AD NUMBER

AD823105

NEW LIMITATION CHANGE

TO

Approved for public release, distribution unlimited

FROM

Distribution authorized to U.S. Gov't. agencies and their contractors; Critical Technology; NOV 1967. Other requests shall be referred to Army Missile Command, Redstone Arsenal, AL 35809.

AUTHORITY

usamc ltr, 13 mar 1973

APPLICATION OF THE FINITE ELEMENT METHOD TO HEAT CONDUCTION IN SOLIDS

by

E. B. Becker
C. H. Parr

U. S. ARMY MISSILE COMMAND
Redstone Arsenal, Alabama 35809

Contracts
DA-01-021 AMC-11536(Z)
DAAH01-67-C-0947

ROHM AND HAAS COMPANY
REDSTONE RESEARCH LABORATORIES
HUNTSVILLE, ALABAMA 35807

November 1967 Technical Report S-117

APPLICATION OF THE FINITE ELEMENT METHOD TO HEAT CONDUCTION IN SOLIDS

by

E. B. Becker
C. H. Parr

U. S. ARMY MISSILE COMMAND
Redstone Arsenal, Alabama 35809

Contracts
DA-01-021 AMC-11536(Z)
DAAH01-67-C-0947

Distribution Limited

STATEMENT #2 UNCLASSIFIED

This document is subject to special export controls and each transmittal to foreign governments or foreign nationals may be made only with prior approval of Hq U.S. Army Missile Comd. Redstone Arsenal, Ala.

ROHM AND HAAS COMPANY
REDSTONE RESEARCH LABORATORIES
HUNTSVILLE, ALABAMA 35807

FOREWORD

The work in this report was initiated under Contract DA-01-021 AMC-11536(Z) for exploratory development of propellants for missiles and rockets, and completed under Contract DAAH01-67-C-0947 for exploratory development of solid propulsion technology. Both contracts were under the technical cognizance of Army Propulsion Laboratory and Center, Research and Development Directorate, U. S. Army Missile Command.

The application of finite-element methods to heat-conduction problems is an important way station to the successful application of these methods to more complex time-dependent situations—specifically, to viscoelastic problems of solid propellants and solid-propellant rocket motors.

The work described here has immediate application to propellant grains and rocket nozzles. But this method has general application beyond solid-propulsion technology. Accordingly, with the view that broader distribution will ultimately be authorized, the body of the report contains no allusion to propellants or rockets.

ABSTRACT

A new numerical method for the solution of heat conduction problems in thermally anisotropic, nonhomogeneous bodies of complex geometry was devised which is based on a discretization concept developed in the matrix analysis of structures. This discretization method, commonly referred to as the finite element method, reduces the problem formulation to the solution of a matrix equation for the nodal point temperatures of the assembly of finite elements. The resulting matrix equation is stable for any time step. The method is extremely flexible and easy to apply. The method was applied by writing a computer program for the solution of heat conduction problems in plane, thermally anisotropic, nonhomogeneous bodies.

CONTENTS

CONTENTS (Cont'd)

Section I. INTRODUCTION

The approximate analysis of heat conduction and other diffusion phenomena in bodies of complex geometry has generally been accomplished by using various finite difference techniques, e.g., [1]. These methods suffer from a number of limitations or restrictions which depend on the type of formulation. Explicit finite difference methods, for example, have stability criteria that often make the time increment requirements excessively small, which in turn make computation time excessively large. Regular grid arrays, which yield simple finite difference operators are difficult to adapt to complex boundaries. This problem is compounded when multi-material bodies are considered, since each material interface must be treated as a boundary.

Other types of solution are becoming more common, especially those approximate methods based on variational principles [2]. This fact, coupled with experience and ideas developed in applying variational methods to the matrix analysis of structures, has led to the present development. From this previous experience it was expected that the use of finite element methods would make multi-material bodies and bodies of complex geometry more amenable to solution, as well as providing a compatible nodal point system for coupled usage with numerical stress analysis procedures based on similar concepts.

The present work applies a variational method, along with a discretization concept developed in the matrix analysis of structures, to numerical analysis of heat conduction in thermally anisotropic, nonhomogeneous bodies.[1] This discretization method, commonly referred to as the finite element method, reduces the problem formulation to the solution of a matrix equation for the nodal-point temperatures of the assembly of finite elements.

First, a functional of the temperature field and of its first time derivative is introduced. Then it is shown that when the functional is an extremum, it satisfies the heat conduction equation throughout the body and satisfies general flux boundary conditions over the part of the boundary where the temperature is not specified. Under the assumption of a piecewise linear temperature distribution in a small quadrilateral element which is made up of four triangular elements with linear temperature distributions, the

[1] After the initiation of this work, a similar approach to this problem was published by Nickell and Wilson [3].

variational principle is used to establish a matrix equation for the element in terms of its corner, or nodal-point, temperatures and its boundary conditions. Since this is done in a matrix formulation, the resulting equations for the assemblage of finite elements constituting the body of interest are easily assembled by methods of matrix algebra.

The resulting matrix equation is stable for any time step, thus offering potential advantages over the explicit finite difference methods in computer running time. Each quadrilateral element in the assemblage may have distinct and anisotropic thermal properties. Complex geometries can be approximated as closely as desired with a piecewise linear boundary.

Although the development is done in general terms, the computer program written to demonstrate the method is limited to a plane, nonhomogeneous body whose axes of anisotropy must be in the same Cartesian frame over the body. Internal heat generation is neglected, but adiabatic, constant flux, convective, and temperature boundary conditions may be applied. Extension of the program to general anisotropy, internal heat generation, and axially symmetric bodies can be easily accomplished. Extension to three-dimensional geometries is straightforward in concept but will involve extension of present programming concepts.

Section II. FORMULATION OF THE VARIATIONAL PRINCIPLE[2]

Let Π, a functional of the temperature field $U(x, y, z)$ and the first time derivative of the temperature field $\dot{U}(x, y, z)$, be defined by (1).

$$\Pi(U, \dot{U}) = \int_V \left\{ \tfrac{1}{2} U_{,i} k_{ij} U_{,j} + \rho c U \dot{U} - QU \right\} dV - \int_S n_i q_i U \, dS \,, \tag{1}$$

where

V = volume of the region,

S = boundary of the region,

$k_{ij} \equiv k_{ij}(x, y, z)$ = thermal-conductivity tensor,

$c \equiv c(x, y, z)$ = specific heat,

$\rho \equiv \rho(x, y, z)$ = density,

Q = internal heat-source density,

q_i = heat flux vector across a boundary, and

n_i = unit normal vector.

A comma denotes differentiation with respect to the following subscript, and repeated subscripts imply summation. The quantities k, c and ρ are assumed to be temperature and time independent. Q and q are specified functions of time, and S and V, characterizing the region, do not change.

The variation of $\Pi(U, \dot{U})$ with respect to U (with $\dot{U}$ held constant) is given by

$$\delta\Pi = \frac{\partial \Pi(U + \epsilon\lambda, \dot{U})}{\partial \epsilon} \,,$$

where ϵ is a small parameter and λ is any one of a family of functions that is 0 on the portions of S on which temperature is specified and arbitrary elsewhere. An extremum of the function Π is sought, which implies that $\delta\Pi(U, \dot{U})$ must be zero, i.e.,

---

[2] A similar variational principle for isotropic materials is given in [4].

$$\left.\frac{\delta\Pi(U + \epsilon\lambda,\ \dot{U})}{\partial\epsilon}\right|_{\epsilon=0} = 0 \ .$$

Starting with

$$\Pi(U + \epsilon\lambda,\ \dot{U}) = \int_V \left\{ \tfrac{1}{2}(U + \epsilon\lambda)_{,i}\, k_{ij} (U + \epsilon\lambda)_{,j} + \rho c\,(U + \epsilon\lambda)\dot{U} - Q(U + \epsilon\lambda) \right\} dV$$

$$- \int_S n_i q_i (U + \epsilon\lambda)\, dS \ , \qquad (2)$$

there results

$$\frac{\partial\Pi(U + \epsilon\lambda,\ \dot{U})}{\partial\epsilon} = \int_V \left\{ \left[(U + \epsilon\lambda)_{,i} k_{ij} \lambda \right]_{,j} - k_{ij} (U + \epsilon\lambda)_{,ji} \lambda + \rho c \lambda \dot{U} - Q\lambda \right\} dV - \int_S n_i q_i \lambda\, dS \ . \qquad (3)$$

The volume integral

$$\int_V (U_{,i}\, k_{ij} \lambda)_{,j}\, dV$$

can be transformed into a surface integral

$$\int_V (U_{,i}\, k_{ij} \lambda)_{,j}\, dV = \int_S n_i k_{ij} U_{,j} \lambda\, dS$$

which gives, when (3) is evaluated at $\epsilon = 0$,

$$\left.\frac{\partial\Pi(U + \epsilon\lambda, \dot{U})}{\partial\epsilon}\right|_{\epsilon=0} = \int_V (-k_{ij} U_{,ji} + \rho c \dot{U} - Q)\, \lambda \ dV + \int_S n_i k_{ij} U_{,j}\, \lambda\, dS - \int_S n_i q_i \lambda\, dS = 0. \qquad (4)$$

The vanishing of $\delta\Pi$ requires then, that in V

$$k_{ij}\, U_{,ji} = \rho c \dot{U} - Q \qquad (5)$$

and on S

$$n_i k_{ij} U_{,j} = n_i q_i \quad . \qquad (6)$$

Eq. (5) is the Fourier heat equation and (6) defines the heat flux q at the surface of the body. Therefore a function U which gives an extremum of the functional defined by (1) satisfies both the field equation and boundary equations of transient heat conduction in an anisotropic body.

