of iterations are required, and if the problem must be ended because of convergence failure when the smallest allowable time step is being used.

Heat flow into regular nodes connected to special nodes is corrected to maintain an exact heat balance for each connection.

At intervals determined by input data, summary data are written out, including the number of time steps completed; the total, average, and maximum number of iterations used; and the temperature interpolation factor.

SPECK returns to HEART.

12. TALLY

Subroutine TALLY is entered after the Problem Name card of a continuation problem has been read in, to determine the initial conditions and node classifications for the new problem. Subroutine TALLY is entered to read in the input data in data BLOCK 1, which include output controls; criteria for ending the problem; geometric, unit-system, and accuracy controls; and various system constants. All input data and immediately derived data are written out. Subroutine TALLY is also entered to read in the input items in data BLOCK 9, the initial condition list. All input data and immediately derived data are written out.

After all input data has been read in, TALLY is entered to write out a summary of the number and sizes of input data blocks, data carried over from preceding problems, the number of tables of various types, the numbers of certain types of input items, and the amount of memory used for groups of variables and arrays required for various input-data options. The problem time, the size of the first time step and interpolation factor, and other variables are initially set.

Subroutine TALLY is the time-step control center of the program. At the end of the first time step, TALLY is entered to initially set several variables used thereafter in TALLY. At the end of every time step, TALLY is entered to determine the maximum stable time step for regular nodes. Nodes may be reclassified as special nodes to keep the size of the time step from being limited. The largest temperature change for the time step is found. The results of the time step are accepted or rejected, based

on input data, the largest temperature change, and changes in tabulated quantities during the time step. The size of the next time step is determined based on a number of criteria. The interpolation factor and slope-correction factor for the next time step are determined. The new values of temperatures, phase and reactant concentrations, and their rates of change are found, and heat-balance data are updated. Various criteria for ending the problem are tested. Various criteria for producing output data for the time step are tested. When required, summary data for the time step are written out, including heat-balance data for the system, new values, change and rates of change of temperatures and phase and reactant concentrations, and heat-balance data for the nodes. Other special output data may also be produced if required by input data.

TALLY is entered at entry point TALLY1 when a problem is ended or interrupted, if required by input data, to punch the final conditions of the problem in the format of data BLOCK 9. These cards may be added to the original data deck as a means of continuing the problem.

TALLY calls TRUMP subroutines PATCH, SEEK1, REFER, and TIMER, and returns to HEART.

13. PLOT

Subroutine PLOT is entered to read in the data in data BLOCK 11, which contains the controls and specifications for producing graphical output data. All input data and immediately derived data are written out. For the first time step, plot is entered to initially set the cross-reference table that relates the time plot node list to the main node list.

At intervals determined by input data, PLOT is entered. If required, plots and a table of temperature vs node location are produced. Data are processed and stored to be used at the end of the problem for producing plots and tables of temperature vs time for certain nodes specified in data BLOCK 11. If required, plots and a table of maximum and minimum temperature vs node location are also produced at the end of the problem. If required, a plot of node location vs time for various isotherms is produced at the end of the problem, for the nodes specified in data BLOCK 11.

PLOT uses the dd80 plotting routines to produce graphical output. The inital dd80 calls for identification, character set, and page format are made in HEART.

PLOT calls TRUMP subroutine REFER, and returns to HEART.

14. PRIME

Subroutine PRIME is called by subroutine THERM whenever a "*MAST" control card is read in. If variable MALL on that control card is zero, PRIMES reads in the input data for the optional selection list of master material identification numbers and secondary identification numbers. Input data, including data on the "*MAST" control card, and optional unit conversion factors, are written out, PRIME is used to read in the master materials data list and write out the secondary materials data list, after optionally converting the units to a new system, for either all the materials on the

master list, or just those specified on the optional selection list. In the latter case, the secondary material identification numbers are substituted for the master material identification numbers. All data transferred are written out, as are some summary data and some diagnostic data that may result from errors. The secondary list then becomes the input list for BLOCK 2 items. An option is available for specifying another I/O unit to be used for the input list, after returning to THERM.

PRIME is also used when the optional "*DECKS" control card is used. If a problem is interrupted by Sense Switch 1, PRIME is called by HEART to copy all remaining data decks up to the next end-of-file mark on the input unit (on units NRS, NR, and NB, a card with an "$" in column 1 will be recognized as an end-of-file mark) onto the unit NDATA specified on the "*DECKS" card, using a buffered I/O routine. The file made on unit NDATA may then be saved and used in a later job as in input file.

PRIME returns to the calling routine, which may be HEART ("*DECKS" option) or THERM ("*MAST" option).

15. PATCH

Subroutine PATCH is entered during the input phase of the problem, to test certain input variables in several data blocks. These variables are initially read in as 10 separate Hollerith characters. In PATCH, these characters are tested to see if any of them is a decimal point. If so, the array of 10 characters is converted to a floating point number and assigned to the input variable. If not, a floating point number in the subroutine argument list is assigned to the input variable. A sentinel is also returned, indicating whether or not a decimal point was found.

PATCH is used for several purposes. Data in data BLOCK 1 are substituted for certain input variables when the data field on the input cards is blank. Other input variables have specific values, written into the program, assigned to them when the input data field is blank. Other input variables have data fields in columns 72 through 80 of the input cards, which may contain Hollerith data when produced by certain procedures. It is sometimes convenient to purposely put Hellerith data, as a comment, in the data fields of input variables tested by PATCH, when numerical values of those variables are not needed.

PATCH returns to the calling routine, which may be THERM, FINK, FLOW, SURE, or TALLY.

16. SEEK1

Subroutine SEEK1 is entered only during the data-input phase of the problem, when a data BLOCK 2, 3, 4, 7, 8, or 9 of type B is being read in. The list of identification numbers previously read in for the same data block is searched, in reverse sequential order, to find one equal to the last one read in. If one is found, the sequence number in the list is returned, so that the new input data can be substituted for the old input data. If one is not found, the new input data are added to the end of the list.

This procedure allows input data already in memory to be modified, or new data added, with a minimum number of input data cards. The data to be modified may have been carried over from a preceding data deck, or read in as an item in a separate data block, or as a preceding item in the same data block.

SEEK1 returns to the calling routine, which may be THERM, CHEM, GEN, SURE, or TALLY.

17. SEEK2

Subroutine SEEK2 is entered only during the data input phase of the problem, when a data BLOCK 5, 6, 10 or 12 of type B is read in. SEEK2 has the same function and follows the same procedures as SEEK1, except that the search is for a pair of identification numbers which are the same as the pair just read in.

SEEK2 returns to the calling routine, which may be THERM, FINK, FLOW, or SURE.

18. REFER

Subroutine REFER is entered only during the first time step, to produce cross-reference tables relating various input-data item lists to others. The sequence number in the node list (data BLOCK 4) must be found for all nodes listed in data BLOCKs 1, 5, 6, 8, 9, 10, 11, and 12. The sequence number in the reactant list (data BLOCK 3) must be found for each reactant listed in data BLOCK 2. The sequence number in thematerial list (data BLOCK 2) must be found for each material listed in data BLOCK 4. The sequence number in the boundary-node list (data BLOCK 7) must be found for each boundary node listed in data BLOCK 6. This procedure makes possible the use of arbitrarily chosen identification numbers for each of the four types of items referenced by other data blocks, allowing flexibility in arranging and modifying data decks. Also, the identification numbers may be chosen to convey additional information to the user, such as composition, spatial location, function in the problem, or sequence of modification of data.

The search for each item referenced is made in reverse sequence, so that the latest item in the list is used. If a referenced identification number is not in the list being searched, a diagnostic statement is immediately written out. An error indicator is set, and certain parts of the program are skipped, and the problem is ended at the end of the first time step.

REFER returns to the calling routine, which may be THERM, GEN, CHEM, FINK, FLOW, SURE, TALLY, or PLOT.

19. TIMER

Subroutine TIMER finds the machine time, in seconds, charged to the program since the last call of TIMER, by means of the system subroutine CLOCK1 (CDC/3600) or TICHEK (CDC/6600).

TIMER returns to the calling routine, which may be HEART or TALLY.

C. SYSTEM SUBROUTINES USED IN TRUMP

The various system subroutines used in TRUMP, including in-line and library subroutines, are listed in Section IV.G.6 with a brief indication of their purpose. More detailed descriptions are available in the computer information center at LLL. Many of these are used only for optional features of the program, such as the routines in KITTY for teletype control, the dd80 plotting routines, clock calls, special I/O unit controls, routines in PRIME for buffered I/O of data files, and for convenience features, such as use of CLICHE, ENDCLICHE, USE, and PARAMETER statements, and OFFMON. Most others are standard features, or have equivalents in most versions of FORTRAN, or can easily be replaced with other FORTRAN statements.

D. MEMORY REQUIREMENTS AND PROBLEM SIZE LIMITS

The TRUMP program requires core-memory storage for the program itself, resident and nonresident system subroutines, and data. The latter include data in unlabeled (blank) common, data in labeled common blocks, subscripted arrays listed in DIMENSION statements in subroutines, and unsubscripted variables and constants that are not listed in any COMMON statements.

The dimensions of most arrays are determined by the parameters M1 through M12 specified in one of several PARAMETER statements in CLICHE STORK in the main program. Compilation of one of these PARAMETER statements, and of all optional parts of the TRUMP program, are determined by other parameters specified in CLICHE STORK. The values of these parameters for two versions of TRUMP are listed in Table 5. The corresponding size limits for the lists of data items in various input data blocks, and for input tables, are listed in Table 1. The memory requirements for the subroutines are listed in Table 6. The memory requirements for blank and labeled common blocks are listed in Table 7. The total memory requirements are listed in Table 8. The memory requirements for TRUMP and system subroutines are approximate and subject to occasional change.

Any set of problem-size limits may be used, for which the total memory requirements do not exceed the available memory not required by the program and the system.

Table 5. TRUMP parameter specifications.

Parameter	Subroutines affected	Common blocks affected	Version	
			6600/STD	6600/DS
MV36ST	All	All	0	0
MV66ST	All	All	1	0
MV66DS	All	All	0	1
M1	PLOT	APLOT	200	200
M2	All[a]	BLANK, ACHEMS, APRIM	40	50
M3	CHEM, FLOW, THERM, TALLY	ACHEM, ACHEMS	5	0
M4	All[a]	BLANK, ACHEM, ACHEMS, AFLOW, ASPECK, APLOT	480	600
M5	All[a]	BLANK	1200	1750
M6	SURE, SPECK	ASURE, ASURES	125	150
M7	SURE, SPECK	ASURE	20	20
M8	GEN, THERM, CHEM, FLOW, SURE, TALLY	BLANK	100	150
M9	All[a]	BLANK	12	12
M10	FLOW, SPECK	AFLOW, AFLOWS	80	0
M11	PLOT	APLOT	12	16
M12	THERM, GEN, CHEM, FLOW, SURE	BLANK	480	600
KITTEN	KITTY	NONE	0	0
MACHINE	HEART, PRIME, PATCH, TIMER, KITTY	NONE	1	1
KPRIME	PRIME	APRIM	1	1
KPLOTL	PLOT	APLOT	1	0
KPLOTR	PLOT	APLOT	1	0
KPLOTT	PLOT	APLOT	1	1

[a]Except KITTY, PATCH, TIMER.

Table 6. TRUMP subroutine memory requirements.

Required subroutines	Parameters controlling compilation	Trump version 6600/STD	Trump version 6600/DS
MAIN		31	31
HEART		1432	1432
THERM	M_3	1528	1523
FINK		515	515
SPECK	M_6, M_7, M_{10}	476	406
TALLY	M_3	2052	2009
PATCH		48	48
REFER		104	103
SEEK1		66	64
SEEK2		82	80
Total required		6334	6211
Optional subroutines			
KITTY	MACHINE, KITTEN	82	82
CHEM	M_3	979	6
GEN	M_8	427	427
FLOW	M_{10}, M_3	781	6
SURE	M_6, M_7	1059	1064
PLOT	M_1, M_{11}, KPLOTL, KPLOTR, KPLOTT	1830	811
PRIME	KPRIME	1079	1079
TIMER	MACHINE	17	17
Total optional		6254	3492
Total memory requirements		12,588	9703

Table 7. TRUMP common block characteristics.

Common block	Subroutines in which used (and input data blocks)	Parameters controlling compilation	Equation for variable and array memory requirements	Required memory 6600/STD	Required memory 6600/DS
Required in all versions:					
BLANK[a]	ALL[b] (1, 2, 4, 5, 9)		$162 + (8 + 7M_9)M_2 + 28M_4 + 12M_5 + 3M_9$	31,718	42,598
Optional common blocks:					
APRIM	PRIME (2)[c]	KPRIME	$30 + 2M_2 + 2M_9$	134	154
ACHEM	CHEM (3)	M_3	$(6 + 9M_9)M_3 + 6M_4$	3,450	0
ACHEMS	CHEM, THERM, FLOW, TALLY (3)	M_3	$4M_2 + 4M_4$	2,080	0
ASURE	SURE (6, 7)	M_6, M_7	$(9 + 3M_9)M_6 + (5 + 3M_9)M_7$	6,445	7,570
ASURES	SURE, SPECK (6, 7)	M_6, M_7	$3M_6$	375	450
AGEN	GEN (8)	M_8	$(3 + 3M_9)M_8$	3,900	5,850
AGENS	THERM, GEN, TALLY (8)	M_8	$2M_4$	960	1,200
AFLOW	FLOW (10)	M_{10}	$4M_4 + (6 + 3M_9)M_{10}$	5,280	0
AFLOWS	FLOW, SPECK (10)	M_{10}	$3M_{10}$	240	0
APLOT	PLOT (11)	M_1, M_{11}, KPLOTR	$4M_4$	1,920	0
APLOT	PLOT (11)	M_1, M_{11}, KPLOTT	$M_1 + 3M_{11} + M_1M_{11}$	2,636	3,448
AREM	THERM (12)	M_{12}	$5M_{12}$	2,400	3,000
AREMS	THERM, GEN, CHEM, FLOW, SURE (12)	M_{12}	$8M_4$	3,840	4,800
			Total optional	33,660	26,472
			Total required and optional	65,378	69,070

[a]Contains variables and arrays required for all problems.
[b]Except for KITTY, PATCH, TIMER.
[c]When *MAST control card used.

Table 8. TRUMP total storage requirements.

	Version 6600/STD	Version 6600/DS
TRUMP subroutines	12,588	9,704
Labeled common	35,820	30,922
Blank common	29,558	38,148
System subroutines	16,454	15,800
Total	94,420	94,574
Unused memory	1,260	1,106
Maximum program storage	95,680	95,680
System use	35,392	35,392
Total core memory	131,072	131,072

E. HARDWARE REQUIREMENTS

Input data are punched onto cards with an IBM-029 keypunch machine, or typed on a remote teletype for transmission to the disk files connected to the CDC/6600 or CDC/7600 computers. Program decks and input decks are transferred to 1/2-in. magnetic tape or disk files with the IBM/1401 or CDC/160A card readers. The CDC/6600 (with a 128K, 60-bit-word memory) and the CDC/7600 (with a 64K, 60-bit-word, small-core memory and a 256K, 60-bit-word, large-core memory) are used at LRL, Livermore. A greatly simplified version of TRUMP could be used to solve simple heat-conduction problems in systems of several hundred nodes on an 8K machine. Four 1/2-in. magnetic tape drives are normally used, in addition to those required by the system, for data input, data output, optional interrupt memory dump, and optional output for off-line card punching and dd80 plotting. On the CDC/6600 systems, data input is from disk files. The high-speed tape output is printed on the Radiation printer. Film produced by the dd80 plotters is printed on a high-speed electrostatic printer. Some output is printed on the on-line printers connected to the various computers, and on the user's teletype when running the OCTOPUS controllee version. Off-line card punching is done in the IBM/1401 system. Computer control is handled by a Monitor system.[17] The operator may control running time with Sense Switch 1, which in TRUMP causes an OFFMON tape dump to be made before returning control to the system.

F. MACHINE TIME REQUIREMENTS

The amount of machine time required for any particular TRUMP calculation depends on the size and complexity of the problem, the number of time steps, and iterations for each time step, required to complete the problem, and on the amount and types of output data produced.

Most problems require about 0.5 msec per time step and 0.1 msec per iteration, per calculational item, the latter being the sum of the number of nodes and boundary nodes, the number of surface connections, twice the number of internal thermal and mass-flow connections, and the number of table interpolations required per time step. This generally adds up to about 1 to 5 msec per node per time step.

Data output may slow the problems down if obtained more frequently than every 2 to 5 time steps, if buffered, or every 10 to 50 time steps, if not buffered, depending on the type and amount of numerical and graphical output produced on data-output cycles.

Total times range from a few seconds for small test problems to 30 to 60 min for the largest, most complex problems that have been run.

Every effort has been made to make the program as fast and efficient as possible, while still allowing the user to specify and control the accuracy of the calculation within reasonable limits.