Section III. DISCRETIZATION OF THE PROBLEM

In the preceding development, $\Pi(U, \dot{U})$ is a functional of any functions U and $\dot{U}$ which will satisfy the boundary conditions on S. However, if the choice of U and $\dot{U}$ is restricted such that their only arbitrariness is in certain constants in their formulation, the functional Π becomes a real-valued function. In particular, if the constants are the vector of nodal-point values[3], $\underset{\sim}{u}$ and $\underset{\sim}{\dot{u}}$ of U and $\dot{U}$, Π becomes $\Pi(\underset{\sim}{u}, \underset{\sim}{\dot{u}})$. Finding an extremum of this real-valued function is equivalent to satisfying the following.

$$\frac{\delta \Pi(\underset{\sim}{u}, \underset{\sim}{\dot{u}})}{\delta u_i} = 0 \quad . \tag{7}$$

In the following, the body will be considered to be divided into a number of tetrahedral or plane triangular elements. These are, in some sense, small with respect to the temperature gradient and boundary contours such that the temperature distribution and boundary can be represented by a piecewise linear approximation. The nodal points for the numerical analysis will be the vertices of the elements.

Let the temperature field in an element be given by

$$U(x, y, z, t) = \underset{\sim}{\phi}(x, y, z)\, \underset{\approx}{A}\, \underset{\sim}{u}(t) \tag{8}$$

and the time rate of temperature change in the element be given by

$$\dot{U}(x, y, z, t) = \underset{\sim}{\psi}(x, y, z)\, \underset{\approx}{B}\, \dot{\underset{\sim}{u}}(t) \quad , \tag{9}$$

where $\underset{\sim}{\phi}$ and $\underset{\sim}{\psi}$ are vectors which specify the spatial dependence of U and $\dot{U}$ and $\underset{\sim}{u}$ and $\dot{\underset{\sim}{u}}$ are the vectors of nodal point values.[4] The matrices of constants, $\underset{\approx}{A}$ and $\underset{\approx}{B}$, are defined by the above relationships.

The temperature gradient $U_{,i}$ in the element can be expressed in terms of the nodal-point temperatures by

$$\Delta U \equiv \begin{bmatrix} U_{,x} \\ U_{,y} \\ U_{,z} \end{bmatrix} = \underset{\approx}{D}\, \underset{\sim}{\phi}\, \underset{\approx}{A}\, \underset{\sim}{u} = \underset{\approx}{\phi}'\, \underset{\approx}{A}\, \underset{\sim}{u} \quad . \tag{10}$$

---

[3]The notation $\underset{\approx}{A}$ indicates the matrix A, and $\underset{\sim}{u}$ the vector u.

[4]It should be noted that throughout this development, the fields U and $\dot{U}$ have been taken to be independent. In the computer program, however, $\underset{\sim}{\phi}$ and $\underset{\sim}{\psi}$, and therefore $\underset{\approx}{A}$ and $\underset{\approx}{B}$, were taken to be the same.

Writing Π in terms of nodal point quantities,

$$\Pi(\underset{\sim}{u}, \dot{\underset{\sim}{u}}) = \int_V \left\{ \tfrac{1}{2} \underset{\sim}{u}^T \underset{\sim}{A}^T \underset{\sim}{\phi}'^T \underset{\sim}{k} \underset{\sim}{\phi}' \underset{\sim}{A} \underset{\sim}{u} + \rho c \underset{\sim}{u}^T \underset{\sim}{A}^T \underset{\sim}{\phi}^T \underset{\sim}{\psi} \underset{\sim}{B} \dot{\underset{\sim}{u}} - \underset{\sim}{u}^T \underset{\sim}{A}^T \underset{\sim}{\phi}^T \underset{\sim}{Q} \right\} dV - \int_S \underset{\sim}{u}^T \underset{\sim}{A}^T \underset{\sim}{\phi}^T \underset{\sim}{n} \underset{\sim}{q} \, dS \, . \tag{11}$$

Taking the first variation with respect to $\underset{\sim}{u}$ $\left(\text{i.e., } \frac{\partial \Pi}{\partial \underset{\sim}{u}}\right)$ and setting it equal to zero, there results

$$\delta \Pi = \underset{\sim}{K} \underset{\sim}{u} - \underset{\sim}{Q}^* + \underset{\sim}{C} \dot{\underset{\sim}{u}} - \underset{\sim}{q}^* = 0 \quad , \tag{12}$$

where

$$\underset{\sim}{K} = \int_V \underset{\sim}{A}^T \underset{\sim}{\phi}'^T \underset{\sim}{k} \underset{\sim}{\phi}' \underset{\sim}{A} \, dV \quad , \tag{13}$$

$$\underset{\sim}{C} = \int_V \rho c \, \underset{\sim}{A}^T \underset{\sim}{\phi}^T \underset{\sim}{\psi} \, \underset{\sim}{B} \, dV \quad , \tag{14}$$

$$\underset{\sim}{Q}^* = \int_V \underset{\sim}{Q} \underset{\sim}{A}^T \underset{\sim}{\phi}^T \, dV \quad , \tag{15}$$

and

$$\underset{\sim}{q}^* = \int_S \underset{\sim}{A}^T \underset{\sim}{\phi}^T \underset{\sim}{n} \underset{\sim}{q} \, dS \quad . \tag{16}$$

Boundary conditions of four types will now be considered:

(1) Specified temperature, u_i = constant (boundary segment S_1),

(2) Specified flux, $q = \bar{q}$ (boundary segment S_2),

(3) Convective, $q = h(u_i - u_0)$, where h is a film coefficient and u_0 is the environmental temperature (boundary segment S_3), and

(4) Adiabatic, $q = 0$ (boundary segment S_4).

The boundary integral (16) now becomes

$$\underset{\sim}{q}^* = \bar{\underset{\sim}{q}}^* + \underset{\sim}{H} \underset{\sim}{u} - \underset{\sim}{h}^* \quad , \tag{17}$$

where

$$\underset{\sim}{q}^* = \int_{S_2} \underset{\approx}{A}^T \underset{\sim}{\phi}^T \underset{\approx}{n} \bar{q} \, dS \quad , \tag{18}$$

$$\underset{\approx}{H} = h \int_{S_3} \underset{\approx}{A}^T \underset{\sim}{\phi}^T \underset{\sim}{\phi} \underset{\approx}{A} \, dS \quad , \tag{19}$$

and

$$\underset{\sim}{h}^* = h u_0 \int_{S_3} \underset{\approx}{A}^T \underset{\sim}{\phi}^T \, dS \, . \tag{20}$$

The integral over S_1 is zero since the variation of the functional was specified as zero over that portion of the boundary, and the integral over S_4 is zero since there is no heat flow across the boundary.

To assure the extremum of the functional Π, it is necessary then to find the nodal-point temperatures $\underset{\sim}{u}$ which satisfy the following matrix equation.

$$(\underset{\approx}{K} - \underset{\approx}{H}) \underset{\sim}{u} + \underset{\approx}{C} \dot{\underset{\sim}{u}} - \underset{\sim}{\bar{q}}^* - \underset{\sim}{Q}^* + \underset{\sim}{h}^* = 0 \, . \tag{21}$$

Section IV. SOLUTION OF THE GOVERNING MATRIX EQUATION

1. Solution Method

To solve (21), note that $\underset{\sim}{u}$ and $\dot{\underset{\sim}{u}}$ are functions of time and $\underset{\sim}{Q}^*$, $\underset{\sim}{q}^*$ and $\underset{\sim}{h}^*$ are given functions of time. Let the time variable be restricted to the following set of variables.

$$t_i = i\,\Delta t \quad , \qquad i = 0, 1, 2, \ldots$$

Subscripts "i" in the subsequent development indicate that the subscripted quantities are evaluated at $t = t_i$.

Let (21) be written as

$$\underset{\approx}{K}\,\underset{\sim}{u}_i + \underset{\approx}{C}\,\dot{\underset{\sim}{u}}_i = \underset{\sim}{f}_i \quad , \tag{22}$$

where

$$\underset{\approx}{K} = \underset{\approx}{\kappa} - \underset{\approx}{H}$$

and

$$\underset{\sim}{f}_i = \underset{\sim}{Q}^* + \dot{\underset{\sim}{q}}^* - \underset{\sim}{h}^* \quad .$$

If $\ddot{\underset{\sim}{u}}$ is assumed constant for $t_i \le t \le t_{i+1}$, then $\ddot{\underset{\sim}{u}}_i = [\dot{\underset{\sim}{u}}_{i+1} - \dot{\underset{\sim}{u}}_i]/\Delta t$ and from Taylor's expansion about $t = t_i$

$$\begin{aligned} \underset{\sim}{u}_{i+1} &= \underset{\sim}{u}_i + \Delta t\,\dot{\underset{\sim}{u}}_i + \frac{\Delta t^2}{2}\,\ddot{\underset{\sim}{u}}_i \\ &= \underset{\sim}{u}_i + \Delta t\,\dot{\underset{\sim}{u}}_i + \frac{\Delta t}{2}\,[\dot{\underset{\sim}{u}}_{i+1} - \dot{\underset{\sim}{u}}_i] \\ &= \underset{\sim}{u}_i + \frac{\Delta t}{2}\,[\dot{\underset{\sim}{u}}_{i+1} + \dot{\underset{\sim}{u}}_i] \quad . \end{aligned} \tag{23}$$

Eqs. (22) and (23) now are sufficient to determine $\underset{\sim}{u}_{i+1}$ and $\dot{\underset{\sim}{u}}_{i+1}$ in terms of $\underset{\sim}{u}_i$ and $\dot{\underset{\sim}{u}}_i$. Solving (23) for $\dot{\underset{\sim}{u}}_{i+1}$ yields

$$\dot{\underset{\sim}{u}}_{i+1} = \frac{2}{\Delta t}\,[\underset{\sim}{u}_{i+1} - \underset{\sim}{u}_i] - \dot{\underset{\sim}{u}}_i \quad . \tag{24}$$

Substituting this value into (22) gives

$$\left(\underset{\approx}{K} + \frac{2}{\Delta t}\,\underset{\approx}{C}\right)\underset{\sim}{u}_{i+1} = \underset{\sim}{f}_{i+1} + \underset{\approx}{C}\left[\frac{2\underset{\sim}{u}_i}{\Delta t} + \dot{\underset{\sim}{u}}_i\right] . \tag{25}$$

Also from (22),

$$\dot{\underset{\sim}{u}}_i = \underset{\approx}{C}^{-1} \underset{\sim}{f}_i - \underset{\approx}{C}^{-1} \underset{\approx}{K} \underset{\sim}{u}_i \quad . \tag{26}$$

Substituting (26) into (25) results in

$$\left(\underset{\approx}{K} + \frac{2}{\Delta t} \underset{\approx}{C}\right) \underset{\sim}{u}_{i+1} = \left(\frac{2}{\Delta t} \underset{\approx}{C} - \underset{\approx}{K}\right) \underset{\sim}{u}_i + \underset{\sim}{f}_i + \underset{\sim}{f}_{i+1} \quad . \tag{27}$$

A simpler computation results by rewriting (27) as

$$\left(\underset{\approx}{K} + \frac{2}{\Delta t} \underset{\approx}{C}\right) \underset{\sim}{u}_{i+1} = - \left(\underset{\approx}{K} + \frac{2}{\Delta t} \underset{\approx}{C}\right) \underset{\sim}{u}_i + \frac{4}{\Delta t} \underset{\approx}{C} \underset{\sim}{u}_i + \underset{\sim}{f}_i + \underset{\sim}{f}_{i+1} \tag{28}$$

or

$$\left(\underset{\approx}{K} + \frac{2}{\Delta t} \underset{\approx}{C}\right) \left(\frac{\dot{\underset{\sim}{u}}_{i+1} + \dot{\underset{\sim}{u}}_i}{2}\right) = \frac{2}{\Delta t} \underset{\approx}{C} \underset{\sim}{u}_i + \left(\frac{\underset{\sim}{f}_i + \underset{\sim}{f}_{i+1}}{2}\right) \quad , \tag{29}$$

where $\underset{\sim}{u}_{i+1}$ is found from the auxiliary calculation[5]

$$\underset{\sim}{u}_{i+1} = 2\left(\frac{\underset{\sim}{u}_{i+1} + \underset{\sim}{u}_i}{2}\right) - \underset{\sim}{u}_i \quad . \tag{30}$$

For the solution of the heat flow in a multi-element body, it is necessary to assemble the element matrix equations (29) into a single matrix equation. This assembly is a complex task which can be performed in an efficient manner by a computer. The general method of assembly for matrix equations is given in [5], Section 7.2.

2. Stability of Solution Technique

To study the stability of the solution technique defined above by (27), i.e., the effect on the numerical solution of an error introduced at some step, consider the vector $\underset{\sim}{u}_i$ which satisfies exactly the relation

$$\left(\underset{\approx}{K} + \frac{2}{\Delta t} \underset{\approx}{C}\right) \underset{\sim}{u}_{i+1} + \left(\underset{\approx}{K} - \frac{2}{\Delta t} \underset{\approx}{C}\right) \underset{\sim}{u}_i = \underset{\sim}{f}_i + \underset{\sim}{f}_{i+1} \quad . \tag{31}$$

---

[5] Also note that if Δt is very large (29) reduces to $\underset{\approx}{K} \underset{\sim}{u}_i = \underset{\sim}{f}_i$, the steady-state form. Thus the steady-state solution can be obtained in one iteration from the computer program simply by making the single time increment very large.

Suppose that, at some step (i+1) in the calculation of $\underset{\sim}{u}$, an error (say round-off error) is introduced in the calculation (31), which can be rewritten in terms of the incorrect value $\underset{\sim}{u}^*$ as

$$\left(\underset{\approx}{K} + \frac{2}{\Delta t}\,\underset{\approx}{C}\right)\underset{\sim}{u}^*_{i+1} + \left(\underset{\approx}{K} - \frac{2}{\Delta t}\,\underset{\approx}{C}\right)\underset{\sim}{u}^*_{i} = \underset{\sim}{f}_i + \underset{\sim}{f}_{i+1} \quad . \tag{32}$$

Then subtracting (31) from (32) results in a recursive relation (33) for the error in $\underset{\sim}{u}$ at step N. $(N > i+1)$. Let $\underset{\sim}{e}_N \equiv \underset{\sim}{u}_N - \underset{\sim}{u}^*_N$; then

$$\left(\underset{\approx}{K} + \frac{2}{\Delta t}\,\underset{\approx}{C}\right)\underset{\sim}{e}_N + \left(\underset{\approx}{K} - \frac{2}{\Delta t}\,\underset{\approx}{C}\right)\underset{\sim}{e}_{n-1} = \underset{\sim}{0} \quad , \quad N = i+1,\ i+2,\ \ldots . \tag{33}$$

Solving (33) for $\underset{\sim}{e}_N$ results in

$$\underset{\sim}{e}_N = \left(\underset{\approx}{K} + \frac{2}{\Delta t}\,\underset{\approx}{C}\right)^{-1}\left(-\underset{\approx}{K} + \frac{2}{\Delta t}\,\underset{\approx}{C}\right)\underset{\sim}{e}_{N-1} \equiv \underset{\approx}{\Lambda}\,\underset{\sim}{e}_{N-1} \quad .$$

It follows inductively that

$$\underset{\sim}{e}_N = \underset{\approx}{\Lambda}^n \underset{\sim}{e} \quad , \quad n = N - (i+1)$$

where $\underset{\sim}{e}$ is the error introduced at n = 0. Let λ^* be the absolute value of the largest element of the m×m matrix $\underset{\approx}{\Lambda}$. Then

$$(m\lambda^*)^N \underset{\sim}{e} \geq \underset{\approx}{\Lambda}^N \underset{\sim}{e} \quad .$$

Consider solutions of the form

$$\underset{\sim}{e}_N = (m\lambda^*)^N \underset{\sim}{e} = \lambda^n \underset{\sim}{e} \quad , \quad n = N - (i+1) \tag{34}$$

where λ is a positive constant. The error so defined is greater than or equal to the true error. Substituting (34) into (33),

$$\left[\left(\underset{\approx}{K} + \frac{2}{\Delta t}\,\underset{\approx}{C}\right)\lambda + \left(\underset{\approx}{K} - \frac{2}{\Delta t}\,\underset{\approx}{C}\right)\right]\underset{\sim}{e} = \underset{\sim}{0} \quad ,$$

or, rearranging things slightly,

$$\left[\underset{\approx}{K}(\lambda + 1) - \frac{2}{\Delta t}\,\underset{\approx}{C}\,(1 - \lambda)\right]\underset{\sim}{e} = \underset{\sim}{0} \quad . \tag{35}$$

Defining

$$\omega = \frac{2}{\Delta t}\left(\frac{1-\lambda}{1+\lambda}\right) ,$$

(35) can be written

$$\left[\underset{\approx}{K} - \omega \underset{\approx}{C}\right] \underset{\sim}{e} = \underset{\sim}{0} \quad . \tag{36}$$

If $\underset{\approx}{C}$ is a positive definite matrix, then according to Wilkinson [6, p. 34] the eigenvalues ω are all positive. This, in turn, requires that $|\lambda| < 1$. It follows from (34) that the error will decrease as N increases. Since $|\lambda| < 1$ for <u>any</u> value of $\Delta t > 0$, the solution scheme is unconditionally stable if $\underset{\approx}{C}$ is positive definite. This property of $\underset{\approx}{C}$ is dependent on the assumed forms for ϕ_i, which, as indicated in Section V.1, give a positive definite $\underset{\approx}{C}$ for the forms assumed in the present development.