G. TRUMP GLOSSARY

```
* TRUMP GLOSSARY.  VERSICN 4/7/72 .
*        (ITEMS NEW CR REVISED AFTER MAY 1, 1968 HAVE / IN COLUMN 1.)

      1.  GENERAL DEFINITIONS.

      BLOCK ITEM    AN INPUT DATA DESCRIPTION OF A MATERIAL, REACTANT,
      BLOCK ITEM      NODE, INTERNAL THERMAL OR MASS FLOW CONNECTION,
      BLOCK ITEM      EXTERNAL THERMAL CONNECTION, BOUNDARY NODE, HEAT
      BLOCK ITEM      GENERATION TABLE, NODE INITIAL CONDITION, OR REMOTE
      BLOCK ITEM      PRCPERTY DEPENDENCE.
      BOUNDARY      PERTAINING TO SPECIFIED CONDITIONS AT AN EXTERNAL
      BOUNDARY        SURFACE, OR TO AN EXTERNAL TEMPERATURE (BOUNDARY
      BOUNDARY        NODE) THERMALLY CONNECTED TO THE SYSTEM.
      CONDUCTANCE   RATE CF HEAT FLOW PER UNIT TEMPERATURE DIFFERENCE.
      CONNECTION    A MEANS OF HEAT TRANSPORT BETWEEN TWO NODES, OR
      CONNECTION      BETWEEN A SURFACE NODE AND AN EXTERNAL TEMPERATURE.
/     CONVERGENCE   ACCEPTANCE OF RESULTS OF ITERATIVE CALCULATION OF THE
      CONVERGENCE     HEAT BALANCE FOR INTER-CONNECTED SPECIAL NODES.  SEE
      CONVERGENCE     TVARY, HSUM, ESUM, EMAX, ERRS.
      DATA BLCCK    A GRCUP OF INPUT DATA OF A PARTICULAR TYPE, OR A LIST
      DATA BLCCK      OF BLOCK ITEMS, PRECEDED BY A BLOCK NUMBER CARD AND
      DATA BLOCK      FOLLOWED BY A BLANK CARD.  BLOCK ITEMS NEED NOT BE
      DATA BLCCK      ON THE SAME INPUT UNIT AS THE BLOCK NUMBER CARD.
      DATA BLOCK      SEE BLOCK, ABLOCK, NB, *INPUT.
      DATA DECK     INPUT DATA FOR A SINGLE PROBLEM, CONSISTING OF A
      DATA DECK       PROBLEM NAME CARD, DATA BLOCKS, AND A DATA END CARD.
      EXTERNAL      PERTAINING TO A BOUNDARY NODE OR CONNECTION BETWEEN
      EXTERNAL        A NCDE AND A BOUNDARY NODE.
      INTERFACE     A SURFACE COMMON TO TWO NODES, THROUGH WHICH HEAT MAY
      INTERFACE       BE TRANSPORTED BY CONDUCTION, CONVECTION, RADIATION
      INTERFACE       OR MASS FLOW.
      INTERNAL      PERTAINING TO A NODE OR CONNECTION BETWEEN NODES.
/     ITERATICN     A REPETITIVE STEP IN THE CALCULATION OF THE HEAT BALANCE
      ITERATICN       FOR INTER-CONNECTED SPECIAL NODES.
      MATERIAL      A SUBSTANCE WITH DENSITY, HEAT CAPACITY, AND THERMAL
      MATERIAL        CONDUCTIVITY.  IT MAY ALSO HAVE LATENT HEAT AT A
      MATERIAL        SPECIFIED TEMP, AND CONTAIN CHEMICAL REACTANTS.
      NODAL PCINT   A REPRESENTATIVE POINT WITHIN A NODE, FROM WHICH
      NODAL PCINT     DISTANCES TO INTERFACES ARE MEASURED TO DESCRIBE
      NODAL PCINT     INTERNAL THERMAL AND MASS FLOW CONNECTIONS.
      NODE          A VOLUME ELEMENT OF A PARTICULAR MATERIAL.
      PHASE         A MATERIAL STATE SEPARATED FROM ANOTHER STATE BY
      PHASE           A LATENT HEAT EFFECT AT A SPECIFIED TEMPERATURE.
/     REACTANT      A MATERIAL COMPONENT WITH A HEAT OF REACTION, COLLISION
      REACTANT        FREQUENCY, ACTIVATION ENERGY, AND REACTION ORDER,
      REACTANT        FOR AN ARRHENIUS DECOMPOSITION REACTION.
      REGULAR       CLASSIFICATION OF NODES FOR WHICH THE HEAT BALANCE IS
      REGULAR         BASED CN TEMPERATURES AT THE BEGINNING OF EACH TIME
      REGULAR         STEP FOR THERMAL AND MASS FLOW CONNECTIONS WITH
      REGULAR         OTHER REGULAR NODES.
      SPECIAL       CLASSIFICATION OF NODES FOR WHICH THE HEAT BALANCE IS
      SPECIAL         BASED CN AVERAGE TEMPS DURING THE TIME STEP.  SEE
      SPECIAL         REGULAR, NTYPE, KS, KSPEC, FOR, SPECK.
/     STEADY STATE A CONDITICN IN WHICH REMAINING TIME DEPENDENCIES ARE
/     STEADY STATE   INSIGNIFICANT COMPARED WITH A SPECIFIED ALLOWABLE ERROR.
      SURFACE       PERTAINING TO A NODE CONNECTED TO A BOUNDARY NODE, TO
      SURFACE         THE CONNECTION, OR THE AREA WHICH IS CONNECTED.
      TIME STEP     A SMALL INTERVAL OF TIME FOR WHICH A PSEUDO STEADY
      TIME STEP       STATE CALCULATION IS MADE TO DETERMINE CHANGES IN
      TIME STEP       TIME-DEPENDENT VARIABLES.
      TRANSIENT     PERTAINING TO VARIATION WITH TIME, NON-STEADY STATE.
*                                                                        **

      2.  TRUMP CONTROL CARDS.

      BLCCK         BLOCK NUMBER CARD, WITH BLOCK NUMBER (SEE IBLOCK) IN
      BLOCK           COLS. 6-7, BLOCK TYPE (SEE MOD) IN COL. 8.  COLS.
      BLCCK           9-80 MAY CONTAIN A TITLE (SEE ABLOCK).  AN INPUT
      BLCCK           UNIT OTHER THAN NR MAY BE SPECIFIED FOR THE BLOCK
      BLCCK           ITEMS (SEE *INPUTNB).
```

```
ENDED-      DATA END CARD (MUST FOLLOW LAST DATA BLOCK).  COLUMNS
ENDED-        6-7 MUST BE EITHER -1 OR -2, AND THE PROBLEM WILL
ENDED-        EITHER BE INTERRUPTED OR ENDED, RESP., IF SENSE
ENDED-        SWITCH 1 IS TURNED ON BEFORE THE PROBLEM ENDS.
ENDED-        IF INTERRUPTED, PROBLEM MAY BE CONTINUED WITH ONMON.
(BLANK)     A BLANK CARD INDICATES THE END OF A LIST OF BLOCK
(BLANK)       ITEMS IN BLOCKS 2, 3, 4, 5, 6, 7, 8, 9, 10, AND 12,
(BLANK)       AND THE END OF MATERIAL SELECTION LIST IN BLOCK 2.
(BLANK)       EXTRA BLANKS MAY BE INSERTED ANYWHERE EXCEPT WITHIN
(BLANK)       BLOCK ITEM LISTS.

*ASSIGN     REASSIGN LOGICAL I/O UNIT. IF COL 8 IS 2, 3, OR 4 A CALL
*ASSIGN       ASSIGN WILL BE EXECUTED WITH 2, 3, OR 4 ARGS.
*ASSIGN       SPECIFIED IN COLS 9-32 (1X, I2, 1X, I2, A10, 1X, I6).
*ASSIGN       IF COL 8 IS 8, CALL KEEP80 (8RTRUMPDDA) WILL BE
*ASSIGN       EXECUTED, TO GIVE THE DD80 OUTPUT FILES TO
*ASSIGN       THE USER.

*CHECK      DATA DECK PREPROCESSING CONTROL CARD (PRECEDING DATA
*CHECK        DECK).  CAUSES NEXT PROBLEM TO END AFTER THE FIRST
*CHECK        TIME STEP.  USED FOR CHECKING INPUT DATA.
*DECKS      UNUSED DATA DECKS CONTROL CARD (PRECEDING DATA DECK).
*DECKS        UNIT NUMBER (SEE NDATA) IS SPECIFIED IN COLUMNS 7-8
*DECKS        AND MUST HAVE BEEN ASSIGNED TO A TAPE, DISC, OR
*DECKS        OTHER FILE BY A *T, *MOUNT, OR *ASSIGN CARD.  USED
*DECKS        TO CAUSE ALL UNUSED DATA DECKS UP TO AN END-OF-FILE
*DECKS        MARK OR CARD WITH A $ IN COLUMN 1, TO BE COPIED
*DECKS        ONTO UNIT NDATA IF THE JOB IS INTERRUPTED BY SENSE
*DECKS        SWITCH 1.  LAST DATA DECK SHOULD BE FOLLOWED BY A
*DECKS        *INPUTNR CARD FOR USE WHEN NDATA IS USED FOR INPUT.
*DISCON     RESTART CONTROL CARD (IMMEDIATLY FOLLOWING EITHER
*DISCON       /*     LOAD(1,5,ONMONS)/ OR /*     LOAD(1,5,ONMON)/
*DISCON       MONITOR CONTROL CARD).  USED TO CAUSE THE PROBLEM,
*DISCON       IF ANY, PREVIOUSLY INTERRUPTED BY SENSE SWITCH 1,
*DISCON       AND SAVED ON THE RESTART TAPE LOADED ON UNIT 5A,
*DISCON       TO BE ENDED (FINAL OUTPUT WILL BE PRODUCED).
*FINIS      TAPE FINISH CONTROL CARD (FOLLOWING DATA DECK).  UNIT
*FINIS        IS SPECIFIED IN COLUMNS 7-8, TAPE VAULT NUMBER IN
*FINIS        COLUMNS 10-14.    USED TO HAVE TAPE ENDFILED AND
*FINIS        UNLOADED TO MAKE UNIT AVAILABLE FOR ANOTHER TAPE.
*INPUTNB    INPUT UNIT CONTROL SYMBOLS, IN COLUMNS 9-14 OF BLOCK
*INPUTNB      NUMBER CARD, WITH UNIT NUMBER FOR INPUT OF BLOCK
*INPUTNB      ITEMS (SEE NB, NR) IN COLUMNS 15-16.  USED WHEN NB
*INPUTNB      IS NOT NR.
*INPUTNR    INPUT UNIT CONTROL CARD (PRECEDING DATA DECK), WITH
*INPUTNR      UNIT NUMBER FOR INPUT OF DATA DECK (SEE NR,NRS) IN
*INPUTNR      COLUMNS 7-8.  USED WHEN NR IS NOT NRS, OR TO CHANGE
*INPUTNR      NR BACK TO NRS.  A *INPUTNR CARD SHOULD FOLLOW THE
*INPUTNR      LAST DATA DECK WHEN A *DECKS CARD IS USED, SO THAT
*INPUTNR      INPUT WILL BE CHANGED TO THE DESIRED UNIT AFTER
*INPUTNR      USING THE TAPE FROM UNIT NDATA AS AN INPUT TAPE.
*MAST       MATERIAL DATA LIST CONTROL CARD (IN BLOCK 2), TO HAVE
*MAST         DATA FOR SELECTED MATERIALS (SEE MATP, MATS) OR ALL
*MAST         DATA (SEE MALL) COPIED FROM UNIT NREAD TO UNIT
*MAST         NLIST, WITH AN OPTION TO CONVERT UNITS (SEE MUNCH),
*MAST         AND AN OPTION TO THEN USE NLIST FOR INPUT OF BLOCK
*MAST         2 ITEMS (SEE NEST).  FORMAT FOR INPUT OF *MAST,
*MAST         NREAD, NLIST, NEST, MALL, MUNCH, XD, XC, XK, XT,
*MAST         AND XTP IS A5,5I5,5E10.3.
*MOUNT      TAPE MOUNT CONTROL CARD (PRECEDING DATA DECK).  UNIT
*MOUNT        IS SPECIFIED IN COLUMNS 7-8, TAPE VAULT NUMBER IS
*MOUNT        SPECIFIED IN COLUMNS 10-14.  USED TO HAVE DIFFERENT
*MOUNT        TAPES ASSIGNED CONSECUTIVELY TO SAME UNIT.
*OFFMON     OFFMON CONTROL CARD (FOLLOWING DATA DECK).  IF READ,
*OFFMON       MEMORY AND POINTERS WILL BE SAVED ON THE TAPE ON
*OFFMON       UNIT 5A, ALLOWING RESTART OF THE PROBLEM AND USE OF
*OFFMON       DATA IN MEMORY FOR CONTINUATION PROBLEMS, BY USE OF
*OFFMON       EITHER OF THE MONITOR CONTROL CARDS (SEE MANUALS)
*OFFMON       /*     LOAD(1,5,ONMON)/, /*     LOAD(1,5,ONMONS)/.
*OUTPUT     OUTPUT UNIT CONTROL CARD (PRECEDING DATA DECK), WITH
*OUTPUT       UNIT NUMBER FOR NUMERICAL OUTPUT DATA (SEE MW, M)
*OUTPUT       IN COLUMNS 8-10.  USED WHEN M IS NOT MW, OR TO
*OUTPUT       CHANGE M BACK TO MW.  M IS USUALLY C03 OR 10C.
```

```
      *SPLIT        JCB COMPLETION CONTROL CARD (FOLLOWING DATA DECK). IF
      *SPLIT          READ, PROGRAM WILL CALL EXIT.  (REPLACES *OFFMON.)
      *(NAME)       PROBLEM NAME CARD.  COLUMN 1 MUST BE AN *.  COLUMN 72
      *(NAME)         CONTROLS USE OF DATA FROM PRECEDING PROBLEM (SEE K).
/     $             SAME AS AN END-OF-FILE MARK WHEN IN COLUMN 1 OF A
      $               CARD RECORD READ IN FROM UNIT NRS, NR, OR NB WHILE
      $               TRANSFERRING DATA IN SUBROUTINE PRIME.
*                                                                        **

      3.  VARIABLES AND PARAMETERS.

      AA(N)         INITIAL W/F REACTANT A IN NOTE(N).  BLANK = ALONE.
      A(N)          WEIGHT FRACTION REACTANT A, OR LOW-TEMPERATURE PHASE, IN
      A(N)            NODE(N).  SEE ALONE, AA(N), KA(N), KEM(N), HMELT(N).
      ABLOCK        DESCRIPTIVE INFORMATION PUNCHED ON BLOCK NUMBER CARD.
      ABLOCK          MAY INCLUDE *INPUTNB IN COL. 9-16 TO SPECIFY AN
      ABLOCK          INPUT UNIT NB FOR THE BLOCK ITEMS INSTEAD OF NR.
      ADD           DIFFERENCE BETWEEN SUCCESSIVE VALUES OF DRAD IN A
      ADD             SEQUENCE OF NODES OR CONNECTIONS PRODUCED BY NSEQ.
      AKEM(N)       DESCRIPTIVE NAME FOR CHEMICAL REACTANT KEM(N).
      ALONE         INITIAL WEIGHT FRACTION REACTANT A ASSIGNED TO ALL NODES
      ALONE           IN ANY BLOCK 4 READ IN AFTER BLOCK 1, AND SUBSTITUTED FOR
      ALONE           UNSPECIFIED AA(N) IN ANY BLOCK 9 READ IN AFTER BLOCK 1.
      AMAT(N)       DESCRIPTIVE NAME FOR MATERIAL WITH NUMBER MAT(N).
      APROP(N)      NAME OF A PROPERTY WHICH MAY HAVE REMOTE DEPENDENCE.
      AREA(N)       INTERFACE AREA BETWEEN NOD1(N) AND NOD2(N), EQUALS
      AREA(N)         (GEOM*DLONG*DRAD**KSYM)*SCALE**2.  SEE KD.
      AREAS(N)      AREA OF NODS(N) EXPOSED TO BOUNDARY NODE NODSB(N),
      AREAS(N)        EQUALS (GEOM*DLONG*DRAD**KSYM)*SCALE**2.  SEE KD.
      AX            TEMPORARY VALUE OF AA(N).
      BB(N)         INITIAL W/F REACTANT B IN NOTE(N).  BLANK = BONE.
      B(N)          WEIGHT FRACTION REACTANT B IN NODE(N).  SEE BONE, BB(N),
      B(N)            KB(N), KEM(N).
      BET           VALUE OF A TABULATED PROPERTY, USED TO FIND CHANGE.
      BETA          VALUE OF A TABULATED CHEMICAL PROPERTY OF CHEM A.
      BETB          VALUE OF A TABULATED CHEMICAL PROPERTY OF CHEM B.
/     BETW          VALUE OF W(N) BEFORE FINDING NEW VALUE FROM TABLE.
      BIG           CONSTANT, WITH VALUE 0.99999999E12.
      BONE          INITIAL WEIGHT FRACTION REACTANT B ASSIGNED TO ALL NODES
      BONE            IN ANY BLOCK 4 READ IN AFTER BLOCK 1, AND SUBSTITUTED FOR
      BONE            UNSPECIFIED BB(N) IN ANY BLOCK 9 READ IN AFTER BLOCK 1.
      BX            TEMPORARY VALUE OF BB(N).
      CA            TEMPORARY VALUE OF CLOCKA.
      CAP(N)        HEAT CAPACITY OF NODE(N) AT T(N), SUMTIM.
      CAPMS         TOTAL HEAT CAPACITY OF A MATERIAL.
      CAPS          TOTAL HEAT CAPACITY OF SYSTEM.
      CAPT(J,N)     HEAT CAPACITY OF MAT(N) AT TVARC(J,N).
      CB            TEMPORARY VALUE OF CLOCKB.
      CLOCKA        MACHINE LETTER AND DATE, IN BCD, FROM CALL CLOCK.
      CLOCKB        TIME, IN HOURS AND THOUSANDS OF HOURS, FROM CLOCK.
/     COMDATE       DATE OF LAST COMPILATION OF TRUMP PROGRAM.
      CON(N)        THERMAL CONDUCTIVITY OF NODE(N) AT T(N), SUMTIM.
      CONE          VALUE GIVEN A NON-DECIMAL BCD NUMBER NXX IN PATCH.
      CON(J,N)      THERMAL CONDUCTIVITY OF MAT(N) AT TVARK(J,N).
/     CORD(N)       REACTION ORDER OF KEM(N).  PROGRAM WILL USE 1.0 IF
      CORD(N)         FIELD IS BLANK, SO ZERO ORDER MUST BE PUNCHED 0.0.
      CORR          ENERGY CONTENT OF MATERIAL AT 0 DEG, REL TO TVARC(1).
      DA(N)         CHANGE IN W/F OF REACTANT A IN NODE(N) IN TIME STEP.
      DATE          MACHINE LETTER AND DATE OF LAST USE OF PROGRAM DECK.
      DATEX         MACHINE LETTER AND DATE OF LAST USE OF RESTART TAPE.
      DAX           CHANGE IN A(N) DUE TO PHASE CHANGE.
      DB(N)         CHANGE IN W/F OF REACTANT B IN NODE(N) IN TIME STEP.
      DDA(N)        ESTIMATED RATE OF CHANGE OF A(N) IN NEXT TIME STEP.
      DDB(N)        ESTIMATED RATE OF CHANGE OF B(N) IN NEXT TIME STEP.
      DDT(N)        ESTIMATED RATE OF CHANGE OF T(N) IN NEXT TIME STEP.
      DELF1(N)      RELATIVE DISTANCE FROM NODAL POINT OF NODF1(N) TO
      DELF1(N)        MASS FLOW INTERFACE WITH NODF2(N), CONVERTED TO WEIGHT
      DELF1(N)        FACTOR DELF2(N)/(DELF1(N) + DELF2(N)) AFTER INPUT.
      DELF2(N)      RELATIVE DISTANCE FROM NODAL POINT OF NODF2(N) TO
      DELF2(N)        MASS FLOW INTERFACE WITH NODF1(N), CONVERTED TO WEIGHT
      DELF2(N)        FACTOR DELF1(N)/(DELF1(N) + DELF2(N)) AFTER INPUT.
/     DELT          TIME STEP, CONTROLLED BY TVARY, SMALL, DELTO,
```

```
      DELT          DELTMX, TIMEP, AND TIMAX.  ALSO SEE KS, KSPEC.
/     DELTG       LAST ACCEPTED TIME STEP COMPLETED.
      DELTMX      MAXIMUM ALLOWABLE TIME STEP, NO LARGER THAN EITHER
      DELTMX        DELTO OR 2/3 OF THE SMALLEST VALUE OF SLIM(N) FOR
      DELTMX        REGULAR NODES (NTYPE(N) = 0).  SEE KSPEC,KS.
      DELTMX        ABSOLUTE LIMITS ARE FROM 1.E-10 TO 1.E12.
/     DELTO       MAXIMUM ALLOWABLE TIME STEP.  BLANK = 1.0E12.  MUST BE
      DELTO         SPECIFIED IF KSPEC IS POSITIVE.
/     DELTS       LAST TIME STEP COMPLETED.
/     DELTSS      LAST ACCEPTED TIME STEP COMPLETED BEFORE DELTS.
      DEL1(N)     AVERAGE LENGTH OF HEAT FLOW PATH FROM NODAL POINT IN
      DEL1(N)       NOD1(N) TO INTERFACE WITH NOD2(N).  SEE SCALE.
      DEL2(N)     AVERAGE LENGTH OF HEAT FLOW PATH FROM NODAL POINT IN
      DEL2(N)       NOD2(N) TO INTERFACE WITH NOD1(N).  SEE SCALE.
      DENS (N)    DENSITY OF MATERIAL WITH NUMBER MAT(N).
      DEX         TEMPORARY VALUE OF DDT(N).
      DEXA        TEMPORARY VALUE OF DDA(N).
      DEXB        TEMPORARY VALUE OF DDB(N).
      DF(N)       NET HEAT FLOW INTO NODE(N) IN TIME STEP DELT.
/     DFB(N)      HEAT FLOW INTO NODE(N) IN TIME STEP DELT.
      DFI(N)      HEAT FLOW FROM NOD2(N) TO NOD1(N) IN TIME STEP DELT.
      DFLOT       TIME FOR MASS FLOW INTO A NODE TO EQUAL THE NODE MASS.
      DFS(N)      HEAT FLOW FROM NODSB(N) TO NODS(N) IN TIME STEP DELT.
      DLONG       AVERAGE LENGTH OF A NODE IN BLOCK 4, AN INTERFACE IN
      DLONG         BLOCK 5, OR A SURFACE IN BLOCK 6.  SEE DRAD.
      DRAD        MEAN DEPTH (KD = 1), OR MEAN RADIUS (KD = 2 OR 3) OF
      DRAD          A NODE IN BLOCK 4, AN INTERFACE IN BLOCK 5, OR A
      DRAD          SURFACE IN BLOCK 6.  SEE NSEQ FOR USE OF NEG VALUE.
      DRAD          ALSO SEE VOL(N), AREA(N), AREAS(N), AND SCALE.
      DRADS       VALUE OF DRAD FOR NODE OR CONNECTION LAST READ IN.
      DS          SUM OF MASS FLOW PATHS DELF1(N) AND DELF2(N).
      DT(N)       CHANGE IN TEMPERATURE IN NODE(N) IN TIME STEP DELT.
/     DTEMP       MAXIMUM TEMPERATURE CHANGE IN TIME STEP DELTS.
/     DTMAX       MAXIMUM CHANGE IN ANY TABULATED PROPERTY, IN EQUIVALENT
      DTMAX         TEMPERATURE OR TVARY*(PERCENT CHANGE), OR NUTMAX*TVARY,
      DTMAX         IF NUTMAX ITERATIONS WERE DONE, OR 0.025*NUTS*TVARY,
      DTMAX         WHICHEVER IS LARGEST, IN TIME STEP DELTS.
/     DTMAXS      VALUE OF DTMAX FOR TIME STEP DELTS.
      DTX         DIFFERENCE BETWEEN NODE TEMPERATURE AND MELTING TEMP.
      DT12        POSITIVE TEMPERATURE DIFFERENCE BETWEEN TWO NODES.
      DWIDE       AVERAGE WIDTH OF A NODE IN BLOCK 4.  SEE DRAD.
      DX1         FACTOR TO POSITION NODE NUMBER ON TEMP VS TIME PLOT.
      DX2         FACTOR TO POSITION NODE NUMBER ON TEMP VS TIME PLOT.
      DY1         FACTOR TO POSITION NODE NUMBER ON TEMP VS TIME PLOT.
      DY2         FACTOR TO POSITION NODE NUMBER ON TEMP VS TIME PLOT.
      D1          TEMPORARY VALUE OF DELF1(N).
      D2          TEMPORARY VALUE OF DELF2(N).
      EA(N)       ACTIVATION ENERGY OF REACTANT A IN NODE(N).
      EB(N)       ACTIVATION ENERGY OF REACTANT B IN NODE(N).
/     EMAX        LARGEST CHANGE IN THE CALCULATED TEMPERATURE CHANGE OF ANY
      EMAX          FINITE-VOLUME SPECIAL NODE IN ONE ITERATION.  MUST BE
      EMAX          LESS THAN 10.0*ERRS*TVARY FOR CONVERGENCE.
      EOF         INDICATES AN END-OF-FILE HAS BEEN READ, IF POSITIVE.
      ERRCR(N)    NEW ESTIMATED CHANGE IN HEAT FLOW INTO NODE(N).
      ERRCRX(N)   LAST ESTIMATED CHANGE IN HEAT FLOW INTO NODE(N).
/     ERRS        MAXIMUM ALLOWABLE FRACTION OF TVARY FOR THE CHANGE IN THE
      ERRS          CALCULATED AVERAGE TEMPERATURE CHANGE OF INTER-CONNECTED
      ERRS          SPECIAL NODES IN ONE ITERATION, FOR CONVERGENCE.
/     ESUM        CHANGE IN THE CALCULATED TOTAL HEAT CONTENT OF INTER-
      ESUM          CONNECTED SPECIAL NODES IN ONE ITERATION.
      ET(J,N)     ACTIVATION ENERGY OF KEM(N) AT TVARE(J,N).
/     EX          EXTRAPOLATION OF TEMP OR TIME FOR TABLE LOOK-UP,
      EX            OR A CORRECTION TO THE TEMPERATURE CHANGE OF A
      EX            SPECIAL NODE FOR ONE ITERATION STEP.
      F(N)        TOTAL HEAT FLOW INTO NODE(N) AT SUMTIM.
      FB(N)       TOTAL HEAT ADDED TO SYSTEM FROM BOUNDARY NODE NODB(N).
      FI(N)       TOTAL HEAT FLOW FROM NOD2(N) TO NOD1(N) AT SUMTIM.
/     FIN         DUMMY LIST ARGUMENT USED FOR WRITE STATEMENTS WITH NO
      FIN           LIST, TO AVOID FIN STATEMENT REQUIRED IN LRLTRAN
      FIN           BUT NOT ALLOWED IN CHAT.
      FINIS       LARGEST TIME TO BE PLOTTED ON CRT PLOT TIME SCALES.
      FLAPS(N)    TOTAL MASS FLOW FROM NODF1(N) TO NODF2(N) AT SUMTIM.
      FLEX        FINAL CORRECTION TO HEAT FLUX BETWEEN NODES (SPECK),
```

```
      FLEX          OR REACTANT FLOW BETWEEN NODES (FLOW).
      FLEX1        HEAT CONTENT OF MASS FLOW SOURCE NODE.
      FLEX2        HEAT CONTENT OF MASS FLOW SINK NODE.
      FLINT(N)     NET MASS FLOW RATE INTO NODE(N) DURING TIME STEP.
      FLIPS(N)     NET TOTAL MASS FLOW INTO NODE(N) UP TO SUMTIM.
      FLOPS(N)     NET TOTAL MASS FLOW OUT OF NODE(N) UP TO SUMTIM.
      FLOUT(N)     NET MASS FLOW RATE OUT OF NODE(N) DURING TIME STEP.
      FLOW         NET AMOUNT OF HEAT FLOW INTO NODE OR SYSTEM.
/     FLOWN(N)     MASS FLOW RATE FROM NODF1(N) TO NODF2(N).  MAY BE NEGATIVE
      FLOWN(N)       TO INDICATE FLOW IN OPPOSITE DIRECTION.  IF NOT SPECIFIED,
      FLOWN(N)       FONE WILL BE USED BY PROGRAM.
/     FLOWT(J,N)   MASS FLOW RATE FROM NODF1(N) TO NODF2(N) AT TVARFL(J,N).
      FLOWT(J,N)     MAY BE NEGATIVE TO INDICATE FLOW IN OPPOSITE DIRECTION.
      FLOWX        MASS FLOW FROM NODF1(N) TO NODF2(N) DURING TIME STEP.
      FLUX         NET TOTAL HEAT FLOW INTO ALL NODES IN SYSTEM.
      FLUXS        NET TOTAL HEAT FLOW INTO SYSTEM FROM BOUNDARY NODES.
/     FLUXSS       HEAT FLOW INTO SYSTEM FROM BOUNDARY NODES
      FLUXSS         DURING PRECEDING TIME STEP.
      FONE         FLOW RATE FOR ANY MASS FLOW CONNECTIONS WITH FLOWN(N)
      FONE           BLANK IN ANY BLOCK 10 READ IN AFTER BLOCK 1.
/     FOR          INTERPOLATION FACTOR BETWEEN TEMPERATURES AT THE BEGINNING
      FOR            AND END OF THE TIME STEP.  FOR IS SET TO 1.0 FOR THE
      FOR            FIRST TIME STEP AND AFTER REJECTED TIME STEPS.  FOR OTHER
      FOR            TIME STEPS, FOR IS 1.0 IF KSPEC IS 2, 0.5 IF KSPEC IS 3,
      FOR            OR BETWEEN 0.57 AND 1.0, DEPENDING ON RAST.
      FORD         FACTOR FOR ESTIM. CHANGES, = FOR*DELT OR 0.5*DELT.
      FRAC1        CONSTANT IN EXPRESSION USED TO CONVERT PROBLEM NODE
      FRAC1          NUMBERS (KD = 1) OR NODE RADII (KD = 2 OR 3) TO CRT
      FRAC1          PLOT NODE LOCATION SCALES, AS FOLLOWS...
      FRAC1          RPLOT (OR PRAD) = FRAD1 + FRAD2*NODE (OR RADIUS).
      FRAC2        SEE FRAD1.  IF NOT SPECIFIED, PROGRAM USES 1.0.
      FS(N)        TOTAL HEAT FLOW FROM NODSB(N) TO NODS(N) AT SUMTIM.
      FTEMP1       CONSTANT IN EXPRESSION USED TO CONVERT PROBLEM TEMPERATURES
      FTEMP1         TO CRT PLOT SCALE TEMPERATURES, AS FOLLOWS...
      FTEMP1         TPLOT (OR PTEMP) = FTEMP1 + FTEMP2*T
      FTEMP2       SEE FTEMP1.  IF NOT SPECIFIED, PROGRAM USES 1.0.
      FTIME1       CONSTANT IN EXPRESSION USED TO CONVERT PROBLEM TIMES
      FTIME1         TO CRT PLOT TIMES, AS FOLLOWS...
      FTIME1         PTIME = FTIME1 + FTIME2*SUMTIM.
      FTIME2       SEE FTIME1.  IF NOT SPECIFIED, PROGRAM USES 1.0.
      FX           AVERAGE RATE OF HEAT FLOW ACROSS A CONNECTION.
      FX1          AVERAGE MASS FLOW RATE INTO NODE(N) UP TO SUMTIM.
      FX2          AVERAGE MASS FLOW RATE OUT OF NODE(N) UP TO SUMTIM.
/     FXX          HEAT FLOW RATE ACROSS A THERMAL CONNECTION
      FXX            DURING PRECEDING TIME STEP.
      F1(N)        TEMPORARY NAME OF FIRST ITEM OF INPUT TABLE PAIR.
      F2(N)        TEMPORARY NAME OF SECOND ITEM OF INPUT TABLE PAIR.
      F3(N)        TEMPORARY NAME OF SLOPE BETWEEN INPUT TABLE PAIRS.
      GG(N)        INITIAL HEAT GEN RATE IN NOTE(N).  BLANK = GONE.
      G(N)         HEAT GEN. RATE IN NODE(N).  SEE GONE, GG(N), GT(J,N).
      GENS         NET TOTAL AMOUNT OF HEAT GENERATED IN SYSTEM.
      GEOM         GEOMETRIC FACTOR, 1.0 FOR KD = 1, 2.0*PI FOR KD = 2,
      GEOM           AND 4.0*PI FOR KD = 3 (PI = 3.14159265).
      GONE         CONSTANT HEAT GENERATION RATE ASSIGNED TO ALL NODES IN ANY
      GONE           BLOCK 4 READ IN AFTER BLOCK 1, AND SUBSTITUTED FOR ANY
      GONE           UNSPECIFIED GG(N) IN ANY BLOCK 9 READ IN AFTER BLOCK 1.
      GONE           WILL NOT BE USED FOR NODES LISTED IN BLOCK 8.
      GS           NET TOTAL HEAT GENERATION RATE IN SYSTEM.
      GT(J,N)      HEAT GENERATION RATE IN NODG(N) AT TVARG(J,N).  IF
      GT(J,N)        LTABG(N) IS -1, 0, OR 1, GT IS THE GEN RATE AT TIME
      GT(J,N)        ZERO, DECAYING WITH HALF-LIFE TVARG(1,N).
      GX           TEMPORARY VALUE OF GG(N).
      H(N)         HEAT ADDED TO NODE(N) UP TO SUMTIM.
      HEAT         SENSIBLE HEAT CONTENT OF SYSTEM AT SUMTIM.
      HEFT(N)      MASS OF NODE(N).
      HEFTMS       TOTAL MASS OF A MATERIAL.
      HEFTS        TOTAL MASS OF SYSTEM.
      HEX          HEAT TRANSFERRED BETWEEN NODES IN TIME DELT.
      HINT(N)      INTERFACE CONDUCTANCE BETWEEN NODES NOD1(N) AND NOD2(N).
      HINT(N)        BLANK = 1.0E12 IF RINT IS BLANK, 1.0E-24 IF RINT NOT BLANK.
      HINTS        INPUT VALUE OF HINT(N).
      HMELT(N)     LATENT HEAT ABSORBED AT TMELT(N) BY MATERIAL MAT(N),
      HMELT(N)       SET TO ZERO IF TMELT(N) NOT BETWEEN TMIN AND TMAX.
      HMELTX(N)    INPUT VALUE OF HMELT(N).
      HMX          TEMPERATURE CHANGE EQUIVALENT TO LATENT HEAT EFFECT.
      HONE         SURFACE CONDUCTANCE FOR ANY EXTERNAL CONNECTIONS WITH
```

```
      HONE          HSURE(N) BLANK, IN A BLOCK 6 READ IN AFTER BLOCK 1.
      HOUR        TIME, IN HOURS, OF LAST USE OF PROGRAM DECK.
      HOURX       TIME, IN HOURS, OF LAST USE OF RESTART TAPE.
      HSUM        TOTAL HEAT CAPACITY OF ALL SPECIAL NODES WHICH HAVE
      HSUM          INTERNAL CONNECTIONS WITH OTHER SPECIAL NODES.
      HSURE(N)    SURFACE CONDUCTANCE BETWEEN NODS(N) AND NODSB(N).
      HSURT(J,N)  SURF. COND. OF NODS(N) AT TVARH(J,N).  BLANK = HONE.
      HX          TEMPORARY VALUE OF HSURE(N).
      I           ARRAY SUBSCRIPT.
      IBLOCK      BLOCK NUMBER, 1 FOR CONTROLS, LIMITS, AND CONSTANTS,
      IBLOCK        2 FOR MATERIALS, 3 FOR REACTANTS, 4 FOR NODES,
      IBLOCK        5 FOR INTERNAL THERMAL CONNECTIONS, 6 FOR EXTERNAL
      IBLOCK        THERMAL CONNECTIONS, 7 FOR BOUNDARY NODES,
      IBLOCK        8 FOR VARIABLE HEAT GENERATION NODES, 9 FOR INITIAL
      IBLOCK        CONDITIONS, 10 FOR MASS FLOW CONNECTIONS,
      IBLOCK        11 FOR GRAPHICAL OUTPUT CONTROLS AND NODES,
      IBLOCK        12 FOR NODES WITH REMOTE PROPERTY DEPENDENCE.
/     IBLOCL      VALUE OF IBLOCK ON DATA END CARD.
/     IER         INDICATES ERROR OCCURRED IN OOODROP, IF POSITIVE.
/     INFRA       INDICATES, IF 1, THAT NORMAL VALUE OF RPLOT(N) WILL BE
      INFRA         MULTIPLIED BY 0.5*SUMTIM**(-0.5).  SEE LOGR.
      INTERP      INDICATES TEMP VS NODE LOCATION PLOT WILL HAVE
      INTERP        INTERPOLATION BETWEEN POINTS, IF NONZERO.
/     INT1-INT2   INTEGERS IN COL. 24 OR 29, AND 25 OR 30, RESP.,
      INT1-INT2     FOUND ON RECORD ON MASTER MATERIAL DATA LIST.
/     IOCHECK     INDICATES A PARITY ERROR READING DATA, IF POSITIVE.
/     IPRINT      INDICATES THAT DATA PRINTOUTS WILL BE MADE WHENEVER
      IPRINT        THE NUMBER OF TIME STEPS IS EVENLY DIVISIBLE BY
      IPRINT        IPRINT, IN ADDITION TO THE FIRST, SECOND, AND LAST
      IPRINT        TIME STEPS, AND OTHER TIME STEPS ACCORDING TO
      IPRINT        TIMEP.  SEE KDATA, NUM, IRITE, ITAPE, TIMEP, JPIC,
      IPRINT        MW, M, AND *OUTPUT.
/     IRITE       INDICATES DATA WILL BE WRITTEN ON UNIT ITAPE WHENEVER
      IRITE         KCYC IS EVENLY DIVISIBLE BY IRITE, IN ADDITION TO
      IRITE         THE FIRST, SECOND, AND LAST TIME STEPS.
/     ITAPE       OUTPUT UNIT ON WHICH TIME AND NODE NUMBERS AND
      ITAPE         TEMPERATURES WILL BE WRITTEN  (SEE IRITE).  MUST BE
      ITAPE         SPECIFIED BY A *T OR *MOUNT CONTROL CARD PRECEDING
      ITAPE         THE DATA DECK, AND ADDITIONAL UNITS USED IN THE
      ITAPE         SAME RUN CONTROLLED BY *MOUNT AND *FINIS CARDS.
      ITEMS(N)    NUMBER OF CROSS-REFERENCED ITEMS IN DATA BLOCK N.
      ITHERM      INDICATES NODE LOCATION VS TIME ISOTHERM PLOT
      ITHERM        WILL BE MADE, IF NON-ZERO.
      J           ARRAY SUBSCRIPT.
/     JPIC        INDICATES THAT POINTS WILL BE SAVED FOR THE FINAL PLOT
      JPIC          OF TEMPERATURE VS TIME (IF NUP1 IS POSITIVE), AND PLOTS
      JPIC          OF TEMPERATURE VS NODE LOCATION WILL BE MADE (IF JPIC IS
      JPIC          NEGATIVE), WHENEVER THE NUMBER OF TIME STEPS IS EVENLY
      JPIC          DIVISIBLE BY JPIC, IN ADDITION TO THE FIRST, SECOND, AND
      JPIC          LAST TIME STEPS, AND TIME STEPS DETERMINED BY TIMEP.
      J1          TEMPORARY VALUE OF NODMAT(NOD1(N)) OR MAT(N).
      J2          TEMPORARY VALUE OF NODMAT(NOD2(N)).
/     K           IN COLUMN 72 OF PROBLEM NAME CARD, IF 2, 3, OR 4,
      K             INDICATES NEW PROBLEM WILL USE ALL DATA FROM THE
      K             PRECEDING PROBLEM, INCL. EITHER THE INITIAL (K = 2)
      K             OR FINAL (K = 3 OR 4) TEMPERATURES AND REACTANT
      K             CONCENTRATIONS, AND START AT EITHER THE INITIAL
      K             (K = 2 OR 3) OR FINAL (K = 4) PROBLEM TIME.
      K             (ALSO USED FOR ARRAY SUBSCRIPTS, FLAGS IN SEEK1 AND
      K             SEEK2, TEMPORARY VALUES OF SUBSCRIPTED INTEGERS.)
      KA(N)       IDENTIFICATION NUMBER OF REACTANT A IN MAT(N).  WILL NOT
      KA(N)         BE USED IF HMELT(N) IS NOT ZERO.
      KAX(N)      INPUT VALUE OF KA(N).
/     KB(N)       IDENTIFICATION NUMBER OF REACTANT B IN MAT(N).  MAY BE USED
      KB(N)         WHETHER KA(N) IS ZERO OR NOT, OR WHEN HMELT(N) IS NOT ZERO.
/     KBLR        CONTROL SENTINEL IN BUFFERING IN UNUSED DATA DECKS.
      KBX(N)      INPUT VALUE OF KB(N).
/     KCHAR(N)    CHARACTER IN A CARD RECORD BUFFERED IN FROM MASTER
      KCHAR(N)      MATERIAL PROPERTIES TAPE IN PRIME.
/     KCOPY       INDICATES NUMBER OF CARD RECORDS TO BE COPIED FROM MASTER
      KCOPY         MATERIAL PROPERTIES TAPE TO SECONDARY TAPE.
```

```
/     KCYC        INDICATES PROBLEM IS IN DATA INPUT PHASE (IF -1), OR IN
      KCYC          INITIALIZATION PHASE, INCLUDING INITIAL TIME STEP OF
      KCYC          1.E-12 (IF 0), OR THE NUMBER OF TIME STEPS COMPLETED.
/     KD          GEOMETRIC SYMMETRY INDICATOR, 1 FOR NONSYMMETRIC,
      KD            2 FOR AXISYMMETRIC, 3 FOR CENTRISYMMETRIC.  USED TO
      KD            CONTROL CALCULATION OF NODE VOLUMES AND CONNECTION
      KD            AREAS FOR BLOCKS 4, 5, AND 6 READ IN AFTER BLOCK 1.
      KD            (MORE THAN ONE BLOCK 1 MAY BE USED, TO CHANGE KD.)
      KD            SEE GEOM, KSYM, SCALE, VOL(N), AREA(N), AREAS(N).
/     KDATA       CONTROLS AMOUNT OF DATA ON PRINTOUTS, MINIMUM IF
      KDATA         NEGATIVE, NORMAL IF ZERO, AND MAXIMUM IF POSITIVE.
      KDATA         MAXIMUM DATA IS ALWAYS PRODUCED ON THE FIRST
      KDATA         AND LAST TIME STEPS.  SEE IPRINT, NUM, JPIC, TIMEP.
      KDATAX      INPUT VALUE OF KDATA.
/     KDR         OCTAL EQUIVALENT OF CHARACTER $.
      KEM(N)      IDENTIFICATION NUMBER OF REACTANT WITH INDEX N.
      KGOOD       NUMBER OF SUCCESSIVE TIME STEPS NOT REPEATED, OR
      KGOOD         NOT INVOLVING CHANGES TO SPECIAL NODES, AFTER 2ND.
/     KITTEN      PARAMETER CONTROLLING COMPILATION OF SUBROUTINE KITTY
      KITTEN        FOR CONTROLLEE VERSION, 0 TO SKIP, 1 TO COMPILE.
      KINT        INTERVAL BETWEEN POINTS TO BE LABELED ON CRT PLOTS.
/     KLOOGE      INTEGER CONSTANT INDICATING COMPUTER COMPILING
      KLOOGE        PROGRAM, I.E., 3600, 6600, OR 7600.
/     KMAT        INDEX OF MATERIAL IN LIST OF MATP, MATS.
      KNOCK       NO. OF THERMAL CONN. BETWEEN PAIRS OF SPECIAL NODES.
      KNOFL       NO. OF MASS FLOW CONN. BETWEEN PAIRS OF SPECIAL NODES.
/     KPLOTL      PARAMETER CONTROLLING COMPILATION OF PLOT STATEMENTS
      KPLOTL        FOR MAKING ISOTHERM PLOTS, 1 TO COMPILE, 0 TO SKIP.
      KPLOTL        (KPLOTT MUST ALSO BE 1 TO COMPILE.)
/     KPLOTR      PARAMETER CONTROLLING COMPILATION OF PLOT STATEMENTS
      KPLOTR        FOR MAKING TEMPERATURE VS RADIUS PLOTS.  1 TO
      KPLOTR        COMPILE, 0 TO SKIP.
/     KPLOTT      PARAMETER CONTROLLING COMPILATION OF PLOT STATEMENTS
      KPLOTT        FOR MAKING TEMPERATURE VS TIME AND ISOTHERM PLOTS.
      KPLOTT        1 TO COMPILE, 0 TO SKIP.
/     KPRIME      PARAMETER CONTROLLING COMPILATION OF EXECUTABLE
      KPRIME        STATEMENTS IN PRIME.  1 TO COMPILE, 0 TO SKIP.
/     KR          INDICATES NODE LOCATION SCALE ON CRT PLOT IS LINEAR IF 2,
      KR            OR AS SPECIFIED IN INPUT DATA IF 1 (SEE LOGR).
      KS          INDICATES THAT NODE(N) IS A SPECIAL NODE.  SEE KSPEC.
/     KSEC        CLOCK TIME IN SECONDS, FROM CALL CLOCK1 (360C ONLY).
/     KSECS       MACHINE TIME CHARGED TO THE PROBLEM, IN SECONDS,
      KSECS         MEASURED FROM JUST BEFORE READING THE PROBLEM
      KSECS         NAME CARD FOR EACH DATA DECK.
/     KSECZ       CLOCK TIME IN SECONDS, FROM LAST CLOCK1 CALL.
/     KSPEC       INDICATES REGULAR NODES WILL BE CHANGED TO SPECIAL
      KSPEC         NODES AS NEEDED TO ALLOW LARGER TIME STEPS (0),
      KSPEC         BEFORE FIRST TIME STEP (1, 2, OR 3), OR NOT AT
      KSPEC         ALL, AND STEADY STATE CRITERIA CANNOT END PROBLEM
      KSPEC         (-1).  CAN FIX INTERPOLATION FACTOR (FOR) AT 1.0
      KSPEC         (2) OR AT 0.5 (3).  NORMAL VALUE OF KSPEC IS ZERO.
      KSPEC       DELTO MUST BE SPECIFIED IF KSPEC IS POSITIVE.
      KSYM        SYMMETRY FACTOR, 1 IF KD IS 1 OR 2, 2 IF KD IS 3.
      KT          INDICATES TEMP SCALE IS C, K, F, OR R, IF 1, 2, 3, 4.
      KTEM        INDICATES TEMPERATURE SCALE ON CRT PLOT IS LINEAR IF 2, OR
      KTEM          AS SPECIFIED IN INPUT DATA IF 1 (SEE LOGTEM).
      KTIM        INDICATES TIME SCALE ON CRT PLOT IS LINEAR IF 2, OR AS
      KTIM          SPECIFIED IN INPUT DATA IF 1 (SEE LOGTIM).
/     KWIT        INDICATES TIME STEP TO BE REPEATED (-1), SUMTIM EXCEEDS
      KWIT          TIMAX (1), TEMP OVER TMAX (2), OR UNDER TMIN (3),
      KWIT          STEADY STATE REACHED (4), INPUT ERROR (5),
      KWIT          *DISCON CARD READ IN (6), KCYC OVER MCYC (7), KSECS
      KWIT          OVER MSEC (8), MISSING BLOCK 2 OR 4 (9), ITERATION
      KWIT          CONVERGENCE FAILURE (10), NUMBER OF BLOCK ITEMS TOO
      KWIT          LARGE (11), OR TABLE LENGTH OVER MS (12).
      K1          TEMPORARY VALUE OF NLOOK(N,K) OR KA(N).
      K2          TEMPORARY VALUE OF NLOOK(N,K) OR KB(N).
/     K36ST       PARAMETER CONTROLLING CHOICE OF OTHER PARAMETERS WHEN
      K36ST         COMPILING.  1 FOR 3600/STD VERSION, 0 FOR OTHERS.
/     K66DS       PARAMETER CONTROLLING CHOICE OF OTHER PARAMETERS WHEN
      K66DS         COMPILING.  1 FOR 6600/DS VERSION, 0 FOR OTHERS.
/     K66ST       PARAMETER CONTROLLING CHOICE OF OTHER PARAMETERS WHEN
```

```
      K66ST          COMPILING.  1 FOR 6600/STU VERSION, 0 FOR OTHERS.
      L            TEMPORARY VALUE OF NLOOK(N,K).
      LA           TEMPORARY VALUE OF KA(NODMAT(N)).
      LABEL        INDICATES TABLE HEADINGS TO BE REWRITTEN, IF NONZERO.
      LAGTEM       TEMP SCALE INDICATOR, EQUALS LOGTEM OR ZERO.
      LAGTIM       TIME SCALE INDICATOR, EQUALS LOGTIM OR ZERO.
/     LASR         INDICATES NODE LOCATION SCALE IS LINEAR IF 0, LOG IF 1.
      LASTEM       INDICATES TEMPERATURE SCALE IS LINEAR IF 0, LOG IF 1.
      LASTIM       INDICATES TIME SCALE IS LINEAR IF 0, LOG IF 1.
      LB           TEMPORARY VALUE OF KB(NODMAT(N)).
      LBH          INDICATES INPUT DATA FIELD FOR HINT(N) BLANK IF ZERO.
      LBR          INDICATES INPUT DATA FIELD FOR RINT(N) BLANK IF ZERO.
      LBX          INDICATES INPUT DATA FIELD FOR NXX BLANK IF ZERO.
      LIST         INPUT ARRAY WHICH MUST BE RECALCULATED IN SUBR REFER.
      LISTR        ARRAY CONTAINING NUMBERS TO WHICH ARRAY LIST REFERS.
      LISTX        INPUT VALUES OF ARRAY LIST.
/     LOGR         MAXIMUM NUMBER OF LOG CYCLES ON NODE LOCATION SCALES ON CRT
      LOGR           PLOTS.  LINEAR SCALES WILL BE USED IF 0, BOTH LOG AND
      LOGR           LINEAR IF PRECEDED BY A MINUS SIGN.  ADD 100 TO GET
      LOGR           INTERPOLATION BETWEEN POINTS ON TEMPERATURE VS NODE
      LOGR           LOCATION PLOT.  ALSO SEE FRAD1, FRAD2.
      LOGR           ADD 1000 TO MULTIPLY NODE LOCATION SCALE BY THE FACTOR
      LOGR           0.5*SUMTIM**(-0.5).
      LOGTEM       MAXIMUM NUMBER OF LOG CYCLES ON TEMPERATURE SCALES ON CRT
      LOGTEM         PLOTS.  LINEAR SCALES WILL BE USED IF 0, BOTH LOG AND
      LOGTEM         LINEAR IF PRECEDED BY A MINUS SIGN.  ADD 100 TO DELETE
      LOGTEM         GRID LINES FROM ALL CRT PLOTS.  ALSO SEE FTEMP1, FTEMP2.
      LOGTIM       MAXIMUM NUMBER OF LOG CYCLES ON TIME SCALES ON CRT PLOTS.
      LOGTIM         LINEAR SCALES WILL BE USED IF 0, BOTH LOG AND LINEAR IF
      LOGTIM         PRECEDED BY A MINUS SIGN.  ADD 100 TO OBTAIN CRT PLOT OF
      LOGTIM         NODE LOCATION VS TIME (ISOTHERM PLOT) FOR NODES IN LIST
      LOGTIM         NODEP(N).  ALSO SEE FTIME1, FTIME2.
      LOSR         INDICATES NODE LOCATION SCALE LINEAR (0) OR LOG(1).
      LOSTEM       INDICATES CRT TEMP SCALE LINEAR (0) OR LOG (1).
      LOSTIM       INDICATES CRT TIME SCALE LINEAR (0) OR LOG (1).
      LTAB         ABSOLUTE VALUE OF A TABLE LENGTH.
      LTABC(N)     LENGTH OF TABLE OF CAPT(J,N) VS TVARC(J,N) FOR
      LTABC(N)       MAT(N), IS NEG FOR TIME TABLE, POS FOR TEMP TABLE.
      LTABE(N)     LENGTH OF TABLE OF ET(J,N) VS TVARE(J,N) FOR KEM(N),
      LTABE(N)       IS NEG FOR TIME TABLE, POS FOR TEMP TABLE.
      LTABFL(N)    LENGTH OF TABLE,  FLOWT(J,N) VS TVARFL(J,N), NODF1(N)
      LTABFL(N)      TO NODF2(N), NEG FOR TIME TABLE, POS FOR TEMP TABLE.
      LTABG(N)     LENGTH OF TABLE OF GT(J,N) VS TVARG(J,N) FOR NODG(N),
      LTABG(N)       IS NEG FOR TIME TABLE, POS FOR TEMP TABLE, -1, 0,
      LTABG(N)       OR 1 FOR EXPONENTIALLY DECAYING HEAT GENERATION
      LTABG(N)       WITH RATE GT(1,N) AT TIME 0, HALF-LIFE TVARG(1,N).
      LTABH(N)     LENGTH OF TABLE OF HSURT(J,N) VS TVARH(J,N) FOR
      LTABH(N)       NODS(N), IS NEG FOR TIME TABLE, POS FOR TEMP TABLE.
      LTABK(N)     LENGTH OF TABLE OF CONT(J,N) VS TVARK(J,N) FOR
      LTABK(N)       MAT(N), IS NEG FOR TIME TABLE, POS FOR TEMP TABLE.
      LTABQ(N)     LENGTH OF TABLE OF QT(J,N) VS TVARQ(J,N) FOR KEM(N),
      LTABQ(N)       IS NEG FOR TIME TABLE, POS FOR TEMP TABLE.
      LTABT(N)     LENGTH OF TABLE OF TEMPB(J,N) VS TIMEB(J,N), NODB(N).
      LTABT(N)       MAKE 100 TO GET SINUSOIDAL VARIATION OF TB(N)
      LTABT(N)       AROUND TEMPB(1,N) WITH AMPLITUDE TEMPB(2,N), PERIOD
      LTABT(N)       TIMEB(1,N), AND PHASE ADVANCE TIME TIMEB(2,N).
      LTABZ(N)     LENGTH OF TABLE OF ZT(J,N) VS TVARZ(J,N) FOR KEM(N),
      LTABZ(N)       IS NEG FOR TIME TABLE, POS FOR TEMP TABLE.
      LXX          INDICATES INPUT DATA FIELD FOR NXX BLANK IF ZERO.
      L1 - L3      TEMPORARY VALUE OF A TABLE LENGTH.
      M            UNIT NUMBER FOR OUTPUT OF CALCULATIONAL RESULTS.  INITIALLY
      M              MW.  MAY BE CHANGED BY SPECIFYING IN COLUMNS 8-10 OF
      M              *OUTPUT CARD PRECEDING DATA DECK.
/     MACHINE      PARAMETER SET TO ZERO TO COMPILE ON THE 3600, SET TO
      MACHINE        1 TO COMPILE ON THE 6600.  TESTED BY DIF STATEMENTS
      MACHINE        FOR COMPILATION OF MACHINE-SPECIFIC STATEMENTS.
/     MALL         INDICATES, IF 1 OR MORE, THAT ALL DATA ON UNIT NREAD
      MALL           (MASTER MATERIAL DATA LIST) WILL BE WRITTEN ON
      MALL           UNIT NLIST (SECONDARY MATERIAL LIST) AFTER ANY UNIT
      MALL           CONVERSION REQUIRED (SEE MUNCH).
/     MALT         INDICATES ALL RECORDS IN FIRST FILE ON NREAD WILL BE
      MALT           WRITTEN ON NLIST WITHOUT CONVERSION, IF POSITIVE.
```

```
      MAR            SIZE OF ARRAY LISTR IN SUBROUTINE REFER.
/     MASK1-MASK6    LITERALS USED TO MASK OUT VARIOUS PARTS OF 8- AND 10-
      MASK1-MASK6      CHARACTER WORDS IN PRIME, TO COMPARE CHARACTERS.
      MAT(N)         ICENTIFICATION NUMBER OF MATERIAL WITH INDEX N.
/     MATCH          NUMBER OF MATERIALS FOR WHICH DATA HAS BEEN
      MATCH            COPIED FROM THE MASTER MATERIALS DATA LIST TO THE
      MATCH            SECONDARY DATA LIST, WITH OPTIONAL UNIT CONVERSION.
/     MATP(N)        MASTER ICENTIFICATION NUMBER OF A MATERIAL ON THE
      MATP(N)          MASTER MATERIALS DATA LIST ON UNIT NREAD, FOR WHICH
      MATP(N)          DATA IS TO BE WRITTEN ON THE SECONDARY MATERIALS
      MATP(N)          DATA LIST ON UNIT NLIST.
/     MATS(N)        ICENTIFICATION NUMBER TO BE SUBSTITUTED FOR THE
      MATS(N)          MASTER IDENTIFICATION NUMBER WHEN DATA IS WRITTEN
      MATS(N)          ON THE SECONDARY MATERIALS DATA LIST ON UNIT NLIST.
      MATS(N)          IF BLANK OR ZERO, MASTER NUMBER WILL BE USED.
      MAX            SIZE OF ARRAYS LIST AND LISTX IN SUBROUTINE REFER.
      MAX            HIGHEST INDEX OF PART OF TABLE BEING SEARCHED.
/     MCYC           MAXIMUM ALLOWED NUMBER OF TIME STEPS (LIMIT ON KCYC).
      MCYC             NEGATIVE WILL STOP PROBLEM AT END OF FIRST TIME
      MCYC             STEP, ZERO OR BLANK IS IGNORED.
/     MF             TOTAL NUMBER OF TIMES DTEMP OR DTMAX HAS EXCEEDED TVARY
      MF               IN TIME STEPS FOR WHICH DELT WAS EQUAL TO SMALL.
/     MICS           MACHINE TIME CHARGED TO THE PROBLEM SINCE THE LAST
      MICS             CALL OF TIMER, IN SECONDS.  (ON 66CO, OBTAINED IN
      MICS             MICROSECONDS FROM TICHEK IN TIMER, IMMEDIATLY
      MICS             CONVERTED TO SECONDS.)
      MID            CENTRAL INDEX OF PART OF TABLE BEING SEARCHED.
      MIN            LOWEST INDEX OF PART OF TABLE BEING SEARCHED.
      MOD            CHARACTER IN COLUMN 8 OF BLOCK NUMBER CARD,
      MOD              BLANK FOR DELETING OLD ITEMS AND ADDING NEW ITEMS,
      MOD              /A/ FOR SAVING OLD ITEMS AND ADDING NEW ITEMS,
      MOD              /B/ FOR MODIFYING OLD ITEMS AND ADDING NEW ITEMS.
      MODS           INDICATES MOD IS A (+1), B(-1), OR NEITHER (0).
/     MS             NUMBER OF TIME STEPS FOR WHICH DELT WAS EQUAL TO DELTMX
      MS               AND DTEMP AND DTMAX WERE LESS THAN TVARY, SINCE LAST
      MS               REGULAR NODE WAS RECLASSIFIED AS A SPECIAL NODE.
/     MSEC           MAXIMUM ALLOWED USE OF MACHINE TIME, IN SECONDS.
      MSEC             NEGATIVE WILL STOP PROBLEM AT END OF FIRST TIME
      MSEC             STEP, ZERO OR BLANK IS IGNORED.
/     MSS            TOTAL NUMBER OF TIMES DTEMP AND DTMAX WERE LESS THAN TVARY
      MSS              IN TIME STEPS FOR WHICH DELT WAS EQUAL TO DELTMX.
/     MUNCH          INDICATES UNITS OF MASTER MATERIALS DATA LIST ON UNIT
      MUNCH            NREAD WILL BE CONVERTED FROM CM-G-SEC-CAL-DEG C TO
      MUNCH            M-KG-SEC-JOULE-DEG K (2), FT-LB-HR-BTU-DEG F (3),
      MUNCH            IN-LB-SEC-BTU-DEG F (4), M-TON-DAY-GCAL-DEG C (5),
      MUNCH            OR CONVERSION FACTORS XD, XC, XK, XT, XTP, AND XH
      MUNCH            WILL BE USED (6) BEFORE WRITING DATA ON UNIT NLIST,
      MUNCH            OR WILL NOT BE CONVERTED (BLANK, ZERO, OR 1).
      MW             NORMAL MONITOR OUTPUT UNIT NUMBER (03 AT LRL).
/     M1 - M12       PARAMETERS CONTROLLING DIMENSIONS OF ARRAYS FOR
      M1 - M12         CRT TIMES (M1), MATERIALS (M2), REACTANTS (M3),
      M1 - M12         NODES (M4), INTERNAL CONNECTIONS (M5), SURFACE
      M1 - M12         CONNECTIONS (M6), BOUNDARY NODES (M7), HEAT GEN
      M1 - M12         NODES(M8), TABLE LENGTH (M9), MASS FLOW (M10),
      M1 - M12         CRT PLOT NODES (M11), AND REMOTE PROPERTIES (M12).
      M1 - M12         PARAMETERS M1, M3, M6, M7, M8, M10, M11, AND M12
      M1 - M12         ALSO CONTROL COMPILATION OF THE ASSOCIATED OPTIONAL
      M1 - M12         PARTS OF THE PROGRAM, SKIPPED IF PARAMETER IS ZERO.
      N              ARRAY SUBSCRIPT.
      NACRT          INDICATES TYPE OF SCALES ON CRT ISOTHERM PLOT.
      NADD           DIFF BETWEEN SUCCESSIVE VALUES OF NODE  IN SEQUENCE.
      NADF1          DIFF BETWEEN SUCCESSIVE VALUES OF NODF1 IN SEQUENCE.
      NADF2          DIFF BETWEEN SUCCESSIVE VALUES OF NODF2 IN SEQUENCE.
      NADG           DIFF BETWEEN SUCCESSIVE VALUES OF NODG  IN SEQUENCE.
      NADPR          DIFF BETWEEN SUCCESSIVE VALUES OF NPROP IN SEQUENCE.
      NADP1          DIFF BETWEEN SUCCESSIVE VALUES OF NODP1 IN SEQUENCE.
      NADP2          DIFF BETWEEN SUCCESSIVE VALUES OF NODP2 IN SEQUENCE.
      NADS           DIFF BETWEEN SUCCESSIVE VALUES OF NODS  IN SEQUENCE.
      NADSB          DIFF BETWEEN SUCCESSIVE VALUES OF NODSB IN SEQUENCE.
      NAD1           DIFFERENCE, AFTER MULTIPLICATION BY 10**NZ, BETWEEN
      NAD1             VALUES OF NAD1 IN SEQUENCE OF CONNECTIONS.
      NAD2           DIFFERENCE, AFTER MULTIPLICATION BY 10**NZ, BETWEEN
```

```
      NAD2          VALUES OF NAD2 IN SEQUENCE OF CONNECTIONS.
/     NAME        80-CHARACTER TITLE ON PROBLEM NAME CARD.  FIRST
      NAME          CHARACTER MUST BE AN *.  CHARACTER 72 CONTROLS USE
      NAME          OF DATA FROM PRECEDING PROBLEM (SEE K).
/     NAMES       PROBLEM NAME AT TIME LAST DATA END CARD WAS READ IN.
      NART        INDICATES TYPE OF SCALES ON TEMP VS NODE LOC. PLOT.
      NATI        INDICATES TYPE OF SCALES ON TEMP VS TIME PLOT.
      NB          INPUT UNIT NUMBER FOR BLOCK ITEMS, NR OR NUMBER IN
      NB            COLUMNS 15-16 OF BLOCK NUMBER CARD, WHEN COLUMNS
      NB            9-14 ARE /*INPUT/, OR IF *MAST CONTROL CARD IS USED
      NB            IN BLOCK 2, AFTER DATA TRANSFER, NB WILL BE NEST,
      NB            IF LATTER IS SPECIFIED, OR NLIST, IF NEST IS NOT
      NB            SPECIFIED AND NLIST IS NOT MW, M, OR 100.
      NB1         INTEGER WHICH SEEK1 OR SEEK2 IS TO SEARCH FOR.
      NB2         INTEGER WHICH SEEK2 IS TO SEARCH FOR, ALONG WITH NB1.
      NC          ARRAY SUBSCRIPT OF NXX IN PATCH.
/     NCARD       NUMBER OF CARD RECORDS READ IN FROM UNIT NREAD.
/     NCARDS      NUMBER OF CARD RECORDS WRITTEN ON UNIT NLIST.
      NCHECK      INDICATES /*CHECK/ CARD READ IN, IF NON-ZERO.
      NCUR        NUMBER OF VALUES OF TCONT USED FOR ISOTHERMS ON NODE
      NCUR          LOCATION VS TIME PLOT.  VARIES FROM 10 TO 40.
      NCYC        VALUE OF KCYC FOR WHICH CRT PLOT IS BEING MADE.
      NDATA       OUTPUT UNIT ON WHICH UNUSED DATA DECKS WILL BE SAVED,
      NDATA         SPECIFIED IN COLUMNS 7-8 OF /*DECKS/ CARD.
      NDATE       BCD WORD INDICATING MACHINE LETTER, MONTH, AND DAY.
      NDISC       INDICATES A /*DISCONTINUE/ CARD HAS BEEN READ IN, IF
      NDISC         NONZERO, AND PREVIOUSLY INTERRUPTED PROBLEM ENDED.
      NDOT        INDICATES TIME DERIVATIVES DDT(N), DDA(N), DDB(N) WILL BE
      NDOT          USED (NDOT = 0), OR MAINTAINED AT 0.0 (NDOT NOT 0).
      NDUMP       INDICATES SS1 SHOULD END A PROBLEM, IF 1.
/     NEST        UNIT NUMBER TO BE USED FOR INPUT OF BLOCK 2 ITEMS
      NEST          AFTER DATA HAS BEEN WRITTEN ON UNIT NLIST (SEE
      NEST          *MAST).  BLANK = NLIST, IF NLIST IS NOT MW, M, OR
      NEST          100.  OTHERWISE, BLANK = NB.
/     NEWBL(N)    NUMBER OF TIMES DATA BLOCK N HAS BEEN READ IN.  1000
      NEWBL(N)      IS ADDED EACH TIME A REFERENCED BLOCK IS READ IN
      NEWS        NUMBER OF REG NODES MADE SPECIAL IN ONE TIME STEP.
      NGRID       INDICATES DELETION OF CRT PLOT GRID LINES, IF 1.
      NHOUR       SIX-CHARACTER WORD INDICATING CLOCK TIME IN HOURS.
      NINT        NUMBER OF EQUALLY SPACED ISOTHERMS ON NODE LOCATION
      NINT          VS TIME PLOT.  VARIES FROM 10 TO 40.
      NIT         NUMBER OF NODES WITH INITIAL VALUES TT, AA, BB, GG.
      NKEM        NUMBER OF REACTANTS DESCRIBED IN INPUT BLOCK 3.
      NLIST       SECONDARY MATERIALS DATA LIST I/O UNIT (SEE *MAST).
      NLIST         BLANK = 4.  NLIST WILL BE REWOUND AND BECOME THE
      NLIST         NEXT INPUT UNIT FOR BLOCK 2 ITEMS AFTER IT HAS BEEN
      NLIST         WRITTEN ON, UNLESS NEST HAS BEEN SPECIFIED OR NLIST
      NLIST         IS MW, M, OR 100 (SEE NEST, NRS, NR, NB).  NLIST
      NLIST         WILL NOT BE REWOUND BEFORE OR AFTER BEING WRITTEN
      NLIST         ON IF IT IS PRECEDED BY A MINUS SIGN.  WHEN PRIME
      NLIST         IS CALLED FOR THE *DECKS OPTION, NLIST IS NDATA.
      NLOCK(N,K)  NODE WHOSE TEMP DET. PROP K FOR NODE N (SEE NPROP).
      NMAT        NUMBER OF MATERIALS DESCRIBED IN INPUT BLOCK 2.
/     NMATS       NUMBER OF MATERIALS IN LIST OF SELECTED MATERIALS
      NMATS         WHICH FOLLOWS *MAST CARD IN BLOCK 2 WHEN MALL = 0.
      NMELT       NUMBER OF MATERIALS WITH LATENT HEAT OF PHASE CHANGE.
      NN          INDICATES TABLE LOOK-UP FOR REACTANT A (1) OR B (2).
      NOCON       NUMBER OF INTERNAL NODE CONNECTIONS (INPUT BLOCK 5).
      NODB(N)     NUMBER OF BOUNDARY NODE WITH INDEX N.
      NODES       NUMBER OF BOUNDARY NODES (INPUT BLOCK 7).
      NODE(N)     IDENTIFICATION NUMBER OF THE NODE WITH INDEX N.
      NODEP(N)    NODE FOR WHICH A TEMPERATURE VS TIME PLOT WILL BE MADE,
      NODEP(N)      AND IF 100 ADDED TO LOGTIM, WILL BE USED TO PRODUCE A
      NODEP(N)      NODE LOCATION VS TIME (ISOTHERM) PLOT.
      NODES       NUMBER OF NODES DESCRIBED IN INPUT BLOCK 4.
      NODF1(N)    NUMBER OF NODE FROM WHICH MASS FLOWS TO NODF2(N).
      NODF2(N)    NUMBER OF NODE TO WHICH MASS FLOWS FROM NODF1(N).
      NODG(N)     NUMBER OF A NODE WITH VARIABLE HEAT GENERATION.
      NODMAT(N)   IDENTIFICATION NUMBER OF THE MATERIAL IN NODE(N).
      NODPS       TOTAL NUMBER OF NODES OF A MATERIAL.
```

```
      NODP1(N)      NODE FOR WHICH T(NODP2(N)) DETERMINES PROP NPROP(N).
      NODP2(N)      NODE WHOSE TEMP DETERMINES PROP NPROP(N) OF NODP1(N).
      NODS(N)       NUMBER OF SURFACE NODE CONNECTED TO NODSB(N).
      NODSB(N)      NUMBER OF BOUNDARY NODE CONNECTED TO NODS(N).
      NOD1(N)       NUMBER OF INTERNAL OR SURFACE NODE CONN. TO NOD2(N).
      NOD2(N)       NUMBER OF INTERNAL OR SURFACE NODE CONN. TO NOD1(N).
      NOFF          INDICATES A /*OFFMON/ CARD HAS BEEN READ IN, IF 1.
      NOFLOW        NUMBER OF MASS FLOW CONNECTIONS (INPUT BLOCK 10).
      NOGEN         NUMBER OF NODES WITH HEAT GENERATION.
      NOPOWS        NO. OF SURFACE-BOUNDARY CONN. WITH NON-ZERO POWER(N).
      NORAD         NUMBER OF NODE CONNECTIONS WITH NON-ZERO RINT(N).
      NORADS        NO. OF SURFACE-BOUNDARY CONN. WITH NON-ZERO RSURE(N).
      NOSCON        NUMBER OF EXTERNAL CONNECTIONS (INPUT BLOCK 6).
      NOSPEC        NUMBER OF SPECIAL NODES IN SYSTEM.
/     NOSR          INDICATES BOTH LOG AND LINEAR NODE LOCATION SCALES WILL
      NOSR            BE USED ON CRT PLOTS IF 2, ONLY LINEAR SCALES IF 1.
/     NOSRT         INDICATES BOTH LOG AND LINEAR NODE LOCATION SCALES WILL
      NOSRT           BE USED ON ISOTHERM PLOTS IF 2, ONLY LINEAR IF 1.
      NOSTEM        INDICATES BOTH LOG AND LINEAR TEMPERATURE SCALES WILL BE
      NOSTEM          USED ON CRT PLOTS IF 2, ONLY LINEAR IF 1.
      NOSTIM        INDICATES BOTH LOG AND LINEAR TIME SCALES WILL BE USED ON
      NOSTIM          CRT PLOTS IF 2, ONLY LINEAR IF 1.
      NOTE(N)       NUMBER OF NODE WITH INIT. COND. TT, AA, BB, AND GG.
/     NOW           INDICATES PRINTOUTS ARE BEING PRODUCED FOR THE TIME STEP
      NOW             JUST COMPLETED, IF NONZERO.
      NOXE(N)       INPUT VALUE OF NOTE(N).
      NOXEP(N)      INPUT VALUE OF NODEP(N).
      NOXF1(N)      INPUT VALUE OF NODF1(N).
      NOXF2(N)      INPUT VALUE OF NODF2(N).
      NOXG(N)       INPUT VALUE OF NODG(N).
      NOXMAT(N)     INPUT VALUE OF NODMAT(N).
      NOXP1(N)      INPUT VALUE OF NODP1(N).
      NOXP2(N)      INPUT VALUE OF NODP2(N).
      NOXS(N)       INPUT VALUE OF NODS(N).
      NOXSB(N)      INPUT VALUE OF NODSB(N).
      NOX1(N)       INPUT VALUE OF NOD1(N).
      NOX2(N)       INPUT VALUE OF NOD2(N).
      NP            TEMPORARY VALUE OF NPROP(N).
/     NPER          BCD WORD CONTAINING PERCENTAGES OF MACHINE TIME
      NPER            CHARGED TO CPU, I/O, AND SYSTEM CALLS SINCE THE
      NPER            LAST CALL OF TIMER (6600 ONLY, /8H UNKNOWN/ ON 3600).
/     NPERS         VALUE OF NPER OBTAINED JUST BEFORE OFFMON CALLED.
      NPLOT         NO. OF TEMP VS NODE LOCATION CRT PLOTS MADE.
      NPRINT        NO. OF TEMP VS NODE NUMBER PRINTOUTS MADE.
      NPROB         PROBLEM NUMBER, COUNTED FROM LAST /*DATA/, /*OFFMON/,
      NPROB           OR /*DISCONTINUE/ CARD.
      NPROP(N)      NO. OF PROPERTY EVALUATED AT TEMP OF NODE NODP2(N),
      NPROP(N)        USED FOR NODE NODP1(N).  IS 1 FOR CAPT, 2 FOR CONT,
      NPROP(N)        3 FOR QT, 4 FOR ZT, 5 FOR ET, 6 FOR GT, 7 FOR
      NPROP(N)        FLOWT, AND 8 FOR HSURT.
      NPUNCH        INDICATES FINAL VALUES OF T, A, B, AND CONSTANT
      NPUNCH          VALUES OF G, WILL BE PUNCHED IN THE FORMAT OF BLOCK
      NPUNCH          9 WHEN THE PROBLEM IS ENDED OR INTERRUPTED, TO
      NPUNCH          ALLOW A CONTINUATION PROBLEM TO BE RUN.
      NR            INPUT UNIT NUMBER FOR DATA DECKS, NRS OR NUMBER IN
      NR              COLUMNS 7-8 OF /*INPUT/ CARD.
      NREACT        NUMBER OF NODES WITH AN INITIAL CONCENTRATION A OR B, OR
      NREACT          OF A MATERIAL WITH HMELT(N) NOT ZERO.  BEFORE FIRST TIME
      NREACT          STEP, NKEM IS ADDED TO NREACT.
/     NREAD         MASTER MATERIAS DATA LIST INPUT UNIT (SEE *MAST).
      NREAD           NREAD WILL NOT BE REWOUND BEFORE OR AFTER BEING
      NREAD           READ IF IT IS PRECEDED BY A MINUS SIGN.  WHEN PRIME
      NREAD           IS CALLED FOR THE *DECKS OPTION, NREAD IS NR.
      NREC          NUMBER OF TAPE CHECK ERRORS IN BUFFERING IN A RECORD.
      NRECS         NUMBER OF RECORDS BUFFERRED IN WITH TAPE CHECK ERRORS.
      NREG          NUMBER OF REGULAR NODES IN SYSTEM.
      NRS           NORMAL MONITOR INPUT UNIT NUMBER (02 AT LRL).
/     NS            ARRAY SUBSCRIPT OF P(N).
      NSAVE         INDICATES A /*DECKS/ CARD HAS BEEN READ IN, IF 1, AND
      NSAVE           UNUSED DATA DECKS ARE TO BE SAVED ON UNIT NDATA.
      NSEG          NUMBER OF BLOCK ITEMS TO BE GENERATED IN ADDITION TO THE
      NSEG            ITEM DESCRIBED.  NODE NUMBERS WILL BE INCREMENTED BY THE
```

```
      NSEG          AMOUNTS NADD, NAD1, NAD2, NADS, NADSB, ETC., AS APPLICABLE,
      NSEG          AND IF DRAD IS PRECEDED BY A MINUS SIGN, DRAD VALUES WILL
      NSEG          BE INCREMENTED BY THE DIFFERENCE BETWEEN DRAD AND THE
      NSEG          VALUE OF DRAD FOR THE PRECEDING BLOCK ITEM.
      NSKIP       NUMBER OF CRT PLOT TIMES FOR WHICH NO DATA WAS PLOTTED, DUE
      NSKIP         TO LIMIT M1 ON TOTAL NUMBER OF PLOT TIMES.
      NSTOP       INDICATES PROBLEM TO END WHEN TIME STEP COMPLETE.
/     NSTORE      NUMBER OF WORDS OF MEMORY USED BY ARRAYS THAT HAVE SIZES
      NSTORE        FIXED BY PARAMETERS M1 TO M12.
      NTABLE      TOTAL NUMBER OF TABLES IN INPUT DATA.
      NTABS       NUMBER OF NODES WITH REMOTELY DEPENDENT PROPERTIES.
      NTARP       INDICATES TEMP VS NODE LOC. PLOTS TO BE MADE, IF 1.
      NTYPE(N)    NODE TYPE, REGULAR (0), SURFACE (1), ZERO-VOLUME (2),
      NTYPE(N)      KS=1 SPECIAL (3), KSPEC=0 CONVERTED SPECIAL (4),
      NTYPE(N)      OR KSPEC=1 CONVERTED SPECIAL (5).
      NTYPES(N)   INITIAL NODE TYPE .  SEE NTYPE(N).
/     NUM         NUMBER OF A NODE FOR WHICH T, DDT, A, B, G, AND SUMTIM WILL
      NUM           BE WRITTEN ON OUTPUT UNIT M EVERY TIME STEP.  THIS IS IN
      NUM           ADDITION TO THE DATA WRITTEN OUT FOR ALL NODES AT
      NUM           INTERVALS CONTROLLED BY IPRINT AND TIMEP.
      NUMX        INPUT VALUE OF NUM.
/     NUP         INDICATES CRT PLOTS WILL BE MADE, IF POSITIVE.  EQUAL
      NUP           TO NUP1 + NTARP.
/     NUP1        NUMBER OF NODES WHICH WILL BE ON TEMP VS TIME PLOTS,
      NUP1          AND ISOTHERM PLOTS, IF LOGTIM IS OVER 100.  SPECIFIED
      NUP1          BY LIST NODEP(N) IN BLOCK 11.
/     NUTMAX      MAXIMUM ALLOWABLE NUMBER OF ITERATIONS OF THE HEAT BALANCE
      NUTMAX        EQUATIONS FOR INTER-CONNECTED SPECIAL NODES (80).
/     NUTS        NUMBER OF ITERATIONS COMPLETED BEFORE CONVERGENCE.  WILL
      NUTS          CAUSE NEXT TIME STEP TO BE REDUCED IF MORE THAN 40.
      NUTSUM      TOTAL NUMBER OF ITERATIONS USED IN PROBLEM.
      NUTX        MAXIMUM NUMBER OF ITERATIONS USED IN ANY TIME STEP.
      NVARC       NUMBER OF MATERIALS WITH VARIABLE HEAT CAPACITY.
      NVARE       NUMBER OF REACTANTS WITH VARIABLE ACTIVATION ENERGY.
      NVARFL      NUMBER OF MASS FLOW CONNECTIONS WITH VARIABLE FLOW.
      NVARG       NUMBER OF NODES WITH VARIABLE HEAT GENERATION.
      NVARH       NUMBER OF SURFACE NODES WITH VARIABLE SURFACE COND.
      NVARK       NUMBER OF MATERIALS WITH VARIABLE THERMAL COND.
      NVARQ       NUMBER OF REACTANTS WITH VARIABLE HEAT OF REACTION.
      NVART       NUMBER OF BOUNDARY NODES WITH VARIABLE TEMPERATURE.
      NVARZ       NUMBER OF REACTANTS WITH VARIABLE COLLISION FREQ.
/     NW          NUMBER OF WORDS IN EACH RECORD BUFFERED IN (8 OR 9 ON
      NW            6600, 10 ON 3600).
/     NWBL12      MEMORY USED FOR ADJUSTABLE ARRAYS FOR BLOCK 12 OPTION.
      NWBL12        EQUALS 8*M4 + 5*M12.
/     NWCHEM      MEMORY USED FOR ADJUSTABLE ARRAYS FOR BLOCK 3 OPTION.
      NWCHEM        EQUALS 4*M2 + (6 + 9*M9)*M3 + 10*M4.
/     NWFLOW      MEMORY USED FOR ADJUSTABLE ARRAYS FOR BLOCK 10 OPTION.
      NWFLOW        EQUALS 4*M4 + (3 + 3*M9)*M8.
/     NWGEN8      MEMORY USED FOR ADJUSTABLE ARRAYS FOR BLOCK 8 OPTION.
      NWGEN8        EQUALS 2*M4 + (3 + 3*M9)*M8.
/     NWINDR      INDICATES, IF NEGATIVE, THAT UNIT NREAD WILL NOT BE
      NWINDR        REWOUND BEFORE OR AFTER IT IS READ.
/     NWINDW      INDICATES, IF NEGATIVE, THAT UNIT NLIST WILL NOT BE
      NWINDW        REWOUND BEFORE OR AFTER IS IS WRITTEN ON.
/     NWPLOT      MEMORY USED FOR ADJUSTABLE ARRAYS FOR BLOCK 11 OPTION.
      NWPLOT        EQUALS 4*M4 + 3*M11 + (1 + M1)*M11.
/     NWREQD      MEMORY USED FOR REQUIRED ADJUSTABLE ARRAYS.
      NWREQD        EQUALS (8 + 7*M9)*M2 + 28*M4 + 12*M5 + 3*M9.
/     NWSURE      MEMORY USED FOR ADJUSTABLE ARRAYS FOR BLOCK 6 OPTION.
      NWSURE        EQUALS (12 + 3*M9)*M6 + (5 + 3*M9)*M7.
      NX          TEMPORARY VALUE OF NEWBL(N), OUTPUT LINE COUNTER.
      NXX(N)      A BCD NUMBER TO BE TESTED FOR A DECIMAL PT. IN PATCH.
      NX1(N)      INPUT NUMBER HINT(N), HSURE(N), OR TT(N) IN BCD FORM.
      NX2(N)      INPUT NUMBER RINT(N), RSURE(N), OR AA(N) IN BCD FORM.
      NX3(N)      INPUT NUMBER POWER(N) OR BB(N) IN BCD FORM.
      NX4(N)      INPUT NUMBER GG(N) IN BCD FORM.
      NZ          INDICATES NAD1, NAD2 ARE TO BE MULTIPLIED BY 10**NZ.
      N1 - N2     TEMPORARY VALUES OF VARIOUS INTEGERS.
/     P(N)        A WORD BUFFERED INTO MEMORY, IN PRIME.
/     PIX         NUMBER OF TIME STEPS ESTIMATED TO REMAIN BEFORE TIMAX
      PIX           REACHES AN INTEGRAL MULTIPLE OF TIMEP.
```

```
      PONE          FREE CONVECTION EXPONENT FOR ANY EXTERNAL CONNECTION
      PONE            WITH POWER(N) BLANK, IN ANY BLOCK 6 READ IN AFTER
      PONE            BLOCK 1.
      POUT          FLOATING POINT VALUE GIVEN A BCD NUMBER NXX IN PATCH.
      POWER(N)      FREE CONVECTION EXPONENT FOR EXTERNAL CONNECTION
      POWER(N)        BETWEEN NODS(N) AND NODSB(N), USED IN EQUATION
      POWER(N)        OF FORM HSURE*(T1 - T2)**POWER.  BLANK = PONE.
      PRAC(N)       EQUAL TO RPLOT(J), WHERE J = NODEP(N).
      PRMAX         LARGEST NODE LOCATION ON ISOTHERM PLOT SCALE.
      PRMIN         SMALLEST NODE LOCATION ON ISOTHERM PLOT SCALE.
      PROP(N)       TABULATED CAPACITY OR CONDUCTIVITY, USED IN PRIME.
/     PS(N)         A WORD BUFFERED OUT, IN PRIME.
      PTEMP(J,N)    CRT PLOT TEMP = FTEMP1+FTEMP2*T(NODEP(J)), N = NPLOT.
      PTIME(N)      CRT PLOT TIME = FTIME1 + FTIME2*SUMTIM, N = NPLOT.
/     PX            TEMPORARY VALUE OF POWER(N) OR CORD(N).
      P1 - P5       TEMPORARY VALUES OF SEVERAL INPUT VARIABLES.
      QA(N)         HEAT OF REACTION OF REACTANT A IN NODE(N).
      QB(N)         HEAT OF REACTION OF REACTANT B IN NODE(N).
      QT(J,N)       HEAT OF DECOMPOSITION OF KEM(N) AT TVARQ(J,N).
      RAD           RADIATION COEFFICIENT AT NODE CONNECTIONS.
      RADIUS(N)     RADIUS OF NODE(N), DRAD*SCALE (POS) IN BLOCK 4.
      RATE          AVERAGE RATE OF HEAT FLOW INTO NODE OR SYSTEM.
/     RATG          VALUE OF TVARY/DTEMP IN TIME STEP DELTG.
/     RATIO         RATIO OF DELT TO DELTS, BEFORE RESTRICTING DELT TO LIMITS
      RATIO           SMALL AND DELTMX.  RATIO IS TVARY/MAX1F(DTEMP,DTMAX),
      RATIO           RESTRICTED TO THE RANGE FROM 0.5 TO 2.0.
/     RAT1          VALUE OF TVARY/DTEMP IN TIME STEP DELTS.
/     RAT2          VALUE OF TVARY/DTEMP IN TIME STEP DELTSS.
/     RDELT         VALUE OF (DELT + DELTS)/(DELTS + DELTSS).
      REX           NODE(N), IF KD IS 1, RADIUS(N), IF KD IS 2 OR 3.
      RGAS          GAS CONSTANT, EQUAL TO 1.987 CAL/MOLE DEG K.
      RINT(N)       RADIATION VIEW FACTOR FOR CONNECTION BETWEEN NOD1(N)
      RINT(N)         AND NOD2(N), INCLUDING EFFECTS OF GEOMETRY AND
      RINT(N)         SURFACE EMISSIVITY.  IF NEGATIVE, POS. VALUE USED
      RINT(N)         AS FREE CONVECTION EXPONENT, HINT*(T1 - T2)**RINT.
      RINTC         FREE CONVECTION EXPONENT DERIVED FROM RINT.
      RINTR         RADIATION TRANSPORT COEFFICIENT DERIVED FROM RINT.
      RINTS         INPUT VALUE OF RINT(N).
      RMAX          LARGEST VALUE OF RPLOT(N) TO BE USED ON CRT PLOTS.
/     RMAXR         LARGEST VALUE OF RPLOT(N) TO BE USED ON CRT PLOTS WHEN
      RMAXR           INFRA IS ZERO.
      RMIN          SMALLEST VALUE OF RPLOT(N) TO BE USED ON CRT PLOTS.
/     RMINR         SMALLEST VALUE OF RPLOT(N) TO BE USED ON CRT PLOTS WHEN
      RMINR           INFRA IS ZERO.
      RONE          RADIATION VIEW FACTOR FOR ANY EXTERNAL CONNECTION
      RONE            WITH RSURE(N) BLANK, IN ANY BLOCK 6 READ IN AFTER
      RONE            BLOCK 1.
/     RPLOT(N)      NODE LOCATION = FRAD1 + FRAD2*REX, WHERE REX = NODE(N),
      RPLOT(N)       IF KD = 1, OR RADIUS(N), IF KD IS 2 OR 3.  RPLOT(N)
      RPLOT(N)        IS MULTIPLIED BY 0.5*SUMTIM**(-0.5) IF INFRA IS 1.
      RSURE(N)      RADIATION VIEW FACTOR FOR EXTERNAL CONNECTION BETWEEN
      RSURE(N)        NODS(N) AND NODSB(N), INCLUDING EFFECTS OF GEOMETRY
      RSURE(N)        AND SURFACE EMISSIVITY.  BLANK = RONE.
      RX            TEMPORARY VALUE OF RSURE(N).
      SCALE           READ IN, TO CHANGE SCALE.
/     SCALE         LINEAR SCALE FACTOR USED IN CALCULATING ALL LENGTHS,
      SCALE           AREAS, AND VOLUMES FOR BLOCKS 4, 5, AND 6 READ IN
      SCALE           AFTER BLOCK 1.  BLANK = 1. MORE THAN ONE BLOCK 1 MAY BE
      SET           TIME OR TEMP FOR TABLE LOOK-UP.
      SETD          FACTOR DETERMINING EXPONENTIAL DECAY OF HEAT GEN.
      SETS          FACTOR DETERMINING EXPONENTIAL DECAY OF HEAT GEN.
      SIGMA         STEPHAN-BOLZMAN CONSTANT FOR THERMAL RADIATION, WITH VALUE
      SIGMA           1.355E-12 FOR KT = 1 OR 2, OR 0.173E-8 FOR KT = 3 OR 4.
      SLIM(N)       MAXIMUM STABLE TIME STEP FOR A REGULAR NODE(N),
      SLIM(N)         EQUALS CAP(N)/ZIP(N).
      SLOC(J,N)     SLOPE BETWEEN ENTRIES J AND J-1 IN CAPT(J,N) TABLE.
      SLOE(J,N)     SLOPE BETWEEN ENTRIES J AND J-1 IN ET(J,N) TABLE.
      SLOFL(J,N     SLOPE BETWEEN ENTREES J AND J-1 IN FLOWT(J,N) TABLE.
      SLOG(J,N)     SLOPE BETWEEN ENTRIES J AND J-1 IN GT(J,N) TABLE.
      SLOH(J,N)     SLOPE BETWEEN ENTRIES J AND J-1 IN HSURT(J,N) TABLE.
      SLOK(J,N)     SLOPE BETWEEN ENTRIES J AND J-1 IN CONT(J,N) TABLE.
/     SLOPS         RATIO OF MAXIMUM RATES OF TEMPERATURE CHANGE IN TIME STEPS
```

```
      SLOPS          DELTS AND DELTSS, (RAT1*DELTS)/(RAT2*DELTSS).
      SLQQ(J,N)    SLOPE BETWEEN ENTRIES J AND J-1 IN QT(J,N) TABLE.
      SLQT(J,N)    SLOPE BETWEEN ENTRIES J AND J-1 IN TEMPB(J,N) TABLE.
      SLQZ(J,N)    SLOPE BETWEEN ENTRIES J AND J-1 IN ZT(J,N) TABLE.
/     SMALL        MINIMUM ALLOWABLE TIME STEP.  BLANK = 1.E-12, LATER
      SMALL          REPLACED WITH 0.01*DELTMX, AS LONG AS AT LEAST 1/4 OF THE
      SMALL          NODES ARE REGULAR NODES.
      SMALT        INPUT VALUE OF SMALL OR 1.0E-12, WHICHEVER IS LARGER.
/     SPEED        FACTOR CONTROLLING ACCELERATION OF ITERATION METHOD (0.2).
      START        SMALLEST TIME TO BE USED ON CRT PLOT TIME SCALES.
      SUMTIM       TOTAL PROBLEM TIME, STARTING AT TAU.  LIMIT IS TIMAX.
      S1           ITERATION ACCELERATION FACTOR, EITHER SPEED OR 0.0.
      S2           ITERATION ACCELERATION FACTOR, EQUALS S1 + 1.
      T(N)         TEMPERATURE IN NODE(N).  SEE TONE, TT(N).
      TAU          INITIAL VALUE OF TIME, AND LOWER LIMIT ON CRT SCALE.
      TB(N)        TEMPERATURE AT BOUNDARY NODE NODB(N).  SEE TBONE,
      TB(N)          LTABT(N), TEMPB(J,N), AND TIMEB(J,N).
      TBASE        TEMPERATURE DIFFERENCE BETWEEN SCALE IN USE AND ABSOLUTE
      TBASE          SCALE, 273.15 FOR KT = 1, 0.0 FOR KT = 2 OR 4, AND
      TBASE          460.0 FOR KT = 3.
      TBS(N)       AVERAGE VALUE OF TB(N) IN A TIME STEP.
      TCONT(N)     TEMPS USED FOR ISOTHERMS ON NODE LOCATION VS TIME
      TCONT(N)       PLOT.  INTERVAL IS TINT, NUMBER IS NCUR, 10 TO 40.
      TCONT(N)       TCONT(2) IS INITIALLY THE HIGHEST TEMP TO BE USED.
      TEMPAD       AVERAGE TEMPERATURE OF SYSTEM (CONSTANT CAP(N)).
      TEMPB(J,N)   TEMPERATURE AT BOUNDARY NODE NODB(N) AT TIMEB(J,N).
      TEMPB(1,N)   AVERAGE VALUE OF TB(N), IF LTABT(N) IS 100.
      TEMPB(2,N)   AMPLITUDE OF VARIATION OF TB(N), IF LTABT(N) IS 100.
      TEMPER       CHANGE IN AVG TEMP OF SYSTEM DUE TO NET SURFACE FLUX.
      TEMPLE       CHANGE IN AVG TEMP DUE TO HEAT GENERATION.
      TEX          ABSOLUTE TEMPERATURE DETERMINING REACTION RATE.
      TEX          DUMMY ARGUMENT IN CALL CARTMM AND CALL LOGMM.
      THIGH        LARGEST TEMPERATURE TO BE PLOTTED FOR A CRT NODE.
/     TIM          REMAINING TIME BETWEEN SUMTIM AND NEXT HIGHER INTEGER
      TIM            MULTIPLE OF TIMEP FOR WHICH A PRINTOUT WILL BE MADE.
/     TIMS         REMAINING TIME BETWEEN SUMTIM AND NEXT HIGHER INTEGER
      TIMS           MULTIPLE OF TIMEP.
      TIMAX        MAXIMUM ALLOWED PROBLEM TIME (UPPER LIMIT ON SUMTIM).
      TIMAX          NEGATIVE WILL STOP PROBLEM AT END OF FIRST TIME
      TIMAX          STEP, ZERO OR BLANK IS IGNORED.
      TIMEB(J,N)   TIME AT WHICH BOUNDARY NODE NODB(N) IS AT TEMPB(J,N).
      TIMEB(1,N)   SINE WAVE PERIOD OF TB(N), IF LTABT(N) IS 100.
      TIMEB(2,N)   SINE WAVE ADVANCE TIME OF TB(N), IF LTABT(N) IS 100.
/     TIMEP        INDICATES THAT DATA PRINTOUTS AND CRT PLOTS WILL BE MADE AT
      TIMEP          PROBLEM TIMES WHICH ARE INTEGER MULTIPLES OF TIMEP, IN
      TIMEP          ADDITION TO THE FIRST, SECOND, AND LAST TIME STEPS,
      TIMEP          AND OTHER TIME STEPS ACCORDING TO IPRINT AND JPIC.  ALSO
      TIMEP          SEE KDATA, NUM, IRITE, ITAPE, MW, M, AND *OUTPUT.
      TINT         INTERVAL BETWEEN ISOTHERMS ON NODE LOCATION VS TIME
      TINT           PLOT, EQUAL TO 5.0*TVARY*FTEMP2*2.0**N, WITH N SUCH
      TINT           THAT FROM 10 TO 40 ISOTHERMS ARE PLOTTED.
      TLOW         SMALLEST TEMPERATURE TO BE PLOTTED FOR A CRT NODE.
      TMAX         MAXIMUM ALLOWABLE TEMPERATURE.  REPLACED WITH 1.0E12
      TMAX           IF NOT LARGER THAN TMIN.  HMELT(N) WILL BE SET TO
      TMAX           ZERO FOR MATERIALS WITH TMELT(N) ABOVE TMAX.
      TMAXP(N)     MAXIMUM VALUE OF TPLOT(N) DURING PROBLEM.
      TMAX1        MAXIMUM TEMPERATURE IN SYSTEM AT SUMTIM.
      TMAX1        LARGEST TEMPERATURE TO BE USED FOR A CRT PLOT SCALE.
      TMELT(N)     TEMPERATURE AT WHICH MAT(N) HAS LATENT HEAT HMELT(N).
      TMIN         MINIMUM ALLOWABLE TEMPERATURE.  REPLACED WITH -1.0E12
      TMIN           IF NOT LESS THAN TMAX.  HMELT(N) WILL BE SET TO
      TMIN           ZERO FOR MATERIALS WITH TMELT(N) BELOW TMIN.
      TMINP(N)     MINIMUM VALUE OF TPLOT(N) DURING PROBLEM.
      TMIN1        MINIMUM TEMPERATURE IN SYSTEM AT SUMTIM.
      TMIN1        SMALLEST TEMPERATURE TO BE USED FOR A CRT PLOT SCALE.
      TMS          AVERAGE TEMPERATURE OF A MATERIAL (CONSTANT CAP(N)).
      TONE         INITIAL TEMPERATURE ASSIGNED TO ALL NODES IN ANY BLOCK 4
      TONE           READ IN AFTER BLOCK 1, AND SUBSTITUTED FOR UNSPECIFIED
      TONE           TT(N) IN ANY BLOCK 9 READ IN AFTER BLOCK 1.
      TPLOT(N)     CRT PLOT TEMP = FTEMP1 + FTEMP2*T(N).  SEE RPLOT(N).
      TPMAX        MAXIMUM OF ALL TPLOT VALUES IN PROBLEM.
```

```
      TPMIN          MINIMUM OF ALL TPLOT VALUES IN PROBLEM.
      TRAN(N)        THERMAL CONDJCTANCE BETWEEN NOD1(N) AND NOD2(N).
      TRANS(N)       THERMAL CONDJCTANCE BETWEEN NODS(N) AND NODSB(N).
      TT(N)          INITIAL TEMPERATJRE IN NOTE(N).  BLANK = TONE.
      TVARC(J,N)     TEMP CR TIME AT WHICH MAT(N) HAS CAPT(J,N).
      TVARE(J,N)     TEMP CR TIME AT WHICH KEM(N) HAS ET(J,N).
      TVARFL(J,N)    TEMP CR TIME AT WHICH FLOWN(N) IS FLOWT(J,N).
      TVARG(1,N)     HALF-LIFE OF HEAT GEN IN NODG(N,) WHEN LTABG(N) IS 0.
      TVARG(J,N)     TEMP CR TIME AT WHICH NODG(N) HAS GT(J,N).
      TVARH(J,N)     TEMP CR TIME AT WHICH NODS(N) HAS HSLRT(J,N).
      TVARK(J,N)     TEMP CR TIME AT WHICH MAT (N) HAS CONT(J,N).
      TVARP(N)       TABLE TEMPERATURE, USED IN PRIME.
      TVARQ(J,N)     TEMP CR TIME AT WHICH KEM (N) HAS QT(J,N).
/     TVARY          TEMPERATURE ACCURACY CONTROL.  TIME STEPS ADJUSTED TO
      TVARY            KEEP MAXIMJM TEMPERATURE CHANGE NEAR TVARY, AND NO
      TVARY            MORE THAN 2.0*TVARY.  CONTROLS ITERATION CONVERGENCE
      TVARY            (SEE EMAX, ESUM, HSUM, ERRS).  CRITERIA FOR STEADY
      TVARY            STATE INCLJDE TWO CONSECUTIVE TIME STEPS WITH DTEMP
      TVARY            LESS THAN 0.001*TVARY.
      TVARZ(J,N)     TEMP CR TIME AT WHICH KEM (N) HAS ZT(J,N).
      TX             HEAT CONTENT OF SYSTEM, BASED ON CONSTANT CAP(N), OR
      TX               TIME INTERVAL SUMTIM-TAU, OR TEMPORARY VALUE OF TT(N).
      T1             EST. AVG. ABSOLUTE TEMPERATURE OF A SURFACE NODE.
      T2             EST. AVG. ABSOLUTE TEMPERATURE OF A BOUNDARY NODE.
      VOL(N)         VOLUME OF NODE(N), BASED ON INPUT IN BLOCKS 1 AND 4,
      VOL(N)           EQUALS (GEOM*DWIDE*DLONG*DRAD**KSYM)*SCALE**3.
      VOLMS          TOTAL VOLUME OF A MATERIAL.
      VOLS           TOTAL VOLUME OF ALL NODES IN SYSTEM.
      W(N)           SENSIBLE AND LATENT HEAT CONTENT OF NODE(N) ABOVE 0.
      WMS            TOTAL HEAT CONTENT OF A MATERIAL.
      WT(J,N)        SENS. HEAT CONTENT OF MAT(N) AT TVARC(J,N), ABOVE 0.
/     XC             CONVERSION FACTOR FOR HEAT CAPACITY (SEE MUNCH).
      XC               NEW CAPT = XC*(OLD CAPT).
/     XD             CONVERSION FACTOR FOR DENSITY (SEE MUNCH).
      XD               NEW DENS = XD*(OLD DENS).
/     XH             CONVERSION FACTOR FOR LATENT HEAT (SEE MUNCH).
      XH               XH = XC*XT, NEW HMELT = XH*(OLD HMELT).
/     XK             CONVERSION FACTOR FOR CONDUCTIVITY (SEE MUNCH).
      XK               NEW CONT = XK*(OLD CONT)
      XMAP           X COORDINATE OF NODE NUMBER ON TEMP VS TIME PLOT.
/     XT             CONVERSION FACTOR FOR TEMPERATURE (SEE MUNCH).
      XT               NEW TEMP = XT*(OLD TEMP) + XTP.
/     XTP            CONVERSION FACTOR FOR TEMPERATURE SCALE (SEE MUNCH).
      XTP              NEW TEMP = XT*(OLD TEMP) + XTP.
      YMAP           Y COORDINATE OF NODE NUMBER ON TEMP VS TIME PLOT.
      ZA(N)          NATURAL LOG OF COLL. FREQ. OF REACTANT A IN NODE(N).
      ZB(N)          NATURAL LOG OF COLL. FREQ. OF REACTANT B IN NODE(N).
      ZIP(N)         OVERALL CONDJCTANCE FOR NODE(N), EQUAL TO SUM OF
      ZIP(N)           CONDUCTANCES OF ALL INTERNAL AND EXTERNAL THERMAL
      ZIP(N)           CONNECTIONS TO NODE(N), AND CONDUCTANCES OF ALL
      ZIP(N)           MASS FLOW CONNECTIONS LEADING INTO NODE(N).
      ZT(J,N)        NATL LOG COLLISION FREQUENCY OF KEM(N) AT TVARZ(J,N).
*                                                                          **
      4.  TRUMP COMMON BLOCKS.

/     ACHEM          COMMON BLOCK OF ARRAYS USED ONLY IN CHEM (M3 NOT 0).
/     ACHEMS         COMMON BLOCK OF ARRAYS USED IN CHEM, THERM, FLOW,
      ACHEMS           AND TALLY, REQJIRED ONLY IF CHEM USED (M3 NOT 0).
/     AFLOW          COMMON BLOCK OF ARRAYS USED ONLY IN FLOW (M10 NOT 0).
/     AFLOWS         COMMON BLOCK OF ARRAYS USED IN FLOW AND SPECK,
      AFLOWS           REQUIRED ONLY IF FLOW IS USED (M10 NOT 0).
/     AGEN           COMMON BLOCK OF ARRAYS USED ONLY IN GEN (M8 NOT 0).
/     AGENS          COMMON BLOCK OF ARRAYS USED IN THERM, GEN, AND TALLY
      AGENS            (M8 NOT 0).
/     APLOT          COMMON BLOCK OF ARRAYS USED ONLY IN PLOT (M1 AND M11
      APLOT            NOT ZERO, KPLOTL, KPLOTR, KPLOTT NOT ALL ZERO).
/     APRIM          COMMON BLOCK OF ARRAYS USED ONLY IN PRIME (KPRIME
      APRIM            NOT ZERO).
/     AREM           COMMON BLOCK OF ARRAYS USED ONLY IN THERM (M12 NOT 0).
/     AREMS          COMMON BLOCK OF ARRAYS USED IN THERM, GEN, CHEM, FLOW,
      AREMS            AND SURE (M12 NOT 0).
```

```
/     ASURE        COMMON BLOCK OF ARRAYS USED ONLY IN SURE (M6, M7 NOT
      ASURE          BOTH ZERO).
/     ASURES       COMMON BLOCK OF ARRAYS USED IN SURE AND SPECK,
      ASURES         REQUIRED ONLY IF SURE IS USED (M6, M7 NOT ZERO).
/     GOBCOM       COMMON BLOCK OF SYMBOLS RECEIVED FROM TTY IN KITTY
      GOBCOM         (MACHINE NOT ZERO, K66ST OR K66DS NOT ZERO).
      (BLANK)      COMMON BLOCK CONTAINING ARRAYS AND UNSUBSCRIPTED
      (BLANK)        VARIABLES USED IN ONE OR MORE NON-OPTIONAL TRUMP
      (BLANK)        SUBROUTINES.
*                                                                      **
      5.  TRUMP SUBROUTINES AND CLICHES.

/     CHEM         SUBROUTINE OF TRUMP FOR CHEMICAL REACTION.  USES DATA
      CHEM           FROM BLOCK 3 (OPTIONAL, SKIPPED IF M3 ZERO).
      FINK         SUBROUTINE OF TRUMP FOR HEAT FLOW BETWEEN NODES BY
      FINK           CONDUCTION, CONVECTION, AND RADIATION.  USES DATA
      FINK           FROM BLOCK 5.  TREATS ALL NODES AS REGULAR NODES.
/     FLOW         SUBROUTINE OF TRUMP FOR MASS FLOW BETWEEN NODES.
      FLOW           USES DATA FROM BLOCK 10.  TREATS ALL NODES AS
      FLOW           REGULAR NODES (OPTIONAL, SKIPPED IF M10 ZERO).
/     GEN          SUBROUTINE OF TRUMP FOR INTERNAL HEAT GENERATION.
      GEN            USES DATA FROM BLOCKS 1, 8, AND 9 (OPTIONAL,
      GEN            SKIPPED IF M8 IS ZERO).
      HEART        SUBROUTINE OF TRUMP FOR READING IN TRUMP CONTROL CARDS,
      HEART          PROBLEM NAME CARDS, BLOCK NUMBER CARDS, DATA END CARDS,
      HEART          AND MAKING SUBROUTINE CALLS FOR DATA INPUT AND TIME STEP
      HEART          CALCULATIONS.  CONTROLS PROBLEM INTERRUPTION AND RESTART,
      HEART          MONITORS CLOCK.
      KITTY        SUBROUTINE OF TRUMP/6600 FOR CONTROLLEE ENTRY POINT,
      KITTY          SAVING CONTROLLEE FILE, GETTING DATA FILE NAME.
      KITTY          USED ONLY AT LRL IN OCTOPUS CONTROLLEE VERSION.
      PATCH        SUBROUTINE OF TRUMP TO CONVERT A NUMBER FROM BCD TO
      PATCH          FLOATING POINT IF IT CONTAINS A DECIMAL POINT, OR
      PATCH          SUBSTITUTING A SPECIFIED FLOATING POINT NUMBER.
/     PLOT         SUBROUTINE OF TRUMP FOR MAKING CRT PLOTS AND TABLES
      PLOT           OF TEMP VS NODE LOCATION, TEMP VS TIME, AND NODE
      PLOT           LOCATION VS TIME.  CONTROLLED BY BLOCK 11 INPUT
      PLOT           (OPTIONAL, SKIPPED IF M1 OR M11 IS ZERO, PARTS ARE
      PLOT           SKIPPED IF KPLOTR, KPLOTL, OR KPLOTT IS ZERO).
/     PRIME        SUBROUTINE OF TRUMP FOR READING IN LIST OF SELECTED
      PRIME          MATERIALS AND COPYING THEIR PROPERTIES FROM THE
      PRIME          MASTER MATERIALS DATA LIST TO A SECONDARY DATA LIST
      PRIME          AND INTO MEMORY.  ALSO USED FOR COPYING UNUSED DATA
      PRIME          DECKS FROM THE INPUT UNIT ONTO ANOTHER UNIT TO BE
      PRIME          SAVED (OPTIONAL, SKIPPED IF KPRIME IS ZERO).
      REFER        SUBROUTINE OF TRUMP TO FIND ARRAY SUBSCRIPTS OF
      REFER          MATERIALS, REACTANTS, NODES, OR BOUNDARY NODES
      REFER          REFERRED TO BY IDENTIFICATION NUMBER IN ANOTHER
      REFER          DATA BLOCK, AND TO WRITE OUT DIAGNOSTIC STATEMENTS
      REFER          WHENEVER ONE CANNOT BE FOUND.
      SEEK1        SUBROUTINE OF TRUMP FOR FINDING AN IDENTIFICATION
      SEEK1          NUMBER OF A MATERIAL, REACTANT, NODE, OR BOUNDARY
      SEEK1          NODE IN A BLOCK ITEM LIST.
      SEEK2        SUBROUTINE OF TRUMP FOR FINDING THE IDENTIFICATION
      SEEK2          NUMBERS OF A PAIR OF NODES IN A BLOCK ITEM LIST.
      SPECK        SUBROUTINE OF TRUMP FOR HEAT FLOW BETWEEN SPECIAL
      SPECK          NODES AND OTHER NODES OR BOUNDARY NODES.  ITERATES
      SPECK          TO SOLVE SET OF IMPLICIT DIFFERENCE EQUATIONS WHEN
      SPECK          SPECIAL NODES ARE CONNECTED TO EACH OTHER.
/     STORE        CLICHE CONTAINING LAST PART OF BLANK COMMON, USED IN
      STORE          HEART, THERM, GEN, CHEM, FINK, FLOW, SURE, SPECK,
      STORE          TALLY, AND PLOT.  DEFINED IN MAIN.
/     STORK        CLICHE CONTAINING PARAMETER AND DATA STATEMENTS, AND
      STORK          BANK STATEMENTS (3600 ONLY).  DEFINED IN MAIN.
/     STORM        CLICHE CONTAINING FIRST PART OF BLANK COMMON, USED IN
      STORM          ALL SUBROUTINES BUT PATCH.  DEFINED IN MAIN.
/     SURE         SUBROUTINE OF TRUMP FOR HEAT FLOW BETWEEN SURFACE
      SURE           NODES, BOUNDARY NODES BY RADIATION, FREE AND FORCED
      SURE           CONVECTION.  USES DATA FROM BLOCKS 6 AND 7.  TREATS
      SURE           ALL SURFACE NODES AS REGULAR NODES.  SEE SPECK.
      SURE           (OPTIONAL, SKIPPED IF M6 OR M7 IS ZERO.)
      TALLY        SUBROUTINE OF TRUMP FOR INITIALIZATION, TOTALING AND
```

```
      TALLY          CHECKING RESULTS, FINDING NEW DELTMX AND DELT,
      TALLY          CONVERTING REGULAR TO SPECIAL NODES, TESTING FOR
      TALLY          ENDING THE PROBLEM.  USES DATA FROM BLOCKS 1 AND 9.
      TALLY1       ENTRY POINT IN TALLY FOR PUNCHING BLOCK 9 CARDS.
      THERM        SUBROUTINE OF TRUMP FOR MATERIAL PROPERTIES AND NODE
      THERM          DESCRIPTIONS.  USES DATA FROM BLOCKS 2 AND 4.
      THERM1       ENTRY POINT IN THERM FOR EST. LATENT HEAT EFFECTS.
      THERM2       ENTRY POINT IN THERM FOR FINAL LATENT HEAT EFFECTS.
/     TIMER        SUBROUTINE OF TRUMP FOR OBTAINING TIME CHARGED TO THE
      TIMER          PROBLEM, IN SECONDS, SINCE LAST CALL OF TIMER.
      TIMER          CALLS TICHEK (6600) OR CLOCK1 (3600).
      TRUMP        A COMPUTER PROGRAM FOR TRANSIENT AND STEADY STATE
      TRUMP          TEMPERATURE DISTRIBUTIONS IN MULTIDIMENSIONAL
      TRUMP          SYSTEMS WITH CONDUCTION, CONVECTION, RADIATION,
      TRUMP          HEAT GENERATION, CHEMICAL REACTION, PHASE CHANGE,
      TRUMP          AND MASS FLOW, AND VARIABLE PROPERTIES AND
      TRUMP          BOUNDARY CONDITIONS.
*                                                                      **
      6.  SYSTEM SUBROUTINES AND STATEMENTS.

      ABSF(X)      FINDS ABSOLUTE VALUE OF A FLOATING POINT ARGUMENT X.
      ASSIGN       ASSIGNS LOGICAL I/O CHANNEL TO AN I/O HARDWARE TYPE.
      BANK X (N)   ASSIGNS STORAGE OF SUBROUTINE OR LABELLED COMMON X TO
      BANK X (N)     3600 MEMORY BANK N.
      BSPACE(K)    BACKSPACES UNIT K ONE RECORD.
/     BUFFER IN    READS DATA IN, USING A BUFFER, ACCORDING TO ARGUMENT LIST.
/     BUFFER OUT   WRITES OUT DATA, USING A BUFFER, ACCORDING TO ARGUMENT LIST.
      CARIMM       FINDS MAXIMUM AND MINIMUM ARRAY VALUES (LCPLOT).
/     CHAR X       TYPE SPECIFICATION DEFINING X AS A 6-BIT INTEGER.
      CLICHE X     DEFINES GROUP OF STATEMENTS FOR LATER INSERTION IN
      CLICHE X       THE PROGRAM BY MEANS OF A /USE X/ STATEMENT.
      CLOCK        FINDS MACHINE LETTER, DATE AND HOUR.
/     CLOCK1       FINDS CLOCK TIME IN SECONDS (3600 ONLY).
      CRTID        IDENTIFIES DD80 OUTPUT (LCPLOT AND DD80).
      DATA         SPECIFIES VALUES OF VARIABLES, TO BE SET AT LOAD TIME.
      DECODE       CONVERTS ARGUMENT LIST FROM ONE FORMAT TO ANOTHER.
/     DIF(K)A,B,C  CAUSES COMPILER TO SKIP ALL STATEMENTS THROUGH
      DIF(K)A,B,C    STATEMENT A, B, OR C IF K IS -, 0, OR +, RESP.
      ENDCLICHE    END OF A CLICHE DEFINED IN A PRECEDING STATEMENT.
      ENDFILE K    WRITES AN END-OF-FILE ON OUTPUT UNIT K.
      ENTRY X      DEFINES LOCATION WHERE SUBROUTINE WILL BE ENTERED
      ENTRY X        WHEN A /CALL X/ STATEMENT IS EXECUTED.
/     EQUIVALENCE  ASSIGNS SAME MEMORY LOCATION TO TWO VARIABLES.
      ERRORS       SETS FLOATING POINT UNDERFLOWS TO ZERO.
      EXIT         ENDS JOB, TRANSFERS CONTROL TO MONITOR.
      EXPF(X)      FINDS EXPONENTIAL FUNCTION OF ARGUMENT X.
      FRAME        RESTORES FRAME ON DD80 PLOT FILM (DD80).
      FSPACE(K)    FORWARD SPACES UNIT K TO NEXT RECORD.
      IF(ECF,-     TESTS TO SEE IF AN END-OF-FILE HAS BEEN READ.
      IF(ICCHECK,- TESTS TO SEE IF A TAPE CHECK ERROR HAS OCCURRED.
      IF(UNIT,-    TESTS TO SEE IF BUFFER OPERATION IS COMPLETE.
      IF(X)A,B,C   CAUSES EXECUTION TO JUMP TO STATEMENT A, B, OR C, IF
      IF(X)A,B,C     X IS -, 0, OR +.  X MAY BE AN EXPRESSION.
      INTEGER X    TYPE SPECIFICATION DEFINING ARGUMENT X AS AN INTEGER.
/     INTF(X)      FINDS INTEGER EQUAL TO OR LESS THAN X.
      LOGMM        MAXIMUM AND POSITIVE MINIMUM ARRAY VALUES (LCPLOT).
      MAPX         SETS DD80 PLOT SCALE POSITIONS AND RANGES (DD80).
      MAX1F(X,Y)   FINDS MAXIMUM VALUE OF FLOATING POINT ARGUMENTS X, Y.
      MIN1F(X,Y)   FINDS MINIMUM VALUE OF FLOATING POINT ARGUMENTS X, Y.
/     MODF(X,Y)    FINDS MODULUS OF FLOATING POINT ARGUMENT X WITH
      MODF(X,Y)      RESPECT TO FLOATING POINT ARGUMENT Y.
      OFFMON       DUMPS MEMORY AND POINTERS ONTO TAPE FOR RESTART.
      ONMON        RESTORES MEMORY AND POINTERS FROM OFFMON TAPE.
      ONMONS       SAME AS ONMON, BUT UNLOADS TAPE BEFORE EXECUTING.
      UODRCP       COPIES CONTROLLEE FILE, EXECUTES COPY.
      UOL206       BUFFERS OUT ARGUMENT LIST ONTO TAPE MW.
      PARAMETER    SPECIFIES CONSTANT VALUES TO BE SUBSTITUTED FOR
      PARAMETER      ARGUMENTS DURING COMPILATION OF PROGRAM.
      PLOTE        COMPLETES DD80 OUTPUT FILE, GIVES TO SYSTEM (DD80).
      POINTC       PLOTS CHARACTERS AT DATA POINTS (DD80).
      PRINT L,     PRINTS ARGUMENT LIST ON ON-LINE PRINTER, USING FORMAT L.
      PUNCH L,     WRITES ARGUMENT LIST ON AN OUTPUT TAPE, TO BE PUNCHED
```

```
      PUNCH L,        OFFLINE, ACCORDING TO FORMAT L (LIMIT 72 CHARACTERS).
      RCCNTR        MAKES CONTOUR PLOT (DD80).
/     READ(K,L)     READS ARGUMENT LIST FROM UNIT K ACCORDING TO FORMAT L.
      REWIND K      REWINDS UNIT K TO LOAD POINT.
      SETCH         SETS CRT PLOT POSITION FOR BCD OUTPUT (DD80).
      SETCHM        DETERMINES DD80 CHARACTER SET (DD80).
      SETLCH        SETS SCALED CRT PLOT POSITION FOR BCD OUTPUT (DD80).
      SETPCH        DETERMINES SIZE, ORIENTATION, TYPE SET OF CHARACTERS
      SETPCH          PLOTTED BY POINTC.
      SINF(X)       FINDS SINE FUNCTION OF ARGUMENT X.
/     TICHEK        RETURNS MACHINE TIME CHARGE AND PERCENTAGES OF THAT
      TICHEK          TIME USED BY CPU, I/O, AND SYSTEM SINCE LAST CALL.
      TICHEK          TIME IN MICROSECONDS, PERCENTS IN BCD.  (6600 ONLY)
      TRACEP        TRACES CURVE BETWEEN DATA POINTS (DD80).
      UNLOAD K      REWINDS AND UNLOADS UNIT K.
      USE X         INSERTS CLICHE X INTO PROGRAM AND COMPILES.
/     WRITE(K,L)    WRITES ARGUMENT LIST ON UNIT K ACCORDING TO FORMAT L.
      WRTEOF(K)     WRITES END-OF-FILE MARK ON UNIT K.
      XABSF(K)      FINDS ABSOLUTE VALUE OF AN INTEGER ARGUMENT K.
      XMAXOF(K,L)   FINDS MAXIMUM VALUE OF TWO INTEGER ARGUMENTS K AND L.
      XMINOF(K,L)   FINDS MINIMUM VALUE OF TWO INTEGER ARGUMENTS K AND L.
      XMODF(K,L)    FINDS MODULUS OF INTEGER K WITH RESPECT TO INTEGER L.
/     X.INT.Y       RESULT HAS POS BITS WHERE BOTH X, Y HAVE POS BITS.
/     X.SHL.Y       RESULT IS X SHIFTED LEFT Y BITS.
/     X.SHR.Y       RESULT IS X SHIFTED RIGHT Y BITS.
/     X.UN.Y        RESULT HAS POS BITS WHERE EITHER X OR Y HAS POS BIT.
/     .LOC.X        MEMORY ADDRESS OF VARIABLE X.
```