Section V. FORMULATION OF ELEMENT CONDUCTIVITY, SPECIFIC HEAT, AND OTHER MATRICES

1. Triangular Element Matrices

In the sequel, specialization to a plane Cartesian system will be made in which case the field relations expressed by (6) and (9) reduce to

$$U(x, y, t) = \underset{\sim}{\phi}(x, y)\, \underset{\approx}{A}\, \underset{\sim}{u}(t) \tag{37}$$

and

$$\dot{U}(x, y, t) = \underset{\sim}{\psi}(x, y)\, \underset{\approx}{B}\, \dot{\underset{\sim}{u}}(t) \quad . \tag{38}$$

In the absence of other motivations, it is convenient to let

$$\underset{\sim}{\phi}(x, y) \equiv \underset{\sim}{\psi}(x, y) = \{1, x, y\} \quad , \tag{39}$$

$$\underset{\approx}{\phi}'(x, y) \equiv \underset{\approx}{\psi}'(x, y) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad , \tag{40}$$

and

$$\underset{\approx}{A} \equiv \underset{\approx}{B} \quad .$$

Using the nomenclature of Fig. 1,

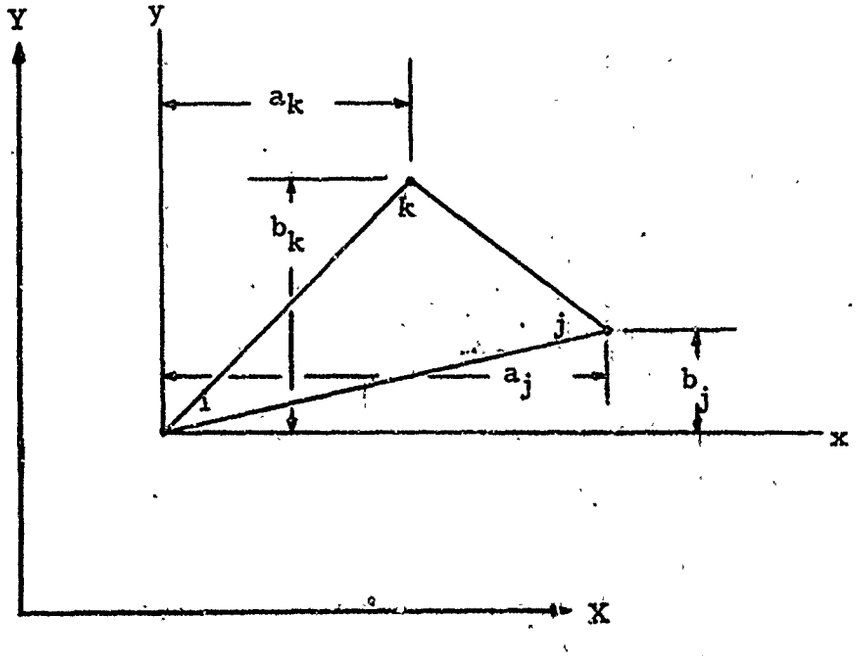


FIG. 1. PLANE TRIANGULAR ELEMENT

the matrix $\underset{\sim}{A}$, from its definition in (37) and (39), can be found to be

$$\underset{\sim}{A} = \frac{1}{a_j b_k - a_k b_j} \begin{bmatrix} a_j b_k - a_k b_j & 0 & 0 \\ b_j - b_k & b_k & -b_j \\ a_k - a_j & -a_k & a_j \end{bmatrix} . \qquad (41)$$

Eqs. (37)-(41) are sufficient to define a linear, spatial, temperature field and a linear, spatial, temperature-rate field in terms of the nodal point values of the temperature and temperature rate, respectively.

Since $\underset{\sim}{A}$ and $\underset{\sim}{\phi}'$ are constant over the element, (13)-(16) may be written

$$\underset{\sim}{K} = \underset{\sim}{A}^T \underset{\sim}{\phi}'^T \underset{\sim}{k} \underset{\sim}{\phi}' \underset{\sim}{A} A \quad , \qquad (42)$$

$$\underset{\sim}{C} = \rho c \underset{\sim}{A}^T \left[\int_A \underset{\sim}{\phi}^T \underset{\sim}{\phi} \, dA \right] \underset{\sim}{A} \quad , \qquad (43)$$

$$Q^* = Q \underset{\sim}{A}^T \int_A \underset{\sim}{\phi}^T \, dA \quad , \qquad (44)$$

and

$$\underset{\sim}{q}^* = \underset{\sim}{A}^T \int_S \underset{\sim}{\phi}^T \underset{\sim}{n} q \, dS \quad , \qquad (45)$$

if $\underset{\sim}{K}$, ρc, and Q are also taken as constant in an element.

These integrals are easily evaluated in terms of the first and second moments of area.

The boundary integrals in (18)-(20) also simplify to

$$\underset{\sim}{\bar{q}}^* = \underset{\sim}{A}^T \int_{S_2} \underset{\sim}{\phi}^T \underset{\sim}{n} \bar{q} \, dS \quad , \qquad (46)$$

$$\underset{\sim}{H} = h \underset{\sim}{A}^T \left[\int_{S_3} \underset{\sim}{\phi}^T \underset{\sim}{\phi} \, dS \right] \underset{\sim}{A} \quad , \qquad (47)$$

and

$$\underset{\sim}{h}^* = h u_0 \underset{\sim}{A}^T \int_{S_3} \underset{\sim}{\phi}^T \, dS \quad . \qquad (48)$$

The coefficient matrices appearing in (29) may now be written

$$\underset{\approx}{K} = \underset{\approx}{A}^T \underset{\approx}{\phi}'^T \underset{\approx}{k} \underset{\approx}{\phi}' \underset{\approx}{A} A - h \underset{\approx}{A}^T \left[\int_{S_3} \underset{\sim}{\phi}^T \underset{\sim}{\phi} \, dS \right] \underset{\approx}{A}$$

$$= \underset{\approx}{A}^T \left[\underset{\approx}{\phi}'^T \underset{\approx}{k} \underset{\approx}{\phi}' A - h \int_{S_3} \underset{\sim}{\phi}^T \underset{\sim}{\phi} \, dS \right] \underset{\approx}{A} \quad , \tag{49}$$

$$\underset{\approx}{C} = \rho c \, \underset{\approx}{A}^T \left[\int_A \underset{\sim}{\phi}^T \underset{\sim}{\phi} \, dA \right] \underset{\approx}{A} \quad , \tag{50}$$

and

$$\underset{\sim}{f} = Q \underset{\approx}{A}^T \int_A \underset{\sim}{\phi}^T dA + \underset{\approx}{A}^T \int_{S_2} \underset{\sim}{\phi}^T \underset{\sim}{n} \bar{q} \, dS - h u_0 \underset{\approx}{A}^T \int_{S_3} \underset{\sim}{\phi}^T dS$$

$$= \underset{\approx}{A}^T \left[Q \int_A \underset{\sim}{\phi}^T dA + \int_{S_2} \underset{\sim}{\phi}^T \underset{\sim}{n} \bar{q} \, dS - h u_0 \int_{S_3} \underset{\sim}{\phi}^T dS \right] . \tag{51}$$

In the development of the computer program ϕ was taken in the linear form

$$\phi(x, y) = \{1, x, y\} \quad ,$$

and

$$\phi'(x, y) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} .$$

In this case, for a triangular element, (50) becomes

$$\underset{\approx}{C} = \frac{\rho c A}{12} \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}$$

which is positive definite as required for stability in Section IV. 2.

2. Quadrilateral Element Matrices

It is convenient in terms of programming logic to work with a quadrilateral element. For this purpose a quadrilateral element composed of four triangular elements, as shown in Fig. 2, was used in the present computer program. The four triangles are determined by defining the coordinates of the common point to be the average of the coordinates of the other four points. The common point is eliminated from explicit representation by the following procedure.

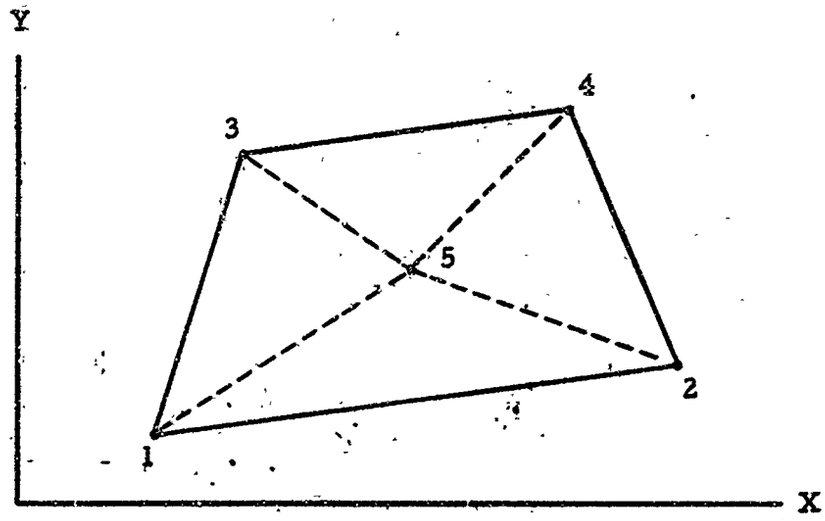


FIG. 2. QUADRILATERAL ELEMENT

The matrix equation for this quadrilateral element can be expressed by assembling the matrix equations for each triangular element by addition of terms at each nodal point in the manner used in the direct stiffness method of structural analysis.[6] Eq. (22) for a quadrilateral element takes the form

$$\begin{bmatrix} K_{11} & K_{12} & K_{13} & K_{14} & K_{15} \\ K_{21} & K_{22} & K_{23} & K_{24} & K_{25} \\ K_{31} & K_{32} & K_{33} & K_{34} & K_{35} \\ K_{41} & K_{42} & K_{43} & K_{44} & K_{45} \\ K_{51} & K_{52} & K_{53} & K_{54} & K_{55} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{bmatrix} + \begin{bmatrix} C_{11} & C_{12} & C_{13} & 0 & C_{55} \\ C_{21} & C_{22} & 0 & C_{24} & C_{25} \\ C_{31} & 0 & C_{33} & C_{34} & C_{35} \\ 0 & C_{42} & C_{43} & C_{44} & C_{45} \\ C_{51} & C_{52} & C_{53} & C_{54} & C_{55} \end{bmatrix} \begin{bmatrix} \dot{u}_1 \\ \dot{u}_2 \\ \dot{u}_3 \\ \dot{u}_4 \\ \dot{u}_5 \end{bmatrix}$$

$$= \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{bmatrix} . \qquad (52)$$

[6] See [5], Section 7.2 for a description of this assembly method.

Rewriting this, there results

$$\begin{bmatrix} [K_{ij}] & [K_{i5}] \\ [K_{5j}] & [K_{55}] \end{bmatrix} \begin{bmatrix} [u_i] \\ [u_5] \end{bmatrix} + \begin{bmatrix} [C_{ij}] & [C_{i5}] \\ [C_{5j}] & C_{55} \end{bmatrix} \begin{bmatrix} [\dot{u}_j] \\ \dot{u}_5 \end{bmatrix} = \begin{bmatrix} [f_j] \\ f_5 \end{bmatrix}, i,j=1,2,3,4 , \tag{53}$$

where $[K_{ij}]$ and $[C_{ij}]$ represent the 4×4 submatrices of the complete matrices in (52) and $[C_{i5}]$ and $[K_{5j}]$ are column and row vectors. The subscripts i, j now represent nodal points, instead of time increments. Eq. (53) can then be written as two equations,

$$[K_{ij}][u_j] + [K_{i5}]u_5 + [C_{ij}][\dot{u}_j] + [C_{i5}]\dot{u}_5 = [f_j] \tag{54}$$

and

$$[K_{5j}][u_j] + K_{55}\, u_5 + [C_{5j}][\dot{u}_j] + C_{55}\, \dot{u}_5 = f_5 \quad . \tag{55}$$

The interior nodal point quantities u_5 and $\dot{u}_5$ cannot be eliminated from (54) by use of (55) as it stands. However, if the specific heat matrix $\underset{\approx}{C}(5 \times 5)$ is approximated by lumping the heat capacities at the four external nodal points, $\underset{\approx}{C}$ becomes a diagonal matrix[7] with $C_{55} = 0$ and (54) can be written

$$[K_{ij}][u_j] + [K_{i5}]u_5 + [C_{ij}][\dot{u}_j] = [f_j] \tag{56}$$

and

$$[K_{5j}][u_j] + K_{55}\, u_5 = f_5 \quad , \tag{57}$$

in which $[C_{ij}]$ is now the (4×4) submatrix of the diagonal-lumped specific heat matrix.

Solving (57) for u_5 and substituting into (56), there results

$$[K_{ij}][u_j] + [K_{i5}]K_{55}^{-1}\{f_5 - [K_{5j}][u_j]\} + [C_{ij}][\dot{u}_j] = [f_i] \tag{58}$$

or

$$\{[K_{ij}] - [K_{i5}]K_{55}^{-1}[K_{5j}]\}\ [u_j] + [C_{ij}][\dot{u}_j] = [f_i] - [K_{i5}]K_{55}^{-1}f_5 . \tag{59}$$

[7] This form of the specific heat matrix is also positive definite.

This equation, (59), above is analogous to (22) except that $\underset{\sim}{K}$ and $\underset{\sim}{C}$ are now 4×4 matrices and $\underset{\sim}{f}$ is a 4×1 vector.

Section VI. COMPUTER PROGRAMS

1. Description

The organization and coding of the present computer programs rely heavily on concepts developed previously in finite element structural analysis programs, particularly those described in [5]. Two programs are described below. AMG042 is the heat-conduction program, and AMG042P is an associated plot program which may be used to aid in reducing the output data to graphical form.

Program AMG042, which has been written to effect the solution of the matrix equations formulated in Section V, is somewhat more restricted than that development. Although the steps necessary to generalize the program are obvious, these are not necessarily trivial. Presently the directions of anisotropy of conductivity of each element must all lie in the same Cartesian frame. Likewise there is no provision for internal heat generation. However the material properties may vary from element to element.

The development has, in general, been applicable to bodies of fairly arbitrary shape. However, the necessity of employing a formal solution method consistent with minimum effort in data input has resulted in some restraints in the computer program. The network of quadrilaterals needed for solution was regularized with a two-dimensional nodal-point identification array, which then provided a systematic framework for solution of the matrix equation. This grid method was first developed for stress analysis purposes, and, although it is described in some detail in Section VI.2, a more comprehensive treatment is given in [6]. Aside from the requirements on grid network, some further restrictions are imposed by the boundary condition subroutines which are described below.

In setting up the program logic, it became obvious that including completely general time-dependent boundary-condition options for arbitrary geometry would be extremely difficult. Therefore, it was decided to handle the boundary conditions by separate short routines to be prepared for each class of problems. The boundary condition subroutines included in this report are written to apply only to a rectangular nodal-point identification array. This does not imply that the program in its present form is limited to a rectangular region.

The sequence of operations of AMG042 is given by the flow chart shown in Fig. 3. The coefficients of the complete matrix equation are assembled from the coefficients of each quadrilateral in a manner analogous to the direct stiffness method of structural analysis. See [6], p. 28, for a more detailed description of the assembly process. Modifications for boundary conditions are made in a similar manner.

2. Mesh Layout and Generation

The requirement of closely approximating the contours of complex regions, together with the desirability of a fine mesh size and its attendant high accuracy, makes the use of a large number of nodal points desirable. The program allows the user to employ a maximum of 496 quadrilateral nodal points. Obviously the layout and specification for the program of the locations of such a number of points is a tedious and time-consuming job in which the probability of human error is high. To minimize this effort and to preserve as much general utility as possible, a scheme for the internal (to the program) generation of much of the required data has been incorporated in the program. This same scheme has been used previously in stress analysis programs [6]. Certain restrictions are imposed on the layout of the nodal points, but the reduction in the effort required to effect the solution of a given problem adequately compensates for these restrictions.

To lay out a nodal-point system for the body to be analyzed, the region of the x-y plane constituting the body is covered (insofar as any curved boundaries will permit) with an array of convex quadrilaterals.[8] Each vertex of a quadrilateral is called a nodal point or node. Each nodal point is identified by an ordered pair of positive integers, denoted by (I, J). The nodes may thus be thought of as a subset of the lattice points in the I-J plane. Nodes with common second member J are said to lie in the same row, although this implies nothing about their location in the x-y plane.

The scheme for mesh generation may be thought of as representing a one-to-one mapping from the I-J plane into the x-y plane. Fig. 4 illustrates this mapping. The points in the I-J plane

[8] The use of a quadrilateral element with a vertex angle greater than 180° may result in erroneous calculations for that element. A vertex angle of 180°, which is acceptable, gives the quadrilateral the appearance of a triangle.