7. CHANGES IN TRUMP SINCE MAY 1, 1968.
(CHANGES PRECEDED BY * ARE REQUIRED IN ALL VERSIONS.)

```
  * 5/8/68     CHANGED W(N) TABLE INTERPOLATION FROM LINEAR TO QUADRATIC.
  * 5/8/68     REPLACED SLOW(M9,M2) WITH SLOW(M9), MADE RELATED CORRNS.
  * 5/14/68    CHANGED TALLY TO RESTORE SMALL TO SMALT IF POSSIBLE
    5/14/68       AFTER SMALL REDUCED BELOW SMALT FOR STABILITY.
  * 5/28/68    REPLACED SLOW(M9) WITH F3(M9), CORRECTED NSTORE IN TALLY.
  * 5/28/68    CORRECTED ENTHALPY REFERENCE CORRECTION IN THERM.
    5/28/68    ADDED PATCH TEST OF HMELT INPUT BEFORE ASSIGNING VALUE.
    5/28/68    MOVED ZEROING OF KA, KB WHEN HMELT NOT ZERO TO INPUT.
    7/11/68    PUT LABELLED COMMON ATHERM AND ATALLY INTO BLANK COMMON.
    7/12/68    ADDED HMELTX(M2), TMAX, AND TMIN TO COMMON.
    9/30/68    MOVED HMELTX(M2) TO COMMON LIST USED ONLY IN THERM.
    9/30/68    MODIFIED THERM TO DELETE PHASE CHANGE CALCULATION AND
    9/30/68       TABLE PRINTOUTS OF CONCENTRATIONS FOR NODES OF
    9/30/68       MATERIALS WITH TMELT NOT BETWEEN TMIN AND TMAX.
    11/1/68    ADDED *MAST CONTROL CARD OPTION IN BLOCK 2 INPUT IN
    11/1/68       THERM, AND NEW SUBROUTINE PRIME.  THESE ALLOW DATA
    11/1/68       FOR SELECTED MATERIALS TO BE TRANSFERRED FROM MASTER
    11/1/68       MATERIAL DATA TAPE TO SECONDARY TAPE, WITH OPTIONAL
    11/1/68       UNIT CONVERSION, AND USE OF THE SECONDARY TAPE FOR
    11/1/68       INPUT OF BLOCK 2 ITEMS.
    11/8/68    CHANGED HEART TO PREVENT CALL PLOT WHEN KWIT = -1.
  * 11/11/68   CHANGED MAXIMUM NUMBER OF ITERATIONS FROM 200 TO 80.
    11/15/68   CHANGED THERM TO ALLOW ZERO TEMP INTERVAL IN C AND K
    11/15/68      TABLES, AND MAKE KWIT = 12 FOR NEGATIVE INTERVALS.
    12/4/68    CHANGED THERM, CHEM, FLOW TO ALLOW MATERIALS TO HAVE
    12/4/68       A CHEMICAL REACTANT SPECIFIED WITH KB WHEN HMELT IS
    12/4/68       NONZERO.  IN LATTER CASE, KA IS SET TO ZERO.
    1/8/69     REMOVED STACKS SUBROUTINE, CHANGED HEART AND PRIME SO
    1/8/69        THAT PRIME WOULD DO JOB PREVIOUSLY DONE BY STACKS.
    1/8/69     CHANGED ALL STATEMENTS OF TYPE READ INPUT TAPE AND
    1/8/69        WRITE OUTPUT TAPE INTO READ(M,N) AND WRITE(M,N) FORMS.
    1/8/69     PUT LABELLED COMMON BLOCKS AGEN, ACHEM, ACHEMS, AFINK,
    1/8/69        AFLOW, AFLOWS, ASURE, ASURES, ASPECK, AND APLOT
    1/8/69        INTO SUBROUTINES WHICH USE THEM.  ADDED LABELLED
    1/8/69        COMMON APRIM TO SUBROUTINE PRIME.
    1/8/69     REPLACED CLOCK1 CALLS WITH TICHEK CALLS IN 6600 VERSIONS.
    1/8/69     REMOVED SENSE SWITCH 3 TEST OPTION IN HEART/6600.
    1/8/69     ADDED IRITE, ITAPE SPECIAL OUTPUT OPTION TO TALLY/STD.
    1/9/69     ADDED DIF STATEMENTS TO ALLOW CHOICE OF SELECTION OF
    1/9/69        OPTIONAL FEATURES IN TRUMP TO BE MADE AUTOMATICALLY
    1/9/69        DURING COMPILATION ACCORDING TO THE VALUES OF THE
```

```
  1/9/69        INTEGERS IN THE PARAMETER STATEMENT IN CLICHE STORAGE.
  1/10/69    ADDED DUMMY ARGUMENT FIN TO ALL WRITE STATEMENTS WITH
  1/10/69       NO ARGUMENT LIST, TO AVOID INCOMPATIBILITY BETWEEN
  1/10/69       LRLTRAN AND CHAT ON USE OF /FIN/ STATEMENT.  FIN IS
  1/10/69       MADE A BLANK WORD IN /DATA/ STATEMENT BELOW, AND
  1/10/69       SPECIFICATION /A1/ ADDED TO CORRESPONDING FORMATS.
  1/10/69    CHANGED NSTORE CALCULATION TO SHOW TOTALS DEPENDING ON
  1/10/69       EACH PARAMETER WHICH MAY BE MADE ZERO.
  1/14/69    ADDED PARAMETER KPRIME TO STORAGE, TO ALLOW OPTION OF
  1/14/69       COMPILING CONTENTS OF SUBROUTINE PRIME OR NOT.
  1/24/69    ADDED ARRAY CORD(M3) TO COMMON ACHEM IN CHEM, INPUT
  1/24/69       IN BLOCK 3, TO SPECIFY REACTION ORDER P IN
  1/24/69       RATE EQUATION DA/DT = - A**P *EXP(Z - E/RT).
  1/30/69    ADDED SUBROUTINE TIMER FOR USE IN HEART AND TALLY TO GET
  1/30/69       MACHINE TIME CHARGED TO EACH PROBLEM.
  1/30/69    REPLACED CLICHE STORAGE WITH CLICHES STORK, STORM, AND STORE.
  1/30/69    PUT ALL PARAMETER STATEMENTS INTO CLICHE STORK.
  2/10/69    ADDED NEW CONTINUATION PROBLEM OPTION.  4 IN COLUMN 72
  2/10/69       OF PROBLEM NAME CARD LIKE OPTION 3, EXCEPT TAU IS SET
  2/10/69       TO SUMTIM OF PRECEDING PROBLEM.
  2/14/69    ADDED INPUT OPTION TIMEP IN BLOCK 1, TO GET PRINTOUTS AT
  2/14/69       SPECIFIED PROBLEM TIME INTERVALS.
* 2/14/69    ADDED NEW CONVERGENCE CRITERION FOR ITERATION SCHEME, SO
  2/14/69       ERROR IN TEMPERATURE CHANGE OF ANY FINITE-VOLUME NODE
  2/14/69       CANNOT BE MORE THAN 10 TIMES AVERAGE ERROR.
* 2/15/69    ADDED NEW CONTROL TO REDUCE DELT WHENEVER MORE THAN 40
  2/15/69       ITERATIONS ARE REQUIRED FOR CONVERGENCE IN SPECK.
  2/17/69    ADDED NEW VARIABLES DTEMP, DELTG, RATG, RATS, MSS,
  2/17/69       CHANGED METHOD OF CALCULATION OF RAT1, RAT2, DELTSS,
  2/17/69       DELTSX, FOR, AND STEADY STATE CRITERIA, TO FURTHER
  2/17/69       OPTIMIZE CONTROL OF THE TIME STEP AND ACCURACY OF THE
  2/17/69       INITIAL GUESS FOR THE ITERATION SCHEME.
  2/20/69    ADDED DTEMP, DTMAX, MSS, NUTS, NUTSUM, NUTX TO PRINTOUTS.
* 2/22/69    ADDED SAFEGUARDS TO CHEM REACTION CALCULATION TO PREVENT
  2/22/69       DIVISION BY ZERO, NEGATIVE CONCENTRATIONS.
  2/25/69    CHANGED PLOT TO MAKE ANY CONVERSION OF LOG SCALES TO
  2/25/69       LINEAR ONLY TEMPORARY, AND TO ALWAYS INCLUDE ANY
  2/25/69       NEGATIVE TEMPERATURES IN PLOT OF MAXIMUM AND MINIMUM
  2/25/69       NODE TEMPERATURES VS NODE LOCATION.
  3/3/69     CHANGED TALLY, PLOT TO GET CRT PLOTS AT TIME INTERVALS
  3/3/69        TIMEP, IN ADDITION TO THOSE EVERY JPIC TIME STEPS.
  3/6/69     REMADE ONMON RESTART TAPES CE403 AND CE404 FOR TRUMP/3600,
  3/6/69        CE408 AND CE409 FOR TRUMP/6600/STD, AND CE411 AND CE412
  3/6/69        FOR TRUMP/6600/DS.
  3/11/69    ADDED VARIABLES NOSR, NOSRT, KR, LASR TO PLOT,  CHANGED
  3/11/69       TO PRODUCE BOTH LOG AND LINEAR NODE LOCATION SCALES
  3/11/69       WHEN LOGR IS NEGATIVE, ELIMINATE ALL DUPLICATE PLOTS
  3/11/69       WHEN LOG SCALES ARE REJECTED.
  3/11/69    CHANGED SEQUENCE OF DELT ADJUSTMENTS IN TALLY TO IMPROVE
  3/11/69       EFFICIENCY, LOGIC.  REMADE ALL ONMON RESTART TAPES.
  3/12/69    CHANGED PLOT, HEART TO ALLOW JPIC = 0, STILL GET PLOTS ON
  3/12/69       CYCLES 0, 1, AND LAST, AND WHEN DETERMINED BY TIMEP.
  3/18/69    ADDED OPTION OF ADDING 1000 TO LOGR, TO MAKE INFRA 1, TO
  3/18/69       MULTIPLY NODE LOCATION SCALE BY 0.5*SUMTIM**(-0.5).
  3/22/69    CHANGED PLOT TO ELIMINATE DUPLICATE TABLES.
  3/22/69    REMADE 3600 RESTART TAPES CE403 AND CE404, AND 6600 RESTART
  3/22/69       TAPES CE408 AND CE409.
* 4/3/69     CHANGED FLOW TO ALLOW SPECIFICATION OF NEGATIVE FLOW RATES.
* 4/4/69     CHANGED FLOW TO CORRECT ERROR PROPORTIONAL TO FRACTIONAL CHANGE
  4/4/69        IN HEAT CAPACITY OF FLOWING MATERIAL, WHEN CAPT TABULATED.
  4/4/69     CHANGED METHOD OF CALCULATION OF FLINT AND FLOUT IN FLOW TO
  4/4/69        ELIMINATE ACCUMULATION OF NUMERICAL TRUNCATION ERRORS.
  4/4/69     CHANGED METHOD OF CALCULATION OF ZIP IN THERM, FINK, FLOW, AND
  4/4/69        SURE TO ELIMINATE ACCUMULATION OF NUMERICAL TRUNCATION
  4/4/69        ERRORS AFFECTING FOURTH OR FIFTH SIGNIFICANT FIGURE.
  4/4/69     REMADE ALL ONMON RESTART TAPES.
* 4/18/69    CORRECTED ERROR IN REFER CAUSING INCORRECT REFERENCES IN A
  4/18/69       CONTINUATION PROBLEM WITH A NEW BLOCK 2, 3, 4, OR 7 WITH
  4/18/69       A DIFFERENT NUMERICAL SEQUENCE OF IDENTIFICATION NUMBERS,
  4/18/69       AND USING OLD BLOCK 1, 4, 5, 6, 8, 9, 10, 11, OR 12.
  4/18/69    REMOVED ARRAYS DEPENDENT ON M8 FROM BLANK COMMON, PUT INTO
```

```
 4/18/69      GEN, THERM, AND TALLY AS LABELLED COMMON AGENS.
 4/18/69   REMOVED ARRAYS DEPENDENT ON M12 FROM BLANK COMMON, PUT INTO
 4/18/69      THERM AS LABELLED COMMON AREM, INTO THERM, CHEM, GEN,
 4/18/69      FLOW, AND SURE AS LABELLED COMMON AREMS.
 4/18/69   REMOVED COMMON AFINK FROM FINK, PUT INTO BLANK COMMON.
 4/18/69   REMOVED COMMON ASPECK FROM SPECK, PUT INTO BLANK COMMON.
 5/2/69    CHANGED TIM, PIX CALCULATION IN TALLY TO SPEED UP.
 5/20/69   CHANGED ALL TABLE SLOPE CALCULATIONS TO ALLOW TWO CONSECUTIVE
 5/20/69      TABLE TEMPERATURES OR TIMES TO BE THE SAME, BUT MAKE
 5/20/69      KWIT = 12 IF IN DESCENDING ORDER.
*3/6/70    ADDED 1.0E-12 to HMX FOR ZV NODE IN THERM2,
 3/6/70      TO PREVENT DIVISION BY ZERO, RESULTING IN A
 3/6/70      DUMP.
 7/22/70   CHANGED HEART TO END PROBLEM SOONER IF
 7/22/70     KWIT = 11 OR 12.
*2/4/71    CORRECTED BRANCHES IN TALLY TO ALLOW
 2/4/71      OUTPUT ON TAPE ITAPE EVERY IRITE TIME STEPS.
 2/4/71      ADDED HEADER RECORD CONTAINING PROBLEM
 2/4/71      NAME TO OUTPUT ON TAPE ITAPE.
*2/4/71    CHANGED FORMAT OF OUTPUT OF NEWBL ARRAY
 2/4/71       IN TALLY, TO PREVENT OVERFLOW.
*4/14/71   ADDED TEST IN TALLY TO SEE IF BLOCK 7 IS
 4/14/71     MISSING WHEN BLOCK 6 IS USED.  CHANGED
 4/14/71     FORMAT IN HEART TO LIST THE ERROR UNDER
 4/14/71     KWIT = 9.
 7/1/71    CHANGED THERM, FINK, AND SURE TO CORRECT
 7/1/71      THE CALCULATION OF VOL(N), AREA(N), AND
 7/1/71      AREAS(N), RESP., WHEN KD IS 3 AND SCALE IS
 7/1/71      NOT 1.0.
*7/1/71    CHANGED TALLY TO CORRECT THE TESTS FOR
 7/1/71      PRODUCING OUTPUT BASED ON TIMEP, WHICH
 7/1/71      FAILED IF SUMTIM WAS NEGATIVE.  DELETED
 7/1/71      THE *ASSIGN8 OPTION TO PRODUCE TWO DISK
 7/1/71      FILES IN TWO CABINETS.
 7/1/71    ADDED AN OPTION FOR THE USER TO KEEP THE
 7/1/71      DD80 OUTPUT FILES AT THE END OF THE RUN,
 7/1/71      BY PUTTING A CARD WITH *ASSIGN8 IN
 7/1/71      COLUMNS 1-8 IN FRONT OF THE FIRST PROBLEM
 7/1/71      NAME CARD.  THE DD80 FILES WILL HAVE
 7/1/71      8-CHARACTER NAMES, BEGINNING WITH TRUMPD.
 7/1/71    REMADE ONMON RESTART TAPES FOR
 7/1/71      TRUMP/6600/STD (CE408 AND CE409), AND FOR
 7/1/71      TRUMP/6600/DS (CE411 AND CE412).
*2/10/72   ADDED 1.0E-24 TO DIVISORS CONTAINING BET IN
 2/10/72     THERM, TO AVOID DIVISION BY ZERO FOLLOWING
 2/10/72     CERTAIN TYPES OF INPUT DATA ERRORS.
*2/10/72   ADDED A TEST IN FINK TO AVOID INCREMENTING
 2/10/72     ARRAY FI AFTER TIME STEPS FOR WHICH
 2/10/72     KWIT WAS -1.
*2/10/72   CORRECTED ROUTINES SEEK1 AND SEEK2 TO
 2/10/72     SET K = 0 WHEN THE NUMBER OR NUMBERS
 2/10/72     SEARCHED FOR ARE NOT FOUND, WHEN AN
 2/10/72     INPUT DATA BLOCK OF TYPE B IS USED.  THESE
 2/10/72     ROUTINES PREVIOUSLY ALWAYS SET K = 1
 2/10/72     WHICH COULD CAUSE ERRORS UNDER CERTAIN
 2/10/72     IMPROBABLE CIRCUMSTANCES.
 4/7/72    ADDED ARRAY DFB, AND CODING IN FINK AND SURE, TO PRINT
             HEAT FLOW RATES FOR LAST TIME STEP FOR INTERNAL CONNECTIONS,
             EXTERNAL CONNECTIONS.
```

MAN.

H. TRUMP PROGRAM LISTING

```
*     FORTRAN            TRUMP/MASTER/MAIN                                  *  11
C********VERSION 5/20/69.                                                   *  12
CCC                                                                         *  13
CCC   DEVELOPED BY ARTHUR L. EDWARDS, LAWRENCE RADIATION LABORATORY,        *  14
CCC   P.O. BOX 808, LIVERMORE, CALIFORNIA, L-437, BLDG T1C2, RM 115,        *  15
CCC   PHONE 447-1100 EXTENSION 7886.                                        *  16
CCC                                                                         *  17
C********PRESENTLY AVAILABLE SUBROUTINES INCLUDE HEART, THERM, FINK,        *  18
C********     SPECK, TALLY, PATCH, REFER, SEEK1, AND SEEK2 (REQUIRED),      *  19
C********     KITTY, GEN, CHEM, FLOW, SURE, PLOT, PRIME, AND TIMER (OPT)    *  20
CCC                                                                         *  21
C********LANGUAGE IS LRLTRAN (LRL VERSION OF FORTRAN), AND CHAT.            *  22
CCC                                                                         *  23
C********STATEMENTS MOST LIKELY REQUIRING CHANGES FOR OTHER LANGUAGES...    *  24
C********     PRINT, PUNCH, .INT., .UN., DIF,                               *  25
C********     PARAMETER, CLICHE, DECODE, SUBSCRIPTED SUBSCRIPTS,            *  26
C********     STATEMENTS OF TYPE /A=B=C/, IF TESTS WITH ARGUMENT OF TYPE    *  27
C********     /N-1H*/, DO-LOOP INDICES USED OUTSIDE LOOP, AND FORMATS OF    *  28
C********     TYPES E AND 1PE USED INTERCHANGABLY.                          *  29
C********STATEMENT LABELS USED IN DIF STATEMENTS DELIMIT GROUPS OF          *  30
C********     STATEMENTS WHICH ARE TO BE COMPILED OR NOT DEPENDING ON       *  31
C********     THE VALUE OF THE PARAMETER IN THE INITIAL DIF STATEMENT.      *  32
C********     THIS ALLOWS SELECTION OF OPTIONAL PARTS OF THE PROGRAM TO     *  33
C********     BE DETERMINED BY THE PARAMETERS IN CLICHE STORK.              *  34
CCC                                                                         *  35
C********ALL CLICHES USED IN PROGRAM ARE DEFINED IN MAIN.                   *  36
CCC                                                                         *  37
C********STORK USED IN MAIN, HEART, THERM, GEN, CHEM, FINK, FLOW, SURE,     *  38
C********     SPECK, TALLY, PLOT, PRIME, PATCH, KITTY, AND TIMER.           *  39
      CLICHE STORK                                                          *  40
C********PARAMETERS CONTROLLING SIZE OF VARIABLE ARRAYS AND COMPILATION     *  41
C********     OF CORRESPONDING PARTS OF THE PROGRAM.  .   .   .   .   .     *  42
C********     M1 = CRT PLOT TIMES (BLOCK 11), M2 = MATERIALS (BLOCK 2),     *  43
C********     M3 = REACTANTS (BLOCK 3), M4 = NODES (BLOCKS 4 AND 9),        *  44
C********     M5 = INTERNAL CONNECTIONS (BLOCK 5),                          *  45
C********     M6 = SURFACE CONNECTIONS (BLOCK 6), M7 = BOUNDARY NODES       *  46
C********     (BLOCK 7), M8 = HEAT GENERATING NODES (BLOCKS 8, 1 AND 9),    *  47
C********     M9 = TABLE LENGTH (BLOCKS 2, 3, 6, 7, 8, AND 10),             *  48
C********     M10 = MASS FLOW CONNECTIONS (BLOCK 10), M11 = CRT PLOT        *  49
C********     NODES (BLOCK 11), AND M12 = NODES WITHPROPERTIES DEPENDENT    *  50
C********     ON REMOTE TEMPERATURES (BLOCK 12).                            *  51
C********TO ELIMINATE BLOCK 3, SET M3 = 0, RECOMPILE MAIN, THERM, CHEM,     *  52
C********     FLOW (IF M10 NOT 0), AND TALLY.                               *  53
C********TO ELIMINATE BLOCKS 6 AND 7, SET M6 = 0, M7 = 0, AND RECOMPILE     *  54
C********     MAIN, SURE, AND SPECK.                                        *  55
C********TO ELIMINATE BLOCK 8 (AND CONSTANT G SPECIFIED IN BLOCKS 1 OR      *  56
C********     9), SET M8 = 0, AND RECOMPILE MAIN, THERM, GEN, AND TALLY.    *  57
C********TO ELIMINATE BLOCK 10, SET M10 = 0, AND RECOMPILE MAIN, FLOW,      *  58
C********     AND SPECK.                                                    *  59
C********TO ELIMINATE BLOCK 11, SET M1 = 0, M11 = 0, AND RECOMPILE MAIN     *  60
C********     AND PLOT.                                                     *  61
C********TO ELIMINATE BLOCK 12, SET M12 = 0, AND RECOMPILE MAIN, THERM,     *  62
C********     CHEM (IF M3 NOT 0), GEN (IF M8 NOT 0), FLOW (IF M10 NOT 0)    *  63
C********     AND SURE (IF M6 IS NOT 0).                                    *  64
CCC                                                                         *  65
C********PARAMETERS CONTROLLING COMPILATION OF MACHINE-SPECIFIC            *  66
C********     STATEMENTS AND SOME OPTIONAL PARTS OF PROGRAM.....            *  67
C********     MACHINE IS 0 FOR 3600, 1 FOR 6600 OR 7600.                    *  68
C********     KPRIME IS POSITIVE TO COMPILE SUBROUTINE PRIME.               *  69
C********     KPLOTR IS POSITIVE TO COMPILE TEMP VS RADIUS OPTION IN        *  70
C********     SUBROUTINE PLOT.  (M1 AND M11 MUST ALSO BE POSITIVE.)         *  71
C********     KPLOTT IS POSITIVE TO COMPILE TEMP VS TIME OPTION IN          *  72
C********     SUBROUTINE PLOT.  (M1 AND M11 MUST ALSO BE POSITIVE.)         *  73
C********     KPLOTL IS POSITIVE TO COMPILE RADIUS VS TIME OPTION IN        *  74
C********     SUBROUTINE PLOT (KPLOTT, M1, AND M11 MUST ALSO BE POSITIVE    *  75
C********PARAMETERS DETERMINING MACHINE VERSION DESIRED.  MAKE ALL ZERO     *  76
C********     EXCEPT ONE DESIRED.                                           *  77
      PARAMETER (MV36ST = 0, MV66ST = 0, MV66DS = 1)                        *  78
CCC   PARAMETER (MV36ST = 0, MV66ST = 1, MV66DS = 0)                        *  79
```

MAIN.

```
CCC   PARAMETER (MV36ST = 1, MV66ST = 0, MV66DS = 0)
C********PARAMETER DETERMINING COMPILATION OF KITTY FOR FOR SPECIAL
C********     CONTROLLE VERSION OF TRUMP (0 TO SKIP, 1 TO COMPILE).
      PARAMETER (KITTEN = 0)
      DIF(MV36ST)982,982,981
981   CONTINUE
C********VERSION 3600/STD
      PARAMETER(M1 = 200, M2 = 25, M3 = 4, M4 = 250, M5 = 600, M6 = 125,
     1   M7 = 20, M8 = 50, M9 = 12, M10 = 80, M11 = 12, M12 = 250)
      PARAMETER(MACHINE = 0,KPRIME = 1,KPLOTL = 1,KPLOTR = 1,KPLOTT = 1)
C--------ASSIGN CDC-3600 BANK 1 STORAGE.
      BANK HEART(1)
      BANK THERM(1)
      BANK GEN(1)
      BANK AGEN(1)
      BANK FINK(1)
      BANK AGENS(1)
      BANK FLOW(1)
      BANK AFLOW(1)
      BANK AFLOWS(1)
      BANK SURE(1)
      BANK ASURE(1)
      BANK ASURES(1)
      BANK SPECK(1)
      BANK AREM(1)
      BANK TALLY(1)
      BANK PLOT(1)
      BANK APLOT(1)
      BANK THERM1(1)
      BANK THERM2(1)
      BANK TALLY1(1)
      BANK AREM(1)
982   CONTINUE
      DIF(MV66ST)984,984,983
983   CONTINUE
C********VERSION 6600/STD
      PARAMETER(M1= 200, M2=  40, M3=  5, M4=  480, M5= 1200, M6=  125,
     1          M7=  20, M8= 100, M9= 12, M10=  80, M11=  12, M12= 480)
      PARAMETER(MACHINE = 1,KPRIME = 1,KPLOTL = 1,KPLOTR = 1,KPLOTT = 1)
984   CONTINUE
      DIF(MV66DS)986,986,985
985   CONTINUE
C********VERSION 6600/DS
      PARAMETER (M1=200, M2= 50, M3=  0, M4= 600, M5=1750, M6=150,
     1  M7= 20, M8=150, M9= 12, M10=  0, M11=  16, M12= 600)
      PARAMETER(MACHINE = 1,KPRIME = 1,KPLOTL = 0,KPLOTR = 0,KPLOTT = 1)
986   CONTINUE
C--------END OF CLICHE STORK.
      ENDCLICHE
      USE STORK
CCC
C********STORE USED IN MAIN, HEART, THERM, GEN, CHEM, FINK, FLOW, SURE,
C********     SPECK, TALLY, PLOT, PRIME, REFER, SEEK1, AND SEEK2.
      CLICHE STORE
C********BLANK (UNLABELLED) COMMON BLOCKS.
CCC   INTEGER VARIABLES CONTROLLING I/O CHANNELS, DATA INPUT OPTIONS.
      COMMON  KLUDGE, NRS, NR, NB, MW, M, NDATA, IBLOCK, MOD, KWIT,
     1        ITEMS(15), NEWBL(15), NAME(20)
C********FIN IS BLANK WORD ADDED TO ALL FINAL LINES OF OUTPUT LISTS NOT
C********     HAVING ANY OTHER LIST ELEMENTS, TO AVOID INCOMPATIBILITY
C********     BETWEEN VARIOUS LANGUAGES IN USE AT LRL.
      DATA (FIN = 0H)
C--------END OF CLICHE STORE.
      ENDCLICHE
      USE STORE
CCC
C********STORM USED IN MAIN, HEART, THERM, GEN, CHEM, FINK, FLOW, SURE,
C********     SPECK, TALLY, AND PLOT.
      CLICHE STORM
C********COUNTERS FOR BLOCK ITEMS, VARIOUS TYPES OF INPUT.
C********THIS COMMON STATEMENT MUST NOT BE ALTERED IN ANY WAY.
      COMMON  NCSPEC(1),NMAT,NKEM,NODES,NOCON,NOSCON,NOOBS,NVARG,NIT,
     1        NCFLOW,NUPL,NTABS,NVARC,NVARK,NMELT,NREACT,NVARQ,
```

MAN.

```
     2        NVARZ,NVARE,NORAD,NVARH,NORADS,NOPOWS,NVART,NVARFL
CCC   OTHER INTEGER VARIABLES USED IN TWO OR MORE SUBROUTINES.
      COMMON  JPIC,KCYC,KC,KDATA,KSECS,KSYM,NOGEN,NOW,NPRCB,NUP,NUTS
CCC   FLOATING POINT VARIABLES USED IN TWO OR MORE SUBROUTINES.
      COMMON  ALONE, BONE, CAPS, CLOCKA, CLOCKB, DELT, DELTS,
     1        OSTAB, DTMAX, FONE, FOR, GEOM, GONE, HONE, PONE,
     2        RONE, SCALE, SIGMA, SMALL, SUMTIM, TAU, TBASE,
     3        TMAX, TMIN, TONE, TVARY
CCC
C********REMAINDER OF CLICHE STORM IS NOT NEEDED IN SUBROUTINE HEART.
CCC
C********OPTIONAL ARRAYS NOT NEEDED FOR TEMPERATURE CALCULATION ARE...
C********     DF, DFI, DFS, F, FB, FI, FS, H, FLAPS, FLINT, FLIPS,
C********     FLOPS, FLOUT, AND IF SUBR FLOW NOT USED, W AND WT, AND IF
C********     SUBROUTINE PRIME NOT USED, MATP AND MATS.
C********ARRAYS ONLY NEEDED FOR REMOTE TEMPERATURE DEPENDENCE OPTION ARE
C********     NLOOK, NODP1, NODP2, NOXP1, NOXP2, AND NPROP.
C********ARRAYS ONLY NEEDED FOR VARIOUS CRT PLOTTING OPTIONS ARE...
C********     NODEP, NOXEP, PRAD, PTEMP, PTIME, RADIUS, RPLOT, TCONT,
C********     TMAXP, TMINP, AND TPLOT.
C********ARRAYS USED ONLY TO SAVE CALCULATION TIME IN TIME STEPS ARE...
C********     SLOC, SLOE, SLOFL, SLOG, SLOH, SLOK, SLOQ, SLOT, AND SLOZ.
CCC
CCC   VARIABLE ARRAYS USED IN TWO OR MORE SUBROUTINES (REQUIRED).
      COMMON  HMELT(M2), LTABK(M2),
     1        A(M4), CAP(M4), CON(M4), DA(M4), DDA(M4), DOT(M4),
     2        DF(M4), DT(M4), HEFT(M4), NODE(M4),
     3        NODMAT(M4), NTYPE(M4), RADIUS(M4), SLIM(M4), T(M4),
     4        VOL(M4), W(M4), ZIP(M4),
     5        DFI(M5), FI(M5), NOD1(M5), NOD2(M5), TRAN(M5),
     6        F1(M9), F2(M9), F3(M9), NX1(10), NX2(10), NX3(10), NX4(10)
C********USED ONLY IN THERM.
C********COULD BE REPLACED BY A DIMENSION STATEMENT IN THERM.
CCC (FORMERLY LABELLED COMMON ATHERM IN UCRL-14754 REV 1 OF MAY 1,1968)
      COMMON  AMAT(M2), CAPT(M9,M2), CONT(M9,M2), DENS(M2), HMELTX(M2),
     1        LTABC(M2), MAT(M2), SLOC(M9,M2), SLOK(M9,M2),
     2        TMELT(M2),TVARC(M9,M2),TVARK(M9,M2),WT(M9,M2),NOXMAT(M4)
C********USED ONLY IN TALLY.
C********COULD BE REPLACED BY A DIMENSION STATEMENT IN TALLY.
CCC (FORMERLY LABELLED COMMON ATALLY IN UCRL-14754 REV 1 OF MAY 1,1968)
      COMMON  AA(M4), F(M4), H(M4), NOTE(M4),NOXE(M4), NTYPES(M4),TT(M4)
C********USED ONLY IN FINK.
C********COULD BE REPLACED BY A DIMENSION STATEMENT IN FINK.
CCC (FORMERLY LABELLED COMMON AFINK  IN UCRL-14754 REV 1 OF MAY 1,1968)
      COMMON  AREA(M5), DEL1(M5), DEL2(M5), HINT(M5),
     1        NOX1(M5), NOX2(M5), RINT(M5)
C********USED ONLY IN SPECK.  COULD BE REPLACED BY A DIMENSION STATEMENT
CCC (FORMERLY LABELLED COMMON ASPECK IN UCRL-14754 REV 1 OF MAY 1,1968)
      COMMON  ERROR(M4), ERRORX(M4)
C--------END OF CLICHE STORM.
      ENDCLICHE
C********CLICHE STORM USED HERE ONLY TO CHECK DURING COMPILATION.
      USE STORM
C********NORMAL INPUT AND OUTPUT UNITS.
      NRS = 2
      MW = 3
CCC   IDENTIFY OUTPUT WHEN LOADING FROM FORTRAN, BINARY, OR CHAIN TAPE.
      COMDATE = 8H 5/20/69
      WRITE (   MW,5800 ) COMDATE
      PRINT 5800 , COMDATE
      CALL HEART
C********RETURNS HERE ONLY WHEN /*SPLIT/ CARD IS READ IN FROM DATA DECK.
C********EXIT RELEASES FILES, I/O CHANNELS, RETURNS CONTROL TO MONITOR.
5800  FORMAT(10(1H*),19HCOMPILATION DATE IS,2X,A8)
      CALL EXIT
      END

*     CARDS COLUMN       KITTY
*     LIST 8
*     FORTRAN            KITTY/6600 (ENTRY POINT FOR TRUMP CONTROLLEE.)
      SUBROUTINE KITTY
C********VERSION 4/18/69.
```

KITTY

KITTY

HEART

```
C********COMPILED ONLY FOR CONTROLLEE VERSION FOR LIVERMORE TIMESHARING
C********     SYSTEM, WHEN PARAMETER KITTEN IS POSITIVE.
C********INITIAL ENTRY POINT WHEN EXECUTED ON 6600 WITH * CONTROLLEE CAR
      USE STORK
      USE STORK
      DIF(KITTEN)982,982,981
981   CONTINUE
      DIF(MACHINE)92,92,91
91    CONTINUE
      COMMON /GOBCOM/-INTEGER-X(3)
      DATA (FIN = 0H)
CCC   COPY TRUMP INTO +TRUMP, MAKE +TRUMP NEW CONTROLLEE.
      CALL COPRCP(5RTRUMP,6R+TRUMP,IER)
      IF(IER)4700, 6100,4700
 6100 CONTINUE
CCC   GET DATA FILE NAME FROM TELETYPE XEQ LINE.
      TYPE ADDRESS ZERO
      ZERO = 0
      NSYMBOL = (10048.SHL.18).UN.(.LOC.NSYM)
      NSYM = (20018.SHL.48).UN.((.LOC.BAD).SHL.30).UN.(10CB.SHL.18).UN.
     1   (.LOC.NFILE)
      X = NSYMBOL
      GO TO ZERO
  BAD GO TO 4700
      GO TO BAD
      IF(NFILE - 3RTTY)2100, 6105,2100
 6105 CALL ASSIGN(2,3)
CCC   LOGICAL I/O CHANNEL 2 IS ASSIGNED TO USERS TTY.
CCC   LOGICAL I/O CHANNEL 59 IS ASSIGNED TO USERS TTY.
      WRITE (   59,5800 )    FIN
      GO TO 2105
2100  CALL ASSIGN(2,0,NFILE,0)
CCC   LOGICAL I/O CHANNEL 2 IS ASSIGNED TO DISK FILE NFILE, NFILE OPENED
      CALL ASSIGN(14,16)
C ASSIGN /PUNCH/ OUTPUT TO SYSTEM PUNCH FILE.
2105  CALL HEART
      GO TO 4705
4700  WRITE (   59,5805 )    FIN
4705  CALL EXIT
5800  FORMAT(10X,17HSTART TYPING DATA,A1)
5805  FORMAT(10X,32HRETYPE, FORMAT  TRUMP DATA / T V,A1)
92    CONTINUE
982   CONTINUE
      RETURN
      END
```

```
*     CARDS COLUMN       HEART
*     LIST 8
*     FORTRAN            HEART   TRUMP LOGIC CONTROL SUBROUTINE.
      SUBROUTINE HEART
C********VERSION 5/20/69.
C********HEART IS A REQUIRED SUBROUTINE.
      INTEGER ABLOCK
      DIMENSION ABLOCK(12), NAMES(20)
      USE STORK
      USE STORE
      USE STORM
CCC
      NAMES(1) = 0H
      KLUDGE = 6600
      IF(MACHINE) 6100, 6100,2100
 6100 KLUDGE = 3600
C********CLOCK  CALLS IN HEART ARE OPTIONAL, FOR INFORMATION ONLY.
2100  CALL CLOCK(HOUR,DATE)
      NOFF = 1
C********ERRORS CAUSES ALL UNDERFLOWS (LARGE NEGATIVE EXPONENT) TO BE
C********SET TO ZERO.  THIS IS REQUIRED.
2105  CALL ERRORS
C********CC80 CRT SUBROUTINES CRTID, FRAME, SETCHM, SETCH, AND PLOTE
C********USED ONLY AT LRL.  REMOVE IF SUBROUTINE PLOT IS REPLACED WITH A
C********DUMMY SUBROUTINE.  FIND SUBSTITUTES IF PLOTTING OPTIONS ARE
C********TO BE KEPT.
      CALL CRTID(2HAE,1)
```