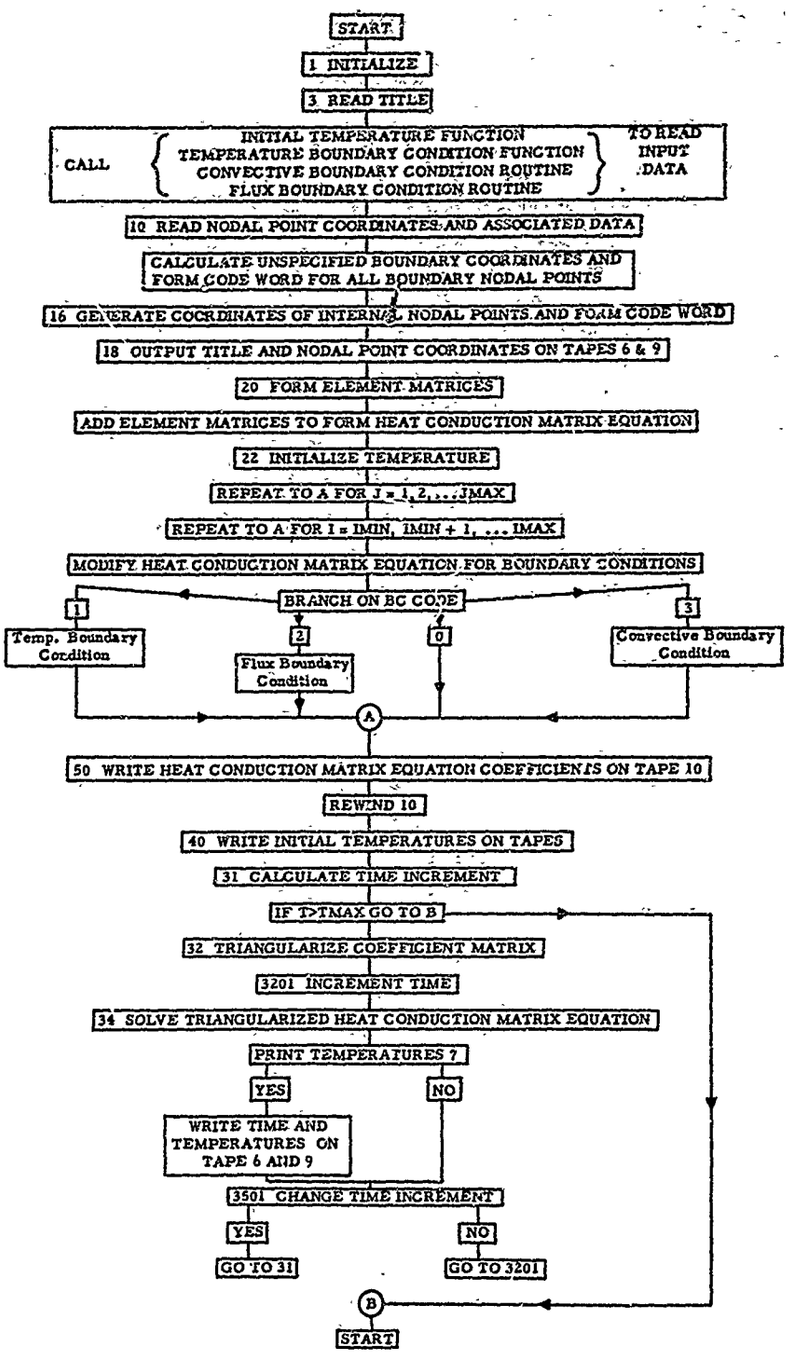


FIG. 3. FLOW CHART FOR AMG042

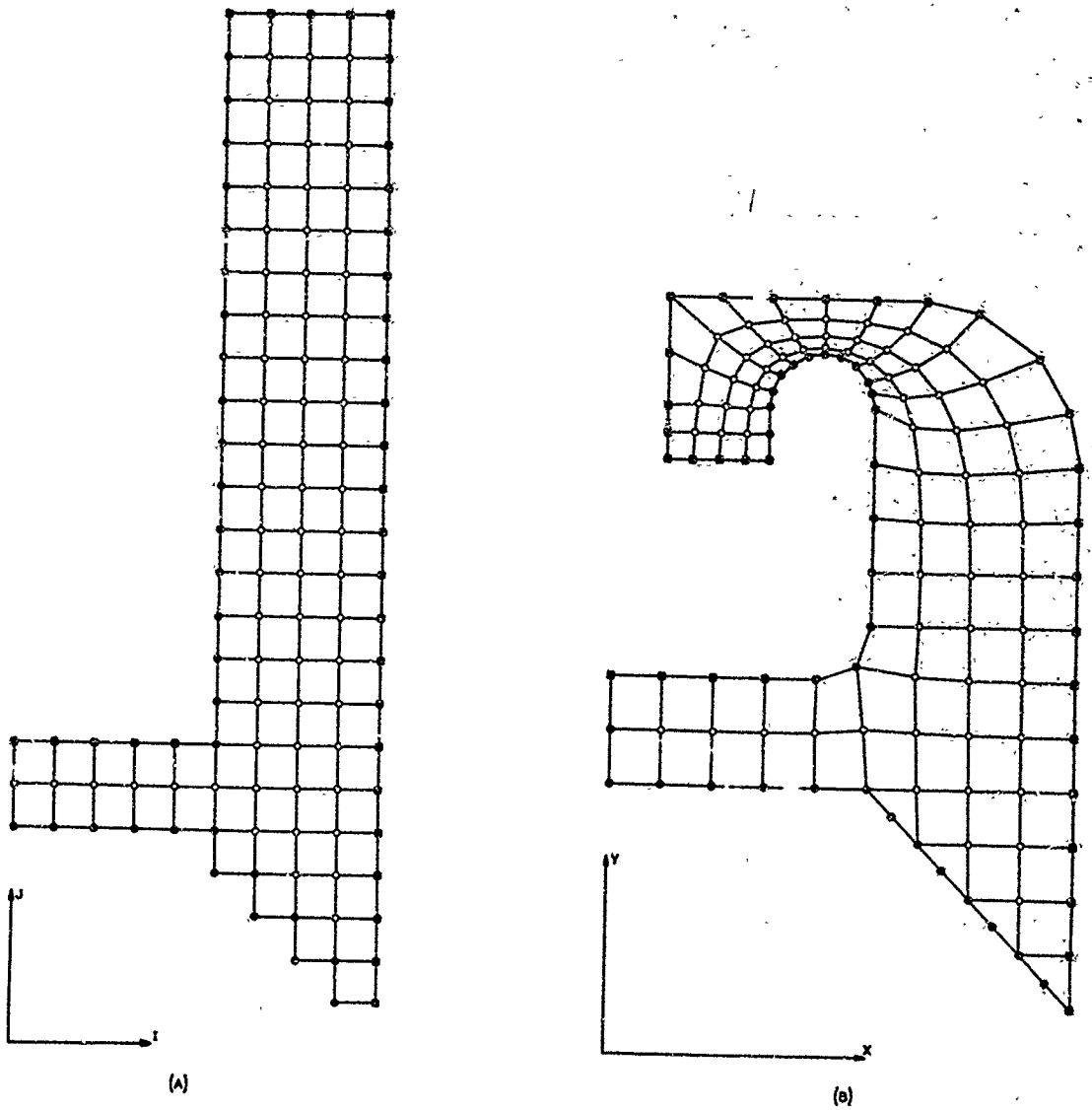


FIG. 4. DEMONSTRATION GRID

are shown in Fig. 4a and their image points in the x-y plane are shown in Fig. 4b. It can be seen that the inverse images of the quadrilaterals in the x-y plane are squares in the I-J plane. Each quadrilateral (or continuum element) is identified by the I, J coordinates of the node whose inverse image lies at the lower left-hand vertex of the inverse image of the quadrilateral. Thus the nodes which are vertices of element (I, J) are the nodes (I, J), (I+1, J), (I, J+1), and (I+1, J+1). It may be noted that not every boundary node need have an element associated with it. In Fig. 4, circles represent nodes associated with elements and squares those which are not associated with any element. The unfilled circles represent nodes whose coordinates were generated by the program.

An important restriction, which is due to the bookkeeping procedure used in the program to assemble the element stiffness into the stiffness for the entire structure, may be phrased thus: if, in any given row, IMIN and IMAX are respectively the least and greatest value of I for which there is a node, then there must be a node in that row for each I such that IMIN $\leq$ I $\leq$ IMAX. For the present program IMAX $\leq$ 16 and JMAX $\leq$ 31. The limiting values of IMAX and JMAX may be varied by changing the appropriate dimensions in the COMMON statements so as to stay within the capacity of the computer. All nodal points that define the boundary must have their coordinates specified, and any other nodal points may either be specified or calculated by the internal generation scheme.

The mesh generation is accomplished in the following manner. A data card containing the values of I, J and the x, y coordinates is input to the computer for each node whose coordinates are to be specified. Such nodes must include at least all nodes on the boundary of the region of interest, as well as on any interfaces between regions of different materials. As many other nodes as the user may desire may have their coordinates specified, but no others are necessary. As the data cards are read, a list is compiled of the minimum and maximum values of I for each J, and each node for which coordinates have been input is identified and the coordinates are stored.

An option is included to permit the input of straight-line segments, corresponding to I = constant or J = constant, which are to be divided into equal increments. The I, J corresponding to the smallest I (or smallest J) is input in the first position on the card, with the I, J corresponding to the largest I (or largest J) being input in the second position. Corresponding x, y coordinates are input into the first and second coordinate positions. The line segment

is internally divided and assigned equally spaced nodal points. Note that TYPE, BCCODE, and IJCODE (described in Appendix A, 1.a) must be the same for all nodal points. If only a single code is to be input, the second I and J positions and the second coordinate positions are left blank. A polar-coordinate input option is also provided.

After all the desired nodal point cards have been input, the coordinates for all unspecified nodes which have I in the interval IMIN < I < IMAX for the proper J, are calculated for all J. The calculation, or mapping, of the coordinates is achieved by solving twice the finite-difference analogue of Laplace's equation on the lattice points in the I-J plane. First, the x coordinates of the boundary points are used as boundary values of the unknown harmonic function, and the functional values obtained on the interior points are taken as the x-coordinates of the corresponding image points in the x-y plane. A similar procedure yields the y-coordinates of the unspecified nodes. It should be noted that, in general, this method tends to yield nodal points with uniform spacing. If this is not deemed desirable, some nodal points interior to the region may have their coordinates specified to control the distribution of the remaining points.

Section VII. ILLUSTRATIVE PROBLEMS

Several problems to illustrate the utility and accuracy of the program have been solved and, when possible, compared with formal solutions. Since these are of the form of illustrations, the problems are posed in dimensionless form wherein any consistent set of units may be inferred. Unless otherwise stated, K, ρ, and c were taken to be unity.

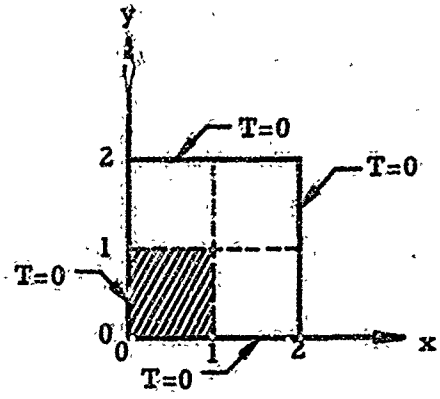
1. Specified Boundary Temperatures

To demonstrate the accuracy of the solution technique, the problem of an isotropic, homogeneous 2×2 square initially at a uniform temperature of 1 with boundaries held at 0 was solved. For a one-quarter symmetric section of the square, a 14×14 grid was used. The resulting temperature distributions along three constant coordinate lines are shown in Fig. 5 and compared with theoretical results from Carslaw and Jaeger [7]. Agreement is quite good.

2. Convective Boundary Condition

The problem of a hollow, circular cylinder with convective boundary conditions was run. Because of the assumed symmetry it was only necessary to run a sector-shape geometry with adiabatic boundary conditions on the straight sides and convective conditions on the inner and outer boundaries. A 45° sector was used, with an inner radius of 0.25 and outer radius of 1. The convective boundary conditions $h(T-T_0)$ used were 35.0 on the inner boundary and 70.0 on the outer boundary. Initial temperature was zero, and the environment temperature was 1.

Fig. 6 illustrates the comparison with the results from a finite-difference program [8]. As can be seen, agreement is essentially perfect. For this particular run, the time increment for the finite element solution was taken as 0.001 while the time increment was 0.000125 for the finite difference solution. When the time increment for the finite element solution was taken 10 times larger, 0.01, the oscillations shown in Fig. 7 occurred. Note that the boundary temperature, as indicated from results with smaller time increments, should have reached over 90% of its total temperature change during the first time increment of 0.01. Despite this crudeness and the resulting oscillations near the boundary, the solution near the center of the slab is fairly accurate for all times and the solution near the boundaries becomes more accurate for increasing time as the oscillation dies away. This is illustrated