HEART

HEART

```
      CALL FRAME                                                        * 298
      CALL SETCHM(1)                                                    * 299
      CALL SETCH( 6.0,64.0,1,0,0,0)                                     * 300
C********REMCVE FOLLCWING STATEMENT IF OFFMON-ONMON OPTION IS REMCVED.  * 301
      IF(NOFF) 6105,4400, 6105                                          * 302
 6105 CONTINUE                                                          * 303
C********START HERE FOR NEW SERIES OF PROBLEMS.                         * 304
2110  NPRCB = 0                                                         * 305
      NR = NRS                                                          * 306
      M = MW                                                            * 307
      NSAVE = NCATA = 0                                                 * 308
CCC   INITIALIZE PROBLEM TIMING, SET TOTAL TIME CHARGED TC ZERO.        * 309
      CALL TIMER(NPER,MICS)                                             * 310
      KSECS = 0                                                         * 311
CCC   START HERE FOR EACH ACCITIONAL PROBLEM IN A STACK OF CATA DECKS.  * 312
2115  NPRCB = NPROB + 1                                                 * 313
      KCYC = -1                                                         * 314
CCC   INITIALIZE CONTROL PARAMETERS.                                    * 315
      IBLCCK = KWIT = NSTOP = NDISC = NDUMP = NOFF = NOW = NCHECK = 0   * 316
      DO 2120  N = 1,15                                                 * 317
      NEWBL(N) = 0                                                      * 318
 2120 CONTINUE                                                          * 319
      CALL SETCH( 6.0,64.0,1,0,0,0)                                     * 320
      WRITE (    M,5890 )    FIN                                        * 321
CCC   SAVE CLD MACHINE, HOUR IDENTIFIER, GET NEW IDENTIFIER.            * 322
      CB = CLCCKB                                                       * 323
      CA = CLCCKA                                                       * 324
      CALL CLCCK(CLOCKB,CLOCKA)                                         * 325
CCC   FIND TCTAL TIME CHARGEC TO PROBLEM.                               * 326
      CALL TIMER(NPER,MICS)                                             * 327
      KSECS = KSECS + MICS                                              * 328
      WRITE (    M,5845 )CLOCKA,KSECS,NPER                              * 329
      WRITE (   MW,5845 )CLOCKA,KSECS,NPER                              * 330
      PRINT        5845 ,CLOCKA,KSECS,NPER                              * 331
      PRINT 5885                                                        * 332
      WRITE (    M,5805 )    FIN                                        * 333
      WRITE (   MW,5805 )    FIN                                        * 334
C--------READ IN CONTROL CARCS AND PROBLEM NAME CARO.                   * 335
 2125 READ  (   NR,5875 ) NAME                                          * 336
      WRITE (   MW,5875 )NAME                                           * 337
      PRINT 5875 , NAME                                                 * 338
      IF(NAME(1) - 1H*)2125, 6110,2125                                  * 339
 6110 IF(NAME(2) - 5HSPLIT) 6115,4600, 6115                             * 340
 6115 IF(NAME(2) - 5HDISCO) 6120,2125, 6120                             * 341
 6120 IF(NAME(2) - 5HDECKS)2130, 6125,2130                              * 342
 6125 NSAVE = 1                                                         * 343
      DIF(MACHINE)972,972,971                                           * 344
 971  CONTINUE                                                          * 345
      WRITE (63,5905) NAME(3)                                           * 346
      READ  (63,5910) NCATA                                             * 347
 972  CONTINUE                                                          * 348
      DIF(MACHINE)973,973,974                                           * 349
 973  CONTINUE                                                          * 350
  910 FORMAT(I2,1X,R5)                                                  * 351
      DECCDE 5905,(NAME(3)),910,(NDATA)                                 * 352
 974  CONTINUE                                                          * 353
      WRITE (   MW,5940 )NCATA                                          * 354
      PRINT 5940 ,NDATA                                                 * 355
      GO TC 2125                                                        * 356
 2130 IF(NAME(2) - 5HOFFMO) 6130,4320, 6130                             * 357
 6130 IF(NAME(2) - 5HINPUT)2135, 6135,2135                              * 358
 6135 DIF(MACHINE)976,976,975                                           * 359
 975  CONTINUE                                                          * 360
      WRITE (63,5905) NAME(3)                                           * 361
      REAC  (63,5910) NR                                                * 362
 976  CONTINUE                                                          * 363
      DIF(MACHINE)977,977,978                                           * 364
 977  CONTINUE                                                          * 365
      DECODE 5905,(NAME(3)),910,(NR)                                    * 366
 978  CONTINUE                                                          * 367
      GO TC 2125                                                        * 368
 2135 IF(NAME(2) - 5HOUTPU)2140, 6140,2140                              * 369
```

HEART

```
 6140 DIF(MACHINE)9710,9710,979                                          * 370
979   CONTINUE                                                          * 371
      WRITE (63,5905) NAME(3)                                           * 372
      READ  (63,5920) M                                                 * 373
9710  CONTINUE                                                          * 374
      DIF(MACHINE)9711,9711,9712                                        * 375
9711  CONTINUE                                                          * 376
  920 FORMAT(1X,I3)                                                     * 377
      DECODE 5905,(NAME(3)),920,(M)                                     * 378
9712  CONTINUE                                                          * 379
      GO TO 2125                                                        * 380
2140  IF(NAME(2) - 5HCHECK)2145, 6145,2145                              * 381
 6145 NCHECK = 1                                                        * 382
      GO TO 2125                                                        * 383
2145  IF(NAME(2) - 5HFINIS)2155, 6150,2155                              * 384
 6150 CONTINUE                                                          * 385
CCC   ENDFILE AND UNLOAD TAPE WITH LABEL N2 ASSIGNED TO I/O CHANNEL N1. * 386
C********ASSIGN STATEMENTS BELOW ARE USED TO CONTROL HANDLING OF TAPES  * 387
C********AND DISC FILES, ALL OPTIONAL PROCEDURES.  THESE MAY BE        * 388
C********ELIMINATED, OR SUBSTITUTES FOUND.                              * 389
      DIF(MACHINE)982,982,981                                           * 390
981   CONTINUE                                                          * 391
      WRITE (63,5905) NAME(3), NAME(4)                                  * 392
      READ  (63,5910) N1, N2                                            * 393
      CALL WRTECF (N1)                                                  * 394
      CALL ASSIGN(N1,1,N2,-2)                                           * 395
982   CONTINUE                                                          * 396
      DIF(MACHINE)983,983,984                                           * 397
983   CONTINUE                                                          * 398
      DECODE 5905,(NAME(3),NAME(4)),910,(N1,N2)                         * 399
      CALL WRTECF (N1)                                                  * 400
      CALL UNLOAD (N1)                                                  * 401
      WRITE (    MW,2150 )N1,N2                                         * 402
      PRINT 2150 ,N1,N2                                                 * 403
2150  FORMAT(//,11HUNLOAD UNIT,I3,12H, LABEL TAPE,1X,R5,1CH AND SAVE.)  * 404
984   CONTINUE                                                          * 405
      GO TO 2125                                                        * 406
2155  IF(NAME(2) - 5HMOUNT)2165, 6155,2165                              * 407
 6155 CONTINUE                                                          * 408
C MOUNT NEW TAPE.                                                       * 409
CCC   ASSIGN LOGICAL I/O CHANNEL N1 TO TAPE LABELLED N2.                * 410
      DIF(MACHINE)986,986,985                                           * 411
985   CONTINUE                                                          * 412
      WRITE (63,5905) NAME(3), NAME(4)                                  * 413
      READ  (63,5910) N1, N2                                            * 414
      CALL ASSIGN(N1,1,N2)                                              * 415
986   CONTINUE                                                          * 416
      DIF(MACHINE)987,987,988                                           * 417
987   CONTINUE                                                          * 418
      DECODE 5905,(NAME(3),NAME(4)),910,(N1,N2)                         * 419
      WRITE (    MW,2160 )N1,N2                                         * 420
      PRINT 2160 ,N1,N2                                                 * 421
      PAUSE                                                             * 422
2160  FORMAT(//,9HLOAD UNIT,I3,10H WITH TAPE,1X,R5,15H AND HIT START.)  * 423
988   CONTINUE                                                          * 424
      GO TO 2125                                                        * 425
2165  IF(NAME(2) - 5HASSIG)2195, 6160,2195                              * 426
 6160 DIF(MACHINE)9810,9810,989                                         * 427
989   CONTINUE                                                          * 428
      WRITE (63,5875) (NAME(N), N = 1,12)                               * 429
      READ  (63,5915) NARG, (NX1(K), K = 1,8)                           * 430
      GO TO (2125,2170,2175,2180,2125,2125,2125,2185),NARG              * 431
CCC   ASSIGN LOGICAL I/O CHANNEL NX1(1) TO HARDWARE TYPE NX1(2).        * 432
2170  CALL ASSIGN(NX1(1),NX1(2))                                        * 433
      GO TO 2125                                                        * 434
CCC   ASSIGN I/O CHANNEL NX1(1) TO TAPE WITH WIDTH NX1(2), LABEL NX1(3). * 435
2175  CALL ASSIGN(NX1(1),NX1(2),NX1(3))                                 * 436
      GO TO 2125                                                        * 437
CCC   ASSIGN I/O CHANNEL NX1(1) TO DISK FILE WITH NAME NX1(3), LENGTH   * 438
CCC         OR OPERATION NX1(4).   NX1(2) SHOULD BE C.                  * 439
2180  CALL ASSIGN(NX1(1),NX1(2),NX1(3),NX1(4))                          * 440
      GO TO 2125                                                        * 441
```

HEART

```
CCC  GIVE THE DD80 OUTPUT FILES TO THE USER.
CCC  FILE NAMES WILL HAVE 8 CHARACTERS, BEGINNING TRUMPD.
2185 CALL KEEP80 (8RTRUMPDDA)
     IF (NX1(1) GT. 0) CALL KEEP80(0)
9810 CONTINUE                                                          * 445
     DIF(MACHINE)9811,9811,9812                                        * 446
9811 CONTINUE                                                          * 447
     WRITE (    MW,2190 )    FIN                                       * 448
     PRINT 2190                                                        * 449
2190 FORMAT(/,5X,5H*****,38HREPLACE *ASSIGN WITH *MOUNT OR *FINIS.)    * 450
9812 CONTINUE                                                          * 451
     GO TO 2125                                                        * 452
C--------CONTINUE HERE WHEN A PROBLEM NAME CARD HAS BEEN READ IN.      * 453
2195 WRITE (    MW,5925 )NR,M                                          * 454
     PRINT 5925 ,NR,M                                                  * 455
     WRITE (     M,5830 )HOUR,DATE,NPROB,NAME,CLOCKB,CLOCKA            * 456
     WRITE (     M,5885 )    FIN                                       * 457
     IF(M - MW) 6165,2200, 6165                                        * 458
 6165 WRITE (    MW,5830 )HOUR,DATE,NPROB,NAME,CLOCKB,CLOCKA           * 459
     WRITE (    MW,5845 )CLOCKA,KSECS,NPER                             * 460
     WRITE (    MW,5885 )    FIN                                       * 461
CCC  SET TOTAL ELAPSED PROBLEM TIME TO ZERO FOR THIS PROBLEM.          * 462
CCC  CONTROL CARRYOVER OF DATA FROM PRECEDING PROBLEM.                 * 463
2200 KSECS = 0                                                         * 464
     IF(NAME(16) - 1H2)2205, 6170,2205                                 * 465
 6170 NOW = 2                                                          * 466
     WRITE (     M,5895 )    FIN                                       * 467
     PRINT 5895                                                        * 468
     GO TO 2230                                                        * 469
2205 IF(NAME(16) - 1H3)2210, 6175,2210                                 * 470
 6175 NOW = 3                                                          * 471
     GO TO 2215                                                        * 472
2210 IF(NAME(16) - 1H4)2220, 6180,2220                                 * 473
 6180 NOW = 4                                                          * 474
     TAU = SUMTIM                                                      * 475
2215 WRITE (     M,5900 ) TAU                                          * 476
     PRINT 5900 , TAU                                                  * 477
     GO TO 2230                                                        * 478
2220 WRITE (     M,5840 )    FIN                                       * 479
CCC  ZERO ALL COUNTERS WHEN NO DATA IS CARRIED OVER FROM LAST PROBLEM. * 480
     DO 2225  N = 1,25                                                 * 481
     NOSPEC(N) = 0                                                     * 482
2225 CONTINUE                                                          * 483
     NUP = 0                                                           * 484
CCC  SAVE INITIAL NUMBER OF ITEMS IN EACH BLOCK.                       * 485
2230 DO 2235, N = 1,12                                                 * 486
     ITEMS(N) = NOSPEC(N)                                              * 487
2235 CONTINUE                                                          * 488
     IF(NOW)2245,2245, 6185                                            * 489
 6185 IF(NAMES(1) - 1H*) 6190,2240, 6190                               * 490
 6190 WRITE (     M,5960 )    FIN                                      * 491
     PRINT 5960                                                        * 492
     KWIT = 5
     GO TO 2220                                                        * 493
2240 WRITE (     M,5935 ) NAMES, CB, CA                                * 494
     PRINT 5935 , NAMES, CB, CA                                        * 495
     CALL TALLY                                                        * 496
     NOW = 0                                                           * 497
C--------READ IN BLOCK NUMBER CARDS.                                   * 498
2245 WRITE (     M,5885 )    FIN                                       * 499
2250 READ  (    NR,5860 )ADATA,IBLOCK,MOD,ABLOCK                       * 500
     IF(IBLOCK)2255, 6195,2255                                         * 501
 6195 WRITE (     M,5870 )ABLOCK                                       * 502
     GO TO 2250                                                        * 503
2255 WRITE (     M,5865 )ADATA,IBLOCK,MOD,ABLOCK                       * 504
     IF(IBLOCK)2320,2250, 6200                                         * 505
 6200 MODS = 0                                                         * 506
     IF(MOD - 1HA)2260, 6205,2260                                      * 507
 6205 MODS = 1                                                         * 508
     WRITE (     M,5850 )    FIN                                       * 509
     GO TO 2265                                                        * 510
2260 IF(MOD - 1HB)2265, 6210,2265                                      * 511
 6210 MODS = -1                                                        * 512
     WRITE (     M,5855 )    FIN                                       * 513
```

HEART

```
2265  MOD = MODS                                                        * 514
      NB = NR                                                           * 515
      NEWBL(IBLOCK) = NEWBL(IBLOCK) + 1                                 * 516
      IF(MOD)2270, 6215,2270                                            * 517
 6215 ITEMS(IBLOCK) = 0                                                 * 518
CCC   FIND OUT IF INPUT TAPE FOR BLOCK ITEMS HAS BEEN SPECIFIED.        * 519
2270  IF(ABLOCK(1) - 6H*INPUT)2275, 6220,2275                          * 520

 6220 DIF(MACHINE)9714,9714,9713                                        * 521
9713  CONTINUE                                                          * 522
      WRITE (63,5905) ABLOCK(2)                                         * 523
      READ  (63,5910) NB                                                * 524
9714  CONTINUE                                                          * 525
      DIF(MACHINE)9715,9715,9716                                        * 526
9715  CONTINUE                                                          * 527
      DECODE 5905,(ABLOCK(2)),910,(NB)                                  * 528
9716  CONTINUE                                                          * 529
      WRITE (    MW,5950 )IBLOCK,NB                                     * 530
      PRINT 5950 ,IBLOCK,NB                                             * 531
C--------GO TO SUBROUTINES TO READ IN BLOCK ITEMS.                      * 532
2275  GO TO(2310,2280,2290,2280,2295,2305,2305,2285,2310,2300,2315,2280) * 533
     X,IBLOCK                                                           * 534
2280  CALL THERM                                                        * 535
      GO TO 2245                                                        * 536
2285  CALL GEN                                                          * 537
      GO TO 2245                                                        * 538
2290  CALL CHEM                                                         * 539
      GO TO 2245                                                        * 540
2295  CALL FINK                                                         * 541
      GO TO 2245                                                        * 542
2300  CALL FLOW                                                         * 543
      GO TO 2245                                                        * 544
2305  CALL SURE                                                         * 545
      GO TO 2245                                                        * 546
2310  CALL TALLY                                                        * 547
      GO TO 2245                                                        * 548
2315  CALL PLOT                                                         * 549
      GO TO 2245                                                        * 550
C--------COME HERE AFTER ALL DATA AND DATA END CARD HAVE BEEN READ IN.   * 551
C--------INITIALIZE BEFORE FIRST TIME STEP.                             * 552
2320  IBLOCL = IBLOCK                                                   * 553
      WRITE (     M,5885 )     FIN                                      * 554
      WRITE (     M,5885 )     FIN                                      * 555
CCC   SAVE NAME FOR IDENTIFICATION OF DATA FOR CONTINUATION PROBLEMS.   * 556
      DO 2325, N = 1,20                                                 * 557
      NAMES(N) = NAME(N)                                                * 558
2325  CONTINUE                                                          * 559
CCC   FIND TIME CHARGES FOR INPUT PHASE OF PROBLEM.                     * 560
      CALL TIMER(NPER,MICS)                                             * 561
      KSECS = MICS                                                      * 562
      WRITE (     M,5845 )CLOCKA,KSECS,NPER                             * 563
      PRINT        5845 ,CLOCKA,KSECS,NPER                              * 564
      IF(M - MW) 6225,2330, 6225                                        * 565
 6225 WRITE (    MW,5845 )CLOCKA,KSECS,NPER                             * 566
2330  CALL TALLY                                                        * 567
      IF(KWIT - 9) 6230,4500, 6230                                      * 568
 6230 CALL SPECK                                                        * 569
C INITIALIZE.                                                           * 570
C--------INCREMENT CYCLE COUNTER, DO NEXT TIME STEP.                    * 571
4200  KCYC = KCYC + 1                                                   * 572
CCC   GO TO SUBROUTINES TO DO TIME STEP CALCULATIONS AND WRITE OUT DATA. * 573
C--------RETURN HERE AFTER A TIME STEP IS REJECTED.                     * 574
4205  CALL THERM                                                        * 575
      IF(NCGEN)4210,4210, 6235                                          * 576
 6235 CALL GEN                                                          * 577
4210  IF(NKEM)4215,4215, 6240                                           * 578
 6240 CALL CHEM                                                         * 579
4215  IF(NCCCN)4220,4220, 6245                                          * 580
 6245 CALL FINK                                                         * 581
4220  IF(NCFLOW)4225,4225, 6250                                         * 582
 6250 CALL FLOW                                                         * 583
4225  IF(NCSCON)4230,4230, 6255                                         * 584
```

HEART

```
 6255 CALL SURE                                                          * 585
4230  IF(NMELT)4235,4235, 6260                                            * 586
 6260 CALL THERM1                                                        * 587
4235  IF(NCSPEC)4240,4240, 6265                                           * 588
 6265 CALL SPECK                                                         * 589
4240  IF(NMELT)4245,4245, 6270                                            * 590
 6270 CALL THERM2                                                        * 591
C--------MAKE CRT PLOTS WHEN REQUIRED.                                   * 592
4245  IF(NUP)4255,4255, 6275                                              * 593
 6275 IF(XWIT)4255, 6280, 6280                                            * 594
 6280 IF(NSTOP)4600, 6285,4250                                            * 595
 6285 IF(NCW - 2) 6290,4250, 6290                                         * 596
 6290 IF((KCYC - 1)*(KCYC - 2)) 6295,4250, 6295                           * 597
 6295 IF(JPIC)4255,4255, 6300                                             * 598
 6300 IF(XMODF(KCYC - 1,JPIC)) 6305, 6305,4255                            * 599
 6305  IF(KCYC)4255,4255, 6310                                            * 600
 6310 CONTINUE                                                            * 601
4250  CALL PLOT                                                           * 602
4255  IF(NSTOP)4600, 6315,4500                                            * 603
 6315 IF(NDISC) 6320,4260, 6320                                           * 604
 6320 KWIT = 6                                                            * 605
4260  CALL TALLY                                                          * 606
       IF(KWIT)4205,4300, 6325                                            * 607
 6325 IF((KWIT - 5)*(KWIT - 9)) 6330,4500, 6330                           * 608
 6330 NSTOP = 1                                                           * 609
      GO TO 4200                                                          * 610
C--------TEST FOR PROBLEM PREPROCESSING ONLY.                             * 611
4300  IF(NCHECK)4305,4305, 6335                                           * 612
 6335 KWIT = 6                                                            * 613
      NSTOP = 1                                                           * 614
      GO TO 4200                                                          * 615
C                                                                         * 616
C********INTERRUPT-RESTART CONTROL SECTION.                               * 617
C********REMOVE IF NO SUBSTITUTE FOR OFFMON-ONMON AVAILABLE.              * 618
C                                                                         * 619
C--------TEST FOR PROBLEM INTERRUPTION BY SENSE SWITCH 1.                 * 620
4305  IF(SENSE SWITCH 1)4310,4200                                         * 621
 6340 CONTINUE                                                            * 622
CCC   COME HERE WHEN OFFMON IS TO BE CALLED.                              * 623
4310  IF(IBLOCL + 1) 6345,4315,4315                                       * 624
 6345 KWIT = 6                                                            * 625
      NDUMP = 1                                                           * 626
      GO TO 4260                                                          * 627
C********TALLY1 IS SPECIAL ENTRY POINT IN TALLY.                          * 628
4315  CALL TALLY1                                                         * 629
C PUNCH BLOCK 9.                                                          * 630
      GO TO 4325                                                          * 631
4320  WRITE (    MW,5835 )     FIN                                        * 632
      PRINT 5835                                                          * 633
      NOFF = 1                                                            * 634
      SUMTIM = 0.0                                                        * 635
      NSAVE = NDATA = 0                                                   * 636
4325  WRITE (    MW,5810 )KCYC,SUMTIM,KWIT                                * 637
      PRINT                 5810 ,KCYC,SUMTIM,KWIT                      * 638
CCC   SAVE UNREAD DATA DECKS ON TAPE 6A FOR NEXT JOB.                     * 639
      IF(NSAVE)4330,4330, 6350                                            * 640
 6350 N1 = NR                                                             * 641
      N2 = NDATA                                                          * 642
      WRITE (    MW,5945 ) NDATA                                          * 643
      PRINT 5945 , NDATA                                                  * 644
      CALL PRIME (N1, N2, 1, 0, 0.0, 0.0, 0.0, 0.0, 0.0)                  * 645
4330  WRITE (    MW,5800 )     FIN                                        * 646
CCC   FIND TOTAL TIME CHARGES UP TO TIME OFFMON CALLED.                   * 647
      CALL TIMER(NPER,MICS)                                               * 648
      KSECS = KSECS + MICS                                                * 649
      WRITE (    MW,5845 )CLOCKA,KSECS,NPER                               * 650
      PRINT         5845 ,CLOCKA,KSECS,NPER                               * 651
      IF(N - 100)4335, 6355,4335                                          * 652
 6355 CALL FRAME                                                          * 653
      WRITE (     M,5810 )KCYC,SUMTIM,KWIT                                * 654
      WRITE (     M,5800 )     FIN                                        * 655
      WRITE (     M,5845 )CLOCKA,KSECS,NPER                               * 656
```

1TEART

```
CCC   EMPTY CC80 BUFFER, GIVE CC80 FILE TO SYSTEM.                      * 657
4335  CALL PLCTE                                                        * 658
      WRITE (    MW,5805 )     FIN                                      * 659
      CALL CLCCK(HCURX,CATEX)                                           * 660
CCC   DUMP CCRE MEMCRY CN LCGICAL I/O CHANNEL 5.                        * 661
      CALL CFFMCN                                                       * 662
C PRCBLEM CAN BE RESTARTED FROM 5A WITH ONMON.                          * 663
C*******************THIS IS CNMON RESTART ENTRY POINT. ******************  * 664
      REWIND 5                                                          * 665
      WRITE (    MW,5820 )HOUR,DATE                                     * 666
      PRINT 5820 ,HOUR,CATE                                             * 667
      WRITE (    MW,5815 )HCURX,DATEX                                   * 668
      PRINT 5815 ,HCURX,CATEX                                           * 669
      GO TC 2105                                                        * 670
C--------TEST FCR CCNTINUATION OF INTERRUPTED PROBLEM.                  * 671
CCC   TEST FCR A /*DISCCNTINUE/ RESTART OPTION CARD.                    * 672
4400  READ  (   NRS,5875 )(NX1(N), N = 1,20)                            * 673
      WRITE (    MW,5875 )(NX1(N), N = 1,20)                            * 674
      PRINT 5875 ,(NX1(N), N = 1,20)                                    * 675
      IF(NX1(1) - 1H*)4400, 6360,4400                                   * 676
 6360 NDISC = 0                                                         * 677
      IF(NX1(2) - 5HDISCC)4405, 6365,4405                               * 678
 6365 NDISC = 1                                                         * 679
C********START A NEW SERIES CF PROBLEMS, BEGINNING PROBLEM COUNTER AT 1.  * 680
      WRITE (    MW,5930 )     FIN                                      * 681
      PRINT 5930                                                        * 682
      GO TC 4410                                                        * 683
4405  CALL BSPACE(NRS)                                                  * 684
CCC   INITIALIZE PROBLEM TIMING WHEN RELOADING WITH CNMON OR ONMONS.    * 685
4410  NPERS = NPER                                                      * 686
      CALL TIMER(NPER,MICS)                                             * 687
      WRITE (     M,5845 )CLOCKA,KSECS,NPERS                            * 688
      PRINT         5845 ,CLCCKA,KSECS,NPERS                            * 689
      PRINT 5830 ,HOUR,CATE,NPROB,NAME,CLOCKB,CLOCKA                    * 690
      WRITE (     M,5805 )     FIN                                      * 691
      IF(M - MW)4200, 6370,4200                                         * 692
 6370 WRITE (    MW,5845 )CLCCKA,KSECS,NPERS                            * 693
      WRITE (    MW,5805 )     FIN                                      * 694
C GO CN WITH PREVIOUSLY INTERRUPTED PROBLEM.                            * 695
      GO TC 4200                                                        * 696
C********END CF INTERRUPT-RESTART SECTION.                              * 697
C                                                                       * 698
C--------CCME HERE AT END OF PROBLEM.  (OTHERS MAY FOLLOW.)             * 699
4500  KCYC = KCYC - 1                                                   * 700
      WRITE (     M,5825 )NPRCB,KCYC,KWIT,SUMTIM,CLOCKB,CLCCKA,FIN      * 701
      PRINT 5825, NPROB,KCYC,KWIT,SUMTIM,CLOCKB,CLOCKA,FIN              * 702
      IF(M - MW) 6375,4505, 6375                                        * 703
 6375 WRITE (    MW,5825 )NPROB,KCYC,KWIT,SUMTIM,CLOCKB,CLCCKA,FIN      * 704
4505  IF(NDISC) 6380, 6380,2110                                         * 705
 6380 IF(NDUMP)2115,2115,4320                                           * 706
C--------CCME HERE WHEN A /*SPLIT/ CARD HAS BEEN READ IN FROM DATA DECK.  * 707
4600  WRITE (     M,5955 )     FIN                                      * 708
      IF(M - MW) 6385,4605, 6385                                        * 709
 6385 WRITE (    MW,5955 )     FIN                                      * 710
4605  PRINT 5955                                                        * 711
5800  FORMAT(//10X,40HCALLED OFFMON.  SAVE 5A AND LABEL TRUMP.//////,A1)  * 712
5805  FORMAT(1H1,A1)                                                    * 713
5810  FORMAT( /,10X,6HKCYC =,I6,2X,8HSUMTIM =,1PE12.5,2X,6HKWIT =,I6)   * 714
5815  FORMAT(10X,19HTAPE 5A WAS MADE AT,2(2X,A6))                      * 715
5820  FORMAT(//,10X,37HCNMON RESTART ON 5A.  PROGRAM DATE IS,2(2X,A6))  * 716
5825  FORMAT(10X,10HENDEC PROB,I5,5X,7HKCYC = ,I5,5X,7HKWIT = ,I2,5X,9HS  * 717
     1UMTIM = ,1PE12.5,12X,2(2X,A6),/,10X,92HKWIT...1=TIMAX, 2=TMAX, 3=T  * 718
     2MIN, 4=S.S., 5=DATA, 6=CARD OR KEY, 7=MCYC, 8=MSEC, 9=BLKS 2,4,7,,  * 719
     3  /,15X,72H10=CONVERGENCE FAILURE, 11=PROBLEM SIZE LIMIT, 12=TABLE  * 720
     4 LENGTH CR CRCER.,A1)                                             * 721
5830  FORMAT(1H1,9X,17HTRUMP OUTPUT DATA,13X,12HPROGRAM DATE,2(2X,A6),  * 722
     1   28X,9HCATA DECK,I5,//,10X,A1,14A5,A1,4A2,4X,2(2X,A6))          * 723
5835  FORMAT(//10X,46HNC MCRE DATA DECKS, RESTART FROM 5A WITH CNMON,A1)  * 724
5840  FORMAT(//,10X,41HNC DATA CARRYOVER FROM PRECEDING PROBLEM.,A1)    * 725
5845  FORMAT(10X,14HMACHINE TIME (,A6,4H) = ,I5,2X,                     * 726
     1   33HSECCNDS.  CPU,I/C,SYS PERCENTS = ,A10)                      * 727
5850  FORMAT(38X,34HCATA ACCS TC DATA ALREACY READ IN.,A1)              * 728
```

HEART THERM

```
5855  FORMAT(38X,46HDATA MODIFIES OR ADDS TO DATA ALREADY READ IN,A1)
5860  FORMAT(A5,I2,A1,12A6)
5865  FORMAT(10X,4HDATA,1X,A5,I3,A1,14X,12A6)
5870  FORMAT(38X,12A6)
5875  FORMAT(A1,14A5,A1,4A2)
5880  FORMAT(10X,12A6)
5885  FORMAT(10X,100(1H=),A1)
5890  FORMAT(//////////,A1)
5895  FORMAT(/,10X,36HWILL USE DATA AND INITIAL CONDITIONS,
     1   43H FROM PRECEDING PROBLEM WITH NAME, DATE ---,A1)
5900  FORMAT(/,10X,46HWILL USE DATA AND FINAL CONDITIONS, WITH TAU =,
     1   E13.5,43H FROM PRECEDING PROBLEM WITH NAME, DATE ---)
5905  FORMAT(2A5)
5910  FORMAT(I2,1X,A5)
5915  FORMAT(7X,I1,2(1X,I2,1X,I2,1X,A10,1X,I6))
5920  FORMAT(1X,I3)
5925  FORMAT(10X,12HINPUT UNIT =,I4,16H.  OUTPUT UNIT =,I5,1H.)
5930  FORMAT(10X,79HPREVIOUSLY INTERRUPTED PROBLEM WILL BE ENDED.  DATA
     1FOR LAST TIME STEP FOLLOWS.,A1)
5935  FORMAT(10X,A1,14A5,A1,4A2,4X,2(2X,A6))
5940  FORMAT(10X,39HUNUSED DATA DECKS WILL BE SAVED ON UNIT,I4,1H.)
5945  FORMAT(/,10X,29HFOLLOWING DATA DECKS SAVED ON,I4,1H.,//)
5950  FORMAT(38X,20HINPUT DATA FOR BLOCK,I4,23H WILL BE READ FROM UNIT,
     1   I4,1H.)
5955  FORMAT(10X,17HEND OF TRUMP JOB.,A1)
5960  FORMAT(10(1H*),21HNO DATA IS IN MEMORY.,A1)
      RETURN
CCC   END OF HEART.  RETURN TO MAIN PROGRAM TO CALL EXIT.
CCC   (RETURN TO KITTY IF THIS IS A *CONTROLLEE JOB.)
      END
```

THERM

```
*     CARDS COLUMN       THERM
*     LIST 8
*     FORTRAN            THERM   TRUMP MATERIAL AND NODE SUBROUTINE.
      SUBROUTINE THERM
C********VERSION 5/20/69.
C********THERM IS A REQUIRED SUBROUTINE.
      USE STORK
      USE STORE
      USE STORM
C********REQUIRED WHEN CHEM IS USED, USED IN CHEM, THERM, FLOW, TALLY.
      DIF(M3)92,92,91
91    CONTINUE
      COMMON /ACHEMS/ KA(M2), KAX(M2), KB(M2), KBX(M2),
     1         B(M4), BB(M4), DB(M4), DDB(M4)
92    CONTINUE
      DIF(M8)982,982,981
981   CONTINUE
C********ARRAYS USED ONLY IF SUBROUTINE GEN IS USED, G IN THERM,
C********     GEN, AND TALLY, GG IN TALLY ONLY.
      COMMON /AGENS/  G(M4), GG(M4)
982   CONTINUE
      DIF(M12)972,972,971
971   CONTINUE
C********ARRAYS USED ONLY IF THE REMOTE TEMPERATURE DEPENDENCE OPTION IS
C********     USED (BLOCK 12).  USED IN THERM, CHEM, GEN, FLOW, AND SURE
      COMMON  /AREMS/ NLCOK(M4,8)
C********USED ONLY IF REMOTE TEMPERATURE DEPENDENCE OPTION IS USED
C********     (BLOCK 12).  USED ONLY IN THERM.
      COMMON/AREM/NODP1(M12),NODP2(M12),NOXP1(M12),NOXP2(M12),NPROP(M12)
972   CONTINUE
      INTEGER      A1, AMAT
      DIMENSION APROP(8)
      DATA (APROP(1) = 4HCAPT), (APROP(2) = 4HCONT), (APROP(3) = 2HGT),
     1   (APROP(4) = 2HZT), (APROP(5) = 2HET), (APROP(6) = 2HGT),
     2   (APROP(7) = 5HFLOWT), (APROP(8) = 5HHSURT)
      IF(KCYC) 6100,3100,4200
 6100 IF(IBLOCK - 2) 6105,2135, 6105
 6105 IF(IBLOCK - 4) 6110,2275, 6110
 6110 IF(IBLOCK - 12)2130, 6115,2130
 6115 DIF(M12)914,914,913
 913  CONTINUE
```

TAELM

```
CCC INPUT DATA BLCCK 12.  REMOTE TEMPERATURE DEPENDENCES.
      N = NTABS
      L = 0
      IF(MCD)2100, 6120,2100
 6120 N = NTABS = 0
2100  REAC  (    NB,5860 )N1,N2,NP,NSEQ,NADP1,NADP2,NADPR
      IF(N1) 6125,2130, 6125
 6125 CCNTINUE
2105  IF(MCD) 6130,2110,2110
 6130 CALL SEEK2(N,N1,N2,NCXP1,NOXP2,NTABS,K)
      ITEMS(12) = XMINOF(N - 1,ITEMS(12))
      GC TC 2115
2110  N = NTABS = N + 1
2115  IF(N - M12)2120,2120, 6135
 6135 N = NTABS = M12
      KWIT = 11
      WRITE (    MW,5900 )N,IBLOCK
      PRINT 5900 ,N,IBLCCK
2120  NODP1(N) = NGXP1(N) = N1
      NODP2(N) = NGXP2(N) = N2
      NPRCP(N) =NP
      L = L + 1
      IF(XMCDF(L-1,57))2125, 6140,2125
 6140 WRITE (     M,5855 )     FIN
2125  WRITE (     M,5865 )NCDP1(N),NODP2(N),N,NPROP(N),APRCP(NP)
      IF(NSEC)2100,2100, 6145
 6145 NSEC = NSEQ - 1
      N1 = N1 + NACP1
      N2 = N2 + NACP2
      NP = NP + NACPR
      GO TC 2105
2130  CONTINUE
914   CONTINUE
      RETURN
C COMPLETED BLCCK 12.
CCC INPUT CATA BLOCK  2.  MATERIAL PROPERTY LIST.
2135  N = NMAT
      LABEL = 1
      IF(MCD)2140, 6150,2140
 6150 N = NMELT = NVARC = NVARK = NMAT = 0
2140  REAC  (    NB,5800 )A1,N1,K1,K2,L1,L2,P1,P2,P3,P4,NX1
      IF(N1) 6155,2270, 6155
 6155 CALL PATCH(NX1,0.0,P5,LBX)
CCC   SEE IF *MAST CCNTRCL CARD HAS BEEN REAC IN.
      IF(A1 - 5H*MAST)2150, 6160,2150
 6160 CALL PRIME(N1,K1,L1,L2,P1,P2,P3,P4,P5)
CCC   FINC CUT WHAT NEXT INPUT UNIT FOR BLOCK ITEMS IS.
      IF((K1 - MW)*(K1 - 1001*(K1 - M))2145, 6165,2145
 6165 K1 = NB
2145  NB = K1
      IF(K2 - 2)2140, 6170, 6170
 6170 NB = K2
      GO TC 2140
2150  IF(MCD) 6175,2170,2170
 6175 CALL SEEK1(N,N1,MAT,NMAT,K)
      IF(K)2165,2165, 6180
 6180 IF(NMELT(N)) 6185,2155, 6185
 6185 NMELT = NMELT - 1
2155  IF(LTABC(N)) 6190,2160, 6190
 6190 NVARC = NVARC - 1
2160  IF(LTABK(N)) 6195,2165, 6195
 6195 NVARK = NVARK - 1
2165  ITEMS( 2) = XMINOF(N - 1,ITEMS( 2))
      GO TC 2175
2170  N = NMAT = N + 1
2175  IF(N - M2)2180,2180, 6200
 6200 N = NMAT = M2
      KWIT = 11
      WRITE (    MW,5900 )N,IBLOCK
      PRINT 5900 ,N,IBLCCK
2180  AMAT(N) = A1
      MAT(N) = N1
```

THERM

```
      DIF(M3)94,94,93                                                   * 873
93    CONTINUE                                                          * 874
      KA(N) = K1                                                        * 875
      KAX(N) = K1                                                       * 876
      KB(N) = K2                                                        * 877
      KBX(N) = K2                                                       * 878
94    CONTINUE                                                          * 879
      LTABC(N) = L1                                                     * 880
      LTABK(N) = L2                                                     * 881
      DENS(N) = P1                                                      * 882
      CAPT(1,N) = P2                                                    * 883
      CONT(1,N) = P3                                                    * 884
      THELT(N) = P4                                                     * 885
C********FOLLOWING STATEMENT ADDED AFTER UCRL-14754 REV 1.             * 886
      HMELTX(N) = HMELT(N) = ABSF(P5)                                   * 887
      IF(LABEL)2185,2185, 6205                                          * 888
 6205 LABEL = 0                                                         * 889
      WRITE (     M,5805 )     FIN                                      * 890
2185  WRITE (     M,5810 )AMAT(N),MAT(N),N,K1,K2,LTABC(N),              * 891
     $ LTABK(N),DENS(N),CAPT(1,N),CONT(1,N),THELT(N),HMELT(N)           * 892
      IF(HMELT(N)) 6210,2190, 6210                                      * 893
 6210 NMELT = NMELT + 1                                                 * 894
      DIF(M3)96,96,95                                                   * 895
95    CONTINUE                                                          * 896
      KB(N) = 0                                                         * 897
96    CONTINUE                                                          * 898
2190  LTAB = XABSF(LTABC(N))                                            * 899
      IF(LTAB - 2) 6215,2195,2195                                       * 900
 6215 LTABC(N) = 0                                                      * 901
      GO TO 2250                                                        * 902
2195  NVARC = NVARC + 1                                                 * 903
      LABEL = 1                                                         * 904
      READ  (    NB,5815 )(CAPT(J,N),TVARC(J,N), J=1,LTAB)              * 905
C********NEXT 5 STATEMENTS REVISED SINCE UCRL-14754 REV 1.             * 906
      DO 2205  J = 2,LTAB                                               * 907
      DTX = TVARC(J,N) - TVARC(J-1,N) + 1.0E-12                         * 908
      SLOC(J,N) = (CAPT(J,N) - CAPT(J-1,N))/DTX                         * 909
      IF(DTX) 6220, 6220,2205                                           * 910
 6220 KWIT = 12                                                         * 911
2205  CONTINUE                                                          * 912
      SLOC(1,N) = SLOC(2,N)                                             * 913
CCC   FIND ENERGY REFERRED TO TVARC(1,N)                                * 914
      IF(LTABC(N))2245,2245, 6225                                       * 915
 6225 WT(1,N) = 0.0                                                     * 916
      DO 2215  J = 2,LTAB                                               * 917
      F3(J) = 0.5*(CAPT(J,N) + CAPT(J-1,N))                             * 918
      WT(J,N) = WT(J-1,N) + F3(J)*(TVARC(J,N) - TVARC(J-1,N))           * 919
2215  CONTINUE                                                          * 920
      F3(1) = F3(2)                                                     * 921
CCC   FIND CALCULATED ENERGY AT ZERO DEGREES                            * 922
      DO 2220  J = 1,LTAB                                               * 923
      J = J                                                             * 924
      IF(TVARC(J,N)) 6230,2230,2225                                     * 925
 6230 CONTINUE                                                          * 926
2220  CONTINUE                                                          * 927
2225  CAPX = CAPT(J,N) - 0.5*SLOC(J,N)*TVARC(J,N)                       * 928
      CORR = WT(J,N) - CAPX*TVARC(J,N)                                  * 929
      GO TO 2235                                                        * 930
2230  CORR = WT(J,N)                                                    * 931
CCC   CALCULATE HEAT CONTENT RELATIVE TO ZERO DEGREES.                  * 932
2235  DO 2240  J = 1,LTAB                                               * 933
      WT(J,N) = WT(J,N) - CORR                                          * 934
2240  CONTINUE                                                          * 935
2245  WRITE (     M,5905 )(CAPT(J,N),SLOC(J,N),WT(J,N),F3(J),           * 936
     $   TVARC(J,N), J = 1,LTAB)                                        * 937
      IF(LTAB - M9)2250,2250, 6235                                      * 938
 6235 KWIT = 12                                                         * 939
2250  LTAB = XABSF(LTABK(N))                                            * 940
      IF(LTAB - 2) 6240,2255,2255                                       * 941
 6240 LTABK(N) = 0                                                      * 942
      GO TO 2140                                                        * 943
```

THEM

```
2255  NVARK = NVARK + 1                                                  * 944
       LABEL = 1                                                         * 945
      READ (    N8,5815 )(CONT(J,N), TVARK(J,N), J=1,LTAB)               * 946
      DO 2265  J = 2,LTAB                                                * 947
      DTX = (TVARK(J,N) - TVARK(J-1,N)) + 1.0E-12                        * 948
      SLOK(J,N) = (CONT(J,N) - CONT(J-1,N))/DTX                          * 949
      IF(DTX) 6245, 6245,2265                                            * 950
 6245 KWIT = 12                                                          * 951
2265  CONTINUE                                                           * 952
      SLOK(1,N) = SLOK(2,N)                                              * 953
      WRITE (     M,5880 )(CONT(J,N),SLOK(J,N),TVARK(J,N),J=1,LTAB)      * 954
      IF(LTAB - M9)2140,2140, 6250                                       * 955
 6250 KWIT = 12                                                          * 956
      GO TO 2140                                                         * 957
2270  NEWBL(4) = NEWBL(4) + 1000                                         * 958
      RETURN                                                             * 959
C COMPLETED BLOCK 2.                                                     * 960
CCC INPUT DATA BLOCK  4.  NODE LIST.                                     * 961
2275  N = NODES                                                          * 962
      L = 0                                                              * 963
      IF(MCD)2280, 6255,2280                                             * 964
 6255 N = NODES = NOSPEC = 0                                             * 965
2280  READ (    N8,5820 )N1,NSEQ,NADD,N2,KS,DLONG,DWIDE,DRAD             * 966
      IF(N1) 6260,2330, 6260                                             * 967
 6260 DLONG = DLONG*SCALE                                                * 968
      DWIDE = DWIDE*SCALE                                                * 969
      DRAD = DRAD*SCALE                                                  * 970
      ADD = 0.0                                                          * 971
      IF(DRAD) 6265,2285,2285                                            * 972
 6265 ADD = -(DRAD + DRADS)                                              * 973
      DRAD = DRADS                                                       * 974
2285  IF(MCD) 6270,2295,2295                                             * 975
 6270 CALL SEEK1(N,N1,NODE,NODES,K)                                      * 976
      IF(K)2290,2290, 6275                                               * 977
 6275 IF(NTYPE(N)) 6280,2290, 6280                                       * 978
 6280 NOSPEC = NOSPEC - 1                                                * 979
2290  ITEMS( 4) = XMINOF(N - 1,ITEMS( 4))                                * 980
      GO TO 2300                                                         * 981
2295  N = NODES = N + 1                                                  * 982
2300  IF(N - M4)2305,2305, 6285                                          * 983
 6285 N = NODES = M4                                                     * 984
      KWIT = 11                                                          * 985
      WRITE (    MW,5900 )N,IBLOCK                                       * 986
      PRINT 5900 ,N,IBLOCK                                               * 987
2305  NODE(N) = N1                                                       * 988
      NODMAT(N) = NOXMAT(N) = N2                                         * 989
      DIF(M12)916,916,915                                                * 990
915   CONTINUE                                                           * 991
      DO 2310  J = 1,8                                                   * 992
      NLOCK(N,J) = N                                                     * 993
2310  CONTINUE                                                           * 994
916   CONTINUE                                                           * 995
      T(N) = TONE                                                        * 996
       A(N) = ALONE                                                      * 997
      DIF(M3)98,98,97                                                    * 998
97    CONTINUE                                                           * 999
      B(N) = BONE                                                        *1000
98    CONTINUE                                                           *1001
      DIF(M8)910,910,99                                                  *1002
99    CONTINUE                                                           *1003
      G(N) = GONE                                                        *1004
910   CONTINUE                                                           *1005
      NTYPE(N) = 0                                                       *1006
      DRADS = DRAD = DRAD + ADD                                          *1007
      RADIUS(N) = DRAD                                                   *1008
      VOL(N) = GECM*DLONG*DWIDE* SCALE*(DRAD/SCALE)**KSYM                *1009
      IF(VOL(N))2315, 6290,2315                                          *1010
 6290 VOL(N) = 1.0E-24                                                   *1011
       NTYPE(N) = 2                                                      *1012
       NOSPEC = NOSPEC + 1                                               *1013
       GO TO 2320                                                        *1014
2315  IF(KS) 6295,2320, 6295                                             *1015
```

TTERM

```
 6295 NTYPE(N) = 3                                                      *1016
      NOSPEC = NOSPEC + 1                                               *1017
2320  L = L + 1                                                         *1018
      IF(XMODF(L-1,57))2325, 6300,2325                                  *1019
 6300 WRITE (     M,5825 )     FIN                                      *1020
2325  WRITE (     M,5830 ) NODE(N), N, NODMAT(N), NTYPE(N),             *1021
     $ DLONG,DWIDE,DRAD,VOL(N)                                          *1022
      IF(NSEQ)2280,2280, 6305                                           *1023
 6305 NSEQ = NSEQ - 1                                                   *1024
      N1 = N1 + NACC                                                    *1025
      GO TO 2285                                                        *1026
2330  NX = NEWBL(7)                                                     *1027
      DO 2335  N = 5,12                                                 *1028
      NEWBL(N) = NEWBL(N) + 1000                                        *1029
2335  CONTINUE                                                          *1030
      NEWBL(7) = NX                                                     *1031
      NEWBL(1) = NEWBL(1) + 1000                                        *1032
      RETURN                                                            *1033
C COMPLETED BLOCK 4.  RETURN TO HEART.                                  *1034
3100  IBLOCK = 4                                                        *1035
      CALL REFER(NODMAT,NOXMAT,NODES,MAT,NMAT)                          *1036
      DIF(M12)918,918,917                                               *1037
917   CONTINUE                                                          *1038
      IF(NTABS)3110,3110, 6310                                          *1039
 6310 IBLOCK = 12                                                       *1040
      CALL REFER(NODP1,NOXP1,NTABS,NODE,NODES)                          *1041
      CALL REFER(NODP2,NOXP2,NTABS,NODE,NODES)                          *1042
      IF(KWIT)4500, 6315,4500                                           *1043
 6315 DO 3105  N = 1,NTABS                                              *1044
      J = NODP1(N)                                                      *1045
      K = NPROP(N)                                                      *1046
      NLOCK(J,K) = NODP2(N)                                             *1047
3105  CONTINUE                                                          *1048
918   CONTINUE                                                          *1049
CCC   CALC. NODE MASSES, CAPACITIES, CONDUCTIVITIES.                    *1050
3110  IF(KWIT)4500, 6320,4500                                           *1051
 6320 DO 3125  N = 1,NODES                                              *1052
       J = NODMAT(N)                                                    *1053
      ZIP(N) = 0.0                                                      *1054
      HEFT(N) = VOL(N)*DENS(J)                                          *1055
      IF(HEFT(N)) 6325, 6325,3115                                       *1056
 6325 HEFT(N) = 1.0E-36                                                 *1057
3115  CAP(N) = HEFT(N)*CAPT(1,J)                                        *1058
      IF(CAP(N)) 6330, 6330,3120                                        *1059
 6330 CAP(N) = 1.0E-36                                                  *1060
3120  W(N) = CAP(N)*T(N)                                                *1061
      CON(N) = CONT(1,J)                                                *1062
      IF(CON(N)) 6335, 6335,3125                                        *1063
 6335 CON(N) = 1.0E-24                                                  *1064
3125  CONTINUE                                                          *1065
CCC   FIND AND WRITE OUT TOTAL NODES, VOLUME, MASS, CAP OF EACH MATL.   *1066
      WRITE (     M,5870 )     FIN                                      *1067
      CAPS = HEAT = VOLS = HEFTS = 0.0                                  *1068
      DO 3135  K = 1,NMAT                                               *1069
      NODMS = VOLMS = HEFTMS = CAPMS = WMS = 0.0                        *1070
      DO 3130, N = 1,NODES                                              *1071
       IF(NODMAT(N) - K)3130, 6340,3130                                 *1072
 6340 NODMS = NODMS + 1                                                 *1073
      VOLMS = VOLMS + VOL(N)                                            *1074
      CAPMS = CAPMS + CAP(N)                                            *1075
      HEFTMS = HEFTMS + HEFT(N)                                         *1076
      WMS = WMS + W(N)                                                  *1077
3130  CONTINUE                                                          *1078
      CAPS = CAPS + CAPMS                                               *1079
      VOLS = VOLS + VOLMS                                               *1080
      HEFTS = HEFTS + HEFTMS                                            *1081
      HEAT = HEAT + WMS                                                 *1082
      IF(CAPMS)3135,3135, 6345                                          *1083
 6345 WRITE (     M,5875 )AMAT(K),MAT(K),NODMS,DENS(K),                 *1084
     $CAPT(1,K),VOLMS,HEFTMS,CAPMS,WMS                                  *1085
3135  CONTINUE                                                          *1086
      WRITE (     M,5890 )NODES,VOLS,HEFTS,CAPS,HEAT                    *1087
```

THERM

```
CCC   DELETE PHASE CHANGE CALCULATION FOR MATERIALS WHOSE PHASE CHANGE      *1088
CCC      TEMPERATURE IS NOT BETWEEN TMIN AND TMAX.                          *1089
      NREACT = NREACT - NMELT                                               *1090
      NMELT = 0                                                             *1091
      DO 3140  K = 1,NMAT                                                   *1092
      HMELT(K) = 0.0                                                        *1093
      IF((TMELT(K) - TMIN)*(TMAX - TMELT(K)))3140, 6350, 6350               *1094
 6350 HMELT(K) = HMELTX(K)                                                  *1095
      NMELT = NMELT + 1                                                     *1096
3140  CONTINUE                                                              *1097
      NREACT = NREACT + NMELT                                               *1098
CCC  START EACH NEW PROBLEM WITH PHASES IN EQUILIBRIUM.                     *1099
      IF(NMELT) 6355,4200, 6355                                             *1100
 6355 IF(NAME(16) - 1H3) 6360,4200, 6360                                    *1101
 6360 IF(NAME(16) - 1H4) 6365,4200, 6365                                    *1102
 6365 IF(NAME(15) - 5HA=BL9) 6370,4200, 6370                                *1103
 6370 CONTINUE                                                              *1104
C COLUMNS 67-71, RETAIN BLOCK 9 A.                                          *1105
      DO 3145  N = 1,NODES                                                  *1106
       J = NODMAT(N)                                                        *1107
      IF(HMELT(J)) 6375,3145, 6375                                          *1108
 6375 A(N) = 1.0                                                            *1109
      IF(T(N) - TMELT(J))3145,3145, 6380                                    *1110
 6380 A(N) = 0.0                                                            *1111
3145  CONTINUE                                                              *1112
4200  FORC = FCR*DELT                                                       *1113
      IF(KWIT)4300, 6385, 6385                                              *1114
 6385 IF(NCW)4300,4300, 6390                                                *1115
 6390 IF(KCYC - 1)4300,4205, 6395                                           *1116
 6395 IF(KDATA)4300, 6400, 6400                                             *1117
 6400 CONTINUE                                                              *1118
CCC   FIND, WRITE TOTAL CAP, HEAT CONTENT, AVG TEMP OF EACH MATL.           *1119
4205  WRITE (     M,5885 )     FIN                                          *1120
      DO 4215  K = 1,NMAT                                                   *1121
      CAPMS = WMS = TMS = 0.                                                *1122
      DO 4210  N = 1,NODES                                                  *1123
       IF(NODMAT(N) - K)4210, 6405,4210                                     *1124
 6405 CAPMS = CAPMS + CAP(N)                                                *1125
       WMS = WMS + W(N)                                                     *1126
      TMS = TMS + CAP(N)*T(N)                                               *1127
4210  CONTINUE                                                              *1128
      IF(CAPMS)4215,4215, 6410                                              *1129
 6410 TMS = TMS/CAPMS                                                       *1130
      WRITE (     M,5895 )AMAT(K),MAT(K),CAPMS,WMS,TMS,TMELT(K),HMELT(K)    *1131
4215  CONTINUE                                                              *1132
      WRITE (     M,5835 )     FIN                                          *1133
CCC   WRITE OUT DATA FOR NODES ON PRINTOUT CYCLES.                          *1134
      IF(KCYC - 1)4300,4230, 6415                                           *1135
 6415  IF(KDATA)4300,4300, 6420)                                            *1136
 6420 IF(NVARC + NVARK + NVARH + NORAC + NORACS)4300,4300, 6425              *1137
 6425 CONTINUE                                                              *1138
CCC   WRITE OUT PROPERTIES OF EACH NODE.                                    *1139
4230  WRITE (     M,5840 )     FIN                                          *1140
      DO 4240  N = 1,NODES                                                  *1141
      IF(XMODF(N-1,57))4235, 6430,4235                                      *1142
 6430 WRITE (     M,5845 )     FIN                                          *1143
4235  WRITE (     M,5850 )NODE(N), NOXMAT(N), NTYPE(N), RADILS(N),          *1144
     1   VOL(N), HEFT(N), CAP(N), CON(N), ZIP(N), SLIM(N)                   *1145
4240  CONTINUE                                                              *1146
      WRITE (     M,5835 )     FIN                                          *1147
CCC   INITIALIZE HEAT CONTENT, CONDUCTANCE, CHANGES IN T, A, B, AND FLUX    *1148
4300  DO 4302  N = 1,NODES                                                  *1149
      DA(N) = DT(N) = DF(N) = ZIP(N) = 0.0                                  *1150
      DIF(M3)912,912,911                                                    *1151
911   CONTINUE                                                              *1152
      DB(N) = 0.                                                            *1153
912   CONTINUE                                                              *1154
      IF(LTABC(NODMAT(N)))4302, 6435,4302                                   *1155
 6435 W(N) = CAP(N)*T(N) + HMELT(NODMAT(N))*HEFT(N)*(1.0 - A(N))            *1156
4302  CONTINUE                                                              *1157
      IF(NVARC)4400,4400, 6440                                              *1158
 6440 CONTINUE                                                              *1159
```

THERM

```
CCC   FIND NEW NODE HEAT CAPACITIES, HEAT CONTENT                      *1160
      DO 4345  N = 1,NODES                                             *1161
       J = NODMAT(N)                                                   *1162
       K = N                                                           *1163
      DIF(M12)920,920,919                                              *1164
919   CONTINUE                                                         *1165
      K = NLCCK(N,1)                                                   *1166
920   CONTINUE                                                         *1167
      IF(LTABC(J)) 6445,4345,4305                                      *1168
 6445 SET = SUMTIM + FORC                                              *1169
      GO TO 4310                                                       *1170
4305  EX = FORC*DOT(K)                                                 *1171
      SET = T(K) + EX                                                  *1172
4310  CAPS = CAPS - CAP(N)                                             *1173
       BET = CAP(N)                                                    *1174
      BETW = W(N)                                                      *1175
      MIN = 1                                                          *1176
       MAX  = XABSF(LTABC(J))                                          *1177
4315  MID = (MIN + MAX)/2                                              *1178
       IF(SET - TVARC(MID,J))4320,4330,4325                            *1179
4320  MAX = MID                                                        *1180
       IF(MAX - 2)4330,4315,4315                                       *1181
4325  MIN = MID                                                        *1182
       IF(MAX - MIN - 2)4330,4315,4315                                 *1183
4330  CAPX = CAPT(MID,J) + SLOC(MID+1,J)*(SET - TVARC(MID,J))          *1184
      CAP(N) = HEFT(N)*CAPX                                            *1185
      IF(CAP(N)) 6450, 6450,4335                                       *1186
 6450 CAP(N) = 1.0E-36                                                 *1187
4335  IF(LTABC(J))4340,4345, 6455                                      *1188
 6455 CAPX = 0.5*(CAPT(MID,J) + CAPX - SLOC(MID+1,J)*EX)               *1189
      W(N) = HEFT(N)*(WT(MID,J) + CAPX*(SET - EX - TVARC(MID,J)))      *1190
      W(N) = W(N) + HEFT(N)*HMELT(J)*(1.0 - A(N))                      *1191
4340  CAPS = CAPS + CAP(N)                                             *1192
      BETW = ABSF(BETW - W(N))/(BET + 1.0E-24)                         *1193
      BET = 100.0*ABSF( BET - CAP(N))/(BET + 1.0E-24)                  *1194
      DTMAX = MAX1F(DTMAX,BET*TVARY)                                   *1195
      DTMAX = MAX1F(DTMAX,BETW)                                        *1196
4345  CONTINUE                                                         *1197
CCC   FIND NEW NODE THERMAL CONDUCTIVITIES.                            *1198
4400  IF(NCCON*NVARK)4500,4500, 6460                                   *1199
 6460 DO 4440  N = 1,NODES                                             *1200
       J = NODMAT(N)                                                   *1201
       K = N                                                           *1202
      DIF(M12)922,922,921                                              *1203
921   CONTINUE                                                         *1204
      K = NLCCK(N,2)                                                   *1205
922   CONTINUE                                                         *1206
      IF(LTABK(J)) 6465,4440,4405                                      *1207
 6465  SET = SUMTIM + FORC                                             *1208
       GO TO 4410                                                      *1209
4405  SET = T(K) + FORD*DOT(K)                                         *1210
4410  MIN = 1                                                          *1211
      MAX  = XABSF(LTABK(J))                                           *1212
      BET = CON(N)                                                     *1213
4415  MID = (MIN + MAX)/2                                              *1214
       IF(SET - TVARK(MID,J))4420,4430,4425                            *1215
4420  MAX = MID                                                        *1216
       IF(MAX - 2)4430,4415,4415                                       *1217
4425  MIN = MID                                                        *1218
       IF(MAX - MIN - 2)4430,4415,4415                                 *1219
4430  CON(N) = CONT(MID,J) + SLOK(MID+1,J)*(SET - TVARK(MID,J))        *1220
      IF(CON(N)) 6470, 6470,4435                                       *1221
 6470 CON(N) = 1.0E-24                                                 *1222
4435  BET = 100.0*ABSF(BET-CON(N))/(BET + 1.0E-24)                     *1223
      DTMAX = MAX1F(DTMAX,BET*TVARY)                                   *1224
4440  CONTINUE                                                         *1225
4500  RETURN                                                           *1226
C********ENTRY POINT FOR FIRST ESTIMATE OF PHASE CHANGE.  TO DELETE    *1227
C********PHASE CHANGE CALCULATIONS REMOVE REMAINING STATEMENTS IN THERM, *1228
C********REMOVE ARRAYS TMELT, HMELT, HMELTX FROM COMMON AND REMOVE     *1229
C********THEIR SUBSCRIPTS WHEREVER THEY APPEAR IN THERM, FLOW AND TALLY. *1230
      ENTRY THERM1                                                     *1231
```

THERM
CHEM

```
      IF(KWIT) 6475, 6475,4500                                          *1232
 6475 DO 4605  N = 1,NODES                                              *1233
       J = NCOMAT(N)                                                    *1234
       IF(HMELT(J)) 6480,4605, 6480                                     *1235
 6480 IF(NTYPE(N) - 2)4600, 6485,4600                                   *1236
 6485 DDA(N) = 0.0                                                      *1237
      GO TO 4605                                                        *1238
4600  HMX = HEFT(N)*HMELT(J)/CAP(N)                                     *1239
      DTX = T(N) + DELT*(DOT(N) - HMX*DDA(N)) - TMELT(J)                *1240
CCC   DDA IS TEMPORARY VALUE OF ESTIMATED TEMP CHANGE FROM PHASE CHANGE.*1241
      DDA(N) = HMX*MAX1F(-A(N),MIN1F(-DTX/HMX,1.0-A(N)))                *1242
      DT(N) = DT(N) + DDA(N)                                            *1243
4605  CONTINUE                                                          *1244
       RETURN                                                           *1245
C COMPLETED THERM1.  RETURN TO HEART.                                   *1246
C********ENTRY POINT FOR FINAL CALCULATION OF AMOUNT OF PHASE CHANGE.   *1247
      ENTRY THERM2                                                      *1248
      IF(KWIT) 6490, 6490,4500                                          *1249
 6490 DO 4620  N = 1,NODES                                              *1250
       J = NCOMAT(N)                                                    *1251
       IF(HMELT(J)) 6495,4620, 6495                                     *1252
 6495 DT(N) = DT(N) - DDA(N)                                            *1253
      DDA(N) = 0.0                                                      *1254
      DTX = T(N) + DT(N) - TMELT(J)                                     *1255
      HMX = HEFT(N)*HMELT(J)/CAP(N)                                     *1256
      IF(NTYPE(N) - 2)4615, 6500,4615                                   *1257
 6500 HMX = 1.0E-6*ABSF(DTX) + 1.0E-12                                  *1258
4615  DAX = MAX1F(-A(N),MIN1F(-DTX/HMX,1.0-A(N)))                       *1259
      IF(DAX) 6505,4620, 6505                                           *1260
 6505 DTMAX = MAX1F(DTMAX,ABSF(DT(N)))                                  *1261
      DT(N) = DT(N) + DAX*HMX                                           *1262
      DA(N) = DA(N) + DAX                                               *1263
4620  CONTINUE                                                          *1264
       RETURN                                                           *1265
5800  FORMAT(A5,5I5,4E10.3,10A1)                                        *1266
5805  FORMAT(/10X95HNAME MATL INDEX  KA  KB LTABC LTABK   DENSITY      C *1267
     $APACITY   CONDUCTIVITY   TMELT        HMELT,A1)                   *1268
5810  FORMAT(10X,A5,2I5,2I4,2I6,1P5E13.4)                               *1269
5815  FORMAT (8E10.3)                                                   *1270
5820  FORMAT(5I5,5X,5E10.3)                                             *1271
5825  FORMAT(//,12X,22HNODE INDEX  MATL NTYPE,4X,5HDLONG,10X,5HDWIDE,   *1272
     $   10X,4HDRAD,11X,6HVOLUME,A1)                                    *1273
5830  FORMAT(10X,4I6,1P4E15.5)                                          *1274
5835  FORMAT(10X,100(1H=),A1)                                           *1275
5840  FORMAT(10X,69HNODE DATA.  CHECK TOTAL CONDUCTANCES (ZIP) AND TIME *1276
     1CONSTANTS (SLIM).,/,15X,88HLARGE DIFFERENCES BETWEEN NODES MAY BE *1277
     2DUE TO POOR ZONING, AND MAY PRODUCE POOR RESULTS.,A1)             *1278
5845  FORMAT(/,10X, 95HNODE  MATL NTYPE   RADIUS       VOLUME       MASS *1279
     $     CAPACITY  CONDUCTIVITY    ZIP         SLIM,A1)               *1280
5850  FORMAT(10X,3I5,1X,7E12.4)                                         *1281
5855  FORMAT(//,11X,29HNODP1 NODP2 INDEX NPROP  PROP,A1)                *1282
5860  FORMAT(7I5)                                                       *1283
5865  FORMAT(10X,4I6,2X,A6)                                             *1284
5870  FORMAT(//,10X,16HMATERIAL SUMMARY,//,12X,92HNAME  MATL NODES   DEN *1285
     $SITY      CAPACITY     TOT VOL       TOT MASS      TOT CAP      TOT *1286
     $HEAT,A1)                                                          *1287
5875  FORMAT(10X,A6,2I6,1P6E13.5)                                       *1288
5880  FORMAT(/17X12HCONDUCTIVITY5X5HSLOPE9X5HTVARK/,(15X1P3E15.6))     *1289
5885  FORMAT(10X,13HMATERIAL DATA,//,12X,10HNAME  MATL,5X,7HTOT CAP,    *1290
     1   8X,8HTOT HEAT,7X,8HAVG TEMP,7X,5HTMELT,10X,5HHMELT,A1)         *1291
5890  FORMAT(/,10X,12HSYSTEM TOTAL,I6,26X,1P4E13.5)                     *1292
5895  FORMAT(10X,A6,I6,1P5E15.5)                                        *1293
5900  FORMAT(15(1H*),17HMORE THAN ALLOWED,I5,15H ITEMS IN BLOCK,I3)     *1294
5905  FORMAT(/19X65HCAPACITY       SLOPE          ENERGY          SLOPE *1295
     $        TVARC/,(15X1P5E15.6))                                     *1296
C COMPLETED THERM2.  RETURN TO HEART.                                   *1297
      END                                                               *1298
                                                                        *1299
*     CARDS COLUMN       CHEM                                           *1300
*     LIST 8                                                            *1301
*     FORTRAN            CHEM    TRUMP CHEMICAL REACTION SUBROUTINE.    *1302
      SUBROUTINE CHEM                                                   *1303
```

CHEM

CHEM

```
C********VERSION 5/20/69.                                                  *1304
C********CHEM IS AN OPTIONAL SUBROUTINE.  IF NOT NEEDED, SET M3 = 0.       *1305
      USE STORK                                                            *1306
      USE STORE                                                            *1307
      USE STORM                                                            *1308
      DIF(M3)982,982,981                                                   *1309
981   CONTINUE                                                             *1310
C********USED ONLY IN CHEM.  COULD BE REPLACED BY DIMENSION STATEMENT.     *1311
      COMMON /ACHEM/ AKEM(M3), CORD(M3), ET(M9,M3), KEM(M3), LTABE(M3),    *1312
     1        LTABQ(M3), LTABZ(M3), QT(M9,M3), SLOE(M9,M3), SLOQ(M9,M3),   *1313
     2        SLOZ(M9,M3), TVARE(M9,M3), TVARQ(M9,M3),                     *1314
     3        TVARZ(M9,M3), ZT(M9,M3),                                     *1315
     4        EA(M4), EB(M4), QA(M4), QB(M4), ZA(M4), ZB(M4)               *1316
C********REQUIRED WHEN CHEM IS USED, USED IN CHEM, THERM, FLOW, TALLY.     *1317
      COMMON /ACHEMS/ KA(M2), KAX(M2), KB(M2), KBX(M2),                    *1318
     1        B(M4), BB(M4), DB(M4), DDB(M4)                               *1319
      DIF(M12)972,972,971                                                  *1320
971   CONTINUE                                                             *1321
C********ARRAYS USED ONLY IF THE REMOTE TEMPERATURE DEPENDENCE OPTION IS   *1322
C********     USED (BLOCK 12).  USED IN THERM, CHEM, GEN, FLOW, AND SURE   *1323
      COMMON  /AREMS/ NLOCK(M4,8)                                          *1324
972   CONTINUE                                                             *1325
      IF(KCYC) 6100,3100,4200                                              *1326
 6100 CONTINUE                                                             *1327
CCC INPUT DATA BLOCK  3.  REACTANT PROPERTY LIST.                          *1328
      N = NKEM                                                             *1329
      LABEL = 1                                                            *1330
      IF(MCO)2100, 6105,2100                                               *1331
 6105 N = NKEM = NVARQ = NVARZ = NVARE = 0                                 *1332
 2100 READ  (    N8,5840 )A1,N1,L1,L2,L3,P1,P2,P3,NX1                      *1333
      IF(N1) 6110,2180, 6110                                               *1334
 6110 IF(MCO) 6115,2120,2120                                               *1335
 6115 CALL SEEK1(N,N1,KEM,NKEM,K)                                          *1336
      IF(K)2115,2115, 6120                                                 *1337
 6120 IF(LTABQ(N)) 6125,2105, 6125                                         *1338
 6125 NVARQ = NVARQ - 1                                                    *1339
 2105 IF(LTABZ(N)) 6130,2110, 6130                                         *1340
 6130 NVARZ = NVARZ - 1                                                    *1341
 2110 IF(LTABE(N)) 6135,2115, 6135                                         *1342
 6135 NVARE = NVARE - 1                                                    *1343
 2115 ITEMS( 3) = XMINOF(N - 1,ITEMS( 3))                                  *1344
      GO TO 2125                                                           *1345
 2120 N = NKEM = N + 1                                                     *1346
 2125 IF(N - M3)2130,2130, 6140                                            *1347
 6140 N = NKEM = M3                                                        *1348
      KWIT = 11                                                            *1349
      WRITE (    MW,5835 )N,IBLOCK                                         *1350
      PRINT 5835 ,N,IBLOCK                                                 *1351
 2130 AKEM(N) = A1                                                         *1352
      KEM(N) = N1                                                          *1353
      LTABQ(N) = L1                                                        *1354
      LTABZ(N) = L2                                                        *1355
      LTABE(N) = L3                                                        *1356
      QT(1,N) = P1                                                         *1357
      ZT(1,N) = P2                                                         *1358
      ET(1,N) = P3                                                         *1359
      LXX = 0                                                              *1360
      CALL PATCH(NX1,1.0,P4,LXX)                                           *1361
      CORD(N) = MAX1F(1.0E-24,P4)                                          *1362
      IF(LABEL)2135,2135, 6145                                             *1363
 6145 LABEL = 0                                                            *1364
      WRITE (     M,5845 )     FIN                                         *1365
 2135 WRITE (     M,5850 )AKEM(N),KEM(N),N,LTABQ(N),LTABZ(N),              *1366
     $ LTABE(N), QT(1,N),ZT(1,N),ET(1,N), CORD(N)                          *1367
      LTAB = XABSF(LTABQ(N))                                               *1368
      IF(LTAB - 2) 6150,2140,2140                                          *1369
 6150  LTABQ(N) = 0                                                        *1370
       GO TO 2150                                                          *1371
 2140 NVARQ = NVARQ + 1                                                    *1372
       LABEL = 1                                                           *1373
      READ  (    N8,5800 )(QT(J,N),TVARQ(J,N),   J=1,LTAB)                 *1374
      DO  2145  J=2,LTAB                                                   *1375
```

CHEM

```
      DTX = (TVARC(J,N)-TVARC(J-1,N)) + 1.0E-12                         *1376
      SLOC(J,N) = (CT(J,N)-CT(J-1,N))/DTX                               *1377
      IF(DTX) 6155, 6155,2145                                           *1378
 6155 KWIT = 12                                                         *1379
 2145 CONTINUE                                                          *1380
      SLOC(1,N) = SLCQ(2,N)                                             *1381
      WRITE (     M,5820 )(QT(J,N),SLOQ(J,N),TVARQ(J,N),J=1,LTAB)       *1382
      IF(LTAB - M9)2150,2150, 6160                                      *1383
 6160 KWIT = 12                                                         *1384
 2150 LTAB = XABSF(LTABZ(N))                                            *1385
      IF(LTAB - 2) 6165,2155,2155                                       *1386
 6165  LTABZ(N) = 0                                                     *1387
       GO TO 2165                                                       *1388
 2155 NVARZ = NVARZ + 1                                                 *1389
       LABEL = 1                                                        *1390
      READ  (    NB,5800 )(ZT(J,N),TVARZ(J,N), J=1,LTAB)                *1391
      DO  2160  J=2,LTAB                                                *1392
      DTX = (TVARZ(J,N)-TVARZ(J-1,N)) + 1.0E-12                         *1393
      SLOZ(J,N) = (ZT(J,N)-ZT(J-1,N))/DTX                               *1394
      IF(DTX) 6170, 6170,2160                                           *1395
 6170 KWIT = 12                                                         *1396
 2160 CONTINUE                                                          *1397
      SLOZ(1,N) = SLCZ(2,N)                                             *1398
      WRITE (     M,5825 )(ZT(J,N),SLOZ(J,N),TVARZ(J,N),J=1,LTAB)       *1399
      IF(LTAB - M9)2165,2165, 6175                                      *1400
 6175 KWIT = 12                                                         *1401
 2165 LTAB = XABSF(LTABE(N))                                            *1402
      IF(LTAB - 2) 6180,2170,2170                                       *1403
 6180 LTABE(N) = 0                                                      *1404
      GO TO 2100                                                        *1405
 2170 NVARE = NVARE + 1                                                 *1406
       LABEL = 1                                                        *1407
      READ  (    NB,5800 )(ET(J,N),TVARE(J,N), J=1,LTAB)                *1408
      DO 2175  J = 2,LTAB                                               *1409
      DTX = (TVARE(J,N)-TVARE(J-1,N)) + 1.0E-12                         *1410
      SLOE(J,N) = (ET(J,N)-ET(J-1,N))/DTX                               *1411
      IF(DTX) 6185, 6185,2175                                           *1412
 6185 KWIT = 12                                                         *1413
 2175 CONTINUE                                                          *1414
      SLOE(1,N) = SLCE(2,N)                                             *1415
      WRITE (     M,5830 )(ET(J,N),SLOE(J,N),TVARE(J,N),J=1,LTAB)       *1416
      IF(LTAB - M9)2100,2100, 6190                                      *1417
 6190 KWIT = 12                                                         *1418
      GO TO 2100                                                        *1419
 2180 NEWBL(2) = NEWBL(2) + 1000                                        *1420
      RETURN                                                            *1421
C COMPLETED BLOCK 3.                                                    *1422
 3100 IBLOCK = 2                                                        *1423
      CALL REFER(KA,KAX,NMAT,KEM,NKEM)                                  *1424
      CALL REFER(KB,KBX,NMAT,KEM,NKEM)                                  *1425
      IF(KWIT)4700, 6195,4700                                           *1426
 6195 RGAS = 1.987                                                      *1427
CCC   FIND NODE REACTION HEATS, COLL. FREQ., AND ACTIVATION ENERGIES.   *1428
      DO 3110  N = 1,NODES                                              *1429
       QA(N) = 0.                                                       *1430
       QB(N) = 0.                                                       *1431
      ZA(N) = 0.                                                        *1432
      ZB(N) = 0.                                                        *1433
      EA(N) = 0.                                                        *1434
      EB(N) = 0.                                                        *1435
      LA = KA(NODMAT(N))                                                *1436
      IF(LA)3105,3105, 6200                                             *1437
 6200 QA(N) = CT(1,LA)                                                  *1438
      ZA(N) = ZT(1,LA)                                                  *1439
      EA(N) = ET(1,LA)                                                  *1440
 3105 LB = KB(NODMAT(N))                                                *1441
      IF(LB)3110,3110, 6205                                             *1442
 6205 QB(N) = CT(1,LB)                                                  *1443
      ZB(N) = ZT(1,LB)                                                  *1444
      EB(N) = ET(1,LB)                                                  *1445
 3110 CONTINUE                                                          *1446
 4200 FORC = FOR*DELT                                                   *1447
```

CHBM

```
      IF(NCW)4300,4300, 6210
 6210 IF(KCYC - 1)4300,4205, 6215
 6215 IF(KDATA)4300,4300, 6220
 6220  IF(NVARZ + NVARQ + NVARE)4300,4300, 6225
 6225 CONTINUE
4205  WRITE (     M,5810 )     FIN
      DO 4215  N = 1,NODES
      J = NCDMAT(N)
      IF(KA(J)) 6230, 6230,4210
 6230 IF(KB(J))4215,4215, 6235
 6235 CONTINUE
4210  WRITE (     M,5815 )NODE(N),KAX(J),KBX(J),HEFT(N),
     1   QA(N),ZA(N),EA(N),QB(N),ZB(N),EB(N)
4215  CONTINUE
      WRITE (     M,5805 )     FIN
4300  IF(NVARC)4400,4400, 6240
 6240 CONTINUE
CCC   FIND NEW HEATS OF REACTION.
      DO 4360  N = 1,NODES
       IF(KA(NCDMAT(N))) 6245,4305, 6245
 6245  NN = 1
      K = KA(NCDMAT(N))
       LTAB = LTABQ(K)
       BETA = QA(N)
      GO TO 4310
4305  IF(KB(NCDMAT(N))) 6250,4360, 6250
 6250  NN = 2
      K = KB(NCDMAT(N))
       LTAB = LTABQ(K)
       BETB = QB(N)
4310  IF(LTAB) 6255,4355,4315
 6255  SET = SUMTIM + FCRD
       GO TO 4320
4315  L = N
      DIF(M12)94,94,93
93    CONTINUE
      L = NLOCK(N,3)
94    CONTINUE
      SET = T(L) + FCRD*DDT(L)
4320  MIN = 1
       MAX = XABSF(LTAB)
4325  MID = (MIN + MAX)/2
       IF(SET - TVARQ(MID,K))4330,4340,4335
4330  MAX = MID
       IF(MAX - 2)4340,4325,4325
4335  MIN = MID
       IF(MAX - MIN - 2)4340,4325,4325
4340  SET = QT(MID,K) + SLOQ(MID+1,K)*(SET - TVARQ(MID,K))
      GO TO (4345,4350),NN
4345  QA(N) = SET
       BETA = 100.0*ABSF((BETA - QA(N))/(1.0E-24 + BETA))
      DTMAX = MAX1F(DTMAX,BETA*TVARY)
       GO TO 4305
4350  QB(N) = SET
       BETB = 100.0*ABSF((BETB - QB(N))/(1.0E-24 + BETB))
      DTMAX = MAX1F(DTMAX,BETB*TVARY)
4355  GO TO (4305,4360),NN
4360  CONTINUE
CCC   FIND NEW COLLISION FREQUENCIES.
4400  IF(NVARZ)4500,4500, 6260
 6260 DO 4460  N = 1,NODES
       IF(KA(NCDMAT(N))) 6265,4405, 6265
 6265  NN = 1
      K = KA(NCDMAT(N))
      LTAB = LTABZ(K)
      BETA = ZA(N)
      GO TO 4410
4405  IF(KB(NCDMAT(N))) 6270,4460, 6270
 6270  NN = 2
      K = KB(NCDMAT(N))
       LTAB = LTABZ(K)
       BETB = ZB(N)
```

CHEM

```
4410  IF(LTAB) 6275,4455,4415                                          *1520
 6275  SET = SUMTIM + FCRD                                             *1521
       GO TO 4420                                                      *1522
4415  L = N                                                            *1523
      DIF(M12)96,96,95                                                 *1524
95    CONTINUE                                                         *1525
      L = NLCCK(N,4)                                                   *1526
96    CONTINUE                                                         *1527
      SET = T(L) + FCRD*CDT(L)                                         *1528
4420  MIN = 1                                                          *1529
       MAX = XABSF(LTAB)                                               *1530
4425  MID = (MIN + MAX)/2                                              *1531
       IF(SET - TVARZ(MID,K))4430,4440,4435                            *1532
4430  MAX = MID                                                        *1533
       IF(MAX - 2)4440,4425,4425                                       *1534
4435  MIN = MID                                                        *1535
       IF(MAX - MIN - 2)4440,4425,4425                                 *1536
4440  SET = (ZT(MID,K) + SLCZ(MID+1,K)*(SET - TVARZ(MID,K)))           *1537
      GO TO (4445,4450),NN                                             *1538
4445  ZA(N) = SET                                                      *1539
       BETA = 100.0*ABSF((BETA - ZA(N))/(1.0E-24 + BETA))              *1540
      DTMAX = MAX1F(DTMAX,BETA*TVARY)                                  *1541
       GO TO 4405                                                      *1542
4450  ZB(N) = SET                                                      *1543
      BETB = 100.0*ABSF((BETB - ZB(N))/(1.0E-24 + BETB))               *1544
      DTMAX = MAX1F(DTMAX,BETB*TVARY)                                  *1545
4455  GO TO (4405,4460),NN                                             *1546
4460  CONTINUE                                                         *1547
CCC   FIND NEW ACTIVATION ENERGIES.                                    *1548
4500  IF(NVARE)4600,4600, 6280                                         *1549
 6280 DO 4560  N = 1,NODES                                             *1550
       IF(KA(NODMAT(N))) 6285,4505, 6285                               *1551
 6285  NN = 1                                                          *1552
      K = KA(NODMAT(N))                                                *1553
       LTAB = LTABE(K)                                                 *1554
       BETA = EA(N)                                                    *1555
       GO TO 4510                                                      *1556
4505  IF(KB(NODMAT(N))) 6290,4560, 6290                                *1557
 6290  NN = 2                                                          *1558
      K = KB(NODMAT(N))                                                *1559
       LTAB = LTABE(K)                                                 *1560
      BETB = EB(N)                                                     *1561
4510  IF(LTAB) 6295,4555,4515                                          *1562
 6295  SET = SUMTIM + FCRD                                             *1563
       GO TO 4520                                                      *1564
4515  L = N                                                            *1565
      DIF(M12)98,98,97                                                 *1566
97    CONTINUE                                                         *1567
      L = NLCCK(N,5)                                                   *1568
98    CONTINUE                                                         *1569
      SET = T(L) + FCRD*CDT(L)                                         *1570
4520  MIN = 1                                                          *1571
       MAX = XABSF(LTAB)                                               *1572
4525  MID = (MIN + MAX)/2                                              *1573
       IF(SET - TVARE(MID,K))4530,4540,4535                            *1574
4530  MAX = MID                                                        *1575
       IF(MAX - 2)4540,4525,4525                                       *1576
4535  MIN = MID                                                        *1577
       IF(MAX - MIN - 2)4540,4525,4525                                 *1578
4540  SET = (ET(MID,K) + SLCE(MID+1,K)*(SET - TVARE(MID,K)))           *1579
      GO TO (4545,4550),NN                                             *1580
4545  EA(N) = SET                                                      *1581
      BETA = 100.0*ABSF((BETA - EA(N))/(1.0E-24+BETA))                 *1582
      DTMAX = MAX1F(DTMAX,BETA*TVARY)                                  *1583
       GO TO 4505                                                      *1584
4550  EB(N) = SET                                                      *1585
      BETB = 100.0*ABSF((BETB - EB(N))/(1.0E-24+BETB))                 *1586
      DTMAX = MAX1F(DTMAX,BETB*TVARY)                                  *1587
4555  GO TO (4505,4560),NN                                             *1588
4560  CONTINUE                                                         *1589
CCC   FIND CONCENTRATION AND TEMP CHANGE IN NODES.                     *1590
4600  DO 4630  N = 1,NODES                                             *1591
```

CHEM GEN

```
      K1 = KA(NCDMAT(N))                                                *1592
      K2 = KB(NCDMAT(N))                                                *1593
      TEX = MAX1F(1.0E-12,RGAS*(T(N) + FORC*CCT(N) + TBASE))            *1594
      IF(K1) 6300,4615, 6300                                            *1595
 6300 IF(A(N))4605,4615, 6305                                           *1596
 6305 DEXA = CELT*A(N)**CORC(K1)*EXPF(MIN1F(50.0,ZA(N) - EA(N)/TEX))    *1597
      IF(A(N) - DEXA)4605,4610,4610                                     *1598
4605  DEXA = A(N)                                                       *1599
4610  DT(N) = DT(N) + QA(N)*DEXA*HEFT(N)/CAP(N)                         *1600
      DA(N) = DA(N) - DEXA                                              *1601
4615  IF(K2) 6310,4630, 6310                                            *1602
 6310 IF(B(N))4620,4630, 6315                                           *1603
 6315 DEXB = CELT*B(N)**CORC(K2)*EXPF(MIN1F(50.0,ZB(N) - EB(N)/TEX))    *1604
      IF(B(N) - DEXB)4620,4625,4625                                     *1605
4620  DEXB = B(N)                                                       *1606
4625  DT(N) = DT(N) + QB(N)*DEXB*HEFT(N)/CAP(N)                         *1607
      DB(N) = DB(N) - DEXB                                              *1608
4630  CONTINUE                                                          *1609
5800  FORMAT (8E10.3)                                                   *1610
5805  FORMAT(10X,100(1H=),A1)                                           *1611
5810  FORMAT(10X,13HREACTANT DATA,//,11X,4HNODE,4X,6HA    B,4X,4HMASS,  *1612
     1   8X,2HQA,10X,2HZA,10X,2HEA,10X,2HQB,10X,2HZB,10X,2HEB,A1)       *1613
5815  FORMAT(10X,3I5,1X,1P7E12.4)                                       *1614
5820  FORMAT(/20X2HQT12X5HSLCPE10X5HTVARQ/,(15X,1P3E15.6))              *1615
5825  FORMAT(/20X2HZT12X5HSLCPE10X5HTVARZ/,(15X,1P3E15.6))              *1616
5830  FORMAT(/20X2HET12X5HSLCPE10X5HTVARE/,(15X,1P3E15.6))              *1617
5835  FORMAT(15(1H*),17HMORE THAN ALLOWED,I5,15H ITEMS IN BLOCK,I3)     *1618
5840  FORMAT(A5,4I5,5X,3E10.3,10A1)                                     *1619
5845  FORMAT(//,10X,36HNAME   REACT INDEX LTABQ LTABZ LTABE,8X,2HQT,11X, *1620
     1   2HZT,11X,2HET,11X,5HORDER,A1)                                  *1621
5850  FORMAT(10X,A5,I7,4I6,3X,1P4E13.5)                                 *1622
982   CONTINUE                                                          *1623
4700  RETURN                                                            *1624
CCC   END OF CHEM.  GO BACK TO HEART.                                   *1625
      END                                                               *1626
                                                                        *1627
*     CARDS COLUMN       GEN                                            *1628
*     LIST 8                                                            *1629
*     FORTRAN            GEN     TRUMP HEAT GENERATION SUBROUTINE.      *1630
      SUBROUTINE GEN                                                    *1631
C********VERSION 5/20/69.                                               *1632
C********GEN IS AN OPTIONAL SUBROUTINE.  IF NOT NEEDED, SET M8 = 0.     *1633
      USE STORK                                                         *1634
      USE STORE                                                         *1635
      USE STORM                                                         *1636
      DIF(M8)982,982,981                                                *1637
981   CONTINUE                                                          *1638
C********USED ONLY IN GEN.  COULD BE REPLACED BY A DIMENSION STATEMENT. *1639
      COMMON /AGEN/ GT(M9,M8), LTABG(M8), NODG(M8), NOXG(M8),           *1640
     1         SLOG(M9,M8), TVARG(M9,M8)                                *1641
C********ARRAYS USED ONLY IF SUBROUTINE GEN IS USED, G IN THERM,        *1642
C********     GEN, AND TALLY, GG IN TALLY ONLY.                          *1643
      COMMON /AGENS/  G(M4), GG(M4)                                     *1644
      DIF(M12)972,972,971                                               *1645
971   CONTINUE                                                          *1646
C********ARRAYS USED ONLY IF THE REMOTE TEMPERATURE DEPENDENCE OPTION IS *1647
C********     USED (BLOCK 12).  USED IN THERM, CHEM, GEN, FLOW, AND SURE *1648
      COMMON  /AREMS/ NLCOK(M4,8)                                       *1649
972   CONTINUE                                                          *1650
      IF(KCYC) 6100,3100,4200                                           *1651
 6100 CONTINUE                                                          *1652
CCC INPUT DATA BLOCK  8.  INTERNAL VARIABLE HEAT GENERATION LIST.       *1653
      N = NVARG                                                         *1654
      LABEL = 1                                                         *1655
      IF(MCC)2100, 6105,2100                                            *1656
 6105 N = NVARG = 0                                                     *1657
 2100 READ  (    NB,5815 )N1,NSEQ,NAOG,L1,(F1(J),F2(J),J = 1,3)         *1658
      IF(N1) 6110,2165, 6110                                            *1659
 6110 LTAB = XABSF(L1)                                                  *1660
      IF(LTAB - 2) 6115,2105,2105                                       *1661
 6115 L1 = LTAB = 0                                                     *1662
      IF(F2(1))2125, 6120,2125                                          *1663
```

GEN

GEN

```
 6120 F2(1) = 1.0E-24                                                   *1664
      GO TO 2125                                                        *1665
2105  IF(LTAB - 4)2115, 6125, 6125                                      *1666
 6125 IF(LTAB - M9)2110,2110, 6130                                      *1667
 6130 KWIT = 12                                                         *1668
2110  READ (    N8,5800 )(F1(J), F2(J), J = 4,LTAB)                     *1669
2115  DO 2120  J = 2,LTAB                                               *1670
      DTX = (F2(J) - F2(J-1)) + 1.0E-12                                 *1671
      F3(J) = (F1(J) - F1(J-1))/DTX                                     *1672
      IF(DTX) 6135, 6135,2120                                           *1673
 6135 KWIT = 12                                                         *1674
2120  CONTINUE                                                          *1675
      F3(1) = F3(2)                                                     *1676
2125  IF(MCD) 6140,2130,2130                                            *1677
 6140 CALL SEEK1(N,N1,NCXG,NVARG,K)                                     *1678
      ITEMS( 8) = XMINOF(N - 1,ITEMS( 8))                               *1679
      GO TO 2135                                                        *1680
2130  N = NVARG = N + 1                                                 *1681
2135  IF(N - M8)2140,2140, 6145                                         *1682
 6145 N = NVARG = M8                                                    *1683
      KWIT = 11                                                         *1684
      WRITE (    MW,5830 )N,IBLOCK                                      *1685
      PRINT 5830 ,N,IBLOCK                                              *1686
2140  NODG(N) = NCXG(N) = N1                                            *1687
      LTABG(N) = L1                                                     *1688
      GT(1,N) = F1(1)                                                   *1689
      TVARG(1,N) = F2(1)                                                *1690
      IF(L1)2145, 6150,2145                                             *1691
 6150 LABEL = 1                                                         *1692
      WRITE (    M,5825 )    FIN                                        *1693
2145  IF(LABEL)2150,2150, 6155                                          *1694
 6155 LABEL = 0                                                         *1695
      WRITE (    M,5820 )    FIN                                        *1696
2150  WRITE (    M,5805 )NODG(N),N,LTABG(N),GT(1,N),TVARG(1,N)          *1697
      IF(LTAB)2160,2160, 6160                                           *1698
 6160 DO 2155  J = 2,LTAB                                               *1699
      GT(J,N) = F1(J)                                                   *1700
      TVARG(J,N) = F2(J)                                                *1701
      SLOG(J,N) = F3(J)                                                 *1702
2155  CONTINUE                                                          *1703
      SLOG(1,N) = F3(1)                                                 *1704
      WRITE (    M,5810 )(GT(J,N),SLOG(J,N),TVARG(J,N),J = 2,LTAB)      *1705
2160  IF(NSEQ)2100,2100, 6165                                           *1706
 6165 NSEQ = NSEQ - 1                                                   *1707
      N1 = N1 + NADG                                                    *1708
      GO TO 2125                                                        *1709
2165  RETURN                                                            *1710
C COMPLETED BLOCK 8.                                                    *1711
3100  IF(NVARG)3105,3105, 6170                                          *1712
 6170 IBLOCK = 8                                                        *1713
      CALL REFER(NODG,NCXG,NVARG,NODE,NODES)                            *1714
3105  IF(KWIT)4700, 6175,4700                                           *1715
 6175 DO 3110  N = 1,NODES                                              *1716
       G(N) = G(N)*VOL(N)                                               *1717
3110  CONTINUE                                                          *1718
4200  FORC = FGR*DELT                                                   *1719
      IF(NVARG)4300,4300, 6180                                          *1720
 6180 CONTINUE                                                          *1721
CCC   FIND NEW HEAT GENERATION RATES.                                   *1722
      DO 4255  N = 1,NVARG                                              *1723
       J = NODG(N)                                                      *1724
       K = N                                                            *1725
      DIF(M12)94,94,93                                                  *1726
93    CONTINUE                                                          *1727
      K = NLOCK(J,6)                                                    *1728
94    CONTINUE                                                          *1729
      IF(LTABG(N)) 6185,4235,4205                                       *1730
 6185  SET = SUMTIM + FORD                                              *1731
       GO TO 4210                                                       *1732
4205  SET = T(K) + FORD*DDT(K)                                          *1733
4210  MIN = 1                                                           *1734
       MAX = XABSF(LTABG(N))                                            *1735
```

GEN
FINL

```
      BET = G(J)                                                        *1736
4215  MID = (MIN + MAX)/2                                               *1737
      IF(SET - TVARG(MID,N))4220,4230,4225                              *1738
4220  MAX = MID                                                         *1739
      IF(MAX - 2)4230,4215,4215                                         *1740
4225  MIN = MID                                                         *1741
      IF(MAX - MIN - 2)4230,4215,4215                                   *1742
4230  G(J) = VCL(J)*(GT(MID,N) + SLOG(MID+1,N)*(SET - TVARG(MID,N)))    *1743
      BET = 100.0*ABSF((BET - G(J))/(BET + 1.0E-12))                    *1744
      DTMAX = MAX1F(DTMAX,BET*TVARY)                                    *1745
      GO TO 4255                                                        *1746
4235  SET = 0.69314718/TVARG(1,N)                                       *1747
      SETD = MAX1F(-60.0,MIN1F(SET*DELT,60.0))                          *1748
      SETS = MAX1F(-60.0,MIN1F(SET*SUMTIM,60.0))                        *1749
      IF(ABSF(SETD) - 1.0E-5) 6190, 6190,4240                           *1750
 6190 SET = EXPF(-SETS)*(1.0 - 0.5*SETD)                                *1751
      GO TO 4245                                                        *1752
4240  SET = EXPF(-SETS)*(1.0 - EXPF(-SETD))/(SET*DELT)                  *1753
4245  IF(SET - 1.0E-24) 6195, 6195,4250                                 *1754
 6195 SET = 0.                                                          *1755
4250  G(J) = VCL(J)*GT(1,N)*SET                                         *1756
4255  CONTINUE                                                          *1757
CCC   FIND TEMPERATURE CHANGE FROM INTERNAL HEAT GENERATION.            *1758
4300  DO 4305  N = 1,NODES                                              *1759
4305  DT(N) = DT(N) + DELT*G(N)/CAP(N)                                  *1760
5800  FORMAT (8E10.3)                                                   *1761
5805  FORMAT(10X,3I6,3X,1PE15.6,15X,E15.6)                              *1762
5810  FORMAT(31X,1P3E15.6)                                              *1763
5815  FORMAT (4I5,6E10.3)                                               *1764
5820  FORMAT(//,12X,16HNODG INDEX LTABG,8X,2HGT,12X,5HSLOPE,11X,5HTVARG,  *1765
     1   A1)                                                            *1766
5825  FORMAT(/,10X,66HFOLLOWING NODE HAS G(N) = GT(1,N)*EXPF(-0.69315*SU  *1767
     1MTIM/TVARG(1,N)),A1)                                              *1768
5830  FORMAT(15(1H*),17HMORE THAN ALLOWED,I5,15H ITEMS IN BLOCK,I3)     *1769
982   CONTINUE                                                          *1770
4700  RETURN                                                            *1771
C COMPLETED GEN.  RETURN TO HEART.                                      *1772
      END                                                               *1773
                                                                        *1774
*     CARDS COLUMN       FINK                                           *1775
*     LIST 8                                                            *1776
*     FORTRAN            FINK    TRUMP INTERNAL HEAT FLOW SUBROUTINE.   *1777
      SUBROUTINE FINK                                                   *1778
C********VERSION 5/20/69.                                               *1779
C********FINK IS A REQUIRED SUBROUTINE FOR HEAT CONDUCTION PROBLEMS.    *1780
C********A NUMBER OF PROBLEMS INVOLVING FLOW, CHEMICAL REACTION, HEAT   *1781
C********GENERATION, CONVECTIVE AND RADIATIVE TRANSPORT, ETC., MAY BE   *1782
C********SOLVED WHICH DO NOT REQUIRE FINK.                              *1783
      USE STORK                                                         *1784
      USE STORE                                                         *1785
      USE STORM                                                         *1786
      IF(KCYC) 6100,3100,4200                                           *1787
 6100 CONTINUE                                                          *1788
CCC INPUT DATA BLOCK  5.  INTERNAL THERMAL CONNECTION LIST.             *1789
      N = NCCCN                                                         *1790
      L = 0                                                             *1791
      IF(PCC)2100, 6105,2100                                            *1792
 6105 N = NCCCN = NORAD = 0                                             *1793
 2100 READ  (    NB,5805)N1,N2,NSEQ,NAD1,NAD2,NZ,P1,P2,DLONG,DRAD,NX1,NX2  *1794
      IF(N1) 6110,2155, 6110                                            *1795
 6110 P1 = P1*SCALE                                                     *1796
      P2 = P2*SCALE                                                     *1797
      DLONG = DLONG*SCALE                                               *1798
      DRAD = DRAD*SCALE                                                 *1799
      IF(NZ)2105,2105, 6115                                             *1800
 6115 NZ = 10**NZ                                                       *1801
      NAD1 = NAD1*NZ                                                    *1802
      NAD2 = NAD2*NZ                                                    *1803
 2105 CALL PATCH(NX1,1.E12,HINTS,LBH)                                   *1804
      CALL PATCH(NX2,0.0,RINTS,LBR)                                     *1805
      IF(LBR) 6120,2110, 6120                                           *1806
 6120 NORAD = NORAD + 1 + NSEQ                                          *1807
```

FINK

FMK

```
      IF(LBP) 6125,2115, 6125                                           *1808
 6125 CONTINUE                                                          *1809
2110  IF(HINTS)2120, 6130,2120                                          *1810
 6130 CONTINUE                                                          *1811
2115  HINTS = 1.0E-24                                                   *1812
2120  ADD = 0.0                                                         *1813
      IF(DRAD) 6135,2125,2125                                           *1814
 6135 ADD = -(DRAD + DRADS)                                             *1815
      DRAD = DRADS                                                      *1816
2125  IF(NCC) 6140,2135,2135                                            *1817
 6140 CALL SEEK2(N,N1,N2,NCX1,NOX2,NOCON,K)                             *1818
      IF(K)2130,2130, 6145                                              *1819
 6145 IF(RINT(N)) 6150,2130, 6150                                       *1820
 6150 NORAD = NCRAD - 1                                                 *1821
2130  ITEMS( 5) = XMINOF(N - 1,ITEMS( 5))                               *1822
      GO TO 2140                                                        *1823
2135  N = NCCCN = N + 1                                                 *1824
2140  IF(N - M5)2145,2145, 6155                                         *1825
 6155 N = NCCCN = M5                                                    *1826
      KWIT = 11                                                         *1827
      WRITE (    MW,5835 )N,IBLOCK                                      *1828
      PRINT 5835 ,N,IBLCCK                                              *1829
2145  NOD1(N) = NCX1(N) = N1                                            *1830
      NOD2(N) = NCX2(N) = N2                                            *1831
      DEL1(N) = P1                                                      *1832
      DEL2(N) = P2                                                      *1833
      HINT(N) = HINTS                                                   *1834
      RINT(N) = RINTS                                                   *1835
      DRAD = DRADS = DRAD + ADD                                         *1836
      AREA(N) = GEOM*DLCNG*SCALE*(DRAD/SCALE)**KSYM                     *1837
      L = L + 1                                                         *1838
      IF(XMCDF(L-1,57))2150, 6160,2150                                  *1839
 6160 WRITE (     M,5810 )     FIN                                      *1840
2150  WRITE (     M,5800 )NCD1(N),NOD2(N),N,DEL1(N),DEL2(N),            *1841
     1 DLCNG,DRAD,HINT(N),RINT(N),AREA(N)                               *1842
      IF(NSEC)2100,2100, 6165                                           *1843
 6165 NSEC = NSEQ - 1                                                   *1844
      N1 = N1 + NAC1                                                    *1845
      N2 = N2 + NAC2                                                    *1846
      GO TO 2125                                                        *1847
2155  RETURN                                                            *1848
C COMPLETED BLCCK 5.                                                    *1849
3100  IBLCCK = 5                                                        *1850
      CALL REFER(NOD1,NCX1,NCCON,NODE,NODES)                            *1851
      CALL REFER(NOD2,NCX2,NOCON,NODE,NOCES)                            *1852
      IF(KWIT)4700, 6170,4700                                           *1853
 6170 CONTINUE                                                          *1854
CCC   CALCULATE CONDUCTANCE OF INTERNAL CONNECTIONS.                    *1855
      DO 3105  N = 1,NOCCN                                              *1856
      N1 = NCC1(N)                                                      *1857
      N2 = NCC2(N)                                                      *1858
      T1 = TBASE + T(N1)                                                *1859
       T2 = TBASE + T(N2)                                               *1860
      DT12 = MAX1F(1.0E-24,ABSF(T1 - T2))                               *1861
      RINTR = MAX1F(0.0,RINT(N))*SIGMA                                  *1862
       RINTC = MAX1F(0.0,-RINT(N))                                      *1863
      RAD = RINTR  *(T1 + T2)*(T1*T1 + T2*T2)                           *1864
      RAD = 1.0/(RAD + HINT(N)*DT12**RINTC)                             *1865
      TRAN(N) = AREA(N)/(CEL1(N)/CON(N1) + DEL2(N)/CON(N2) + RAD)       *1866
      DFI(N) = 0.                                                       *1867
      FI(N) = 0.                                                        *1868
3105  CONTINUE                                                          *1869
CCC   FIND TOTAL HEAT FLUX ACROSS EACH INTERNAL CONNECTION.             *1870
4200  IF(KWIT)4208, 4202, 4209                                          *1871
4202  DO 4205  N = 1, NOCON                                             *1872
4205  FI(N) = FI(N) + DFI(N)
4208  IF (NOW)4300,4300, 6175                                           *1873
 6175  IF(KCYC - 1)4300,4210, 6180                                      *1874
 6180  IF(KDATA)4300,4300, 6185                                         *1875
 6185 CONTINUE                                                          *1876
CCC   WRITE OUT PROPERTIES OF EACH INTERNAL CONNECTION.                 *1877
4210  WRITE (     M,5820 )     FIN                                      *1878
      TX = MAX1F(SUMTIM - TAU,1.0E-12)                                  *1879
```

FINK
FLOW

```
      DO 4220  N = 1,NOCCN                                              *1880
      IF(XMCDF(N-1,57))4215, 6190,4215                                  *1881
 6190 WRITE (     M,5825 )     FIN                                      *1882
4215  FX = FI(N)/TX                                                     *1883
      FXX = DFI(N)/(DELTS + 1.0E-24)
      WRITE (     M,5830 ) NOX1(N), NOX2(N), AREA(N),                   *1884
     $ HINT(N), RINT(N), TRAN(N), FI(N), FX, FXX                        *1885
4220  CONTINUE                                                          *1886
      WRITE (     M,5815 )     FIN                                      *1887
CCC   FIND NEW CONDUCTANCES OF INTERNAL CONNECTIONS.                    *1888
4300  IF(NVARK + NCRAD)4400,4400, 6195                                  *1889
 6195  FORC = FCR*DELT                                                  *1890
      DO 4310  N = 1,NOCCN                                              *1891
      N1 = NCC1(N)                                                      *1892
      N2 = NCC2(N)                                                      *1893
      IF(RINT(N))4305, 6200,4305                                        *1894
 6200  J1 = NCDMAT(N1)                                                  *1895
       J2 = NCDMAT(N2)                                                  *1896
      IF(LTABK(J1))4305, 6205,4305                                      *1897
 6205  IF(LTABK(J2)) 6210,4310, 6210                                    *1898
 6210 CONTINUE                                                          *1899
4305  T1 = TBASE + T(N1) + FCRD*DDT(N1)                                 *1900
      T2 = TBASE + T(N2)+FCRC*DDT(N2)                                   *1901
      DT12 = MAX1F(1.0E-24,ABSF(T1 - T2))                               *1902
      RINTR = MAX1F(0.0,RINT(N))*SIGMA                                  *1903
       RINTC = MAX1F(0.0,-RINT(N))                                      *1904
      RAD = RINTR  *(T1 + T2)*(T1*T1 + T2*T2)                           *1905
      RAD = 1.0/(RAD + HINT(N)*DT12**RINTC)                             *1906
      TRAN(N) = AREA(N)/(DEL1(N)/CON(N1) + DEL2(N)/CON(N2) + RAD)       *1907
4310  CONTINUE                                                          *1908
CCC   FIND TEMPERATURE CHANGES IN NODES DUE TO CONDUCTION.              *1909
4400  DO 4405  N = 1,NOCCN                                              *1910
      N1 = NCC1(N)                                                      *1911
      N2 = NCC2(N)                                                      *1912
      ZIP(N1) = ZIP(N1) + TRAN(N)                                       *1913
       ZIP(N2) = ZIP(N2) + TRAN(N)                                      *1914
      HEX = DELT*TRAN(N)*(T(N2) - T(N1))                                *1915
       DFI(N) = HEX                                                     *1916
      DF(N1) = DF(N1) + HEX                                             *1917
       DF(N2) = DF(N2) - HEX                                            *1918
      DT(N1) = DT(N1) + HEX/CAP(N1)                                     *1919
       DT(N2) = DT(N2) - HEX/CAP(N2)                                    *1920
4405  CONTINUE                                                          *1921
5800  FORMAT(10X,I5,I6,I5,1P7E12.4)                                     *1922
5805  FORMAT(2I5,3I3,I1,4E10.3,20A1)                                    *1923
5810  FORMAT(//,11X,93HNCD1 NOD2 INDEX  DEL1        DEL2        DLONG   *1924
     $    DRAD        HINT        RINT        AREA,A1)                 *1925
5815  FORMAT(10X,100(1H=),A1)                                           *1926
5820  FORMAT(10X,53HINTERNAL CONNECTION DATA.  CHECK CONDUCTANCES (TRAN)*1927
     1.,/,15X,94HLARGE DIFFERENCES BETWEEN CONNECTIONS MAY BE DUE TO POO*1928
     2R ZONING, AND MAY PRODUCE POOR RESULTS.,A1)                       *1929
5825  FORMAT(/,10X,100H    NCD1  NOD2    AREA        HINT        RINT  *1930
     $     TRAN        HEAT FLOW   AVG RATE     LAST RATE, A1)          *1931
5830  FORMAT(13X,2I6,1X,7E12.4)                ^^^^^                   *1932
5835  FORMAT(15(1H*),17HMORE THAN ALLOWED,I5,15H ITEMS IN BLOCK,I3)     *1933
4700  RETURN                                                            *1934
C COMPLETED FINK.  RETURN TO HEART.                                     *1935
      END                                                               *1936
                                                                        *1937
*     CARDS COLUMN       FLOW                                           *1938
*     LIST 8                                                            *1939
*     FORTRAN            FLOW    TRUMP MASS FLOW SUBROUTINE.            *1940
      SUBROUTINE FLOW                                                   *1941
C********VERSION 5/20/69.                                               *1942
C********FLOW IS AN OPTIONAL SUBROUTINE.  IF NOT NEEDED, SET MIC = 0.   *1943
      USE STORK                                                         *1944
      USE STORE                                                         *1945
      USE STORM                                                         *1946
      DIF(M10)982,982,981                                               *1947
981   CONTINUE                                                          *1948
C********USED ONLY IN FLOW.  COULD BE REPLACED BY A DIMENSION STATEMENT.*1949
      COMMON /AFLOW/ FLINT(M4), FLIPS(M4), FLOPS(M4), FLOLT(M4),        *1950
     1        DELF1(M10), DELF2(M10), FLAPS(M10), FLOWT(M5,M10),        *1951
```

Flow

```
     2         LTABFL(M10), NCXF1(M10), NOXF2(M10),                          *1952
     3         SLCFL(M9,M10), TVARFL(M9,M10)                                 *1953
C********REQUIRED IF FLOW IS USED, AFLOWS IS USED IN FLOW AND SPECK.         *1954
      COMMON /AFLOWS/ FLOWN(M10), NODF1(M10), NODF2(M10)                     *1955
      DIF(M12)972,972,971                                                    *1956
971   CONTINUE                                                               *1957
C********ARRAYS USED ONLY IF THE REMOTE TEMPERATURE DEPENDENCE OPTION IS     *1958
C********     USED (BLOCK 12).  USED IN THERM, CHEM, GEN, FLOW, AND SURE     *1959
      COMMON  /AREMS/ NLCOK(M4,8)                                            *1960
972   CONTINUE                                                               *1961
      DIF(M3)94,94,93                                                        *1962
93    CONTINUE                                                               *1963
C********REQUIRED WHEN CHEM IS USED, USED IN CHEM, THERM, FLOW, TALLY.       *1964
      COMMON /ACHEMS/ KA(M2), KAX(M2), KB(M2), KBX(M2),                      *1965
     1         B(M4), BB(M4), CB(M4), CCB(M4)                                *1966
94    CONTINUE                                                               *1967
      IF(KCYC) 6100,3100,4200                                                *1968
 6100 CONTINUE                                                               *1969
CCC INPUT DATA BLOCK 10.  MASS FLOW CONNECTION LIST.                         *1970
      N = NCFLOW                                                             *1971
      L = 0                                                                  *1972
      IF(MCC)2100, 6105,2100                                                 *1973
 6105 N = NCFLOW = NVARFL = 0                                                *1974
2100  READ  (    NB,5810 )N1,N2,NSEQ,NADF1,NADF2,L1,NX1,D1,D2                *1975
      IF(N1) 6110,2170, 6110                                                 *1976
 6110 D1 = D1*SCALE                                                          *1977
      D2 = D2*SCALE                                                          *1978
      DS = D1 + D2                                                           *1979
      IF(DS)2105, 6115,2105                                                  *1980
 6115 DX1 = DX2 = 0.5                                                        *1981
      GO TO 2110                                                             *1982
2105  DX1 = D2/DS                                                            *1983
      DX2 = D1/DS                                                            *1984
2110  CALL PATCH(NX1,FONE,FX,LXX)                                            *1985
      LTAB = XABSF(L1)                                                       *1986
      IF(LTAB - 2) 6120,2115,2115                                            *1987
 6120 LTAB = L1 = 0                                                          *1988
      GO TO 2130                                                             *1989
2115  IF(LTAB - M9)2120,2120, 6125                                           *1990
 6125 KWIT = 12                                                              *1991
2120  READ  (    NB,5805 )(F1(J), F2(J), J = 1,LTAB)                         *1992
      DO 2125  J = 2,LTAB                                                    *1993
      DTX = (F2(J) - F2(J-1)) + 1.0E-12                                      *1994
      F3(J) = (F1(J) - F1(J-1))/DTX                                          *1995
      IF(DTX) 6130, 6130,2125                                                *1996
 6130 KWIT = 12                                                              *1997
2125  CONTINUE                                                               *1998
      F3(1) = F3(2)                                                          *1999
      NVARFL = NVARFL + 1 + NSEQ                                             *2000
2130  IF(MCC) 6135,2140,2140                                                 *2001
 6135 CALL SEEK2(N,N1,N2,NCXF1,NOXF2,NOFLOW,K)                               *2002
      IF(K)2135,2135, 6140                                                   *2003
 6140 IF(LTABFL(N)) 6145,2135, 6145                                          *2004
 6145 NVARFL = NVARFL - 1                                                    *2005
2135  ITEMS(10) = XMINOF(N - 1,ITEMS(10))                                    *2006
      GO TO 2145                                                             *2007
2140  N = NCFLOW = N + 1                                                     *2008
2145  IF(N - M10)2150,2150, 6150                                             *2009
 6150 N = NCFLOW = M10                                                       *2010
      KWIT = 11                                                              *2011
      WRITE (    MW,5855 )N,IBLOCK                                           *2012
      PRINT 5855 ,N,IBLOCK                                                   *2013
2150  NODF1(N) = NCXF1(N) = N1                                               *2014
      NODF2(N) = NCXF2(N) = N2                                               *2015
      LTABFL(N) = L1                                                         *2016
      FLOWN(N) = FX                                                          *2017
      DELF1(N) = DX1                                                         *2018
      DELF2(N) = DX2                                                         *2019
      L = L + 1                                                              *2020
      IF(XMODF(L-1,57))2155, 6155,2155                                       *2021
 6155 WRITE (     M,5815 )    FIN                                            *2022
2155  WRITE (     M,5820 )NODF1(N),NODF2(N),N,LTABFL(N),FLOWN(N),D1,D2       *2023
```

```
      IF(LTAB)2165,2165, 6160                                           *2024
 6160 DO 2160  J = 1,LTAB                                               *2025
      FLOWT(J,N) = F1(J)                                                *2026
      TVARFL(J,N) = F2(J)                                               *2027
      SLOFL(J,N) = F3(J)                                                *2028
2160  CONTINUE                                                          *2029
      WRITE (     M,5825 ) (FLOWT(J,N), SLOFL(J,N), TVARFL(J,N),J = 1,LTA  *2030
     XB)                                                                *2031
2165  IF(NSEQ)2100,2100, 6165                                           *2032
 6165 NSEQ = NSEQ - 1                                                   *2033
      N1 = N1 + NADF1                                                   *2034
      N2 = N2 + NADF2                                                   *2035
      GO TO 2130                                                        *2036
2170  RETURN                                                            *2037
C COMPLETED BLOCK 10.                                                   *2038
3100  IBLOCK = 10                                                       *2039
      CALL REFER(NODF1,NOXF1,NOFLOW,NODE,NODES)                         *2040
      CALL REFER(NODF2,NOXF2,NOFLOW,NODE,NODES)                         *2041
      IF(KWIT)4700, 6170,4700                                           *2042
 6170 CONTINUE                                                          *2043
CCC   CHECK MATERIALS, WEIGHT FACTORS OF MASS FLOW CONNECTIONS.         *2044
      DO 3115  N = 1,NOFLOW                                             *2045
      N1 = NODF1(N)                                                     *2046
      N2 = NODF2(N)                                                     *2047
      IF(NTYPE(N1)-2)3105, 6175,3105                                    *2048
 6175 DELF1(N)=1.0                                                      *2049
      DELF2(N)=0.0                                                      *2050
      GO TO 3110                                                        *2051
3105  IF(NTYPE(N2)-2)3110, 6180,3110                                    *2052
 6180 DELF2(N)=1.0                                                      *2053
      DELF1(N)=0.0                                                      *2054
3110  IF(NODMAT(N1) - NODMAT(N2)) 6185,3115, 6185                       *2055
 6185 WRITE (     M,5800 )N,NOXF1(N),NOXF2(N)                           *2056
      PRINT                 5800 ,N,NOXF1(N),NOXF2(N)                  *2057
3115  CONTINUE                                                          *2058
CCC   INITIALIZE NET FLOW AND FLOW RATES FOR EACH NODE.                 *2059
      DO 3120  N = 1,NODES                                              *2060
      FLOPS(N) = FLIPS(N) = 0.0                                         *2061
3120  CONTINUE                                                          *2062
      DO 3125  N = 1,NOFLOW                                             *2063
      FLAPS(N) = 0.0                                                    *2064
3125  CONTINUE                                                          *2065
      DO 3135  N = 1,NOFLOW                                             *2066
      N1 = NODF1(N)                                                     *2067
      N2 = NODF2(N)                                                     *2068
CCC   REORDER CONNECTION IN DIRECTION OF POSITIVE MASS FLOW.            *2069
      IF(FLOWN(N)) 6190,3135,3130                                       *2070
 6190 FLOWN(N) = -FLOWN(N)                                              *2071
      D1 = DELF1(N)                                                     *2072
      NODF1(N) = N2                                                     *2073
      NOXF1(N) = NODE(N2)                                               *2074
      DELF1(N) = DELF2(N)                                               *2075
      NODF2(N) = N1                                                     *2076
      NOXF2(N) = NODE(N1)                                               *2077
      DELF2(N) = D1                                                     *2078
      N1 = NODF1(N)                                                     *2079
      N2 = NODF2(N)                                                     *2080
CCC   FIND NET FLOW RATES IN AND OUT OF EACH NODE.                      *2081
3130  FLOUT(N1) = FLOUT(N1) + FLOWN(N)                                  *2082
      FLINT(N2) = FLINT(N2) + FLOWN(N)                                  *2083
3135  CONTINUE                                                          *2084
CCC   START HERE DURING CYCLING.                                        *2085
4200  FORD = FOR*DELT                                                   *2086
      IF(KWIT)4210, 6195, 6195                                          *2087
 6195 DO 4205  N = 1,NOFLOW                                             *2088
       N1 = NODF1(N)                                                    *2089
       N2 = NODF2(N)                                                    *2090
      FLOWX = DELTS*FLOWN(N)                                            *2091
       FLAPS(N) = FLAPS(N) + FLOWX                                      *2092
      FLOPS(N1) = FLOPS(N1) + FLOWX                                     *2093
       FLIPS(N2) = FLIPS(N2) + FLOWX                                    *2094
4205  CONTINUE                                                          *2095
```

```
4210  IF(NCW)4300,4300, 6200                                          *2096
 6200  IF(KCYC - 1)4300,4215, 6205                                    *2097
 6205 IF(KDATA)4300,4300, 6210                                        *2098
 6210  IF(NVARFL)4300,4300, 6215                                      *2099
 6215 CONTINUE                                                        *2100
4215  WRITE (    M,5835 )    FIN                                      *2101
      TX = MAX1F(SUMTIM - TAU,1.0E-12)                                *2102
      DO 4220  N = 1,NOFLOW                                           *2103
      FX = FLAPS(N)/TX                                                *2104
      WRITE (    M,5840 ) NOXF1(N),NOXF2(N),FLOWN(N),FLAPS(N), FX,    *2105
     1   DELF1(N), DELF2(N)                                           *2106
4220  CONTINUE                                                        *2107
      WRITE (    M,5830 )    FIN                                      *2108
      WRITE (    M,5845 )    FIN                                      *2109
      DO 4225  N = 1,NODES                                            *2110
      IF(FLINT(N) + FLOUT(N)) 6220,4225, 6220                         *2111
 6220 DFLOT = HEFT(N)/MAX1F(FLINT(N),FLOUT(N))                        *2112
      FX1 = FLIPS(N)/TX                                               *2113
      FX2 = FLOPS(N)/TX                                               *2114
      WRITE (    M,5850 )  NODE(N),FLINT(N),FLOUT(N),                 *2115
     $  FLIPS(N),FLOPS(N),FX1, FX2, DFLOT                             *2116
4225  CONTINUE                                                        *2117
      WRITE (    M,5830 )    FIN                                      *2118
CCC   INITIALIZE NET FLOW INTO AND OUT OF EACH NODE EACH TIME STEP.   *2119
4300  DO 4302  N = 1,NODES                                            *2120
      FLINT(N) = FLOUT(N) = 0.0                                       *2121
4302  CONTINUE                                                        *2122
      IF(NVARFL)4400,4400, 6225                                       *2123
 6225 CONTINUE                                                        *2124
CCC   FIND NEW MASS FLOW RATES.                                       *2125
      DO 4345  N = 1,NOFLOW                                           *2126
      N1 = NODF1(N)                                                   *2127
      N2 = NODF2(N)                                                   *2128
      K1 = N1                                                         *2129
      K2 = N2                                                         *2130
      DIF(M12)98,98,97                                                *2131
97    CONTINUE                                                        *2132
       K1 = NLOOK(N1,7)                                               *2133
      K2 = NLOOK(N2,7)                                                *2134
98    CONTINUE                                                        *2135
      IF(LTABFL(N)) 6230,4345,4305                                    *2136
 6230  SET = SUMTIM + FORD                                            *2137
       GO TO 4310                                                     *2138
4305  SET = DELF1(N)*(T(K1)+FORD*DDT(K1))+DELF2(N)*(T(K2)+FORD*DDT(K2))  *2139
4310   BET = FLOWN(N)                                                 *2140
      MIN = 1                                                         *2141
      MAX  = XABSF(LTABFL(N))                                         *2142
4315  MID = (MIN + MAX)/2                                             *2143
       IF(SET - TVARFL(MID,N))4320,4330,4325                          *2144
4320  MAX = MID                                                       *2145
       IF(MAX - 2)4330,4315,4315                                      *2146
4325  MIN = MID                                                       *2147
       IF(MAX - MIN - 2)4330,4315,4315                                *2148
CCC   FIND NEW NET FLOW RATES, CONDUCTANCE CONTRIBUTIONS OF MASS FLOW.  *2149
4330  FLOWN(N) = FLOWT(MID,N) + SLOPL(MID+1,N)*(SET - TVARFL(MID,N))  *2150
CCC   REORDER CONNECTION IN DIRECTION OF POSITIVE MASS FLOW.          *2151
      IF(FLOWN(N)) 6235,4340,4340                                     *2152
 6235 FLOWN(N) = -FLOWN(N)                                            *2153
      D1 = DELF1(N)                                                   *2154
      NODF1(N) = N2                                                   *2155
      NOXF1(N) = NODE(N2)                                             *2156
      DELF1(N) = DELF2(N)                                             *2157
      NODF2(N) = N1                                                   *2158
      NOXF2(N) = NODE(N1)                                             *2159
      DELF2(N) = D1                                                   *2160
      N1 = NODF1(N)                                                   *2161
      N2 = NODF2(N)                                                   *2162
4340  BET = 100.0*ABSF((BET - FLOWN(N))/(BET + 1.0E-12))              *2163
      DTMAX = MAX1F(DTMAX,BET*TVARY)                                  *2164
4345  CONTINUE                                                        *2165
CCC   FIND TEMPERATURE CHANGES DUE TO MASS FLOW.                      *2166
4400  DO 4405  N = 1,NOFLOW                                           *2167
```

Flow
SURE

```
      N1 = NODF1(N)                                                     *2168
      N2 = NODF2(N)                                                     *2169
CCC   FIND CONTRIBUTION OF MASS FLOW TO CONDUCTANCE OF EACH NODE.       *2170
      ZIP(N2) = ZIP(N2) + FLOWN(N)*CAP(N2)/HEFT(N2)                     *2171
      FLINT(N2) = FLINT(N2) + FLOWN(N)                                  *2172
      FLOUT(N1) = FLOUT(N1) + FLOWN(N)                                  *2173
      FLEX1 = W(N1)/HEFT(N1) - FORD*CDA(N1)*HMELT(NODMAT(N1))           *2174
      FLEX2 = W(N2)/HEFT(N2) - FORD*CDA(N2)*HMELT(NODMAT(N2))           *2175
      DT(N1) = DT(N1) + FLOWN(N)*DELF2(N)*(FLEX1 - FLEX2)*DELT/CAP(N1)  *2176
      DT(N2) = DT(N2) + FLOWN(N)*DELF1(N)*(FLEX1 - FLEX2)*DELT/CAP(N2)  *2177
4405  CONTINUE                                                          *2178
      DIF(M3)96,96,95                                                   *2179
95    CONTINUE                                                          *2180
CCC   FIND CHANGE IN CONCENTRATIONS DUE TO MASS FLOW.                   *2181
      IF(NKEM)4700,4700, 6240                                           *2182
 6240 CONTINUE                                                          *2183
C********WARNING*****  MASS FLOW TRANSPORT OF CHEMICAL REACTANTS IS NOT *2184
C********     UNCONDITIONALLY STABLE.  A MAXIMUM TIME STEP (DELTO) MUST *2185
C********     BE SPECIFIED, NO GREATER THAN THE LEAST RESIDENCE TIME.   *2186
      DO 4505  N = 1,NOFLOW                                             *2187
      N1 = NODF1(N)                                                     *2188
      N2 = NODF2(N)                                                     *2189
      FLOWX = DELT*FLOWN(N)                                             *2190
      IF(KA(NODMAT(N1))) 6245,4500, 6245                                *2191
 6245 IF(KA(NODMAT(N2))) 6250,4500, 6250                                *2192
 6250 FLEX = FLOWX*(A(N1) - A(N2) + FORD*(DDA(N1) - DDA(N2)))           *2193
      DA(N1) = DA(N1) + DELF2(N)*FLEX/HEFT(N1)                          *2194
      DA(N2) = DA(N2) + DELF1(N)*FLEX/HEFT(N2)                          *2195
4500  IF(KB(NODMAT(N1))) 6255,4505, 6255                                *2196
 6255 IF(KB(NODMAT(N2))) 6260,4505, 6260                                *2197
 6260 FLEX = FLOWX*(B(N1) - B(N2) + FORD*(DDB(N1) - DDB(N2)))           *2198
      DB(N1) = DB(N1) + DELF2(N)*FLEX/HEFT(N1)                          *2199
      DB(N2) = DB(N2) + DELF1(N)*FLEX/HEFT(N2)                          *2200
4505  CONTINUE                                                          *2201
96    CONTINUE                                                          *2202
                                                                        *2203
5800  FORMAT(/10X,20HMASS FLOW CONNECTION,I5,5X,5HNODES,I5,             *2204
     $  3HAND,I5,5X,27HARE OF DIFFERENT MATERIALS.)                     *2205
5805  FORMAT(8E10.3)                                                    *2206
5810  FORMAT(6I5,10A1,2E10.3)                                           *2207
5815  FORMAT(//,11X,24HNODF1 NODF2 INDEX LTABFL,3X,5HFLOWT,10X,         *2208
     1   5HSLCFL,10X,6HTVARFL,9X,5HDELF1,10X,5HDELF2,A1)                *2209
5820  FORMAT(10X,4I6,1X,1PE15.6,30X,2E15.6)                             *2210
5825  FORMAT(35X,3E15.6)                                                *2211
5830  FORMAT(10X,100(1H=),A1)                                           *2212
5835  FORMAT(10X,14HMASS FLOW DATA,//,10X,75HSOURCE  SINK   FLOW RATE   *2213
     1  NET FLOW    AVG RATE     SOURCE WT     SINK WT,A1)              *2214
5840  FORMAT(10X,2I6,1P5E13.5)                                          *2215
5845  FORMAT(/,10X,99H  NODE   RATE IN      RATE OUT   NET FLOW IN  NET *2216
     1FLOW OUT AVG FLOW IN  AVG FLOW OUT RESIDENCE TIME,A1)             *2217
5850  FORMAT(10X,I6,1P7E13.5)                                           *2218
5855  FORMAT(15(1H*),17HMORE THAN ALLOWED,I5,15H ITEMS IN BLOCK,I3)     *2219
982   CONTINUE                                                          *2220
4700  RETURN                                                            *2221
C COMPLETED FLOW.  RETURN TO HEART.                                     *2222
      END                                                               *2223
                                                                        *2224
*     CARDS COLUMN       SURE                                           *2225
*     LIST 8                                                            *2226
*     FORTRAN            SURE    TRUMP EXTERNAL HEAT FLOW SUB.FORTRAN-400 *2227
      SUBROUTINE SURE                                                   *2228
C********VERSION 5/27/69.                                               *2229
C********SURE IS AN OPTIONAL SUBROUTINE.  IF NOT NEEDED, SET M6 = M7 = 0 *2230
C********(SEE USERS MANUAL FOR METHODS OF OBTAINING VARIOUS TYPES OF    *2231
C********BOUNDARY CONDITIONS WITHOUT USING INPUT DATA BLOCKS 6 OR 7.)   *2232
      USE STORK                                                         *2233
      USE STORE                                                         *2234
      USE STORM                                                         *2235
      DIF(M6)982,982,981                                                *2236
```

SURE

```
981   CONTINUE                                                          *2237
      DIF(M7)984,984,983                                                *2238
983   CONTINUE                                                          *2239
C********USED ONLY IN SURE.  COULD BE REPLACED BY A DIMENSION STATEMENT.  *2240
      COMMON /ASURE/ AREAS(M6), FS(M6), FSURE(M6), FSURT(M9,M6),        *2241
     1        LTABF(M6), NOXS(M6), NOCSB(M6), NOXSB(M6), POWER(M6),     *2242
     2        RSURE(M6), SLOP(M9,M6), TVARF(M9,M6),                     *2243
     3        FB(M7), LTABT(M7), NOCB(M7), SLOT(M9,M7), TB(M7),         *2244
     4        TBS(M7), TEMPB(M9,M7), TIMEB(M9,M7), DFB(M7)              *2245
C********REQUIRED IF SURE IS USED, ASURES IS USED IN SURE AND SPECK.    *2246
      COMMON /ASURES/ DFS(M6), NODS(M6), TRANS(M6)                      *2247
      DIF(M12)972,972,971                                               *2248
971   CONTINUE                                                          *2249
C********ARRAYS USED ONLY IF THE REMOTE TEMPERATURE DEPENDENCE OPTION IS *2250
C********      USED (BLOCK 12).  USED IN THERM, CHEM, GEN, FLOW, AND SURE *2251
      COMMON  /AREMS/ NLCOK(M4,8)                                       *2252
972   CONTINUE                                                          *2253
      IF(KCYC) 6100,3100,4200                                           *2254
 6100 IF(IBLOCK - 7) 6105,2200, 6105                                    *2255
 6105 IF(IBLOCK - 6)4700, 6110,4700                                     *2256
 6110 CONTINUE                                                          *2257
CCC INPUT DATA BLOCK  6.  EXTERNAL THERMAL CONNECTION LIST.             *2258
      N = NCSCON                                                        *2259
      LABEL = L = 0                                                     *2260
      IF(MCC)2100, 6115,2100                                            *2261
 6115 N = NCSCON = NVARH = NORADS = NOPOWS = 0                          *2262
 2100 READ  (    NB,5815 )N1,N2,NSEQ,NADS,NADSB,L1,DLONG,DRAD,NX1,NX2,NX3 *2263
      IF(N1) 6120,2195, 6120                                            *2264
 6120 DLONG = DLONG*SCALE                                               *2265
      DRAD = DRAD*SCALE                                                 *2266
      ADD = 0.0                                                         *2267
      IF(DRAD) 6125,2105,2105                                           *2268
 6125 ADD = -(DRAD + DRADS)                                             *2269
      DRAD = DRADS                                                      *2270
 2105 CALL PATCH(NX1,HONE,HX,LXX)                                       *2271
      CALL PATCH(NX2,RONE,RX,LXX)                                       *2272
      IF(RX) 6130,2110, 6130                                            *2273
 6130 NORADS = NORADS + 1 + NSEQ                                        *2274
 2110 CALL PATCH(NX3,PONE,PX,LXX)                                       *2275
      IF(PX) 6135,2115, 6135                                            *2276
 6135 NOPOWS = NOPOWS + 1 + NSEQ                                        *2277
 2115 LTAB = XABSF(L1)                                                  *2278
      IF(LTAB - 2) 6140,2120,2120                                       *2279
 6140 LTAB = L1 = 0                                                     *2280
      GO TO 2135                                                        *2281
 2120 NVARH = NVARH + 1 + NSEQ                                          *2282
      IF(LTAB - M9)2125,2125, 6145                                      *2283
 6145 KWIT = 12                                                         *2284
 2125 READ  (    NB,5800 )(F1(J), F2(J), J = 1,LTAB)                    *2285
      DO 2130  J = 2,LTAB                                               *2286
      DTX = (F2(J) - F2(J-1)) + 1.0E-12                                 *2287
      F3(J) = (F1(J) - F1(J-1))/DTX                                     *2288
      IF(DTX) 6150, 6150,2130                                           *2289
 6150 KWIT = 12                                                         *2290
 2130 CONTINUE                                                          *2291
      F3(1) = F3(2)                                                     *2292
 2135 IF(MCC) 6155,2155,2155                                            *2293
 6155 CALL SEEK2(N,N1,N2,NOXS,NOXSB,NOSCON,K)                           *2294
      IF(K)2150,2150, 6160                                              *2295
 6160 IF(RSURE(N)) 6165,2140, 6165                                      *2296
 6165 NORADS = NORADS - 1                                               *2297
 2140 IF(LTABF(N)) 6170,2145, 6170                                      *2298
 6170 NVARH = NVARH - 1                                                 *2299
 2145 IF(POWER(N)) 6175,2150, 6175                                      *2300
 6175 NOPOWS = NOPOWS - 1                                               *2301
 2150 ITEMS( 6) = XMINOF(N - 1,ITEMS( 6))                               *2302
      GO TO 2160                                                        *2303
 2155 N = NCSCON = N + 1                                                *2304
 2160 IF(N - M6)2165,2165, 6180                                         *2305
 6180 N = NCSCON = M6                                                   *2306
      KWIT = 11                                                         *2307
      WRITE (    MW,5885 )N,IBLOCK                                      *2308
```

SUBE

```
      PRINT 5885 ,N,IBLCCK                                              *2309
 2165 NODS(N) = NCXS(N) = N1                                            *2310
      NODSB(N) = NCXSB(N) = N2                                          *2311
      LTABH(N) = L1                                                     *2312
      HSURE(N) = HSURT(1,N) = HX                                        *2313
      RSURE(N) = RX                                                     *2314
      POWER(N) = PX                                                     *2315
      DRAC = CRADS = DRAC + ADD                                         *2316
      AREAS(N) = GEOM*DLCNG*SCALE*(DRAD/SCALE)**KSYM                    *2317
      IF(LABEL)2170,2170, 6185                                          *2318
 6185 LABEL = 0                                                         *2319
      GO TC 2175                                                        *2320
 2170 L = L + 1                                                         *2321
      IF(XMCDF(L-1,57))2180, 6190,2180                                  *2322
 6190 CONTINUE                                                          *2323
 2175 WRITE (     M,5820 )     FIN                                      *2324
 2180 WRITE (     M,5825 )NCCS(N),NODSB(N),N,LTABH(N),POWER(N),          *2325
     $ DLCNG,CRAD,HSURE(N),RSURE(N),AREAS(N)                            *2326
      IF(LTAB)2190,2190, 6195                                           *2327
 6195 DO 2185  J = 1,LTAB                                               *2328
      HSURT(J,N) = F1(J)                                                *2329
      TVARH(J,N) = F2(J)                                                *2330
      SLOH(J,N) = F3(J)                                                 *2331
 2185 CONTINUE                                                          *2332
      LABEL = 1                                                         *2333
      WRITE (     M,5830 )(HSURT(J,N),SLOH(J,N),TVARH(J,N),J=1,LTAB)    *2334
 2190 IF(NSEQ)2100,2100, 6200                                           *2335
 6200 NSEC = NSEQ - 1                                                   *2336
      N1 = N1 + NADS                                                    *2337
      N2 = N2 + NACSB                                                   *2338
      GO TC 2135                                                        *2339
 2195 RETURN                                                            *2340

C CCMPLETEC BLCCK 6.                                                    *2341
CCC INPUT DATA BLOCK  7.  EXTERNAL TEMPERATURE (BOUNDARY NCDE) LIST.    *2342
 2200 N = NCDBS                                                         *2343
      LABEL = 1                                                         *2344
      IF(MCC)2205, 6205,2205                                            *2345
 6205 N = NCDBS = NVART = 0                                             *2346
 2205 REAC  (    NB,5835 )N1,L1,(F1(J),F2(J),J = 1,3)                   *2347
      IF(N1) 6210,2265, 6210                                            *2348
 6210 IF(MCD) 6215,2215,2215                                            *2349
 6215 CALL SEEK1(N,N1,NCCB,NCDBS,K)                                     *2350
      IF(K)2210,2210, 6220                                              *2351
 6220 IF(LTABT(N)) 6225,2210, 6225                                      *2352
 6225 NVART = NVART - 1                                                 *2353
 2210 ITEMS( 7) = XMINOF(N - 1,ITEMS( 7))                               *2354
      GC TC 2220                                                        *2355
 2215 N = NCDBS = N + 1                                                 *2356
 2220 IF(N - M7)2225,2225, 6230                                         *2357
 6230 N = NCDBS = M7                                                    *2358
      KWIT = 11                                                         *2359
      WRITE (    MW,5885 )N,IBLOCK                                      *2360
      PRINT 5885 ,N,IBLCCK                                              *2361
 2225 NCDB(N) = N1                                                      *2362
      LTABT(N) = L1                                                     *2363
      DO 2230  J = 1,3                                                  *2364
      TEMPB(J,N) = F1(J)                                                *2365
      TIMEB(J,N) = F2(J)                                                *2366
 2230 CCNTINUE                                                          *2367
      LTABT(N) = LTAB = XABSF(LTABT(N))                                 *2368
      TB(N) = TBS(N) = TEMPB(1,N)                                       *2369
      IF(LTAB - 100)2235, 6235, 6235                                    *2370
 6235 LTAB = 2                                                          *2371
      SLOT(2,N) = 0.0                                                   *2372
      IF(TIMEB(1,N)) 6240, 6240,2235                                    *2373
 6240 TIMEB(1,N) = 1.0E24                                               *2374
 2235 IF(LABEL)2240,2240, 6245                                          *2375
 6245 LABEL = 0                                                         *2376
      WRITE (     M,5840 )     FIN                                      *2377
 2240 WRITE (     M,5845 )NCCB(N),N,LTABT(N),TEMPB(1,N),TIMEB(1,N)      *2378
      IF(LTAB - 2) 6250,2245,2245                                       *2379
```

SURE

```
 6250 LTABT(N) = 0                                                      *2380
      GO TO 2205                                                        *2381
 2245 NVART = NVART + 1                                                 *2382
      IF(LTAB - 4)2250, 6255, 6255                                      *2383
 6255 LABEL = 1                                                         *2384
      READ (    NB,5800 ) (TEMPB(J,N),TIMEB(J,N), J=4,LTAB)             *2385
 2250 IF(LTABT(N) - 100) 6260,2256,2256                                 *2386
 6260 DO  2255  J = 2,LTAB                                              *2387
      DTX = (TIMEB(J,N)-TIMEB(J-1,N)) + 1.0E-12                         *2388
      SLOT(J,N) = (TEMPB(J,N) - TEMPB(J-1,N))/DTX                       *2389
      IF(DTX) 6265, 6265,2255                                           *2390
 6265 KWIT = 12                                                         *2391
 2255 CONTINUE                                                          *2392
 2256 SLOT(1,N) = SLOT(2,N)                                             *2393
      WRITE (    M,5850 )(TEMPB(J,N),SLOT(J,N),TIMEB(J,N),J=2,LTAB)     *2394
      IF(LTAB - M9)2260,2260, 6270                                      *2395
 6270 KWIT = 12                                                         *2396
 2260 IF(LTABT(N) - 100)2205, 6275, 6275                                *2397
 6275 WRITE (    M,5880 )    FIN                                        *2398
      GO TO 2205                                                        *2399
 2265 NEWBL(6) = NEWBL(6) + 1000                                        *2400
      RETURN                                                            *2401
C COMPLETED BLOCK 7.                                                    *2402
 3100 IBLOCK = 6                                                        *2403
      CALL REFER(NODS,NOXS,NOSCON,NODE,NODES)                           *2404
      CALL REFER(NODSB,NOXSB,NOSCON,NODB,NODBS)                         *2405
      IF(KWIT)4700, 6280,4700                                           *2406
 6280 CONTINUE                                                          *2407
CCC   INITIALIZE FLUX, CHANGE NTYPE TO 1, UNLESS ALREADY 2.             *2408
      DO 3105  N=1,NOSCON                                               *2409
      FS(N)=DFS(N)=0.                                                   *2410
      N1=NODS(N)                                                        *2411
      N2=NTYPE(N1)                                                      *2412
      IF(N2 - 2) 6285,3105, 6285                                        *2413
 6285 NTYPE(N1) = 1                                                     *2414
      IF(N2)3105, 6290,3105                                             *2415
 6290 NOSPEC = NOSPEC + 1                                               *2416
 3105 CONTINUE                                                          *2417
      FLUXS = 0.                                                        *2418
CCC   CALC CONDUCTANCE BETWEEN SURFACE AND BOUNDARY NODES.              *2419
      DO 3110  N = 1,NOSCON                                             *2420
       N1 = NODS(N)                                                     *2421
       N2 = NODSB(N)                                                    *2422
      HSURE(N) = HSURT(1,N)                                             *2423
      T1 = TBASE + T(N1)                                                *2424
      T2 = TB(N2) + TBASE                                               *2425
      DT12 = MAX1F(1.0E-24,ABSF(T1 - T2))                               *2426
      RAD = SIGMA*HSURE(N)*(T1 + T2)*(T1*T1 + T2*T2)                    *2427
      TRANS(N) = AREAS(N)*(RAD + HSURE(N)*DT12**POWER(N))               *2428
      FB(N2) = 0.0                                                      *2429
 3110 CONTINUE                                                          *2430
 4200 FORC = FOR*DELT                                                   *2431
      IF(KWIT)4300, 6295, 6295                                          *2432
 6295 CONTINUE                                                          *2433
CCC   FIND TOTAL HEAT FLUX ACROSS EACH EXTERNAL CONNECTION.             *2434
      DO 4205  N = 1,NOSCON                                             *2435
      N2 = NODSB(N)                                                     *2436
      FS(N) = FS(N) + DFS(N)                                            *2437
      DFB(N2) = DFB(N2) + DFS(N)
      FB(N2) = FB(N2) + DFS(N)                                          *2438
 4205 CONTINUE                                                          *2439
      IF(NOW)4300,4300, 6300                                            *2440
 6300  IF(KCYC - 1)4300, 6305, 6305                                     *2441
 6305 WRITE (    M,5865 )    FIN                                        *2442
      TX = MAX1F(SUMTIM - TAU,1.0E-12)                                  *2443
      FLUXS = 0.0                                                       *2444
      FLUXSS = 0.0
      DO 4210  N = 1,NODES                                              *2445
      FX = FB(N)/TX                                                     *2446
      FLUXS = FLUXS + FB(N)                                             *2447
      FLUXSS = FLUXSS + DFB(N)
      FXX = DFB(N)/(DELTS + 1.0E-24)
      WRITE (    M,5870 ) NODB(N), TB(N), FB(N),  FX, FXX               *2448
 4210 CONTINUE                                                          *2449
      FX = FLUXS/TX                                                     *2450
      FXX = FLUXSS/(DELTS + 1.0E-24)
      WRITE (    M,5875 )FLUXS, FX, FXX                                 *2451
```

```
      WRITE (     M,5805 )     FIN                                    *2452
      IF(KCYC - 1)4300,4215, 6310                                     *2453
 6310  IF(KDATA)4300,4300, 6315                                       *2454
 6315 CONTINUE                                                        *2455
4215  WRITE (     M,5810 )     FIN                                    *2456
      DO 4225  N =1,NOSCON                                            *2457
      FX = FS(N)/TX                                                   *2458
      FXX = DFS(N)/(DELTS + 1.0E - 24)
      IF(XMODF(N - 1,57))4220, 6320,4220                              *2459
 6320 WRITE (     M,5855 )     FIN                                    *2460
4220  WRITE (     M,5860 ) NOXS(N), NOXSB(N), AREAS(N),               *2461
     $ HSURE(N), POWER(N),RSURE(N),TRANS(N), FS(N), FX, FXX            *2462
4225  CONTINUE                                                        *2463
      WRITE (     M,5805 )     FIN                                    *2464
CCC   FIND NEW BOUNDARY NODE TEMPERATURES.                            *2465
4300  IF(NVART)4400,4400, 6325                                        *2466
 6325 DO 4335  N = 1,NODES                                            *2467
      IF(LTABT(N)) 6330,4335, 6330                                    *2468
 6330 BET = TB(N)                                                     *2469
      SET = SUMTIM + DELT                                             *2470
      IF(LTABT(N) - 100)4305, 6335, 6335                              *2471
 6335 TB(N) = SINF(6.28318531*(SET + TIMEB(2,N))/(TIMEB(1,N)))        *2472
      TB(N) = TEMPB(1,N) + TEMPB(2,N)*TB(N)                           *2473
      GO TO 4330                                                      *2474
4305  MIN = 1                                                         *2475
      MAX = LTABT(N)                                                  *2476
4310  MID = (MIN + MAX)/2                                             *2477
       IF(SET - TIMEB(MID,N))4315,4325,4320                           *2478
4315  MAX = MID                                                       *2479
       IF(MAX - 2)4325,4310,4310                                      *2480
4320  MIN = MID                                                       *2481
       IF(MAX - MIN - 2)4325,4310,4310                                *2482
4325  TB(N) = TEMPB(MID,N) + SLOT(MID+1,N)*(SET - TIMEB(MID,N))       *2483
4330  TBS(N) = BET + FOR*(TB(N) - BET)                                *2484
      DTMAX = MAX1F(DTMAX,ABSF(BET - TB(N)))                          *2485
4335  CONTINUE                                                        *2486
CCC   FIND NEW SURFACE HEAT TRANSFER COEFFICIENTS.                    *2487
4400  IF(NVARH)4500,4500, 6340                                        *2488
 6340 DO 4435  N = 1,NOSCON                                           *2489
       IF(LTABH(N)) 6345,4435,4405                                    *2490
 6345 SET = SUMTIM + FORD                                             *2491
       GO TO 4410                                                     *2492
4405  N1 = NODS(N)                                                    *2493
       N2 = NODSB(N)                                                  *2494
       K1 = N                                                         *2495
      DIF(M12)94,94,93                                                *2496
93    CONTINUE                                                        *2497
      K1 = NLOOK(N1,8)                                                *2498
94    CONTINUE                                                        *2499
      SET = 0.5*(TBS(N2) + T(K1) + FORD*DDT(K1))                      *2500
4410  MIN = 1                                                         *2501
      MAX = XABSF(LTABH(N))                                           *2502
      BET = HSURE(N)                                                  *2503
4415  MID = (MIN + MAX)/2                                             *2504
       IF(SET - TVARH(MID,N))4420,4430,4425                           *2505
4420  MAX = MID                                                       *2506
       IF(MAX - 2)4430,4415,4415                                      *2507
4425  MIN = MID                                                       *2508
       IF(MAX - MIN - 2)4430,4415,4415                                *2509
4430  HSURE(N) = HSURT(MID,N) + SLOH(MID+1,N)*(SET - TVARH(MID,N))    *2510
      BET = 100.0*ABSF(BET - HSURE(N))/(ABSF(BET) + 1.0E-12)          *2511
      DTMAX = MAX1F(DTMAX,BET*TVARY)                                  *2512
4435  CONTINUE                                                        *2513
CCC   FIND NEW SURFACE BOUNDARY CONDUCTANCES.                         *2514
4500  IF(NVARH + NORADS + NOPOWS)4600,4600, 6350                      *2515
 6350 DO 4510  N = 1,NOSCON                                           *2516
       IF(LTABH(N))4505, 6355,4505                                    *2517
 6355 IF(RSURE(N))4505, 6360,4505                                     *2518
 6360  IF(POWER(N)) 6365,4510, 6365                                   *2519
 6365 CONTINUE                                                        *2520
4505  N1 = NODS(N)                                                    *2521
      N2 = NODSB(N)                                                   *2522
      T1 = TBASE + T(N1) + FORD *DDT(N1)                              *2523
```

SURE
SPECK

```
      T2 = TBASE + TBS(N2)                                              *2524
      DT12 = MAX1F(1.0E-24,ABSF(T1 - T2))                               *2525
      RAD = SIGMA*RSURE(N)*(T1 + T2)*(T1*T1 + T2*T2)                    *2526
      TRANS(N) = AREAS(N)*(RAD + HSURE(N)*DT12**POWER(N))               *2527
4510  CONTINUE                                                          *2528
CCC   FIND TEMPERATURE CHANGES IN SURFACE NODES.                        *2529
4600  DO 4605  N=1,NCSCDN                                               *2530
      N1=NCDS(N)                                                        *2531
      N2=NCDSB(N)                                                       *2532
      ZIP(N1) = ZIP(N1) + TRANS(N)                                      *2533
      HEX = DELT*TRANS(N)*(TBS(N2) - T(N1))                             *2534
      DT(N1) = DT(N1) + HEX/CAP(N1)                                     *2535
      DF(N1) = DF(N1) + HEX                                             *2536
       DFS(N) = HEX                                                     *2537
4605  CONTINUE                                                          *2538
5800  FORMAT (8E10.3)                                                   *2539
5805  FORMAT(10X,100(1H=),A1)                                           *2540
5810  FORMAT(10X,24HEXTERNAL CONNECTION DATA,A1)                        *2541
5815  FORMAT (6I5,2E10.3,30A1)                                          *2542
5820  FORMAT(/,10X,96H  NODS NODSB INDEX LTABH        POWER        DLCNG  *2543
     1      DRAD         HSURE        RSURE        AREAS,A1)           *2544
5825  FORMAT(10X,4I6,4X,1P6E12.4)                                       *2545
5830  FORMAT(/,19X,5HHSURT,10X,5HSLOPE,10X,5HTVARH,/,(15X,3E15.6))      *2546
5835  FORMAT (2I5,10X,6E10.3)                                           *2547
5840  FORMAT(//,12X,16HNODB INDEX LTABT,4X5HTEMPB,10X5HSLOPE,10X5HTIMEB, *2548
     1   A1)                                                            *2549
5845  FORMAT(10X,3I6,1PE15.6,15X,E15.6)                                 *2550
5850  FORMAT(28X,1P3E15.6)                                              *2551
5855  FORMAT(/,10X,96H  NODS NODSB    AREAS        HSURE        POWER   *2552
     $  RSURE        TRANS       HEAT FLOW    AVG RATE,5X, 9HLAST RATE, A1) *2553
5860  FORMAT( 8X, 2I6, 8E12.4)                                          *2554
5865  FORMAT(10X,18HBOUNDARY NODE DATA,//,10X,5H NODB,4X,5HTEMPB,       *2555
     1   8X,9HHEAT FLOW,4X,8HAVG RATE, 5X, 9HLAST RATE, A1)             *2556
5870  FORMAT(10X,I5, 4E13.4)                                            *2557
5875  FORMAT(/,10X,12HSYSTEM TOTAL,6X, 3E13.4)                          *2558
5880  FORMAT(//,18X,65HTB = TEMPB(1) + TEMPB(2)*SINF(2.*PI*(SUMTIM + TIM *2559
     1EB(2))/TIMEB(1)),A1)                                              *2560
5885  FORMAT(15(1H*),17HMORE THAN ALLOWED,I5,15H ITEMS IN BLOCK,I3)     *2561
984   CONTINUE                                                          *2562
982   CONTINUE                                                          *2563
4700  RETURN                                                            *2564
C COMPLETED SURE.  RETURN TO HEART.                                     *2565
      END                                                               *2566
                                                                        *2567
*     CARDS COLUMN       SPECK                                          *2568
*     LIST 8                                                            *2569
*     FORTRAN            SPECK   TRUMP SPECIAL NODE CALC SUBROUTINE.    *2570
      SUBROUTINE SPECK                                                  *2571
C********VERSION 5/27/69.                                               *2572
C********SPECK IS A REQUIRED SUBROUTINE.                                *2573
      USE STORK                                                         *2574
      USE STORE                                                         *2575
      USE STORM                                                         *2576
      DIF(M10)92,92,91                                                  *2577
91    CONTINUE                                                          *2578
C********REQUIRED IF FLOW IS USED, AFLOWS IS USED IN FLOW AND SPECK.    *2579
      COMMON /AFLOWS/ FLOWN(M10), NODF1(M10), NODF2(M10)                *2580
92    CONTINUE                                                          *2581
      DIF(M6)94,94,93                                                   *2582
93    CONTINUE                                                          *2583
C********REQUIRED IF SURE IS USED, ASURES IS USED IN SURE AND SPECK.    *2584
      COMMON /ASURES/ DFS(M6), NODS(M6), TRANS(M6)                      *2585
94    CONTINUE                                                          *2586
      IF(KCYC) 6100,4200,4200                                           *2587
 6100 NUTX = 0                                                          *2588
      NUTSUM = 0                                                        *2589
      NUTMAX = 80                                                       *2590
CCC   INITIALIZE CORRECTIONS TO TEMP CHANGES IN NODES.                  *2591
      DO 3100  N = 1,NODES                                              *2592
      ERRCRX(N) = 0.0                                                   *2593
3100  CONTINUE                                                          *2594
CCC   SET ACCELERATION FACTOR AND CONVERGENCE FACTOR.                   *2595
```

```
      NUTS = 0                                                          *2596
       SPEED = 0.2                                                      *2597
       ERRS = 5.E-5                                                     *2598
       RETURN                                                           *2599
4200  IF(NCW)4205,4205, 6105                                            *2600
 6105 CONTINUE                                                          *2601
C START HERE IF KCYC   -1.                                              *2602
      NCYC = KCYC - 1                                                   *2603
      NUTAVG = NUTSUM/XMAXOF(1,NCYC)                                    *2604
      WRITE (     M,5810 ) NCYC, NUTSUM, NUTAVG, NUTX, FOR              *2605
      IF(KWIT)4205,4205, 6110                                           *2606
 6110 PRINT 5810, NCYC, NUTSUM, NUTAVG, NUTX, FOR                       *2607
      GO TO 4700                                                        *2608
4205  IF(KWIT) 6115, 6115,4700                                          *2609
 6115 FORC = FOR*DELT                                                   *2610
CCC   INITIALIZE CORRECTION FACTORS.                                    *2611
      DO 4210  N = 1,NODES                                              *2612
      ERRCR(N) = 0.0                                                    *2613
      IF(NTYPE(N)) 6120,4210, 6120                                      *2614
 6120  ERRCRX(N) = DELT*DOT(N)                                          *2615
4210  CONTINUE                                                          *2616
CCC   CALC CORRECTION FACTORS FOR SPECIAL NODES, COUNT SPEC-SPEC CONN.  *2617
      KNOCK = KNOFL = KNCT = 0                                          *2618
      IF(NCCON)4230,4230, 6125                                          *2619
 6125 DO 4225  N = 1,NOCON                                              *2620
      N1 = NCC1(N)                                                      *2621
      N2 = NCC2(N)                                                      *2622
      IF(NTYPE(N1))4215, 6130,4215                                      *2623
 6130 IF(NTYPE(N2)) 6135,4225, 6135                                     *2624
 6135 ERRCR(N2) = ERROR(N2) + TRAN(N)*DT(N1)                            *2625
      GO TO 4225                                                        *2626
4215  IF(NTYPE(N2))4220, 6140,4220                                      *2627
 6140 ERRCR(N1) = ERROR(N1) + TRAN(N)*DT(N2)                            *2628
      GO TO 4225                                                        *2629
4220  ERRCR(N1) = ERROR(N1) + TRAN(N)*ERRORX(N2)                        *2630
      ERRCR(N2) = ERROR(N2) + TRAN(N)*ERRORX(N1)                        *2631
       KNOCK = KNOCK + 1                                                *2632
4225  CONTINUE                                                          *2633
4230  CONTINUE                                                          *2634
      DIF(MLO)96,96,95                                                  *2635
95    CONTINUE                                                          *2636
CCC   FIND CORRECTIONS DUE TO MASS FLOW.                                *2637
      IF(NCFLOW)4300,4300, 6145                                         *2638
 6145 DO 4240  N = 1,NOFLOW                                             *2639
      N1 = NCOF1(N)                                                     *2640
      N2 = NCOF2(N)                                                     *2641
      IF(NTYPE(N1))4235, 6150,4235                                      *2642
 6150 IF(NTYPE(N2)) 6155,4240, 6155                                     *2643
 6155 ERRCR(N2) = ERROR(N2) + FLOWN(N)*CAP(N1)*DT(N1)/HEFT(N1)          *2644
      GO TO 4240                                                        *2645
4235  IF(NTYPE(N2)) 6160,4240, 6160                                     *2646
 6160 KNOFL = KNOFL + 1                                                 *2647
      ERRCR(N2)=ERRCR(N2) + FLOWN(N)*CAP(N1)*ERRORX(N1)/HEFT(N1)        *2648
4240  CONTINUE                                                          *2649
96    CONTINUE                                                          *2650
CCC   MAKE FIRST CORRECTION TO TEMP CHANGES IN SPECIAL NODES.           *2651
4300  NUTS = 0                                                          *2652
       ESUM = 0.0                                                       *2653
      HSUM = 1.E-12                                                     *2654
       S1 = SPEED                                                       *2655
       S2 = S1 + 1.                                                     *2656
      IF((KNOCK + KNOFL)*KCYC) 6165, 6165,4305                          *2657
 6165 S1 = 0.0                                                          *2658
      S2 = 1.0                                                          *2659
4305  DO 4310  N = 1,NODES                                              *2660
      IF(NTYPE(N)) 6170,4310, 6170                                      *2661
 6170 EX = ERRCR(N) + S1*ZIP(N)*ERRORX(N)                               *2662
      ERRCR(N) = 0.                                                     *2663
      DT(N) = (CAP(N)*DT(N) + FORD*EX)/(CAP(N) + FORD*S2*ZIP(N))        *2664
      ERRCRX(N) = DT(N)-ERRORX(N)                                       *2665
      ESUM = ESUM + CAP(N)*ABSF(ERRORX(N))                              *2666
      HSUM = HSUM + CAP(N)                                              *2667
```

```
CCC   CAPACITY OF SPECIAL NODES.                                        *2668
4310  CONTINUE                                                          *2669
       HSUM = HSUM*TVARY                                                *2670
C MAX CHANGE IN HEAT CONTENT.                                           *2671
CCC   START ITERATING IF THERE ARE SPEC-SPEC CONNECTIONS, AND           *2672
CCC   RELATIVE CHANGE IN HEAT CONTENT IS GREATER THAN ERRS.             *2673
      IF(ESUM - ERRS*HSUM)4515, 6175, 6175                              *2674
 6175 IF(KNCCK + KNCFL)4515,4515, 6180                                  *2675
 6180 CONTINUE                                                          *2676
4400  NUTS = NUTS + 1                                                   *2677
C CALC CORRECTIONS FOR SPECIAL NODES.                                   *2678
      IF(KNCCK)4410,4410, 6185                                          *2679
 6185 DO 4405  N = 1,NOCCN                                              *2680
      N1 = NCC1(N)                                                      *2681
      N2 = NCC2(N)                                                      *2682
      IF(NTYPE(N1)*NTYPE(N2)) 6190,4405, 6190                           *2683
 6190 CONTINUE                                                          *2684
CCC   BOTH MUST BE SPECIAL.                                             *2685
      ERROR(N1) = ERROR(N1) + TRAN(N)*ERRORX(N2)                        *2686
      ERROR(N2) = ERROR(N2) + TRAN(N)*ERRORX(N1)                        *2687
4405  CONTINUE                                                          *2688
4410  CONTINUE                                                          *2689
      DIF(M10)98,98,97                                                  *2690
97    CONTINUE                                                          *2691
      IF(KNCFL)4500,4500, 6195                                          *2692
 6195 CONTINUE                                                          *2693
CCC   FIND CORRECTIONS DUE TO MASS FLOW.                                *2694
      DO 4415  N = 1,NOFLOW                                             *2695
      N1 = NCDF1(N)                                                     *2696
      N2 = NCDF2(N)                                                     *2697
      IF(NTYPE(N1)*NTYPE(N2)) 6200,4415, 6200                           *2698
 6200 ERROR(N2)=ERROR(N2) + FLOWN(N)*CAP(N1)*ERRORX(N1)/HEFT(N1)        *2699
4415  CONTINUE                                                          *2700
98    CONTINUE                                                          *2701
CCC CORRECT TEMP CHANGE IN SPECIAL NODES.                               *2702
4500  ESUM = 0.                                                         *2703
      EMAX = 0.0                                                        *2704
      DO 4505  N = 1,NODES                                              *2705
      IF(NTYPE(N)) 6205,4505, 6205                                      *2706
 6205 ERRORX(N) = (ERROR(N)+S1*ZIP(N)*ERRORX(N))/(S2*ZIP(N)+CAP(N)/FORD) *2707
      DT(N) = DT(N) + ERRORX(N)                                         *2708
       ERROR(N) = 0.                                                    *2709
      EX = ABSF(ERRORX(N))                                              *2710
      ESUM = ESUM + CAP(N)*EX                                           *2711
      IF(NTYPE(N) - 2) 6210,4505, 6210                                  *2712
 6210 EMAX = MAX1F(EMAX,EX)                                             *2713
4505  CONTINUE                                                          *2714
CCC   STOP ITERATING AFTER NUTMAX CYCLES.                               *2715
      IF(NUTS - NUTMAX)4510, 6215, 6215                                 *2716
 6215 NUTS = 0                                                          *2717
      DTMAX = MAX1F(DTMAX,2.0*TVARY)                                    *2718
      WRITE (   MW,5800 )KCYC                                           *2719
      PRINT 5800 ,KCYC                                                  *2720
      IF(DELT - 2.0*SMALL) 6220, 6220,4515                              *2721
 6220 CONTINUE                                                          *2722
C ITERATION FAILURE.                                                    *2723
      KWIT = 10                                                         *2724
      WRITE (   MW,5805 )    FIN                                        *2725
      PRINT 5805                                                        *2726
      GO TO 4700                                                        *2727
CCC   STOP ITERATING WHEN RELATIVE ERROR IN HEAT CONTENT CHANGES   ERRS. *2728
4510  IF(ESUM - ERRS*HSUM) 6225, 6225,4400                              *2729
 6225 IF(EMAX - 10.0*ERRS*TVARY) 6230, 6230,4400                        *2730
 6230 CONTINUE                                                          *2731
4515  NUTSUM =NUTSUM + NUTS                                             *2732
      NUTX = XMAXOF(NUTX,NUTS)                                          *2733
      DTMAX = MAX1F(DTMAX,0.025*NUTS*TVARY)                             *2734
CCC   CORRECT INTERNAL FLUXES AND FIND CORRECTION FACTORS FOR REG NODES. *2735
      IF(NOCCN)4615,4615, 6235                                          *2736
 6235 DO 4610  N = 1,NOCCN                                              *2737
      N1 = NCC1(N)                                                      *2738
      N2 = NCC2(N)                                                      *2739
```

SPECK TALLY

```
      IF(NTYPE(N1))4600, 6240,4600                                      *2740
 6240 IF(NTYPE(N2)) 6245,4610, 6245                                     *2741
 6245 ERROR(N1) = ERROR(N1) + TRAN(N)*(DT(N2) - DT(N1))                 *2742
      GO TO 4605                                                        *2743
4600  IF(NTYPE(N2))4605, 6250,4605                                      *2744
 6250 ERROR(N2) = ERROR(N2) + TRAN(N)*(DT(N1) - DT(N2))                 *2745
4605  FLEX = FORD*TRAN(N)*(DT(N2) - DT(N1))                             *2746
      DFI(N) = DFI(N) + FLEX                                            *2747
      DF(N1) = DF(N1) + FLEX                                            *2748
      DF(N2) = DF(N2) - FLEX                                            *2749
4610  CONTINUE                                                          *2750
4615  CONTINUE                                                          *2751
      DIF(M10)910,910,99                                                *2752
99    CONTINUE                                                          *2753
      IF(NOFLOW)4625,4625, 6255                                         *2754
 6255 CONTINUE                                                          *2755
CCC   FIND CORRECTIONS TO UPSTREAM REGULAR NODES CONNECTED TO SPECIALS. *2756
      DO 4620  N = 1,NOFLOW                                             *2757
      N1 = NODF1(N)                                                     *2758
      N2 = NODF2(N)                                                     *2759
      IF(NTYPE(N1))4620, 6260,4620                                      *2760
 6260 IF(NTYPE(N2)) 6265,4620, 6265                                     *2761
 6265 ERROR(N1) = ERROR(N1) - FLOWN(N)*CAP(N1)*DT(N1)/HEFT(N1)          *2762
4620  CONTINUE                                                          *2763
910   CONTINUE                                                          *2764
CCC   CALC CORRECTED TEMP CHANGE IN REG NODES CONN TO SPECIAL NODES.    *2765
4625  DO 4630  N = 1,NODES                                              *2766
      IF(NTYPE(N))4630, 6270,4630                                       *2767
 6270 DT(N) = DT(N) + FORD*ERROR(N)/CAP(N)                              *2768
4630  CONTINUE                                                          *2769
      DIF(M6)914,914,911                                                *2770
911   CONTINUE                                                          *2771
      DIF(M7)913,913,912                                                *2772
912   CONTINUE                                                          *2773
      IF(NOSCON)4700,4700, 6275                                         *2774
 6275 CONTINUE                                                          *2775
CCC   CORRECT FLUX OF EXTERNAL CONNECTIONS                              *2776
      DO 4640  N = 1,NOSCON                                             *2777
      N1 = NODS(N)                                                      *2778
      FLEX = FORD*TRANS(N)*DT(N1)                                       *2779
      DF(N1) = DF(N1) - FLEX                                            *2780
       DFS(N) = DFS(N) - FLEX                                           *2781
4640  CONTINUE                                                          *2782
913   CONTINUE                                                          *2783
914   CONTINUE                                                          *2784
4700  RETURN                                                            *2785
5800  FORMAT(/10X,6HKCYC =,I6,5X,84HHEAT BALANCE ITERATION FAILED TO CON *2786
     1VERGE.  THIS MAY BE CAUSED BY A CONNECTED SERIES,/,10X,97HOF NODES *2787
     2 WITH TIME CONSTANTS MUCH SMALLER THAN DELT.  WILL TRY TO REPEAT C *2788
     3YCLE WITH SMALLER DELT.)                                          *2789
5805  FORMAT(/,10X,39HCAN NOT REDUCE DELT.  WILL END PROBLEM.,A1)       *2790
5810  FORMAT(/,10X,6HKCYC =,I6,24H, ITERATIONS...  TOTAL =,I6,          *2791
     1   11H, AVERAGE =,I6,11H, MAXIMUM =,I6,8H.  FOR =,F6.3,2H .)      *2792
C COMPLETED SPECK.  RETURN TO HEART.                                    *2793
      END                                                               *2794
                                                                        *2795
*     CARDS COLUMN       TALLY                                          *2796
*     LIST 8                                                            *2797
*     FORTRAN            TALLY                                          *2798
      SUBROUTINE TALLY                                                  *2799
C********VERSION 5/20/69.                                               *2800
C********TALLY IS A REQUIRED SUBROUTINE.                                *2801
      USE STORK                                                         *2802
      USE STORE                                                         *2803
      USE STORM                                                         *2804
C********REQUIRED WHEN CHEM IS USED, USED IN CHEM, THERM, FLOW, TALLY.  *2805
      DIF(M3)92,92,91                                                   *2806
91    CONTINUE                                                          *2807
      COMMON /ACHEMS/ KA(M2), KAX(M2), KB(M2), KBX(M2),                 *2808
     1         B(M4), BB(M4), DB(M4), CDB(M4)                           *2809
92    CONTINUE                                                          *2810
      DIF(M8)982,982,981                                                *2811
```

TALLY

```
981   CONTINUE                                                          *2812
C********ARRAYS USED ONLY IF SUBROUTINE GEN IS USED, G IN THERM,         *2813
C********     GEN, AND TALLY, GG IN TALLY ONLY.                          *2814
      COMMON /AGENS/  G(M4), GG(M4)                                     *2815
982   CONTINUE                                                          *2816
CCC                                                                     *2817
      IF(KCYC) 6100,3100,4200                                           *2818
 6100 IF(IBLOCK)2400, 6105,2200                                         *2819
 6105 CONTINUE                                                          *2820
CCC   START HERE AFTER PROBLEM NAME CARD HAS BEEN READ IN WITH 2, 3, OR *2821
CCC      4 IN COLUMN 72, FOR SAVING DATA FROM PRECEDING PROBLEM.        *2822
      WRITE (     M,5950 )(I, I = 2,12)                                 *2823
      WRITE (     M,5975 )     FIN                                      *2824
      WRITE (     M,5960 )(ITEMS(I), I = 2,12)                          *2825
      NIT = NODES                                                       *2826
CCC   FIND OUT IF NODE TYPES AND T, A, AND B WILL BE INITIAL OR FINAL   *2827
CCC      VALUES FROM PRECEDING PROBLEM (NAME(16) = 2, OR 3 OR 4, RESP.). *2828
      IF(NOW - 2)2105, 6110,2105                                        *2829
 6110 DO 2100  N = 1,NODES                                              *2830
      IF(NTYPE(N)-NTYPES(N)) 6115,2100, 6115                            *2831
 6115 NTYPE(N)=NTYPES(N)                                                *2832
      NOSPEC=NOSPEC-1                                                   *2833
2100  CONTINUE                                                          *2834
      GO TO 2115                                                        *2835
CCC   CHANGE INTERNAL BLOCK 9 TO FINAL T, A, B OF PREVIOUS PROBLEM.     *2836
2105  DO 2110  N = 1,NODES                                              *2837
      TT(N) = T(N)                                                      *2838
      AA(N) = A(N)                                                      *2839
      DIF(M3)94,94,93                                                   *2840
93    CONTINUE                                                          *2841
      BB(N) = B(N)                                                      *2842
94    CONTINUE                                                          *2843
2110  CONTINUE                                                          *2844
2115  RETURN                                                            *2845
2200  IF(IBLOCK - 9) 6120,2300, 6120                                    *2846
 6120 IF(IBLOCK - 1)4700, 6125,4700                                     *2847
 6125 CONTINUE                                                          *2848
CCC   START HERE AFTER THE BLOCK NUMBER CARD FOR BLOCK 1 IS READ IN.    *2849
CCC INPUT DATA BLOCK  1.  CALCULATION CONTROLS, LIMITS, AND CONSTANTS.  *2850
      READ  (    NB,5865 )IPRINT,NUM,KDATA,KSPEC,MCYC,MSEC,            *2851
     $  NPUNCH,NDCT,IRITE,ITAPE,TIMEP,SCALE                             *2852
      NUMX = NUM                                                        *2853
      KDATAX = KDATA                                                    *2854
      IF(SCALE) 6130, 6130,2205                                         *2855
 6130 SCALE = 1.0                                                       *2856
2205  IF(IRITE)2215,2215, 6135                                          *2857
 6135 IF(ITAPE) 6140, 6140,2210                                         *2858
 6140 ITAPE = 6                                                         *2859
2210  WRITE (     M,51000 ) ITAPE,IRITE                                 *2860
      PRINT         51000 , ITAPE,IRITE                                 *2861
      WRITE(ITAPE, 51020) NAME                                          *2861
2215  WRITE (     M,5890 )IPRINT,NUM,KDATA,KSPEC,MCYC,MSEC,             *2862
     $  NPUNCH,NDCT,IRITE,ITAPE,TIMEP,SCALE                             *2863
      READ  (    NB,5870 )KC,KT,DELTO,SMALL,TVARY,TAL,TIMAX,TMIN,TMAX   *2864
      KD = XMAXOF(1,KC)                                                 *2865
      KT = XMAXOF(1,KT)                                                 *2866
      KSYM = (KC + 3)/3                                                 *2867
      GEOM = 2.0**(KC - 1)*3.1415926**(KC/2)                            *2868
      IF((DELTO - 1.0E-10)*(DELTO - 1.0E12))2220,2220, 6145             *2869
 6145  DELTO = 1.0E12                                                   *2870
2220  SMALL = MAX1F(SMALL,1.0E-12)                                      *2871
       SMALT = SMALL                                                    *2872
      IF(TMAX - TMIN) 6150, 6150,2225                                   *2873
 6150 TMAX =  1.0E12                                                    *2874
      TMIN = -1.0E12                                                    *2875
2225  IF(TVARY) 6155, 6155,2230                                         *2876
 6155 TVARY = MIN1F(5.0,0.005*(TMAX - TMIN))                            *2877
2230  TMIN = TMIN - 0.001*TVARY                                         *2878
       TMAX = TMAX + 0.001*TVARY                                        *2879
      WRITE (     M,5875 )KC,KT,DELTO,SMALL,TVARY,TAL,TIMAX,TMIN,TMAX   *2880
      GO TO(2235,2240,2245,2250,2255),KT                                *2881
2235  TBASE = 273.15                                                    *2882
```

```
      SIGMA = 1.355E-12                                                 *2883
      GO TO 2260                                                        *2884
2240  TBASE = 0.0                                                       *2885
      SIGMA = 1.355E-12                                                 *2886
      GO TO 2260                                                        *2887
2245  TBASE = 460.0                                                     *2888
      SIGMA = 0.173E-8                                                  *2889
      GO TO 2260                                                        *2890
2250  TBASE = 0.0                                                       *2891
      SIGMA = 0.173E-8                                                  *2892
      GO TO 2260                                                        *2893
2255  TBASE = 0.0                                                       *2894
      SIGMA = 1.0                                                       *2895
2260  WRITE (     M,5880 )KC,KSYM,GEOM,SIGMA,TBASE                      *2896
      READ  (    NB,5810 )TONE,ALONE,BONE,GONE,FONE,HONE,RONE,PONE      *2897
      WRITE (     M,5815 )TONE,ALONE,BONE,GONE,FONE,HONE,RONE,PONE      *2898
      RETURN                                                            *2899
C COMPLETED BLOCK 1.  RETURN TO HEART.                                  *2900
CCC   START HERE AFTER THE BLOCK NUMBER CARD FOR BLOCK 9 IS READ IN.    *2901
CCC INPUT DATA BLOCK  9.  INITIAL CONDITIONS LIST.                      *2902
2300  N = NIT                                                           *2903
      L = 0                                                             *2904
      IF(MCC)2305, 6160,2305                                            *2905
 6160 N = NIT = 0                                                       *2906
2305  READ  (    NB,5840 )N1,NSEQ,NADD,NX1,NX2,NX3,NX4                  *2907
      IF(N1) 6165,2335, 6165                                            *2908
 6165 CALL PATCH(NX1,TONE,TX,LXX)                                       *2909
      CALL PATCH(NX2,ALONE,AX,LXX)                                      *2910
      CALL PATCH(NX3,BONE,BX,LXX)                                       *2911
      CALL PATCH(NX4,GONE,GX,LXX)                                       *2912
2310  IF(MCC) 6170,2315,2315                                            *2913
 6170 CALL SEEKI(N,N1,NOXE,NIT,K)                                       *2914
      ITEMS( 9) = XMINOF(N - 1,ITEMS( 9))                               *2915
      GO TO 2320                                                        *2916
2315  N = NIT = N + 1                                                   *2917
2320  IF(N - M4)2325,2325, 6175                                         *2918
 6175 N = NIT = M4                                                      *2919
      KWIT = 11                                                         *2920
      WRITE (    MW,5925 )N,IBLOCK                                      *2921
      PRINT 5925 ,N,IBLOCK                                              *2922
2325  NOTE(N) = NOXE(N) = N1                                            *2923
      TT(N) = TX                                                        *2924
      AA(N) = AX                                                        *2925
      DIF(M3)96,96,95                                                   *2926
 95   CONTINUE                                                          *2927
      BB(N) = BX                                                        *2928
 96   CONTINUE                                                          *2929
      DIF(M8)98,98,97                                                   *2930
 97   CONTINUE                                                          *2931
      GG(N) = GX                                                        *2932
 98   CONTINUE                                                          *2933
      L = L + 1                                                         *2934
      IF(XMODF(L-1,57))2330, 6180,2330                                  *2935
 6180 WRITE (     M,5845 )     FIN                                      *2936
2330  WRITE (     M,5850 )NOTE(N),N,TT(N),AA(N),BX,GX                   *2937
      IF(NSEQ)2305,2305, 6185                                           *2938
 6185 NSEQ = NSEQ - 1                                                   *2939
      N1 = N1 + NADD                                                    *2940
      GO TO 2310                                                        *2941
2335  RETURN                                                            *2942
C COMPLETED BLOCK 9.                                                    *2943
CCC   START HERE AFTER THE DATA END CARD IS READ IN.                    *2944
CCC   SUMMARIZE INPUT DATA, AND INITIALIZE PRIOR TO TIME STEP ZERO.     *2945
2400  WRITE (     M,5950 )(I, I = 2,12)                                 *2946
      WRITE (     M,5985 )(NEWBL(I), I = 2,12)                          *2947
      WRITE (     M,5975 )     FIN                                      *2948
      WRITE (     M,5955 )M2,M3,M4,M5,M6,M7,M8,M4,M10,M11,M12           *2949
      WRITE (     M,5965 )(ITEMS(I), I = 2,12)                          *2950
      WRITE (     M,5970 )(NOSPEC(I), I = 2,12)                         *2951
      NTABLE =NVARC+NVARK + NVARQ+NVARZ+NVARE + NVARH+NVART+NVARG+NVARFL *2952
      WRITE (     M,5945 )NVARC,NVARK,NVARQ,NVARZ,NVARE,NVARH,          *2953
     1   NVART,NVARG,NVARFL,NTABLE                                      *2954
```

TALLY

```
      WRITE (     M,5930 )M9                                          *2955
      IF(NMAT)2405,2405, 6190                                         *2956
 6190 IF(NODES) 2405, 2405, 6192                                      *2957
 6192 IF(NOSCON)2410, 2410, 6194
 6194 IF(NODES)2405, 2405, 2410
2405  KWIT = 9                                                        *2959
      WRITE (    MW,5805 )NMAT,NODES                                  *2960
      PRINT                 5805 ,NMAT,NODES                         *2961
      RETURN                                                          *2962
C MISSING BLOCK 2 OR 4.  RETURN TO HEART.                             *2963
2410  IF(NUM) 6200,2415, 6200                                         *2964
 6200 IBLOCK = 1                                                      *2965
      CALL REFER(NUM,NUMX,1,NODE,NODES)                               *2966
2415  IF(NIT)2425,2425, 6205                                          *2967
 6205 IBLOCK = 9                                                      *2968
      CALL REFER(NOTE,NOXE,NIT,NODE,NODES)                            *2969
      IF(KWIT)2500, 6210,2500                                         *2970
 6210 CONTINUE                                                        *2971
C ASSIGN INITIAL CONDITIONS TO NODES.                                 *2972
      DO 2420  N = 1,NIT                                              *2973
      J = NOTE(N)                                                     *2974
      T(J) = TT(N)                                                    *2975
      A(J) = AA(N)                                                    *2976
      DIF(M3)910,910,99                                               *2977
99    CONTINUE                                                        *2978
      B(J) = BB(N)                                                    *2979
910   CONTINUE                                                        *2980
      DIF(M8)912,912,911                                              *2981
911   CONTINUE                                                        *2982
      G(J) = GG(N)                                                    *2983
912   CONTINUE                                                        *2984
2420  CONTINUE                                                        *2985
CCC   SET UP INTERNAL BLOCK 9 WITH INITIAL CONDITIONS OF THIS PROBLEM.  *2986
2425  DO 2430  N = 1,NODES                                            *2987
      NOTE(N) = N                                                     *2988
      NOXE(N) = NODE(N)                                               *2989
      TT(N) = T(N)                                                    *2990
      AA(N) = A(N)                                                    *2991
      NTYPES(N) = NTYPE(N)                                            *2992
      DIF(M3)914,914,913                                              *2993
913   CONTINUE                                                        *2994
      BB(N) = B(N)                                                    *2995
914   CONTINUE                                                        *2996
      DIF(M8)916,916,915                                              *2997
915   CONTINUE                                                        *2998
      GG(N) = G(N)                                                    *2999
916   CONTINUE                                                        *3000
2430  CONTINUE                                                        *3001
C MAKE ALL NODES SPECIAL IF KSPEC IS POSITIVE.                        *3002
      IF(KSPEC)2500,2500, 6215                                        *3003
 6215 DO 2435  N = 1,NODES                                            *3004
      IF(NTYPE(N))2435, 6220,2435                                     *3005
 6220 NTYPE(N) = 5                                                    *3006
      NOSPEC = NOSPEC + 1                                             *3007
2435  CONTINUE                                                        *3008
CCC   INITIALIZE BEFORE FIRST TIME INCREMENT                          *3009
2500  NOGEN = 0                                                       *3010
      NREACT = 0                                                      *3011
      DO 2515  N = 1,NODES                                            *3012
      F(N) = H(N) = 0.0                                               *3013
      IF(A(N))2505, 6225,2505                                         *3014
 6225 DIF(M3)918,918,917                                              *3015
917   CONTINUE                                                        *3016
      IF(B(N))2505, 6230,2505                                         *3017
 6230 CONTINUE                                                        *3018
918   CONTINUE                                                        *3019
      GO TO 2510                                                      *3020
2505  NREACT = NREACT + 1                                             *3021
2510  CONTINUE                                                        *3022
      DIF(M8)920,920,919                                              *3023
919   CONTINUE                                                        *3024
      IF(G(N)) 6235,2515, 6235                                        *3025
 6235 NOGEN = NOGEN + 1                                               *3026
```

```
920   CONTINUE                                                          *3027
2515  CONTINUE                                                          *3028
CCC   CALCULATE STORAGE REQUIREMENTS FOR PARAMETER-ADJUSTED ARRAYS.     *3029
      NWREQD = (8 + 7*M9)*M2 + 28*M4 + 12*M5 + 3*M9                     *3030
      NWCHEM = 0                                                        *3031
      NWGEN8 = 0                                                        *3032
      NWFLOW = 0                                                        *3033
      NWSURE = 0                                                        *3034
      NWPLOT = 0                                                        *3035
      NWBL12 = 0                                                        *3036
      IF(M3)2520,2520, 6240                                             *3037
 6240 NWCHEM = 4*M2 + (6 + 9*M9)*M3 + 10*M4                             *3038
2520  IF(M8)2525,2525, 6245                                             *3039
 6245 NWGEN8 = 2*M4 + (3 + 3*M9)*M8                                     *3040
2525  IF(M10)2530,2530, 6250                                            *3041
 6250 NWFLOW = 4*M4 + (9 + 3*M9)*M10                                    *3042
2530  IF(M6)2535,2535, 6255                                             *3043
 6255 IF(M7)2535,2535, 6260                                             *3044
 6260 NWSURE = (12 + 3*M9)*M6 + (5 + 3*M9)*M7                           *3045
2535  IF(M1)2540,2540, 6265                                             *3046
 6265 IF(M11)2540,2540, 6270                                            *3047
 6270 NWPLOT = 4*M4 + 3*M11 + (1 + M11)*M1                              *3048
2540  IF(M12)2545,2545, 6275                                            *3049
 6275 NWBL12 = 8*M4 + 5*M12                                             *3050
2545  NSTORE = NWREQD + NWCHEM + NWGEN8 + NWFLOW + NWSURE +NWPLOT+NWBL12 *3051
      WRITE (     M,5935 ) NWREQD,NWCHEM,NWGEN8,NWFLOW,NWSURE,NWPLOT,NWBL *3052
     X12                                                                *3053
      WRITE (     M,5940 ) NSTORE                                       *3054
      WRITE (     M,5980 )NOSPEC,NOGEN,NORAD,NORADS,NMELT,NREACT        *3055
      NREACT = NREACT + NMELT + NKEM                                    *3056
      NOGEN = NOGEN + NVARG                                             *3057
CCC   INITIALIZE BEFORE FIRST TIME STEP (KCYC = 0,  DELT = 1.0E-12).    *3058
      SUMTIM = TAU                                                      *3059
      DELT = 1.0E-12                                                    *3060
      DELTS = DTMAX = 0.0                                               *3061
      FOR = 1.0                                                         *3062
      GO TO 4820                                                        *3063
CCC   START HERE AT THE END OF TIME STEP ZERO (DELT = 1.0E-12).         *3064
3100  IF(KWIT) 6280, 6280,4900                                          *3065
 6280 KGOOD = MF = MS = MSS = NPRINT = 0                                *3066
      DELTG = DTMAX = GENS = 0.0                                        *3067
      RATG = 1.0                                                        *3068
      KDATA = KDATAX                                                    *3069
      DELTMX = MAX1F(1.0E-10,DELTO)                                     *3070
       BIG = 0.99999999E12                                              *3071
      GO TO 4210                                                        *3072
CCC   START HERE WHEN KCYC = 1 OR MORE, TALLY RESULTS OF CALCULATIONS.  *3073
4200  IF(KCYC - 1) 6285, 6285,4205                                      *3074
 6285 DTMAX = 0.                                                        *3075
4205  IF(KWIT) 6290, 6290,4710                                          *3076
 6290  KWIT = 0                                                         *3077
       NOW = 0                                                          *3078
CCC   FIND MAXIMUM STABLE TIME STEP.                                    *3079
      IF(EIG - DELTMX)4235,4235, 6295                                   *3080
 6295 IF(KCYC - 1)4210,4210, 6300                                       *3081
 6300 IF(NTABLE) 6305, 6305,4210                                        *3082
 6305 IF(NSPEC)4235,4210,4210                                           *3083
4210  DELTMX = 1.5E12                                                   *3084
      NREG = 0                                                          *3085
      DO 4220  N = 1,NODES                                              *3086
      SLIM(N) = 1.E24                                                   *3087
      IF(ZIP(N))4215,4215, 6310                                         *3088
 6310  SLIM(N) = MAX1F(1.0E-24,CAP(N)/ZIP(N))                           *3089
4215  IF(NTYPE(N))4220, 6315,4220                                       *3090
 6315 NREG = NREG + 1                                                   *3091
      DELTMX=MIN1F(DELTMX,SLIM(N))                                      *3092
4220  CONTINUE                                                          *3093
CCC   SET MAXIMUM TIME STEP TO 2/3 OF STABILITY LIMIT, OR DELTO,        *3094
CCC      WHICHEVER IS SMALLER, BUT NO SMALLER THAN 1.0E-10.             *3095
      DELTMX = MAX1F(1.0E-10,DELTMX/1.5)                                *3096
      IF(DELTMX - DELTO)4225,4235, 6320                                 *3097
 6320 DELTMX = DELTO                                                    *3098
```

TALLY

```
      GC TC 4235                                                        *3099
CCC   CHANGE NCDES TC SPECIAL NODES IF NECESSARY TO INCREASE DELTMX.    *3100
4225  IF(MS)4235,4235, 6325                                             *3101
 6325 IF(NREG)4235,4235, 6330                                           *3102
 6330 IF(KSPEC)4235, 6335, 6335                                         *3103
 6335 NEWS = 0                                                          *3104
      DO 4230  N = 1,NODES                                              *3105
      IF(NTYPE(N))4230, 6340,4230                                       *3106
 6340 IF(1.8*CELTMX-SLIM(N))4230, 6345, 6345                            *3107
 6345 NTYPE(N)=4                                                        *3108
      NCSPEC=NCSPEC+1                                                   *3109
      NEWS=1                                                            *3110
      WRITE (    MW,5800 )KCYC,NODE(N)                                  *3111
4230  CONTINUE                                                          *3112
CCC   RECALCULATE CELTMX IF ANY NODES RECLASSIFIED.                     *3113
      IF(NEWS)4235,4235, 6350                                           *3114
 6350 KGOCD = 0                                                         *3115
      MS = 0                                                            *3116
      GC TC 4210                                                        *3117
CCC   RESTCRE SMALL TO SMALT, IF SMALL HAS DECREASED BELOW SMALT.       *3118
4235  IF(SMALT - SMALL)4240,4240, 6355                                  *3119
 6355 SMALL = SMALT                                                     *3120
CCC   CALC SMALL IF NOT READ IN, DELTMX NOT 1.E12, NREG NCT   1/4 NCDES. *3121
4240  IF(SMALL - DELTMX)4245, 6360, 6360                                *3122
 6360 SMALL = 0.99999999*DELTMX                                         *3123
4245  IF(1.0 - BIG*SMALL)4300, 6365, 6365                               *3124
 6365  IF(4*NREG - NCDES)4300, 6370, 6370                               *3125
 6370 IF(BIG - DELTMX)4300, 6375, 6375                                  *3126
 6375  SMALL = 0.01*DELTMX                                              *3127
CCC   FINC LARGEST TEMPERATURE CHANGE, EXCLUDING Z-V NODES IF KCYC 0 CR *3128
4300  DTEMP = 1.0E-24                                                   *3129
       DO 4310, N = 1,NCDES                                             *3130
      IF(KCYC - 1) 6380, 6380,4305                                      *3131
 6380 IF(NTYPE(N) - 2) 6385,4310, 6385                                  *3132
 6385 CONTINUE                                                          *3133
4305  DTEMP = MAX1F(DTEMP,ABSF(DT(N)))                                  *3134
4310  CONTINUE                                                          *3135
      DELTSS = DELTG                                                    *3136
      DELTS = DELT                                                      *3137
      RAT2 = RATG                                                       *3138
      RAT1 = TVARY/DTEMP                                                *3139
      RATIC = TVARY/MAX1F(DTMAX,DTEMP)                                  *3140
CCC   REPEAT CYCLE IF MAX TEMP CHANGE MORE THAN DOUBLE TVARY.           *3141
      IF(RATIC - 0.501) 6390, 6390,4315                                 *3142
 6390  IF(DELT - 1.01*SMALL)4315, 6395, 6395                            *3143
 6395 WRITE (    MW,5990 )KCYC,DTEMP,DTMAX,DELT,SUMTIM                  *3144
      KWIT = -1                                                         *3145
       KGOCD = 0                                                        *3146
      FCR = 1.0                                                         *3147
       GO TC 4320                                                       *3148
CCC   INCREMENT PRCBLEM TIME, GOOD CYCLE COUNTER ON ACCEPTED TIME STEPS. *3149
4315  KGOCD = KGOCD + 1                                                 *3150
      SUMTIM = SUMTIM + DELT                                            *3151
      DELTG = DELTS                                                     *3152
      RATG = RAT1                                                       *3153
      DTMAXS = DTMAX                                                    *3154
4320  DTMAX = 0.0                                                       *3155
CCC   USE 1 PERCENT CF CALCULATED TIME STEP, IF ALL NODES ARE SPECIAL.  *3156
      IF(KCYC) 6400, 6400,4325                                          *3157
 6400 IF(NREG) 6405, 6405,4335                                          *3158
 6405 RATIC = 0.01*RATIC                                                *3159
      GC TC 4335                                                        *3160
CCC   CHANGE DELT TC MAKE NEXT DTMAX CLOSER TO TVARY.                   *3161
4325  IF(RATIC-1.0) 6410,4335,4330                                      *3162
 6410 RATIC = MAX1F(0.5,RATIC*RATIO)                                    *3163
      GO TC 4335                                                        *3164
4330  RATIC = MIN1F(2.0,0.5*(1.0 + RATIO))                              *3165
4335  DELT = DELTS*RATIC                                                *3166
CCC   KEEP TIME STEP IN RANGE FROM SMALL TO DELTMX OR DELTC.            *3167
      IF(DELT - SMALL) 6415,4340,4340                                   *3168
 6415 DELT = SMALL                                                      *3169
      MF = MF + 1                                                       *3170
```

```
      GO TO 4400                                                        *3171
4340  IF(DELT - DELTMX)4400,4400, 6420                                  *3172
 6420 DELT = DELTMX                                                     *3173
      MSS = MSS + 1                                                     *3174
      MS = MS + 1                                                       *3175
CCC   RETURN AND REPEAT REJECTED TIME STEPS.                            *3176
4400   IF(KWIT)4900, 6425, 6425                                         *3177
 6425 CONTINUE                                                          *3178
CCC   ADJUST DELT IN RANGE SMALL TO DELTMX TO GET DESIRED PRINTOUT TIMES *3179
      IF(TIMEP)4425,4425, 6430                                          *3180
 6430 TIM = MODF(TIMEP - MODF(SUMTIM, TIMEP), TIMEP)                    *3181
      TIMS = TIM                                                        *3182
CCC   ADJUST DELT BY FACTOR BETWEEN 2/3 AND 3/2 TO GET PRINTOUT TIME.   *3183
      TIM = TIM + INTF(1.0 + (DELT/1.5 - TIM)/TIMEP)*TIMEP              *3184
      PIX = INTF(TIM/DELT + 0.5)                                        *3185
      DELT = TIM/PIX                                                    *3186
CCC   IF DELT OUTSIDE BOUNDS, ADJUST TO NEXT INTEGER FIT TO TIM.        *3187
      IF(DELT - SMALL) 6435,4415,4415                                   *3188
 6435 DELT = TIM/MAX1F(PIX - 1.0,1.0)                                   *3189
4415  IF(DELT - DELTMX)4420,4420, 6440                                  *3190
 6440 DELT = TIM/(PIX + 1.0)                                            *3191
4420  IF(DELT - SMALL) 6445,4425,4425                                   *3192
 6445 DELT = SMALL                                                      *3193
CCC   FOR LAST TIME STEP, ADJUST DELT SO FINAL TIME IS EQUAL TO TIMAX.  *3194
4425  IF(TIMAX)4430,4430, 6450                                          *3195
 6450 IF(SUMTIM + DELT - TIMAX)4430,4430, 6455                          *3196
 6455 DELT = TIMAX - SUMTIM + 1.0E-12                                   *3197
CCC   FIND RATIO OF ESTIMATED TIME DERIVATIVE TO PRESENT VALUE.         *3198
4430  RAST = 1.0                                                        *3199
      IF(KGOOD - 2)4440, 6460, 6460                                     *3200
 6460 CONTINUE                                                          *3201
CCC   USE QUADRATIC EXTRAPOLATION FOR INCREASING MAXIMUM SLOPE,         *3202
CCC      EXPONENTIAL FOR DECREASING MAXIMUM SLOPE.                      *3203
      SLOPS = RAT1*DELTS/(RAT2*DELTSS)                                  *3204
      RDELT = (DELT + DELTS)/(DELTS + DELTSS)                           *3205
      IF(SLOPS - 1.0) 6465,4440,4435                                    *3206
 6465 RAST = 1.0 + RDELT*(1.0 - SLOPS)                                  *3207
      GO TO 4440                                                        *3208
4435  RAST = SLOPS**(-RDELT) + 1.0E-24                                  *3209
CCC   FIND INTERPOLATION FACTOR FOR NEXT TIME STEP.                     *3210
4440  FOR = MAX1F(0.57,MAX1F(1.0,RAST)/(1.0 + RAST))                    *3211
      IF(KSPEC - 2)4500, 6470,4445                                      *3212
 6470 FOR = 1.0                                                         *3213
      GO TO 4500                                                        *3214
4445  FOR = 0.5                                                         *3215
CCC   ON GOOD CYCLES, FIND NEW TEMPERATURES, HEAT CONTENTS, FLUXES.     *3216
CCC   ALSO FIND MAXIMUM AND MINIMUM TEMPERATURES IN SYSTEM.             *3217
4500  GS = 0.0                                                          *3218
       TMAX1 = -1.E8                                                    *3219
       TMIN1 = 1.E8                                                     *3220
      DO 4515  N = 1,NODES                                              *3221
      T(N) = T(N) + DT(N)                                               *3222
      GS = GS + G(N)                                                    *3223
      TMAX1 = MAX1F(TMAX1,T(N))                                         *3224
       TMIN1 = MIN1F(TMIN1,T(N))                                        *3225
      H(N) = H(N) + DT(N)*CAP(N)                                        *3226
       W(N) = W(N) + DT(N)*CAP(N)                                       *3227
      F(N) = F(N) + DF(N)                                               *3228
       DEX = RAST*DT(N)/DELTS                                           *3229
      IF(NTYPE(N)) 6475,4505, 6475                                      *3230
 6475 IF(KCYC - 1) 6480, 6480,4505                                      *3231
 6480 IF(NTYPE(N) - 2) 6485,4515, 6485                                  *3232
 6485 DEX = DEX/(1.0 + DELTS/SLIM(N))                                   *3233
4505  IF(DEX*DDT(N)) 6490,4510,4510                                     *3234
 6490 DEX = DEX*1.0E-24                                                 *3235
4510  DDT(N) = DEX                                                      *3236
4515  CONTINUE                                                          *3237
      GENS = GENS + GS*DELTS                                            *3238
CCC   FIND NEW REACTANT OR PHASE CONCENTRATIONS.                        *3239
      IF(NREACT)4600,4600, 6495                                         *3240
 6495 DO 4530  N = 1,NODES                                              *3241
      A(N) = A(N) + DA(N)                                               *3242
```

TALLY

```
      DEXA = RAST*CA(N)/CELTS                                           *3243
      HMX=HEFT(N)*HMELT(NCCMAT(N))*CA(N)                                *3244
      W(N)=W(N)-HMX                                                     *3245
      H(N)=H(N)-HMX                                                     *3246
      IF(NTYPE(N)) 6500,4520, 6500                                      *3247
 6500 IF(KCYC - 1)4530,4530, 6505                                       *3248
 6505 CONTINUE                                                          *3249
4520  IF(CEXA*CCA(N)) 6510,4525,4525                                    *3250
 6510 DEXA = CEXA*1.0E-24                                               *3251
4525  DDA(N) = CEXA                                                     *3252
4530  CONTINUE                                                          *3253
      DIF(M3)922,922,921                                                *3254
921   CONTINUE                                                          *3255
      IF(NKEM)4600,4600, 6515                                           *3256
 6515 DO 4545  N = 1,NODES                                              *3257
      B(N) = B(N) + CB(N)                                               *3258
      DEXB = RAST*CB(N)/CELTS                                           *3259
      IF(NTYPE(N)) 6520,4535, 6520                                      *3260
 6520 IF(KCYC - 1)4545,4545, 6525                                       *3261
 6525 CONTINUE                                                          *3262
4535  IF(CEXB*CCB(N)) 6530,4540,4540                                    *3263
 6530 DEXB = CEXB*1.0E-24                                               *3264
4540  DDB(N) = CEXB                                                     *3265
4545  CONTINUE                                                          *3266
922   CONTINUE                                                          *3267
CCC   TEST TEMPERATURE LIMIT CRITERIA FOR ENDING PROBLEM.               *3268
4600  IF(TMIN1 - TMIN) 6535,4605,4605                                   *3269
 6535  KWIT = 3                                                         *3270
       GO TO 4710                                                       *3271
4605  IF(TMAX1 - TMAX)4606,4606, 6540                                   *3272
 6540  KWIT = 2                                                         *3273
       GO TO 4710                                                       *3274
CCC   TEST UPPER PROBLEM TIME LIMIT CRITERION FOR ENDING PROBLEM.       *3275
4606  IF(TIMAX)4608,4610, 6545                                          *3276
 6545 IF(SUMTIM - TIMAX)4610, 6550, 6550                                *3277
 6550 CONTINUE                                                          *3278
4608  KWIT = 1                                                          *3279
      GO TO 4710                                                        *3280
CCC   TEST STEADY STATE CRITERIA FOR ENDING PROBLEM.                    *3281
4610  IF(KSPEC)4620, 6555, 6555                                         *3282
 6555 IF(DELTMX - BIG) 6560,4615,4615                                   *3283
 6560 IF(DELTMX - DELTO) 6565,4615,4615                                 *3284
 6565 IF(DELTMX - DELT) 6570, 6570,4620                                 *3285
 6570 CONTINUE                                                          *3286
4615  IF(RAT1 - 1.E3)4620, 6575, 6575                                   *3287
 6575 IF(RAT2 - 1.E3)4620, 6580, 6580                                   *3288
 6580 IF(KCYC - 10)4620, 6585, 6585                                     *3289
 6585 IF(KGECC - 3)4620, 6590, 6590                                     *3290
 6590 KWIT = 4                                                          *3291
       GO TO 4710                                                       *3292
CCC   TEST LIMIT ON NUMBER OF TIME STEPS CRITERION FOR ENDING PROBLEM.  *3293
4620  IF(MCYC)4635,4640, 6595                                           *3294
 6595 IF(KCYC - MCYC)4640, 6600, 6600                                   *3295
 6600 CONTINUE                                                          *3296
4635  KWIT = 7                                                          *3297
      GO TO 4710                                                        *3298
4640  CONTINUE                                                          *3299
CCC   TEST MACHINE TIME LIMIT CRITERION FOR ENDING PROBLEM.             *3300
      IF(MSEC)4645,4700, 6605                                           *3301
 6605 IF(XMCDF(KCYC,2000/NODES))4700, 6610,4700                         *3302
 6610 CALL TIMER(NPER,MICS)                                             *3303
      KSECS = KSECS + MICS                                              *3304
      IF(KSECS - MSEC)4700, 6615, 6615                                  *3305
 6615 CONTINUE                                                          *3306
4645  KWIT = 8                                                          *3307
4700  IF(KWIT)4810, 6620,4710                                           *3308
 6620 IF(KCYC - 1)4710,4710, 6625                                       *3309
 6625 IF(TIMEP)4705,4705, 6630                                          *3310
 6630 CONTINUE                                                          *3311
CCC   FINC OUT IF A PRINTOUT IS REQUIRED BY TIMEP, SUMTIM VALUES.       *3312
      IF(TIMS - DELT/1.5)4702,4702, 6635                                *3313
 6635 IF(TIMEP - TIMS - DELT/1.5) 6640, 6640,4705                       *3314
```

```
 6640 CONTINUE                                                          *3315
4702  NOW = 2                                                           *3316
      GO TO 4712                                                        *3317
CCC   FIND OUT IF A PRINTOUT IS REQUIRED BY IPRINT, KCYC VALUES.        *3318
4705  IF(IPRINT)4755,4755, 6645                                         *3319
 6645 IF(XMODF(KCYC,IPRINT)) 6650, 6650,4755                            *3320
 6650 CONTINUE                                                          *3321
CCC   CALCULATE OVERALL HEAT BALANCE QUANTITIES, WRITE OUT SUMMARY.     *3322
4710  NOW = 1                                                           *3323
4712  NPRINT = NPRINT + 1                                               *3324
      HEAT = 0.                                                         *3325
      FLUX = 0.                                                         *3326
      TX = 0.                                                           *3327
      DO 4715  N = 1,NODES                                              *3328
      FLUX = FLUX + F(N)                                                *3329
      HEAT = HEAT + W(N)                                                *3330
      TX = TX + CAP(N)*T(N)                                             *3331
4715  CONTINUE                                                          *3332
      TEMPAD = TX/CAPS                                                  *3333
      TEMPER = FLUX/CAPS                                                *3334
      TEMPLE = GENS/CAPS                                                *3335
      FX = FLUX/MAX1F(SUMTIM - TAU,1.0E-12)                             *3336
      TX = FX/CAPS                                                      *3337
      WRITE (     M,5820 )NPROB,NAME,CLOCKB,CLOCKA                      *3338
      WRITE (     M,5920 )     FIN                                      *3339
      WRITE (     M,5825 ) NPRINT,KCYC,MF,MSS,KWIT,DELTMX,SMALL,        *3340
     1   TVARY,CTEMP,DTMAXS,NUTS                                        *3341
      WRITE (     M,5830 ) SUMTIM, DELTS, FLUX, TEMPER,                 *3342
     1   FX, TX, TEMPAD, CAPS, HEAT, GS, GENS, TEMPLE                   *3343
      WRITE (     M,5920 )     FIN                                      *3344
      IF(KWIT)4720,4720, 6655                                           *3345
 6655 KDATA = 1                                                         *3346
      PRINT 5885 ,HEAT,FLUX                                             *3347
CCC   WRITE OUT THERMAL DATA FOR NODES.                                 *3348
4720  IF(KCYC - 1)4725,4725, 6660                                       *3349
 6660 IF(KDATA) 6665,4725,4725                                          *3350
 6665 WRITE (     M,5895 ) (NODE(N),T(N), N = 1,NODES)                  *3351
      WRITE (     M,5920 )     FIN                                      *3352
      GO TO 4755                                                        *3353
4725  WRITE (     M,5910 )     FIN                                      *3354
      DO 4735  N = 1,NODES                                              *3355
      IF(XMODF(N+13,58))4730, 6670,4730                                 *3356
 6670 WRITE (     M,5910 )     FIN                                      *3357
4730  GX = 0.0                                                          *3358
      DIF(M8)924,924,923                                                *3359
923   CONTINUE                                                          *3360
      GX = G(N)                                                         *3361
924   CONTINUE                                                          *3362
      WRITE (     M,5915 )NODE(N),T(N),DT(N),DDT(N),GX,  W(N),H(N),F(N) *3363
4735  CONTINUE                                                          *3364
      WRITE (     M,5920 )     FIN                                      *3365
CCC   WRITE OUT PHASE OR REACTANT DATA FOR NODES.                       *3366
      IF(NREACT)4755,4755, 6675                                         *3367
 6675 NX = 0                                                            *3368
      DO 4750  N = 1,NODES                                              *3369
      IF(A(N))4740, 6680,4740                                           *3370
 6680 DIF(M3)926,926,925                                                *3371
925   CONTINUE                                                          *3372
      IF(E(N))4740, 6685,4740                                           *3373
 6685 CONTINUE                                                          *3374
926   CONTINUE                                                          *3375
      GO TO 4750                                                        *3376
4740  NX = NX + 1                                                       *3377
      IF(XMODF(NX - 1,58))4745, 6690,4745                               *3378
 6690 WRITE (     M,5900 )     FIN                                      *3379
4745  CONTINUE                                                          *3380
      BX = 0.0                                                          *3381
      DBX = 0.0                                                         *3382
      DDBX = 0.0                                                        *3383
      DIF(M3)928,928,927                                                *3384
927   CONTINUE                                                          *3385
```

TALLY

```
      BX = B(N)                                                         *3386
      DBX = DB(N)                                                       *3387
      DDBX = DDB(N)                                                     *3388
928   CONTINUE                                                          *3389
      WRITE (     M,5905 ) NODE(N),T(N),A(N),CA(N),DCA(N),BX,DBX,DDBX   *3390
4750  CONTINUE                                                          *3391
      WRITE (     M,5920 )     FIN                                      *3392
CCC   WRITE OUT SPECIAL OUTPUT IF ITAPE, IRITE SPECIFIED.               *3393
4755  IF(IRITE)4800,4800, 6695                                          *3394
 6695 IF(KWIT)4810, 6700,4760                                           *3395
 6700 IF((KCYC - 1)*(XMODF(KCYC,IRITE))) 6705, 6705,4800                *3396
 6705 CONTINUE                                                          *3397
4760  WRITE (ITAPE,51005) SUMTIM                                        *3398
      WRITE (ITAPE,51010) (NODE(N),T(N),N=1,NODES)                      *3399
      WRITE (ITAPE,51015)     FIN                                       *3400
4800  IF(KWIT)4810,4810, 6710                                           *3401
 6710 CONTINUE                                                          *3402
C********ENTRY POINT FOR PRODUCING A PUNCHED CARD DECK OF THE FINAL     *3403
C********CONDITIONS OF THE PROBLEM, FOR LATER USE IN A DATA DECK.       *3404
      ENTRY TALLY1                                                      *3405
      IF(NPUNCH) 6715,4810, 6715                                        *3406
 6715 CONTINUE                                                          *3407
C PUNCH FINAL CONDITIONS IN BLOCK 9 FORMAT.                             *3408
      PUNCH 5860 , (NAME(I), I = 1,6),CLOCKB,CLOCKA                     *3409
      PUNCH 5835 , SUMTIM, NODES                                        *3410
      DO 4805  N = 1,NODES                                              *3411
      BX = 0.0                                                          *3412
      GX = 0.0                                                          *3413
      DIF(M3)930,930,929                                                *3414
929   CONTINUE                                                          *3415
      BX = B(N)                                                         *3416
930   CONTINUE                                                          *3417
      DIF(M8)932,932,931                                                *3418
931   CONTINUE                                                          *3419
      GX = GG(N)                                                        *3420
932   CONTINUE                                                          *3421
      PUNCH 5855 , NODE(N),T(N),A(N),BX,GX                              *3422
4805  CONTINUE                                                          *3423
      N = 0                                                             *3424
      PUNCH 5840 , N                                                    *3425
CCC   WRITE OUT DATA FOR SPECIAL OUTPUT NODE NUM ON NON-PRINTOUT CYCLES. *3426
4810  IF(NOW) 6720, 6720,4815                                           *3427
 6720 IF(NUM)4815,4815, 6725                                            *3428
 6725 N = NUM                                                           *3429
      BX = 0.0                                                          *3430
      GX = 0.0                                                          *3431
      DIF(M3)934,934,933                                                *3432
933   CONTINUE                                                          *3433
      BX = B(N)                                                         *3434
934   CONTINUE                                                          *3435
      DIF(M8)936,936,935                                                *3436
935   CONTINUE                                                          *3437
      GX = G(N)                                                         *3438
936   CONTINUE                                                          *3439
      WRITE (     M,5995 )NUMX,T(N),DDT(N),A(N),BX,GX,SUMTIM            *3440
CCC SET TIME DERIVATIVES TO ZERO BEFORE CYCLE ZERO, AND EVERY CYCLE,    *3441
CCC       IF REQUIRED BY INPUT DATA.                                    *3442
4815  IF(NOCT) 6730,4900, 6730                                          *3443
 6730 CONTINUE                                                          *3444
4820  DO 4825  N = 1,NODES                                              *3445
      DDA(N) = 0.0                                                      *3446
      DDT(N) = 0.0                                                      *3447
      DIF(M3)938,938,937                                                *3448
937   CONTINUE                                                          *3449
      DDB(N) = 0.0                                                      *3450
938   CONTINUE                                                          *3451
4825  CONTINUE                                                          *3452
CCC                                                                     *3453
5800  FORMAT(/10X5HCYCLEI5,10H MADE NODEI5,15H A SPECIAL NODE)          *3454
5805  FORMAT(//10X,25HINPUT ERROR, MATERIALS = ,I5,10H, NODES = ,I5,1H.) *3455
5810  FORMAT (8E10.3)                                                   *3456
5815  FORMAT(//12X89H TONE        ALONE        BONE        GONE        FO *3457
```

TAULT

```
     $NE         HCNE         RONE         PONE/10X1P8E12.4)                    *3458
5820  FORMAT(1H1,9X,17HTRUMP OUTPUT CATA,69X,9HDATA DECK,I5,//,                 *3459
     1   10X,A1,14A5,A1,4A2,4X,2(2X,A6))                                        *3460
5825  FORMAT(/,10X,30HPRINTOUT CYCLE  MF   MSS  KWIT,3X,6HDELTMX,7X,            *3461
     1 5HSPALL,8X,5HTVARY,8X,5HCTEMP,8X,6HCTMAXS,5X,4HNUTS,/,                   *3462
     2  10X,5I6,5E13.5,I5)                                                      *3463
5830  FORMAT(/,10X,87H   TOTAL TIME      TIME STEP       HEAT FLOW     TEMP    *3464
     $ FROM FLUX    FLUX RATE       TEMP RATE,/,8X,1P6E15.5,//,                 *3465
     $10X,89H   AVG TEMP      HEAT CAPACITY    HEAT CONTENT     GEN RATE       *3466
     $    HEAT GEN      TEMP FROM GEN,/,8X,1P6E15.5)                            *3467
5835  FORMAT(28HBLOCK 9.  T, A, B, G AT TIME,20X,1PE13.5,5X,I6)                *3468
5840  FORMAT (3I5,5X,40A1)                                                      *3469
5845  FORMAT(//,12X,10HNCTE INDEX,10X,2HTT,13X,2HAA,13X,2HBB,13X,2HGG,          *3470
     1   A1)                                                                    *3471
5850  FORMAT(10X,2I6,4X,1P4E15.6)                                               *3472
5855  FORMAT(I5,15X,1P4E10.3)                                                   *3473
5860  FORMAT(2H**,5X,A1,5A5,15X,2A6)                                            *3474
5865  FORMAT (10I5,2E10.3)                                                      *3475
5870  FORMAT (2I5,7E10.3)                                                       *3476
5875  FORMAT(//11X,95H  KC   KT    DELTO         SMALL         TVARY            *3477
     $  TAU          TIMAX         TMIN         TMAX,/,10X2I5,1P7E13.5)         *3478
5880  FORMAT(//10X45H   KD KSYM    GEOM          SIGMA         TBASE,/,         *3479
     $ 10X2I5,1P3E13.5)                                                         *3480
5885  FORMAT(10X,12HFINAL HEAT =,1PE15.5,5X,12HFINAL FLOW =,E15.5)              *3481
5890  FORMAT(//10X,60HIPRINT   NUM KDATA KSPEC  MCYC  MSEC NPUNCH NDOT I        *3482
     1RITE ITAPE,3X,5HTIMEP,8X,5HSCALE,/,10X,10I6,2E13.5)                       *3483
5895  FORMAT(//,10X,94H  NCDE    TEMP         NODE    TEMP         NODE         *3484
     $ TEMP         NCDE    TEMP         NODE    TEMP,/,(10X,I6,1PE14.5,        *3485
     $ I6,E14.5,I6,E14.5,I6,E14.5,I6,E14.5))                                    *3486
5900  FORMAT(/12X,93HNODE         TEMP         W/F A         CA            D    *3487
     $DA            W/F B         DB            DDB,A1)                          *3488
5905  FORMAT(10X,I6,3X,1P7E13.4)                                                *3489
5910  FORMAT(/12X,91HNODE         TEMP          DT            CDT          G    *3490
     $EN RATE       W             H             F,A1)                            *3491
5915  FORMAT(10X,I6,3X,7E13.4)                                                  *3492
5920  FORMAT(10X,100(1H=),A1)                                                   *3493
5925  FORMAT(15(1H*),17HMORE THAN ALLOWED,I5,15H ITEMS IN BLOCK,I3)             *3494
5930  FORMAT(/,26X,31HMAXIMUM ALLOWED TABLE LENGTH IS,I4,1H.)                   *3495
5935  FORMAT(/,10X,27HARRAY STORAGE REQUIREMENTS./,15X,                         *3496
     1  51HNWRECD = (8 + 7*M9)*M2 + 28*M4 + 12*M5 + 3*M9     = ,I8,/,15X,       *3497
     2  51HNWCHEM = 4*M2 + (6 + 9*M9)*M3 + 10*M4 (M3 NOT C) = ,I8,/,15X,        *3498
     3  51HNWGEN8 = 2*M4 + (3 + 3*M9)*M8          (M8 NOT C) = ,I8,/,15X,        *3499
     4  51HNWFLCW = 4*M4 + (9 + 3*M9)*M10        (M1C NOT C) = ,I8,/,15X,        *3500
     5  51HNWSURE = (12 + 3*M9)*M6 + (5+3*M9)*M7 (M6 NOT C) = ,I8,/,15X,         *3501
     6  51HNWPLCT = 4*M4 + 3*M11 + (1 + M11)*M1 (M11 NOT C) = ,I8,/,15X,         *3502
     7  51HNWBL12 = 8*M4 + 5*M12                 (M12 NOT C) = ,I8)              *3503
5940  FORMAT(/,15X,30HTCTAL ADJUSTABLE ARRAY STORAGE,19X,2H= ,I8)               *3504
5945  FORMAT(/,10X,6HTABLES,10X,60H  CAPT  CONT    QT    ZT    ET HSURT        *3505
     1TEMPB    GT FLOWT TOTAL,/,26X,10I6)                                       *3506
5950  FORMAT(//10X,21HSUMMARY OF INPUT DATA,/10X12HBLOCK NUMBER,4X11I6)         *3507
5955  FORMAT(10X,12HMAXIMUM SIZE,4X,11I6)                                       *3508
5960  FORMAT(10X,12HINITIAL SIZE,4X,11I6)                                       *3509
5965  FORMAT(10X,15HUNMCDIFIED SIZE,1X,11I6)                                    *3510
5970  FORMAT(10X,12HFINAL SIZE  ,4X,11I6)                                       *3511
5975  FORMAT(10X,12HITEM NAME   ,4X,66H   MAT   KEM  NODE  NOD1  NODS  N        *3512
     1ODB  NCDG  NOTE NCDF1 NODEP NODP1,A1)                                     *3513
5980  FORMAT(/10X,12HOTHER TOTALS,4X,38HNOSPEC NOGEN NORAC NORADS NMELT         *3514
     1NREACT,/,26X,6I6)                                                         *3515
5985  FORMAT (10X, 15HKEY, TIMES READ, 1X, 11I6)                                *3516
5990  FORMAT(10X,17HWILL REPEAT CYCLE,I6,4X,8H DTEMP =,1PE1C.3,                 *3517
     1   8H DTMAX =,E10.3,7H DELT =,E10.3,9H SUMTIM =,E10.3)                    *3518
5995  FORMAT(5(1H*),4HNCDE,I6,2X,13HT,DDT,A,B,G =,1P5E12.4,                     *3519
     1   1X,7HAT TIME,E12.4)                                                    *3520
51000 FORMAT(41H*****TIME, TEMPS WILL BE WRITTEN ON TAPE ,I5,6H EVERY,          *3521
     1   I5,12H TIME STEPS.)                                                    *3522
51005 FORMAT(E16.5)                                                             *3523
51010 FORMAT(I6,E12.5)                                                          *3524
51015 FORMAT(6H999999,A1)                                                       *3525
51020 FORMAT (A1, 14A5, A1, 4A2)                                                *3525
4900  RETURN                                                                    *3526
C COMPLETED TALLY.  RETURN TO HEART.                                            *3527
      END                                                                       *3528
                                                                                *3529
```

PLOT

```
*     CARDS COLUMN       PLOT                                          *3530
*     LIST 8                                                           *3531
*     FORTRAN            PLOT    TRUMP CRT PLOTTING SUBROUTINE.        *3532
      SUBROUTINE PLOT                                                  *3533
C********VERSION 5/20/69.                                              *3534
C********PLOT IS AN OPTIONAL SUBROUTINE.  IF NOT NEEDED, SET M1 = 0.   *3535
C********CIF STATEMENTS CONTROL COMPILATION OF VARIOUS OPTIONS, USING  *3536
C********     PARAMETERS KPLOTR, KPLOTT, AND KPLOTL IN CLICHE STORK.   *3537
C********DDBO ROUTINES SETCHM,FRAME, CARTMM, LOGMM, MAPX, SETPCH, SETCH, *3538
C********POINTC, TRACEP, SETLCH, RCONTR, AND WRITE OUTPUT TAPE 10C, ARE *3539
C********USED ONLY AT LRL.  SUBSTITUTES MUST BE FOUND IF SUBROUTINE PLOT *3540
C********IS TO BE USED, AND EITHER M1 OR M11 IS NOT ZERO.              *3541
CCC   LOG SCALES ARE TEMPORARILY CHANGED TO LINEAR IF ARRAYS HAVE NO   *3542
CCC      POSITIVE VALUES.                                              *3543
CCC   A PLOT IS DELETED IF SCALE LIMITS AGREE TO 6 SIGNIFICANT FIGURES. *3544
CCC   FINAL TIME OF S. S. (KWIT = 4) PROBLEMS IS NOT PLOTTED.          *3545
CCC   TEMP VS TIME POINTS IN EXCESS OF M1 ARE NOT PLOTTED, EXCEPT FOR  *3546
CCC      FINAL TIME STEP.                                              *3547
      USE STORK                                                        *3548
      USE STORE                                                        *3549
      USE STORM                                                        *3550
      DIF(M11)982,982,981                                              *3551
981   CONTINUE                                                         *3552
      DIF(M1)984,984,983                                               *3553
983   CONTINUE                                                         *3554
C********USED ONLY IN PLOT.  COULD BE REPLACED BY A DIMENSION STATEMENT. *3555
      DIF(KPLOTR)986,986,985                                           *3556
985   CONTINUE                                                         *3557
      COMMON /APLOT/ RPLOT(M4), TMAXP(M4), TMINP(M4), TPLOT(M4)        *3558
986   CONTINUE                                                         *3559
      DIF(KPLOTT)9810,9810,987                                         *3560
987   CONTINUE                                                         *3561
      COMMON /APLOT/ NODEP(M11), NOXEP(M11), PTEMP(M1,M11), PTIME(M1)  *3562
      COMMON /APLOT/ PRAC(M11)                                         *3563
      DIF(KPLOTL)989,989,988                                           *3564
988   CONTINUE                                                         *3565
      DIMENSION TCONT(41)                                              *3566
989   CONTINUE                                                         *3567
9810  CONTINUE                                                         *3568
CCC                                                                    *3569
CCC                                                                    *3570
      IF(KCYC - 1) 6100,3100,4200                                      *3571
 6100 IF(IBLOCK - 11)2160, 6105,2160                                   *3572
 6105 CONTINUE                                                         *3573
CCC INPUT DATA BLOCK 11.  GRAPHICAL OUTPUT CONTROLS AND SPECIFICATIONS. *3574
      INFRA = INTERP = ITHERM = NGRIC = NTARP = NUP = 0                *3575
      NOSTEM = NOSTIM = NOSR = NOSRT = 1                               *3576
      READ  (    NB,5820 ) JPIC, LOGR, LOGTEM, LOGTIM,                 *3577
     $ FRAC1, FRAC2, FTEMP1, FTEMP2, FTIME1, FTIME2                    *3578
      IF(JPIC) 6110,2100,2100                                          *3579
 6110 DIF(KPLOTR)9834,9834,9833                                        *3580
9833  CONTINUE                                                         *3581
      JPIC = -JPIC                                                     *3582
      NTARP = 1                                                        *3583
      NUP = 1                                                          *3584
      WRITE (    M,5955 )    FIN                                       *3585
9834  CONTINUE                                                         *3586
2100  IF(LOGR) 6115,2102,2102                                          *3587
 6115 LOGR = -LOGR                                                     *3588
      NOSR = NOSRT = 2                                                 *3589
2102  IF(LOGR - 1000)2104, 6120, 6120                                  *3590
 6120 LOGR = XMODF(LOGR,1000)                                          *3591
      INFRA = 1                                                        *3592
2104   IF(LOGR - 100) 2105, 6125, 6125                                 *3593
 6125 LOGR = XMODF(LOGR,100)                                           *3594
       INTERP = 1                                                      *3595
2105   LOSR = XMINOF(1,LOGR)                                           *3596
      IF(XABSF(LOGTEM) - 100)2110, 6130, 6130                          *3597
 6130 NGRIC =1                                                         *3598
      LOGTEM =XMODF(LOGTEM,100)                                        *3599
2110  IF(LOGTEM) 6135,2115,2115                                        *3600
```

PLOT

```
 6135 NOSTEM = 2                                                        *3601
      LOGTEM = - LOGTEM                                                 *3602
2115  LOSTEM = XMINOF(1,LOGTEM)                                         *3603
      IF(XABSF(LOGTIM)-100)2120, 6140, 6140                             *3604
 6140 ITHERM =1                                                         *3605
      LOGTIM =XMODF(LOGTIM,100)                                         *3606
2120  IF(LOGTIM) 6145,2125,2125                                         *3607
 6145 NOSTIM = 2                                                        *3608
      LOGTIM = -LOGTIM                                                  *3609
2125  LOSTIM = XMINOF(1,LOGTIM)                                         *3610
       IF(FRAC2) 2130, 6150,2130                                        *3611
 6150  FRAC2 = 1.0                                                      *3612
2130  IF(FTEMP2) 2135, 6155,2135                                        *3613
 6155  FTEMP2 = 1.0                                                     *3614
2135  IF(FTIME2) 2140, 6160,2140                                        *3615
 6160  FTIME2 = 1.0                                                     *3616
2140  WRITE (     M,5825 )JPIC,LOGR, LOGTEM,LOGTIM,FRAD1,               *3617
     $ FRAC2,FTEMP1,FTEMP2,FTIME1,FTIME2                                *3618
      READ  (    NB,5860 ) (NX1(J), J=1,M11)                            *3619
      DO 2145  J = 1,M11                                                *3620
      IF(NX1(J)) 6165,2150, 6165                                        *3621
 6165 NUP = NUP + 1                                                     *3622
2145  CONTINUE                                                          *3623
2150  NUP1 = NUP - NTARF                                                *3624
       IF(NUP1) 2160,2160, 6170                                         *3625
 6170 DIF(KPLCTT)9812,9812,9811                                         *3626
9811  CONTINUE                                                          *3627
      DO 2155  J = 1,NUP1                                               *3628
      NODEP(J) = NX1(J)                                                 *3629
      NOXEP(J) = NX1(J)                                                 *3630
2155  CONTINUE                                                          *3631
      WRITE (     M,5865 ) M11,(I, I = 1,16), (NODEP(J),J=1,NUP1)       *3632
9812  CONTINUE                                                          *3633
2160  RETURN                                                            *3634
C COMPLETED BLOCK 11.                                                   *3635
CCC   START HERE AT END OF CYCLE 1.                                     *3636
3100  NPLCT = NSKIP = 0                                                 *3637
      CALL FRAME                                                        *3638
      DIF(KPLCTT)9816,9816,9813                                         *3639
9813  CONTINUE                                                          *3640
      IBLOCK = 11                                                       *3641
      CALL REFER(NODEP,NOXEP,NUP1,NODE,NODES)                           *3642
      IF(KWIT)4705, 6175,4705                                           *3643
 6175 CONTINUE                                                          *3644
CCC   FIND NODE LOCATIONS FOR TEMP VS TIME AND ISOTHERM PLOT NODES.     *3645
      IF(NUP1 - 1)3210,6180, 6180                                       *3646
 6180 DO 3110  N = 1,NUP1                                               *3647
      J = NODEP(N)                                                      *3648
      REX = NODE(J)                                                     *3649
      IF(KO - 2)3105, 6185, 6185                                        *3650
 6185 REX = RADIUS(J)                                                   *3651
 3105 PRAD(N) = FRAD1 + FRAC2*REX                                       *3652
 3110 CONTINUE                                                          *3653
      DIF(KPLCTL)9815,9815,9814                                         *3654
 9814 CONTINUE                                                          *3655
CCC   MAKE SURE NODE LOCATION ARRAY HAS NO ZERO SLOPES (RCONTR BUG).    *3656
      DO 3200, N = 2,NUP1                                               *3657
       IF(PRAD(N) - PRAD(N-1))3200, 6190,3200                           *3658
 6190 PRAD(N) = 1.000001*PRAD(N) + 1.0E-24                              *3659
 3200 CONTINUE                                                          *3660
      CALL CARTMM(NUP1,PRMIN,PRMAX,PRAD,1)                              *3661
      IF(PRMAX) 6195, 6195,3205                                         *3662
 6195  LOGR = LOSR = 0                                                  *3663
      NOSFT = 1                                                         *3664
 3205 IF(LOGR)3210,3210, 6200                                           *3665
 6200  CALL LOGMM(LOGR,NUP1,PRMIN,PRMAX,PRAD,1)                         *3666
 9815 CONTINUE                                                          *3667
 9816 CONTINUE                                                          *3668
 3210 IF(XWIT)4705, 6205,4705                                           *3669
 6205 DIF(KPLCTR)9818,9818,9817                                         *3670
 9817 CONTINUE                                                          *3671
CCC   FIND NODE LOCATIONS FOR CRT PLOTS.                                *3672
```

PLOT

```
      IF(NTARP)4200,4200, 6210
 6210 DO 3305  N = 1,NODES
      REX = NODE(N)
      IF(KC - 2)3300, 6215, 6215
 6215 REX = RADIUS(N)
3300  RPLOT(N) = FRAD1 + FRAD2*REX
      TMAXP(N) = -1.0E24
       TMINP(N) =  1.0E24
3305  CONTINUE
      CALL CARTPM(NODES,RMINR,RMAXR,RPLOT,1)
      IF(RMAXR) 6220, 6220,3310
 6220  LOGR = LCSR = 0
      NOSR = 1
3310  IF(LCGR)4200,4200, 6225
 6225  CALL LCGPM(LCGR,NODES,RMINR,RMAXR,RPLOT,1)
9816  CONTINUE
CCC   START HERE AFTER CYCLE 2 OR LATER.
4200  IF(KWIT - 4) 6230,4215, 6230
 6230 DIF(KPLOTT)9820,9820,9819
9819  CONTINUE
CCC   LIMIT NPLOT TO M1.  SAVE FINAL TEMPERATURES AND TIME.
      NPLOT = NPLOT + 1
      IF(NPLOT - M1)4205,4205, 6235
 6235 NPLOT = M1
      NSKIP = NSKIP + 1
4205  PTIME(NPLOT) = FTIME1 + FTIME2*SUMTIM
CCC   FIND TEMPERATURES AND TIMES FOR CRT PLOTS.
      IF(NUP1)4215,4215, 6240
 6240 DO 4210  N = 1,NUP1
      PTEMP(NPLOT,N) = FTEMP1 + FTEMP2*T(NODEP(N))
4210  CONTINUE
9820  CONTINUE
C
C--------MAKE TEMPERATURE VS NODE LOCATION PLOTS.
C
4215  IF(NTARP)4400,4400, 6245
 6245 DIF(KPLOTR)9822,9822,9821
9821  CONTINUE
      DO 4330  KTEM = 1,NOSTEM
      LAGTEM = LOGTEM
      LASTEM = LOSTEM
      IF(KTEM - 2)4220, 6250, 6250
 6250 LAGTEM = LASTEM = 0
4220  DO 4325  KR = 1,NCSR
      LASR = LCSR
      IF(KR - 2)4225, 6255, 6255
 6255 LASR = 0
4225  DO 4235  N = 1,NODES
       TPLOT(N) = FTEMP1 + FTEMP2*T(N)
      TMAXP(N) = MAX1F(TMAXP(N),TPLOT(N))
      TMINP(N) = MIN1F(TMINP(N),TPLOT(N))
4235  CONTINUE
      RMIN = RMINR
      RMAX = RMAXR
      IF(INFRA)4250,4250, 6260
 6260 TX = 0.5*(ABSF(SUMTIM + 1.0E-12))**(-0.5)
      RMIN = RMINR*TX
      RMAX = RMAXR*TX
      DO 4245  N = 1,NODES
      REX = NODE(N)
      IF(KC - 2)4240, 6265, 6265
 6265 REX = RADIUS(N)
4240  RPLOT(N) = (FRAD1 + FRAD2*REX)*TX
4245  CONTINUE
4250  CALL CARTPM(NODES,TMIN1,TMAX1,TPLOT,1)
      IF(TMAX1) 6270, 6270,4305
 6270 LAGTEM = 0
      LASTEM = 0
      KTEM = 2
4305  IF(LAGTEM)4310,4310, 6275
 6275 CALL LCGPM(LAGTEM,NODES,TMIN1,TMAX1,TPLOT,1)
```

PLOT

```
4310  CALL FRAME                                                        *3744
       IF(RMAX - RMIN - 1.0E-6*ABSF(RMAX))4350,4350, 6280               *3745
 6280 IF(TMAX1 - TMIN1 - 1.0E-6*ABSF(TMAX1))4350,4350, 6285             *3746
 6285 NART = 5 + 3*LASR + 2*LASTEM*(1 - 2*LASR) + 4*NGRID               *3747
      CALL MAPX(NART,RMIN,RMAX,TMIN1,TMAX1,0.210,0.920,0.172,0.999)     *3748
      CALL SETPCH(0,0,0,0, 32)                                          *3749
       CALL PCINTC(1H*,RPLCT,TPLOT,NODES)                               *3750
      IF(INTERP) 6290,4315, 6290                                        *3751
 6290  CALL TRACEP(    RPLCT,TPLOT,NODES,3)                             *3752
4315  CALL SETCH( 7.6,69.0,1,0,0,1)                                     *3753
      WRITE (  100,5810 )    FIN                                        *3754
      CALL SETCH(60.0,5.0,1,0,0,0)                                      *3755
      WRITE (  100,5815 )    FIN                                        *3756
      NCYC = KCYC - 1                                                   *3757
       CALL SETCH(16.0,3.0)                                             *3758
      WRITE (  100,5950 )PTIME(NPLOT),NPLOT,NCYC,NPROB,                 *3759
     $  NAME,CLCCKB,CLOCKA                                              *3760
4325  CONTINUE                                                          *3761
4330  CONTINUE                                                          *3762
      WRITE (   MW,5990 )SUMTIM                                         *3763
      CALL FRAME                                                        *3764
CCC   WRITE CUT TABLE OF CATA PLOTTEC ON TEMP VS NOCE LOCATION PLCT.    *3765
4350  CALL SETCH(16.0,64.0,1,0,0,0)                                     *3766
      NCYC = KCYC - 1                                                   *3767
      WRITE (  100,5950 )PTIME(NPLOT),NPLOT,NCYC,NPROB,NAME,CLOCKB,CLCCK *3768
     XA                                                                 *3769
      WRITE (  100,5970 )    FIN                                        *3770
      WRITE (  100,5800 )    FIN                                        *3771
      WRITE (  100,5975 )(NOCE(N),RPLOT(N),TPLOT(N), N = 1,NODES)       *3772
      WRITE (  100,5800 )    FIN                                        *3773
      IF(N - 100)4400, 6295,4400                                        *3774
 6295 CALL FRAME                                                        *3775
9822  CONTINUE                                                          *3776
4400  IF(KWIT)4705,4705, 6300                                           *3777
 6300 CONTINUE                                                          *3778
C                                                                       *3779
C--------MAKE TEMPERATURE VS TIME PLOTS.                                *3780
C                                                                       *3781
       IF(NUP1)4600,4600, 6305                                          *3782
 6305 DIF(KPLCTT)9826,9826,9823                                         *3783
9823  CONTINUE                                                          *3784
      DO 4465  KTEM = 1,NOSTEM                                          *3785
      LAGTEM = LOGTEM                                                   *3786
      LASTEM = LOSTEM                                                   *3787
      IF(KTEM - 2)4405, 6310, 6310                                      *3788
 6310 LAGTEM = LASTEM = 0                                               *3789
4405  DO 4460  KTIM = 1,NOSTIM                                          *3790
      LAGTIM = LOGTIM                                                   *3791
      LASTIM = LOSTIM                                                   *3792
      IF(KTIM - 2)4410, 6315, 6315                                      *3793
 6315 LAGTIM = LASTIM = 0                                               *3794
4410  TMIN1 = 1.0E24                                                    *3795
       TMAX1 = -1.0E24                                                  *3796
CCC   FINC HIGHEST AND LCWEST TEMPERATURES TO BE PLOTTED.               *3797
      DO 4415  N = 1,NUP1                                               *3798
      CALL CARTMM(NPLOT,TLOW,THIGH,PTEMP(1,N),1)                        *3799
      TMIN1 = MIN1F(TMIN1,TLOW)                                         *3800
       TMAX1 = MAX1F(TMAX1,THIGH) + 1.0E-24                             *3801
4415  CONTINUE                                                          *3802
      DY1 = 1.0                                                         *3803
       DY2 = (TMAX1 - TMIN1)* 0.015                                     *3804
      IF(TMAX1) 6320, 6320,4420                                         *3805
 6320 LAGTEM = LASTEM = 0                                               *3806
      KTEM = 2                                                          *3807
4420  IF(LAGTEM)4435,4435, 6325                                         *3808
 6325 TMIN1 = 1.0E24                                                    *3809
      TLOW = TMAX1*0.1**LAGTEM                                          *3810
      DO 4430  N = 1,NUP1                                               *3811
       DO 4425  J = 1,NPLOT                                             *3812
      IF(PTEMP(J,N) - TLOW)4425, 6330, 6330                             *3813
 6330  TMIN1 = MIN1F(TMIN1,PTEMP(J,N))                                  *3814
```

PLOT

```
4425  CONTINUE                                                          *3815
4430  CONTINUE                                                          *3816
      DY1 = (TMAX1 / TMIN1)**0.015                                      *3817
       DY2 = 0.0                                                        *3818
CCC   FIND HIGHEST AND LOWEST TIMES TO BE PLOTTED.                      *3819
4435  START = PTIME(1)                                                  *3820
       FINIS = PTIME(NPLOT) + 1.0E-24                                   *3821
      DX1 = 1.0                                                         *3822
       DX2 = (START - FINIS)* 0.040                                     *3823
      IF(FINIS) 6335, 6335,4440                                         *3824
 6335 LAGTIM = LASTIM = 0                                               *3825
      KTIM = 2                                                          *3826
4440  IF(LAGTIM)4445,4445, 6340                                         *3827
 6340  PTIME(1) = 0.999*PTIME(2)                                        *3828
      START = MAX1F(PTIME(2),FINIS*0.1**LAGTIM)                         *3829
      DX1 = (START / FINIS)**0.040                                      *3830
       DX2 = 0.0                                                        *3831
CCC   PLOT AND WRITE OUT TEMP VS TIME FOR CRT NODES.                    *3832
4445  CALL FRAME                                                        *3833
      IF(FINIS - START - 1.0E-6*ABSF(FINIS))4470,4470, 6345             *3834
 6345 IF(TMAX1 - TMIN1 - 1.0E-6*ABSF(TMAX1))4470,4470, 6350             *3835
 6350 NATT = 5 + 3*LASTIM + 2*LASTEM*(1 - 2*LASTIM) + 4*NGRID           *3836
      CALL MAPX(NATT,START,FINIS,TMIN1,TMAX1,0.210,0.920,0.172,0.999)   *3837
      KINT = XMAXOF(1,(NPLOT + 4)/5)                                    *3838
      DO 4455  N = 1,NUP1                                               *3839
      CALL POINTC(1H*,PTIME,PTEMP(1,N),NPLOT)                           *3840
      CALL TRACEP(    PTIME,PTEMP(1,N),NPLOT,3)                         *3841
CCC   LABEL CURVES WITH NODE NUMBERS.                                   *3842
      DO 4450  K = 1,NPLOT,KINT                                         *3843
       L = NPLOT - K + 1                                                *3844
      XMAP = CX1*PTIME(L) + CX2                                         *3845
      YMAP = CY1*PTEMP(L,N) + DY2                                       *3846
      CALL SETLCH(XMAP,YMAP)                                            *3847
      WRITE (  100,5860 )NOXEP(N)                                       *3848
4450  CONTINUE                                                          *3849
4455  CONTINUE                                                          *3850
      CALL SETLCH(FINIS,TMAX1)                                          *3851
C CLEARS ANY ERROR FLAGS (CD80 BUG).                                    *3852
      CALL SETCH( 7.6,69.0,1,0,0,1)                                     *3853
      WRITE (  100,5810 )     FIN                                       *3854
      CALL SETCH(74.0,5.0,1,0,0,0)                                      *3855
      WRITE (  100,5830 )     FIN                                       *3856
      CALL SETCH(16.0,3.0)                                              *3857
      WRITE (  100,5870 )NPRCB,NAME,CLOCKB,CLOCKA                       *3858
4460  CONTINUE                                                          *3859
4465  CONTINUE                                                          *3860
      CALL FRAME                                                        *3861
      WRITE (    MW,5880 )(NOXEP(N),N = 1,NUP1)                         *3862
      PRINT                 5880 ,(NOXEP(N),N = 1,NUP1)                 *3863
      IF(NSKIP)4470,4470, 6355                                          *3864
 6355 WRITE (    MW,5875 )NPLOT,NSKIP                                   *3865
      PRINT 5875 ,NPLOT,NSKIP                                           *3866
CCC   WRITE OUT TABLES OF TEMP VS TIME POINTS PLOTTED ON CRT GRAPH.     *3867
4470  CALL SETCH(16.0,64.0)                                             *3868
      WRITE (  100,5870 )NPRCB,NAME,CLOCKB,CLOCKA                       *3869
      WRITE (  100,5800 )     FIN                                       *3870
      DO 4475  N = 1,NUP1                                               *3871
      WRITE (  100,5850 )NOXEP(N),PRAD(N)                               *3872
      WRITE (  100,5855 )(PTEMP(K,N), PTIME(K), K = 1,NPLOT)            *3873
      WRITE (  100,5800 )     FIN                                       *3874
4475  CONTINUE                                                          *3875
C                                                                       *3876
C--------MAKE NODE LOCATION VS TIME (ISOTHERM) PLOTS.                   *3877
C                                                                       *3878
      IF(ITHERM)4600,4600, 6360                                         *3879
 6360 DIF(KPLOTL)9825,9825,9824                                         *3880
 9824 CONTINUE                                                          *3881
      IF(PRMAX - PRMIN - 1.0E-6*ABSF(PRMAX))4600,4600, 6365             *3882
 6365 IF(NUP1 - 1)4600,4600, 6370                                       *3883
 6370 DO 4540  KR = 1,NCSRT                                             *3884
      LASR = LCSR                                                       *3885
      IF(KR - 2)4480, 6375, 6375                                        *3886
```

PLOT

```
 6375 LASR = 0                                                          *3887
4480  DO 4535  KTIM = 1,NOSTIM                                          *3888
      LAGTIM = LOGTIM                                                   *3889
      LASTIM = LOSTIM                                                   *3890
      IF(KTIM - 2)4500, 6380, 6380                                      *3891
 6380 LAGTIM = LASTIM = 0                                               *3892
4500  START = PTIME(1)                                                  *3893
       FINIS = PTIME(NPLCT) + 1.0E-24                                   *3894
      IF(FINIS) 6385, 6385,4505                                         *3895
 6385 LAGTIM = LASTIM = 0                                               *3896
      KTIM = 2                                                          *3897
4505  IF(LAGTIM)4510,4510, 6390                                         *3898
 6390  PTIME(1) = 0.999*PTIME(2)                                        *3899
      START = MAX1F(PTIME(2),FINIS*0.1**LAGTIM)                         *3900
4510  NACRT = 5 + 3*LASTIM + 2*LASR*(1 - 2*LASTIM) + 4*NGRID            *3901
      IF(FINIS - START - 1.0E-6*ABSF(FINIS))4600,4600, 6395             *3902
 6395 CALL FRAME                                                        *3903
      CALL MAPX(NACRT,START,FINIS,PRMIN,PRMAX,0.210,0.920,0.172,0.999)  *3904
      TINT = 5.0*TVARY*ABSF(FTEMP2)                                     *3905
4515  NINT = 0.5 + (TMAX1 - TMIN1)/TINT                                 *3906
       IF(NINT - 10) 6400,4520,4520                                     *3907
 6400 TINT = 0.5*TINT                                                   *3908
       GO TO 4515                                                       *3909
4520  IF(NINT - 40)4525,4525, 6405                                      *3910
 6405  TINT = 2.0*TINT                                                  *3911
       GO TO 4515                                                       *3912
4525  N1 = TMIN1/TINT                                                   *3913
      P1 = N1                                                           *3914
      TCONT(1) = P1*TINT                                                *3915
      N1 = TMAX1/TINT                                                   *3916
      P1 = N1                                                           *3917
      TCONT(2) = P1*TINT                                                *3918
      NINT = 0.5 + (TCONT(2) - TCONT(1))/TINT                           *3919
       NCLR = -(NINT + 1)                                               *3920
CCC   MAKE SURE TIME ARRAY HAS NO ZERO SLOPES (RCONTR BUG).             *3921
      DO 4530  N = 2,NPLCT                                              *3922
       IF(PTIME(N) - PTIME(N-1))4530, 6410,4530                         *3923
 6410 PTIME(N) = 1.000001*PTIME(N-1)                                    *3924
4530  CONTINUE                                                          *3925
      CALL RCONTR(NCUR,TCONT,0,PTEMP,M1,PTIME,1,NPLCT,1,PRAD,1,NUP1,1)  *3926
      CALL SETCH( 7.6,57.0,0,0,0,1)                                     *3927
      WRITE (  100,5815 )    FIN                                        *3928
      CALL SETCH(74.0,5.0,0,0,0,0)                                      *3929
      WRITE (  100,5830 )    FIN                                        *3930
      CALL SETCH(16.0,3.0)                                              *3931
       NCUR = -NCUR                                                     *3932
      WRITE (  100,5900 )TCONT(1),TINT,TCONT(NCUR),NPROB,               *3933
     $ NAME,CLOCKB,CLOCKA                                               *3934
4535  CONTINUE                                                          *3935
4540  CONTINUE                                                          *3936
      WRITE (    MW,5905 )TCONT(1),TINT,TCONT(NCUR)                     *3937
      PRINT                 5905 ,TCONT(1),TINT,TCONT(NCUR)             *3938
9825  CONTINUE                                                          *3939
9826  CONTINUE                                                          *3940
C                                                                       *3941
C--------MAKE MAXIMUM AND MINIMUM TEMPERATURE VS NODE LOCATION PLOT.    *3942
C                                                                       *3943
4600  IF(NTARP)4700,4700, 6415                                          *3944
 6415 DIF(KPLCTR)9828,9828,9827                                         *3945
9827  CONTINUE                                                          *3946
      DO 4630  KTEM = 1,NOSTEM                                          *3947
      LAGTEM = LOGTEM                                                   *3948
      LASTEM = LOSTEM                                                   *3949
      IF(KTEM - 2)4602, 6420, 6420                                      *3950
 6420 LAGTEM = LASTEM = 0                                               *3951
4602  DO 4625  KR = 1,NCSR                                              *3952
      LASR = LCSR                                                       *3953
      IF(KR - 2)4604, 6425, 6425                                        *3954
 6425 LASR = 0                                                          *3955
4604  CALL CARTPM(NODES,TEX,TPMAX,TMAXP,1)                              *3956
      CALL CARTPM(NODES,TPMIN,TEX,TMINP,1)                              *3957
      IF(TPMIN) 6430, 6430,4605                                         *3958
```

PLOT

```
 6430 LAGTEM = C                                                        *3959
      LASTEM = C                                                        *3960
      KTEM = 2                                                          *3961
4605  IF(LAGTEM)4610,4610, 6435                                         *3962
 6435 CALL LCGMM(LAGTEM,NCCES,TPMIN,TEX,TMINP,1)                        *3963
      TPMIN = MAX1F(TPMIN,TPMAX*0.1**LAGTEM)                            *3964
4610  CALL FRAME                                                        *3965
      IF(RMAX - RMIN - 1.0E-6*ABSF(RMAX))4650,4650, 6440                *3966
 6440 IF(TPMAX - TPMIN - 1.0E-6*ABSF(TPMAX))4650,4650, 6445             *3967
 6445 NART = 5 + 3*LASR + 2*LASTEM*(1 - 2*LASR) + 4*NGRID               *3968
      CALL MAPX(NART,RMIN,RMAX,TPMIN,TPMAX,0.210,0.920,0.172,0.999)     *3969
      CALL PCINTC(1H*,RPLCT,TMINP,NODES)                                *3970
      CALL PCINTC(1H*,RPLCT,TMAXP,NODES)                                *3971
       IF(INTERF) 6450,4615, 6450                                       *3972
 6450 CALL TRACEP(     RPLOT,TMINP,NODES,3)                             *3973
      CALL TRACEP(     RPLCT,TMAXP,NODES,3)                             *3974
4615  CALL SETCH( 7.6,59.0,1,0,0,1)                                     *3975
      WRITE (  100,5960 )     FIN                                       *3976
      CALL SETCH(60.0,5.0,1,0,0,0)                                      *3977
      WRITE (  100,5815 )     FIN                                       *3978
      CALL SETCH(16.0,3.0)                                              *3979
      WRITE (  100,5965 )NPRCB,NAME,CLOCKB,CLOCKA                       *3980
4625  CONTINUE                                                          *3981
4630  CONTINUE                                                          *3982
      WRITE (    MW,5995 )     FIN                                      *3983
      PRINT 5995                                                        *3984
CCC   WRITE CUT TABLE OF DATA PLOTTED ON MAX AND MIN TEMP PLOT.         *3985
      CALL FRAME                                                        *3986
4650  CALL SETCH(16.0,64.0,1,0,0,0)                                     *3987
      WRITE (  100,5965 )NPRCB,NAME,CLOCKB,CLOCKA                       *3988
      WRITE (  100,5980 )     FIN                                       *3989
      WRITE (  100,5800 )     FIN                                       *3990
      WRITE (  100,5985 )(NCCE(N),RPLOT(N),TMINP(N),                    *3991
     $  TMAXP(N), N = 1,NODES)                                          *3992
      WRITE (  100,5800 )     FIN                                       *3993
9828  CCNTINUE                                                          *3994
C                                                                       *3995
C--------RESTCRE PAGE, FCRMAT CONTROL, RETURN TO HEART.                 *3996
C                                                                       *3997
4700  WRITE (    MW,5805 )     FIN                                      *3998
      CALL FRAME                                                        *3999
4705  IF(M - 100)4710, 6455, 6455                                       *4000
 6455 CALL SETCH( 6.0,64.0,1,0,0,0)                                     *4001
5800  FORMAT(   100(1H=),A1)                                            *4002
5805  FORMAT( /,10X,20HEND OF CRT PLOT DATA,//,A1)                      *4003
5810  FORMAT(11HTEMPERATURE,A1)                                         *4004
5815  FORMAT(32HNCCE LOCATICN (NUMBER OR RADIUS),A1)                    *4005
5820  FORMAT(4I5,6E10.3)                                                *4006
5825  FORMAT(//,12X,94HJPIC  LOGR LOGTEM LOGTIM     FRAD1        FRAD2  *4007
     $   FTEMP1       FTEMP2       FTIME1       FTIME2,/10X4I6,4X1P6E12.4) *4008
5830  FORMAT(4HTIME,A1)                                                 *4009
      DIF(KPLCTT)9830,9830,9829                                         *4010
9829  CONTINUE                                                          *4011
5850  FORMAT(/,4HNCDE,I6,35H.  TEMPERATURE VS TIME.  LOCATION =,        *4012
     $  1PE13.5,//,8X,4(4HTEMP,8X,4HTIME,8X))                           *4013
5855  FORMAT( 4X,1P8E12.4)                                              *4014
5860  FORMAT(16I5)                                                      *4015
5865  FORMAT(//10X,47HNCCES FOR TEMPERATURE VS TIME CRT PLCTS.  MAX =,  *4016
     1   I3,/,10X,5HINDEX,16I5,/,11X,4HNODE,16I5)                       *4017
5870  FORMAT(27HFIG.   TEMPERATURE VS TIME.,63X, 9HDATA DECK,I5,/,      *4018
     1   A1,14A5,A1,4A2,8X,2(2X,A6))                                    *4019
5875  FORMAT(15(1H*),24HTOC MANY CRT PLOT TIMES.,2X,7HPLOTTED,I5,       *4020
     1   15H TIMES, SKIPPED,I5,7H TIMES.)                               *4021
5880  FORMAT( /,10X,30HPLOTTED TEMP VS TIME FOR NODES,10I6,/,10X,6I6)   *4022
      DIF(KPLCTL)9836,9836,9835                                         *4023
9835  CONTINUE                                                          *4024
5900  FORMAT(43HFIG.   NCDE LOCATION VS TIME FOR ISOTHERMS ,1PE10.3,1H(, *4025
     $ E10.3,1H),E10.3,15X,9HDATA DECK,I5,/,A1,14A5,A1,4A2,8X,2(2X,A6))  *4026
5905  FORMAT( /,10X,43HPLOTTED NODE LOCATION VS TIME FOR ISOTHERMS,     *4027
     $  1X,1PE12.5,1H(,E12.5,1H),E12.5)                                 *4028
9836  CONTINUE                                                          *4029
9830  CONTINUE                                                          *4030
```

PLOT
PRIME

```
      DIF(KPLCTR)9832,9832,9831                                         *4031
9831  CONTINUE                                                          *4032
5950  FORMAT(44HFIG.    TEMPERATURE VS NODE LOCATION AT TIME,1PE13.5,6X,  *4033
     $ 4HPLCT,I5,4X,5HCYCLE,I5,4X,9HDATA DECK,I5,/,                     *4034
     2   A1,14A5,A1,4A2,8X,2(2X,A6))                                    *4035
5955  FORMAT( /,10X,30HWILL MAKE TEMP VS RADIUS PLOTS,A1)               *4036
5960  FORMAT(33H MAXIMUM AND MINIMUM TEMPERATURES,A1)                   *4037
5965  FORMAT(57HFIG.   MAXIMUM AND MINIMUM TEMPERATURES VS NODE LOCATION  *4038
     $.,33X, 9HDATA DECK,I5,/,A1,14A5,A1,4A2,8X,2(2X,A6))               *4039
5970  FORMAT(//,3(30H  NODE  LOCATION   TEMPERATURE,5X),A1)             *4040
5975  FORMAT(3(I6,2E12.4,5X))                                           *4041
5980  FORMAT(//,2(41H  NODE  LOCATION     MIN TEMP    MAX TEMP,11X),A1) *4042
5985  FORMAT(2(I6,3E12.4,10X))                                          *4043
5990  FORMAT( /,10X,38HPLOTTED TEMP VS NODE LOCATION FOR TIME,1PE13.5)  *4044
5995  FORMAT( /,10X,42HPLOTTED MAX AND MIN TEMPS VS NODE LOCATION,A1)   *4045
9832  CONTINUE                                                          *4046
984   CONTINUE                                                          *4047
982   CONTINUE                                                          *4048
4710  RETURN                                                            *4049
CCC   END OF PLOT.  RETURN TO HEART.                                    *4050
      END                                                               *4051
                                                                        *4052
*     CARDS COLUMN       PRIME                                          *4053
*     LIST 8                                                            *4054
*     FORTRAN            PRIME                                          *4055
      SUBROUTINE PRIME (NREAD,NLIST,MALL,MUNCH,XD,XC,XK,XT,XTP)         *4056
C********VERSION 5/20/69.                                               *4057
CCC   PRIME SAVES DATA DECKS ON UNIT NDATA IF *DECKS OPTION USED.       *4058
CCC   PRIME READS IN MATERIAL SELECTOR LIST FROM NB (IF MALL = 0), AND  *4059
CCC      TRANSFERS DATA FOR SELECTED MATERIALS (OR ALL, IF MALL = 1)    *4060
CCC      FROM TAPE NREAD TO TAPE NLIST, WITH OPTIONS (IF MUNCH = 2 OR   *4061
CCC      MORE) FOR CONVERTING UNITS OF DATA BEFORE TRANSFER.            *4062
CCC   NREAD IS REWOUND BEFORE AND AFTER TRANSFER IF NOT NRS, NR, OR NB, *4063
CCC      AND NOT NEGATIVE.                                              *4064
CCC   NREAD IS BACKSPACED AFTER ACTUAL EOF IS REACHED IF NRS, NR, OR NB. *4065
CCC   IF NREAD IS NRS, NR, OR NB, $ IN COL 1 IS RECOGNIZED AS AN EOF.   *4066
CCC   NLIST IS REWOUND BEFORE AND AFTER TRANSFER IF NOT MW, M, OR 100,  *4067
CCC      AND NOT NEGATIVE.  IF NOT MW, M, OR 100, NLIST WILL BE ENDED   *4068
CCC      WITH A BLANK RECORD AND ENDFILED.                              *4069
CCC   BUFFERING RATE IS ABOUT 4500 CARDS PER MINUTE.                    *4070
CCC   TRIES 10 TIMES ON 10 CHECKS BEFORE GIVING UP.                     *4071
CCC   COPIED CARDS ARE ALSO WRITTEN ON TAPE MW (3 AT LRL).              *4072
CCC                                                                     *4073
      USE STORK                                                         *4074
      DIF(KPRIME)910,910,91                                             *4075
91    CONTINUE                                                          *4076
      USE STORE                                                         *4077
CCC   WILL USE STORM ONLY TO GET ARRAY NX1.                             *4078
      USE STORM                                                         *4079
      INTEGER P,PS,A1                                                   *4080
      CHAR KCHAR                                                        *4081
C********LABELLED COMMON APRIM USED ONLY IN PRIME.  COULD BE REPLACED BY *4082
C********     A DIMENSION STATEMENT.                                    *4083
      COMMON /APRIM/   MATP(M2),MATS(M2),P(20),PS(10),PROP(M9),TVARP(M9) *4084
      DIMENSION KCHAR(1)                                                *4085
      EQUIVALENCE (KCHAR,P)                                             *4086
CCC                                                                     *4087
CCC                                                                     *4088
CCC   SET UP BOOLEAN WORDS USED IN MASKING OPERATIONS.                  *4089
CCC   MASK1 KEEPS RIGHT 4 CHAR, MASK2.SHL.24  KEEPS LEFT 6 OF 10 CHAR.  *4090
CCC   MASK3 KEEPS LEFT 6 OF 8 CHAR, MASK4 KEEPS LEFT 2 OF 8 CHAR.       *4091
CCC   MASK5 KEEPS RIGHT 6 OF 8 CHAR, MASK6 KEEPS RIGHT 2 OF 8 CHAR.     *4092
      DATA (MASK1 = 77777777B), (MASK2 = 7777777777777B),               *4093
     1   (MASK3 = 77777777777770000B), (MASK4 = 7777C000CCCCCCC0B),     *4094
     2   (MASK5 = 7777777777777B), (MASK6 = 7777B)                      *4095
CCC                                                                     *4096
CCC   ELIMINATE ERROR WARNINGS ON SUBROUTINE RETURN ARGUMENTS.          *4097
      DATA (NHOUR = 0), (NDATE = 0), (NWR = 0), (NWW = 0)               *4098
CCC                                                                     *4099
CCC   ENTER SUBROUTINE HERE.                                            *4100
CCC                                                                     *4101
CCC   FIND WORDS/CARD RECORD, OCTAL EQUIV. OF RIGHT-ADJ. $.             *4102
```

PRIME

```
      NW = 8                                                            *4103
      KOR = 48                                                          *4104
      IF(KLUDGE - 6600) 6100,2100,2100                                  *4105
 6100 NW = 10                                                           *4106
      KOR = 538                                                         *4107
CCC                                                                     *4108
CCC   SET BLANK OR ZERO NREAD TO NB, BLANK OR ZERO NLIST TO 4.          *4109
CCC   PREVENT REWINDING IF NREAD, NLIST NEGATIVE, OR REGULAR I/O UNITS. *4110
2100  NWINDW = 0                                                        *4111
      IF(NLIST)2110, 6105,2105                                          *4112
 6105 NLIST = 4                                                         *4113
2105  IF((NLIST - MW)*(NLIST - 100)*(NLIST - M)) 6110,2110, 6110        *4114
 6110 REWIND NLIST                                                      *4115
      GO TO 2115                                                        *4116
2110  NLIST = XABSF(NLIST)                                              *4117
      NWINDW = -1                                                       *4118
2115  NWINDR = 0                                                        *4119
      IF(NREAD)2125, 6115,2120                                          *4120
 6115 NREAD = NB                                                        *4121
2120  IF((NREAD - NB)*(NREAD - NR)*(NREAD - NRS)) 6120,2125, 6120       *4122
 6120 REWIND NREAD                                                      *4123
      GO TO 2130                                                        *4124
2125  NREAD = XABSF(NREAD)                                              *4125
      NWINDR = -1                                                       *4126
2130  WRITE (    M,5840 ) NREAD, NLIST, MALL, MUNCH, NWINDR, NWINDW     *4127
      PRINT 5840 ,    NREAD, NLIST, MALL, MUNCH, NWINDR, NWINDW         *4128
CCC                                                                     *4129
CCC   SET UP UNIT CONVERSION FACTORS FOR DATA ON OUTPUT TAPE.           *4130
      MALT = MALL                                                       *4131
      IF(MUNCH - 1)2200,2200, 6125                                      *4132
 6125 IF(MUNCH - 5) 6130,2150,2155                                      *4133
 6130 IF(MUNCH - 3) 6135,2135,2140                                      *4134
 6135 XD = 1000.0                                                       *4135
      XC = 4186.0                                                       *4136
      XK = 418.6                                                        *4137
      XT = 1.0                                                          *4138
      XTP = 273.15                                                      *4139
      GO TO 2155                                                        *4140
2135  XD = 62.43                                                        *4141
      XC = 1.0                                                          *4142
      XK = 242.0                                                        *4143
      GO TO 2145                                                        *4144
2140  XD = 0.03613                                                      *4145
      XC = 1.0                                                          *4146
      XK = 0.005602                                                     *4147
2145  XT = 1.8                                                          *4148
      XTP = 32.0                                                        *4149
      GO TO 2155                                                        *4150
2150  XD = 1.0                                                          *4151
      XC = 0.001                                                        *4152
      XK = 0.00864                                                      *4153
      XT = 1.0                                                          *4154
      XTP = 0.0                                                         *4155
2155  XH = XC*XT                                                        *4156
      MALT = 0                                                          *4157
      WRITE (    M,5800 ) XD,XC,XK,XT,XTP,XH                            *4158
CCC                                                                     *4159
CCC   READ IN MATERIAL SELECTOR LIST.                                   *4160
2200  NMATS = 30000                                                     *4161
      IF(MALL) 6140, 6140,2300                                          *4162
 6140 N = NMATS = 0                                                     *4163
      WRITE (    M,5875 ) NREAD, NLIST                                  *4164
2205  READ  (    NB,5845 ) MATPX, MATSX, NBLANK                         *4165
      IF(MATPX - NBLANK) 6145,2220, 6145                                *4166
 6145 NMATS = N = N + 1                                                 *4167
CCC   PROTECT MEMORY FROM ARRAYS MATP AND MATS OVERFLOWING.             *4168
      IF(NMATS - M2)2210,2210, 6150                                     *4169
 6150 N = NMATS = M2                                                    *4170
      KWIT = 11                                                         *4171
2210  IF(MATSX - NBLANK)2215, 6155,2215                                 *4172
 6155 MATSX = MATPX                                                     *4173
```

```
2215  MATP(N) = MATPX                                                  *4174
      MATS(N) = MATSX                                                  *4175
      WRITE (     M,5805 ) MATPX,MATSX,N                               *4176
      GO TO 2205                                                       *4177
CCC                                                                    *4178
CCC   REORDER LIST INTO ASCENDING PRIMARY MATERIAL NUMBERS.            *4179
2220  IF(NMATS)2300,2300, 6160                                         *4180
 6160 DO 2230  N = 1,NMATS - 1                                         *4181
      DO 2225  NN = N + 1,NMATS                                        *4182
      IF(MATP(N) - MATP(NN))2225,2225, 6165                            *4183
 6165 N1 = MATP(N)                                                     *4184
      MATP(N) = MATP(NN)                                               *4185
      MATP(NN) = N1                                                    *4186
      N1 = MATS(N)                                                     *4187
      MATS(N) = MATS(NN)                                               *4188
      MATS(NN) = N1                                                    *4189
2225  CONTINUE                                                         *4190
2230  CONTINUE                                                         *4191
      WRITE (     M,5865 ) NMATS                                       *4192
CCC                                                                    *4193
CCC   BEGIN BUFFERING IN NREAD, SEARCHING FOR SELECTED MATERIALS.      *4194
CCC   READ IN CARD IMAGE FROM TAPE NREAD (FIRST CARD, AND AFTER IOCHECK) *4195
2300  KMAT = N1 = 1                                                    *4196
      N2 = NW                                                          *4197
      KCOPY = NCARD = NCARDS = MATCH = NRED = 0                        *4198
      WRITE (    MW,5880 ) NREAD, NLIST                                *4199
CCC   CHANGE FORMAT FOR CC80 OUTPUT TO 42 LINES, 85 CHARACTERS.        *4200
      IF(NLIST - 100)2305, 6170,2305                                   *4201
 6170 CALL FRAME                                                       *4202
      CALL SETCH(1.0,42.0,1,0,1,0)                                     *4203
2305  BUFFER IN (NREAD,0) (P(N1),P(N2))                                *4204
CCC   FIND OUT IF READ IS OVER, HOW MANY WORDS WERE READ IN.           *4205
2310  IF(UNIT,NREAD,NWR)2310,2400,2315,2320                            *4206
C  AN END-OF-FILE HAS BEEN REACHED.                                    *4207
2315  IF(EOF,NREAD) 6175,2400                                          *4208
 6175 IF((NREAD - NRS)*(NREAD - NR)*(NREAD - NB))3100, 6180,3100       *4209
 6180 CALL BSPACE (NREAD)                                              *4210
      GO TO 3100                                                       *4211
2320  IF (IOCHECK,NREAD) 6185,2400                                     *4212
 6185 CONTINUE                                                         *4213
C  A TAPE CHECK HAS OCCURRED.  TRY 10 TIMES BEFORE GOING ON.           *4214
      IF(NREC - 10) 6190, 6190,2325                                    *4215
 6190 NREC = NREC + 1                                                  *4216
      CALL BSPACE(NREAD)                                               *4217
      GO TO 2305                                                       *4218
2325  WRITE (    MW,5855 ) NCARD, (P(N), N = N1,N2)                    *4219
      PRINT 5855 ,      NCARD, (P(N), N = N1,N2)                       *4220
CCC                                                                    *4221
CCC   COPY OR TEST GOOD CARD RECORD.                                   *4222
2400  NCARD = NCARD + 1                                                *4223
      IF(KCOPY) 6195, 6195,2600                                        *4224
 6195 K = (18 - NW)*(N1 - 1)                                           *4225
      IF((NREAD - NRS)*(NREAD - NR)*(NREAD - NB))2405, 6200,2405       *4226
 6200 IF(KCHAR(K + 1) - KOR)2405, 6205,2405                            *4227
 6205 NCARD = NCARD - 1                                                *4228
      GO TO 3100                                                       *4229
2405  IF(MALT)2410,2410, 6210                                          *4230
 6210 KCOPY = 1                                                        *4231
      GO TO 2600                                                       *4232
2410  IF(KMAT - NMATS)2415,2415, 6215                                  *4233
 6215 NCARD = NCARD - 1                                                *4234
      GO TO 3100                                                       *4235
CCC                                                                    *4236
2415  CONTINUE                                                         *4237
      DIF(MACHINE)93,93,92                                             *4238
92    CONTINUE                                                         *4239
CCC   CODING FOR 6600 WITH 60-BIT WORDS, ASCII CHARACTER SET FOLLOWS.  *4240
CCC   SEE IF NEXT SELECTED MATERIAL NUMBER IS ON CARD READ IN FROM NREAD *4241
      IF(KCHAR(K + 1)) 6220,2600, 6220                                 *4242
 6220 IF(MALL) 6225, 6225,2425                                         *4243
 6225 IF((MATP(KMAT).INT.MASK1) - (P(N1).INT.MASK1))2465, 6230,2600    *4244
 6230 CONTINUE                                                         *4245
```

```
CCC   SUBSTITUTE MATS(KMAT) FOR MASTER MATERIAL NUMBER.                  *4246
2420  P(N1) = ((P(N1).I.T.(MASK2.SHL.24)).UN.(MATS(KMAT).INT.MASK1))     *4247
CCC   FIND OUT WHAT TABLE LENGTHS ARE.                                   *4248
2425  INT1 = INT2 = 0                                                    *4249
      IF(KCHAR(K + 24))2430,2430, 6235                                   *4250
 6235 INT1 = KCHAR(K + 24) - 208                                         *4251
2430  IF(KCHAR(K + 25))2435,2435, 6240                                   *4252
 6240 INT2 = KCHAR(K + 25) - 208                                         *4253
2435  LTABC = 10*INT1 + INT2                                             *4254
      INT1 = INT2 = 0                                                    *4255
      IF(KCHAR(K + 29))2440,2440, 6245                                   *4256
 6245 INT1 = KCHAR(K + 29) - 208                                         *4257
2440  IF(KCHAR(K + 30))2500,2500, 6250                                   *4258
 6250 INT2 = KCHAR(K + 30) - 208                                         *4259
93    CONTINUE                                                           *4260
CCC                                                                      *4261
      DIF(MACHINE)94,94,95                                               *4262
94    CONTINUE                                                           *4263
CCC   CODING FOR 3600 WITH 48-BIT WORDS,  3600 CHARACTER SET FOLLOWS.    *4264
CCC   SEE IF NEXT SELECTED MATERIAL NUMBER IS ON CARD READ IN FROM NREAD *4265
      IF(KCHAR(K + 1) - 60B) 6255,2600, 6255                             *4266
 6255 IF(MALL) 6260, 6260,2445                                           *4267
 6260 IF(((MATP(KMAT).SHR.12).INT.MASK6) - (P(N1).INT.MASK6))2465, 6265, *4268
     12600                                                               *4269
 6265 IF(((MATP(KMAT).INT.MASK6) - ((P(N1+1).SHR.36).INT.MASK6))2465, 627 *4270
     10, 2600                                                            *4271
 6270 CONTINUE                                                           *4272
CCC   SUBSTITUTE MATS(KMAT) FOR MASTER MATERIAL NUMBER.                  *4273

      P(N1) = ((P(N1).INT.MASK3).UN.((MATS(KMAT).SHR.12).INT.MASK6))     *4275
      P(N1+1) = ((P(N1+1).INT.MASK5).UN.((MATS(KMAT).SHL.36).INT.MASK4)) *4276
CCC   FIND OUT WHAT TABLE LENGTHS ARE.                                   *4277
2445  INT1 = INT2 = 0                                                    *4278
      IF(KCHAR(K + 24) - 60B) 6275,2450, 6275                            *4279
 6275 INT1 = KCHAR(K + 24)                                               *4280
2450  IF(KCHAR(K + 25) - 60B) 6280,2455, 6280                            *4281
 6280 INT2 = KCHAR(K + 25)                                               *4282
2455  LTABC = 10*INT1 + INT2                                             *4283
      INT1 = INT2 = 0                                                    *4284
      IF(KCHAR(K + 29) - 60B) 6285,2460, 6285                            *4285
 6285 INT1 = KCHAR(K + 29)                                               *4286
2460  IF(KCHAR(K + 30) - 60B) 6290,2500, 6290                            *4287
 6290 INT2 = KCHAR(K + 30)                                               *4288
95    CONTINUE                                                           *4289
CCC                                                                      *4290
      GO TO 2500                                                         *4291
CCC  SELECTED MATL NOT ON MASTER TAPE, SKIP TO NEXT SELECTED MATL.       *4292
2465  WRITE (    MW,5860 ) MATP(KMAT),NCARD,NCARDS,(P(N), N = N1,N2)     *4293
      PRINT 5860 ,      MATP(KMAT),NCARD,NCARDS,(P(N), N = N1,N2)        *4294
      KMAT = KMAT + 1                                                    *4295
      GO TO 2405                                                         *4296
CCC                                                                      *4297
CCC   COME HERE WHEN DATA FOR A MATERIAL IS TO BE WRITTEN OUT.        P  *4298
CCC   FIND OUT HOW MANY CARDS ARE TO BE BUFFERED ONTO TAPE NLIST.        *4299
2500  LTABK = 10*INT1 + INT2                                             *4300
      KMAT = KMAT + 1                                                    *4301
      MATCH = MATCH + 1                                                  *4302
      IF(LTABC - 1)2505, 6295,2505                                       *4303
 6295 LTABC = 0                                                          *4304
2505  IF(LTABK - 1)2510, 6300,2510                                       *4305
 6300 LTABK = 0                                                          *4306
2510  KCOPY = 1 + (LTABC + 3)/4 + (LTABK + 3)/4                          *4307
CCC                                                                      *4308
CCC   CONVERT UNITS BEFORE WRITING OUT DATA ONTO SECONDARY TAPE.         *4309
      IF(MUNCH - 2)2600, 6305, 6305                                      *4310
 6305 DIF(MACHINE)97,97,96                                               *4311
96    CONTINUE                                                           *4312
      WRITE (63,5815) (P(N), N = N1,N2)                                  *4313
      READ  (63,5830) A1,J1,K1,K2,L1,L2,P1,P2,P3,P4,NX1                  *4314
97    CONTINUE                                                           *4315
      DIF(MACHINE)98,98,99                                               *4316
```

```
98     CONTINUE                                                          *4317
  830 FORMAT(A5,5I5,5E10.3)                                               *4318
2515  DECODE 5825,(P(N),N=N1,N2),830,(A1,J1,K1,K2,L1,L2,P1,P2,P3,P4,NX1) *4319
99     CONTINUE                                                          *4320
       CALL PATCH(NX1,0.0,P5,LBX)                                        *4321
       P1 = P1*XC                                                        *4322
       P2 = P2*XC                                                        *4323
       P3 = P3*XK                                                        *4324
       P4 = P4*XT + XTP                                                  *4325
       P5 = P5*XH                                                        *4326
       WRITE (NLIST,5830 ) A1,J1,K1,K2,L1,L2,P1,P2,P3,P4,P5              *4327
       IF(NLIST - MW) 6310,2525, 6310                                    *4328
 6310 WRITE (    MW,5830 ) A1,J1,K1,K2,L1,L2,P1,P2,P3,P4,P5               *4329
2525  IF(LTABC)2540,2540, 6315                                           *4330
 6315 READ  (NREAD,5835 ) (PROP(N),TVARP(N), N = 1,LTABC)                *4331
       DO 2530  N = 1,LTABC                                              *4332
       PROP(N) = PROP(N)*XC                                              *4333
       TVARP(N) = TVARP(N)*XT + XTP                                      *4334
2530  CONTINUE                                                           *4335
       WRITE (NLIST,5835 ) (PROP(N),TVARP(N), N = 1,LTABC)               *4336
       IF(NLIST - MW) 6320,2535, 6320                                    *4337
 6320 WRITE (    MW,5835 ) (PROP(N),TVARP(N), N = 1,LTABC)               *4338
2535  IF(LTABC - M9)2540,2540, 6325                                      *4339
 6325 KWIT = 12                                                           *4340
2540  IF(LTABK)2555,2555, 6330                                           *4341
 6330 READ  (NREAD,5835 ) (PROP(N),TVARP(N), N = 1,LTABK)                *4342
       DO 2545  N = 1,LTABK                                              *4343
       PROP(N) = PROP(N)*XK                                              *4344
       TVARP(N) = TVARP(N)*XT + XTP                                      *4345
2545  CONTINUE                                                           *4346
       WRITE (NLIST,5835 ) (PROP(N),TVARP(N), N = 1,LTABK)               *4347
       IF(NLIST - MW) 6335,2550, 6335                                    *4348
 6335 WRITE (    MW,5835 ) (PROP(N),TVARP(N), N = 1,LTABK)               *4349
2550  IF(LTABK - M9)2555,2555, 6340                                      *4350
 6340 KWIT = 12                                                           *4351
2555  NCARDS = NCARDS + KCOPY                                            *4352
       NCARD = NCARD + KCOPY - 1                                         *4353
       KCOPY = 0                                                         *4354
CCC                                                                      *4355
CCC    PREPARE COUNTERS FOR NEXT CARD TO BE READ IN.                     *4356
2600  NRED = 0                                                           *4357
       NS = N1                                                           *4358
       N1 = 12 - N1                                                      *4359
       N2 = N1 + NW - 1                                                  *4360
CCC    READ IN CARD IMAGE FROM TAPE NREAD, AFTER FIRST CARD.             *4361
       BUFFER IN (NREAD,0) (P(N1),P(N2))                                 *4362
2605  IF(KCOPY)2310,2310, 6345                                           *4363
 6345 KCOPY = KCOPY - 1                                                   *4364
CCC    FIND OUT IF LAST WRITE ON NLIST IS COMPLETE.                      *4365
2610  IF(UNIT,NLIST,NWW)2610,2615,2615,2615                              *4366
CCC                                                                      *4367
CCC    SET UP ARRAY TO BE BUFFERED OUT (LAST CARD TO BE BUFFERED IN).    *4368
2615  DO 2620  N = 1,10                                                  *4369
       J = NS + N - 1                                                    *4370
       PS(N) = P(J)                                                      *4371
2620  CONTINUE                                                           *4372
CCC                                                                      *4373
CCC    WRITE OUT CARD IMAGE ON TAPE NLIST.                               *4374
       IF(NLIST - MW) 6350,2635, 6350                                    *4375
 6350 IF(NLIST - 100)2630, 6355,2630                                     *4376
 6355 IF(NW - 8) 6360, 6360,2625                                         *4377
 6360 WRITE (NLIST,5810 ) (PS(N), N = 1,NW)                              *4378
       GO TO 2635                                                        *4379
2625  WRITE (NLIST,5820 ) (PS(N), N = 1,NW)                              *4380
       GO TO 2635                                                        *4381
2630  BUFFER OUT (NLIST,0) (PS(1),PS(NW))                                *4382
CCC    WRITE OUT CARD IMAGE ON TAPE MW (3 AT LRL).                       *4383
2635  CALL COC206 (MW,PS,NW)                                             *4384
       NCARDS = NCARDS + 1                                               *4385
       GO TO 2310                                                        *4386
CCC                                                                      *4387
CCC    COME HERE AFTER TRANSFER IS COMPLETE.                             *4388
```

PRIME
PATCH

```
3100  IF(LNIT,NLIST,NWW)3100,3105,3105,3105                          *4389
CCC   IF NLIST NOT REGULAR OUTPUT UNIT, END WITH BLANK RECORD AND EOF.  *4390
3105  IF((NLIST - MW)*(NLIST - 100)*(NLIST - M)) 6365,3110, 6365     *4391
 6365 WRITE (NLIST,5850 )     FIN                                    *4392
      NCARDS = NCARDS + 1                                            *4393
      CALL WRTECF (NLIST)                                            *4394
      IF(NWINDW)3110, 6370, 6370                                     *4395
 6370 REWIND NLIST                                                   *4396
3110  IF(NWINDR)3115, 6375, 6375                                     *4397
 6375 REWIND NREAD                                                   *4398
3115  WRITE (     M,5870 ) NCARD, NCARDS, MATCH                      *4399
      PRINT 5870 ,     NCARD, NCARDS, MATCH                          *4400
CCC   RESTORE FORMAT FOR DD80 OUTPUT TO 64 LINES, 128 CHARACTERS.    *4401
      IF(NLIST - 100)4700, 6380,4700                                 *4402
 6380 CALL FRAME                                                     *4403
      CALL SETCH(6.0,64.0,1,0,0,0)                                   *4404
CCC                        FORMATS.                                 *4405
5800  FORMAT(/,10X,33HUNIT CONVERSION FACTORS ARE     XD,11X,2HXC,11X,  *4406
     1   2HXK,11X,2HXT,11X,3HXTP,10X,2HXH,/,38X,6E13.5)              *4407
5805  FORMAT(10X,2(1X,R5),I6)                                        *4408
5810  FORMAT(5X,8A10)                                                *4409
5815  FORMAT(8A10)                                                   *4410
5820  FORMAT(5X,10A8)                                                *4411
5825  FORMAT(10A8)                                                   *4412
5830  FORMAT(A5,5I5,5E10.3)                                          *4413
5835  FORMAT(8E10.3)                                                 *4414
5840  FORMAT(//,10X,27H*MAST OR *DECK OPTION USED.,/,30X,            *4415
     1   37HNREAD NLIST  MALL MUNCH NWINDR NWINDW,/,29X,4I6,2I7)     *4416
5845  FORMAT(2R5,40X,R5)                                             *4417
5850  FORMAT(10X,A1)                                                 *4418
5855  FORMAT(15(1H*),22HERROR CHECK AFTER CARD,I5,1H.,/,10X,10A10)   *4419
5860  FORMAT(/,10(1H*),8HMATERIAL,2X,R5,23H IS NOT ON MASTER TAPE.,  *4420
     1   2X,10HCARDS IN =,I6,14H.  CARDS OUT =,I6,1H.,/,10X,         *4421
     2   7HREACHED,3X,10A10)                                         *4422
5865  FORMAT(/,15X,36HTOTAL NUMBER OF SELECTED MATERIALS =,I6,1H.)   *4423
5870  FORMAT(//,15X,10HCARDS IN =,I6,14H.  CARDS OUT =,I6,           *4424
     1   18H.  MATERIALS OUT =,I6,1H.)                               *4425
5875  FORMAT(/,10X,46HMATERIALS SELECTED FOR DATA TRANSFER FROM UNIT,I5,  *4426
     1   2X,7HTO UNIT,I5,1H.,//,12X,16HMATP  MATS INDEX)             *4427
5880  FORMAT(//,10X,26HDATA TRANSFERRED FROM UNIT,I5,2X,             *4428
     1   7HTO UNIT,I5,11H FOLLOWS...,//)                             *4429
910   CONTINUE                                                       *4430
4700  RETURN                                                         *4431
CCC   END OF PRIME, RETURN TO THERM (*MAST OPTION) OR HEART (*DECKS). *4432
      END                                                            *4433
                                                                     *4434
*     CARDS COLUMN       PATCH                                       *4435
*     LIST 8                                                         *4436
*     FORTRAN            PATCH   TRUMP DECIMAL PT FINDING SUBROUTINE. *4437
      SUBROUTINE PATCH (NXX,CONE,POUT,LBX)                           *4438
C********VERSION 5/20/69.                                            *4439
CCC   PATCH STORES CONE IN POUT IF NXX CONTAINS NO DECIMAL POINT,    *4440
CCC      OTHERWISE CONVERTS NXX TO FLOATING POINT AND STORES IN POUT. *4441
      USE STORK                                                      *4442
      DIMENSION NXX(10)                                              *4443
      DO 2100  NC = 1,10                                             *4444
      IF(NXX(NC) - 1H.) 6100,2105, 6100                              *4445
 6100 CONTINUE                                                       *4446
2100  CONTINUE                                                       *4447
      POUT = CONE                                                    *4448
      LBX = 0                                                        *4449
      GO TO 4700                                                     *4450
C********CODING FOR 6600.                                            *4451
      DIF(MACHINE)92,92,91                                           *4452
91    CONTINUE                                                       *4453
2105  WRITE (63,5805) NXX                                            *4454
      READ  (63,5810) POUT                                           *4455
92    CONTINUE                                                       *4456
      DIF(MACHINE)93,93,94                                           *4457
93    CONTINUE                                                       *4458
C********CODING FOR 3600.                                            *4459
  810 FORMAT(E10.3)                                                  *4460
```

DATCH
REFER
SEEK1

```
2105  DECODE 5805,(NXX),810,(POUT)                                    *4461
94    CONTINUE                                                        *4462
      LBX = 1                                                         *4463
5810  FORMAT(E10.3)                                                   *4464
5805  FORMAT(10A1)                                                    *4465
4700  RETURN                                                          *4466
      END                                                             *4467
                                                                      *4468
*     CARDS COLUMN        REFER                                       *4469
*     LIST 8                                                          *4470
*     FORTRAN             REFER   TRUMP CROSS-REFERENCING SUBROUTINE. *4471
      SUBROUTINE REFER(LIST,LISTX,MAX,LISTR,MAR)                      *4472
C********VERSION 5/20/69.                                             *4473
C********REFER IS A REQUIRED SUBROUTINE.                               *4474
CCC   REFER IS USED FOR CROSS-REFERENCING NODE, MATERIAL, REACTANT NOS. *4475
CCC   FINDS J FOR WHICH LISTX(N) = LISTR(J), MAKES LIST(N) = J.       *4476
      DIMENSION LIST(1), LISTR(1), LISTX(1)                           *4477
      USE STORE                                                       *4478
      IF(MAR*MAX)4700,4700, 6100                                      *4479
 6100 IF(NEWBL(IBLOCK))4700,4700, 6105                                *4480
 6105 N1 = 1                                                          *4481
      IF(IBLOCK - 1)4700,3100, 6110                                   *4482
 6110 IF(NEWBL(IBLOCK) - 1000) 6115,3100,3100                         *4483
 6115 N1 = XMINOF(ITEMS(IBLOCK) + 1,MAX)                              *4484
 3100 DO 3120  N = N1,MAX                                             *4485
      DO 3105  J = MAR,1,-1                                           *4486
      IF(LISTX(N) - LISTR(J)) 6120,3115, 6120                         *4487
 6120 CONTINUE                                                        *4488
 3105 CONTINUE                                                        *4489
      IF(IBLOCK - 2)3110, 6125,3110                                   *4490
 6125 IF(LISTX(N)) 6130,3120, 6130                                    *4491
 6130 CONTINUE                                                        *4492
 3110 WRITE (    MW,5800 )N,IBLOCK,LISTX(N)                           *4493
      PRINT 5800 ,N,IBLOCK,LISTX(N)                                   *4494
      KWIT = 5                                                        *4495
      GO TO 3120                                                      *4496
 3115 LIST(N) = J                                                     *4497
 3120 CONTINUE                                                        *4498
 5800 FORMAT(/,15(1H*),8HERROR IN,I5,17H'TH ITEM IN BLOCK,I3,         *4499
     1   52H.  REFERS TO UNSPECIFIED MATL, REACTANT, OR NODE NO.,I6,1H.) *4500
 4700 RETURN                                                          *4501
CCC   END OF REFER.  RETURN TO CALLER.                                *4502
      END                                                             *4503
                                                                      *4504
*     CARDS COLUMN        SEEK1                                       *4505
*     LIST 8                                                          *4506
*     FORTRAN             SEEK1   TRUMP LIST-SEARCHING SUBROUTINE NO. 1. *4507
      SUBROUTINE SEEK1 (N, N1, NB1, MAX, K)                           *4508
C********VERSION 5/20/69.                                             *4509
C********SEEK1 IS A REQUIRED SUBROUTINE.                               *4510
CCC   SEEK1 IS USED FOR FINDING DATA BLOCK ITEMS TO BE ALTERED.       *4511
CCC   SEEK1 LOOKS FOR N1 IN ARRAY NB1, SETS N TO SUBSCRIPT OF NB1 IF  *4512
CCC     N1 FOUND, OTHERWISE SETS N TO ARRAY SIZE PLUS 1.              *4513
      USE STORE                                                       *4514
      DIMENSION NB1(1)                                                *4515
      IF(MAX)2105,2105, 6100                                          *4516
 6100 DO 2100  J = MAX,1,-1                                           *4517
      IF(N1 - NB1(J)) 6105,2110, 6105                                 *4518
 6105 CONTINUE                                                        *4519
 2100 CONTINUE                                                        *4520
 2105 MAX = MAX + 1                                                   *4521
      N = MAX                                                         *4522
      K = 0                                                           *4523
      WRITE (    M,5800 )N1,IBLOCK                                    *4524
      PRINT 5800 ,N1,IBLOCK                                           *4525
      GO TO 4700
 2110 N = J                                                           *4526
      K = 1                                                           *4527
 5800 FORMAT(5(1H*),5HITEM ,I6,6X,33H NOT FOUND, ADDED TO END OF BLOCK, *4528
     1   I4,1H.)                                                      *4529
 4700 RETURN                                                          *4530
      END                                                             *4531
```

SEEK2
TIMER

```
*     CARDS COLUMN       SEEK2
*     LIST 8
*     FORTRAN            SEEK2   TRUMP LIST-SEARCHING SUBROUTINE NO. 2.
      SUBROUTINE SEEK2 (N, N1, N2, NB1, NB2, MAX, K)
C********VERSION 5/20/69.
C********SEEK2 IS A REQUIRED SUBROUTINE.
CCC   SEEK2 IS USED FOR FINDING DATA BLOCK ITEMS TO BE ALTERED.
CCC   SEEK2 LOOKS FOR PAIR OF NUMBERS N1, N2 IN PAIR OF ARRAYS NB1, NB2,
CCC    SETS N TO SUBSCRIPT OF NB1, NB2 IF FOUND, OTHERWISE SETS N TO
CCC   ARRAY SIZE PLUS 1.
      USE STORE
      DIMENSION NB1(1), NB2(1)
      IF(MAX)2105,2105, 6100
 6100 DO 2100  J = MAX,1,-1
      IF(N1 - NB1(J))2100, 6105,2100
 6105 IF(N2 - NB2(J)) 6110,2110, 6110
 6110 CONTINUE
 2100 CONTINUE
 2105 MAX = MAX + 1
      N = MAX
      K = 0
      WRITE (    M,5800 )N1,N2,IBLOCK
      PRINT 5800 ,N1,N2,IBLOCK
      GO TO 4700
 2110 N = J
      K = 1
 5800 FORMAT(5(1H*),5HITEMS,2I6,33H NOT FOUND, ADDED TO END OF BLOCK,
     1   I4,1H.)
 4700 RETURN
      END

*     CARDS COLUMN       TIMER
*     LIST 8
*     FORTRAN            TIMER
      SUBROUTINE TIMER(NPER,MICS)
C********VERSION 5/20/69.
      USE STORK
      DIF(MACHINE)92,92,91
 91   CONTINUE
      CALL TICHEK(NPER,MICS)
      MICS = MICS/1000000
 92   CONTINUE
      DIF(MACHINE)93,93,94
 93   CONTINUE
      DATA (KSECZ = 0)
      CALL CLOCK1(KSEC,NPER)
      MICS = XMODF(86400 + KSEC - KSECZ,86400)
      KSECZ = KSEC
      NPER = 8H UNKNOWN
 94   CONTINUE
      RETURN
      END
```

ACKNOWLEDGMENTS

The author is indebted to the developers of the many computer programs and numerical methods from which various features were adapted to use in TRUMP, to the many users of TRUMP who have made useful suggestions for additions and improvements in the program, to the staff of the Computer Division of UCRL, Livermore, for their continued help and cooperation in the development of TRUMP, and to those with sufficient foresight to provide the opportunity for this effort to be made.

References

1. A. L. Edwards, TRUMP: A Computer Program for Transient and Steady State Temperature Distributions in Multidimensional Systems, Lawrence Livermore Laboratory, Rept. UCRL-14754, Rev. 2 (1969).
2. A. L. Edwards, A Compilation of Thermal Property Data for Computer Heat-Conduction Calculations, Lawrence Livermore Laboratory, Rept. UCRL-50589 (1969).
3. A. L. Edwards, TRUMP/XB2: A Computer Program for Processing the TRUMP Thermal Property Data Compilation, Lawrence Livermore Laboratory, Rept. UCRL-50663 (1969).
4. H. S. Carslaw and J. C. Jaeger, Conduction of Heat in Solids (Oxford at the Clarendon Press, London, 1960), 2nd ed.
5. G. M. Dusinberre, Heat Transfer Calculations by Finite Differences (International Textbook Co., Scranton, Pa., 1961).
6. R. D. Richtmyer and K. W. Morton, Difference Methods in Initial Value Problems (Interscience Publishers Division of John Wiley and Sons, London, 1967).
7. R. B. Bird, W. E. Stewart, and E. N. Lightfoot, Transport Phenomena (John Wiley and Sons, New York, 1960).
8. H. C. Hottel and A. F. Sarofim, Radiative Transport (McGraw-Hill Book Co., New York, 1967).
9. D. Lawrence, Computer Information Center Abstracts, Lawrence Livermore Laboratory, 12th ed. (1968).
10. D. Braff, 7094 Users' Manual, Lawrence Livermore Laboratory, Rept. CIC-M-A-011 (1966).
11. D. Braff, CDC-3600 Users' Manual, Lawrence Livermore Laboratory, Rept. CIC-M-B-001 (1966).
12. F. N. Fritsch, CDC-6600 Monitor 400 Manual, Lawrence Livermore Laboratory, Rept. CIC Manual E-012 (1968).
13. G. E. Theall, FORTRAN-400 Language Manual, Lawrence Livermore Laboratory, Rept. CIC Manual-T-004 (1966).
14. F. N. Fritsch, LRLTRAN Language Manual, Lawrence Livermore Laboratory, Rept. CIC Manual-L-001 (1967).
15. F. N. Fritsch, Introduction to OCTOPUS, Lawrence Livermore Laboratory, Rept. CIC Manual-I-002 (1967).
16. A. Cecil, G. Michael, and F. N. Fritsch, dd80 Programmers Manual (dd80A), Lawrence Livermore Laboratory, Rept. CIC N2.8-002 (1968).
17. J. G. Fletcher and J. D. Lawrence, Livermore Time Sharing System, Lawrence Livermore Laboratory, Rept. CIC-LTSS-1-Ed. 1 (1968) (supersedes parts of references 10-15 above).
18. A. L. Edwards and R. L. Holzman, "Thermal Effects of a Nuclear Explosion in Salt: The Salmon Experiment," Die Naturwissenschaften 55, No. 1, pp. 18-22 (1968).

19. A. L. Edwards, TRUMP Computer Program: Calculation of Transient Laminar Fluid Flow in Porous Media, Lawrence Livermore Laboratory, Rept. UCRL-50664 (1969). (Errata issued Jan. 8, 1971.)

20. V. K. Gabrialson, HEATMESH: A Computer Code for Generating Geometrical Data Required for Studies of Heat Transfer in Axisymmetric Structures, Sandia Laboratories, Livermore, Rept. SCL-DR-67-30 (1967).

21. D. Schauer, FED: A Computer Program to Generate Geometric Input for the Heat Transfer Code TRUMP, Lawrence Livermore Laboratory, Rept. UCRL-50816 (1970).

22. G. W. Evans, R. J. Brousseau, and R. Keirstead, Instability Considerations for Various Difference Equations Derived from the Diffusion Equation, Lawrence Livermore Laboratory, Rept. UCRL-4476 (1954).

23. R. Lobel, GIFT Computer Program for Calculating One-Dimensional Geometric Input for the TRUMP Heat Transfer Code, Lawrence Livermore Laboratory, Engineering Note END 71-11, May 1971.

24. D. Schauer, Memorandum to TRUMP Users, Lawrence Livermore Laboratory, June 22, 1971.

DISTRIBUTION

LRL Internal Distribution		External Distribution	
A. L. Edwards	40	TID-4500 Distribution, UC-32 Mathematics and Computers	207
D. Montan			
R. Rozsa			
D. Schauer			
LBL Library			
TID File	50		

LOF:rd

NOTICE

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

Printed in USA. Available from the National Technical Information Center, National Bureau of Standards, U.S. Department of Commerce, Springfield, Virginia 22151
Price: Printed Copy $3.00; Microfiche $0.95.