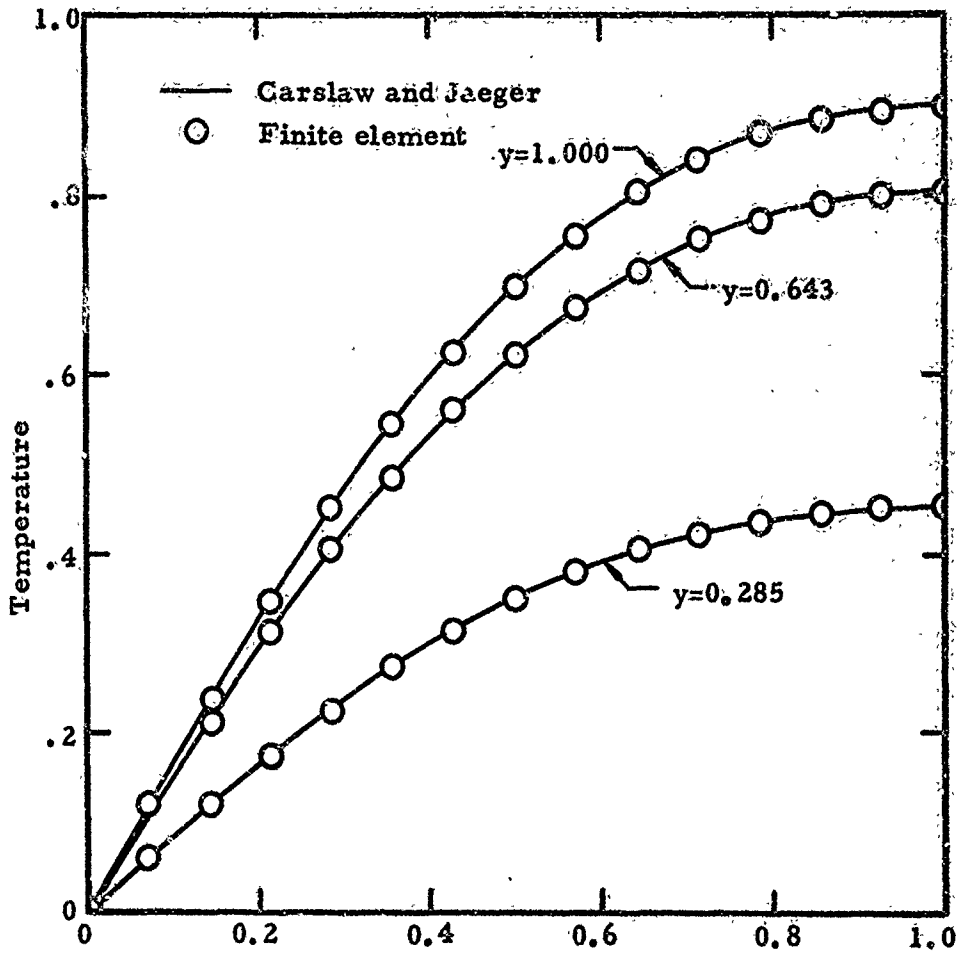


FIG. 5. TEMPERATURE IN A SQUARE

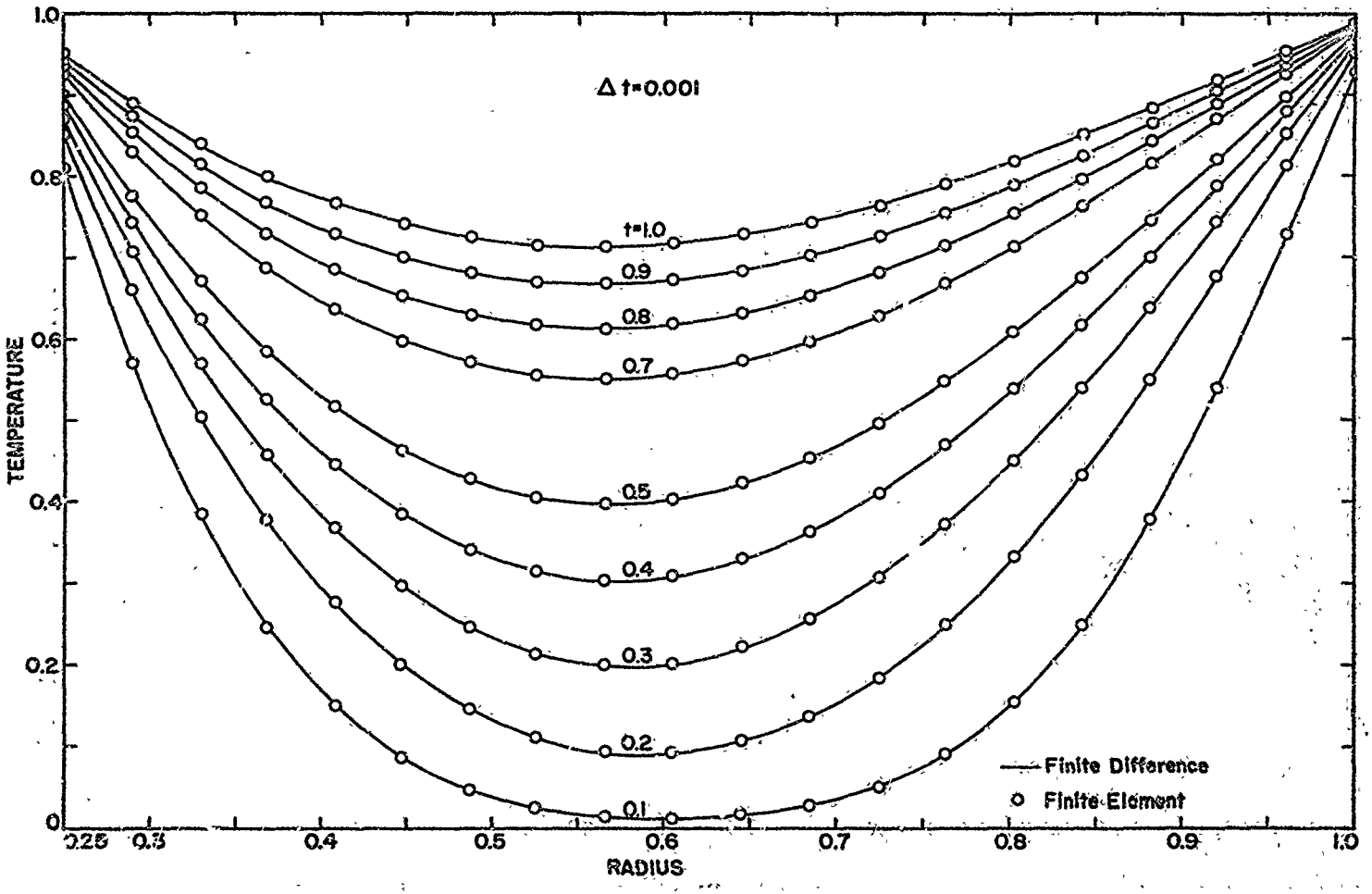


FIG. 6. HEAT CONDUCTION IN A CYLINDER WITH CONVECTIVE BOUNDARY CONDITIONS

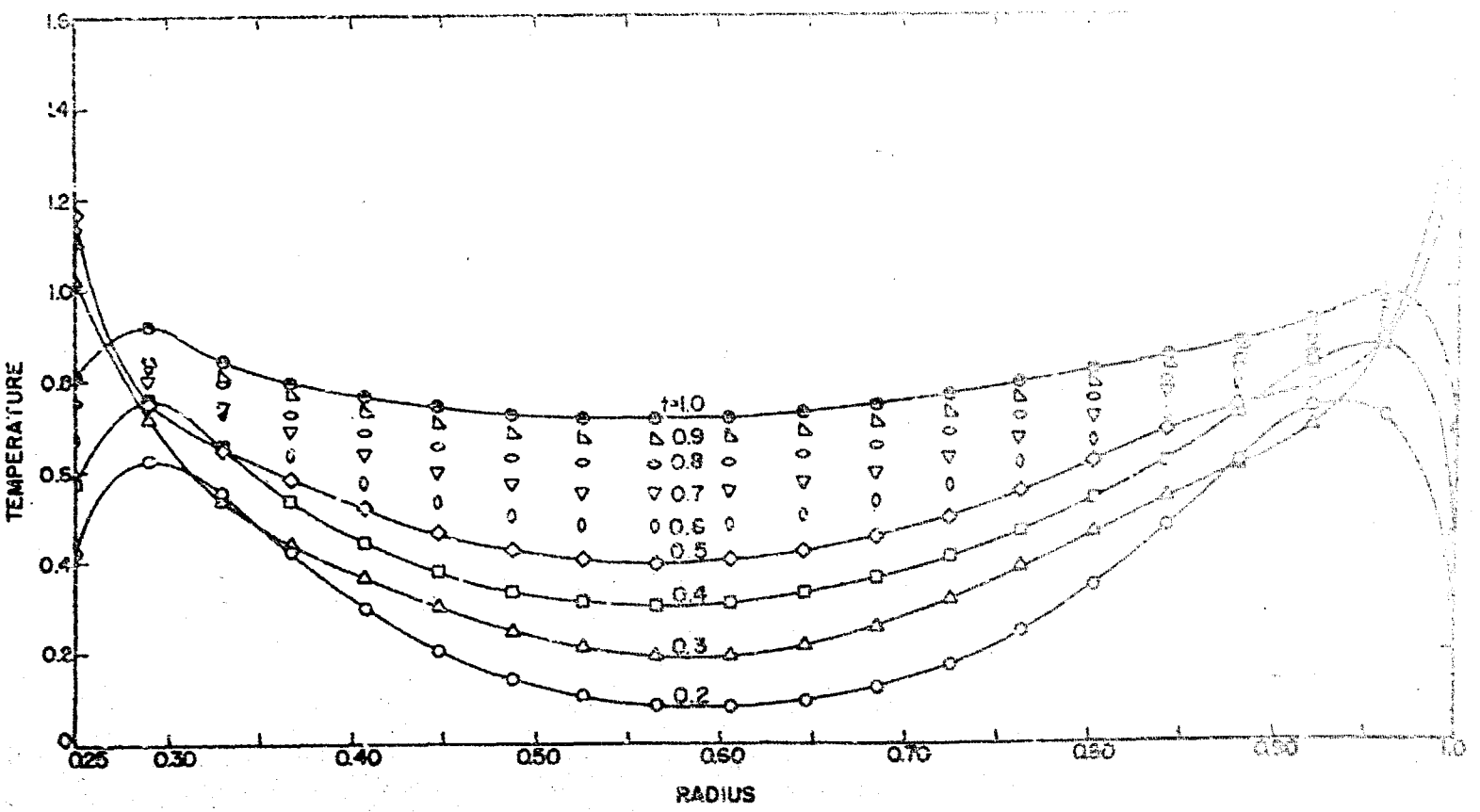


FIG. 7. OSCILLATIONS DUE TO LARGE TIME STEP IN HEATING OF A CYLINDER

in Fig. 8 in which the temperature at r = 0.25, r = 0.60, and r = 1.0 are plotted versus time.

3. Flux Boundary Condition

The behavior of constant-flux boundary conditions were investigated for a rectangular slab with a constant flux supplied at two opposite faces while the other two faces were adiabatic. The slab was initially at a temperature of zero. The results are illustrated in Fig. 9. Only one-half of the slab is illustrated. The center is on the left of the figure. The lines are from the series solution of Carslaw and Jaeger [7].

4. Nonhomogeneous Properties

An axisymmetric cylinder with conductivity and specific heat which vary inversely with radius was studied. Initial temperature of the cylinder was given as zero and the internal and external boundary were subjected to a temperature of 1 at time t = 0. The results are compared in Fig. 10 with the formal solution.

If $K = \frac{K_0}{r}$ and $\rho c = \frac{\rho_0 c_0}{r}$, then for an axisymmetric cylinder the heat-conduction equation becomes

$$\frac{1}{r} \frac{d}{dr} \left(r K \frac{du}{dr} \right) = \rho c \frac{du}{dt}$$

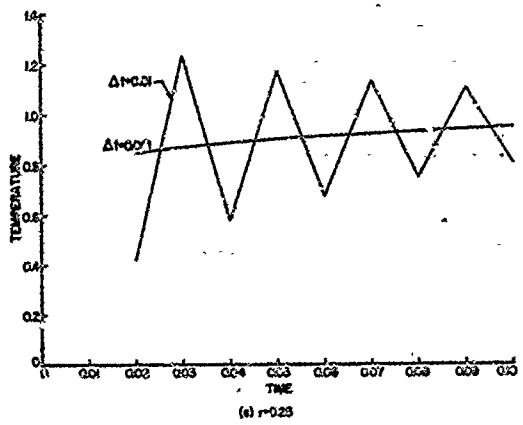
which reduces to

$$K_0 \frac{d^2u}{dr^2} = \rho_0 c_0 \frac{du}{dt}$$

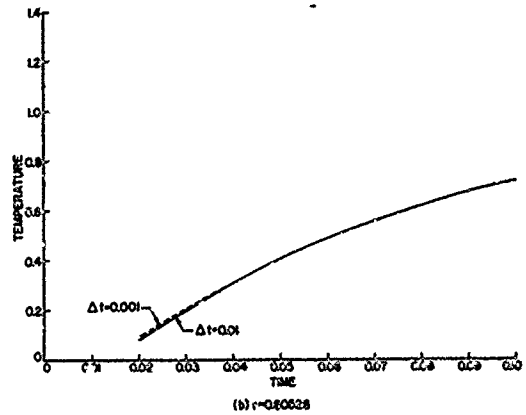
which is the equation for a homogeneous slab. This solution was obtained from [7, p. 101] to plot in Fig. 10. K_0 was taken to be unity and the product $\rho_0 c_0$ was taken to be 5. Agreement with the formal solution is very good despite the crude mesh of nine radial increments.

5. Anisotropic Conductivity

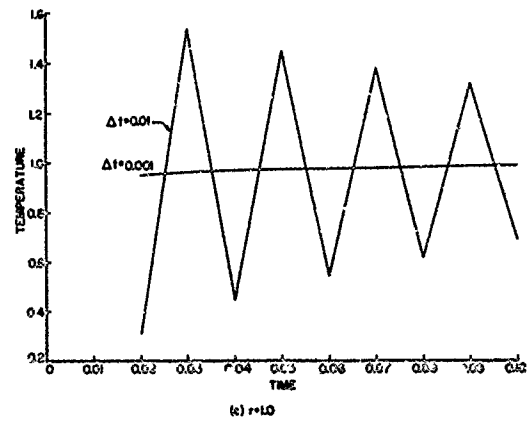
The quadrilateral shown in Fig. 11, with the conductivity in the x-direction equal to 4 times the conductivity in the y-direction, was used to check the anisotropic features of the program. This can be checked with an isotropic body by the following analogy. Let k_x, k_y, x, and y represent the conductivity, and coordinates



(a) r=0.25



(b) r=0.50625



(c) r=1.0

FIG. 8. TEMPERATURE OSCILLATIONS AT VARIOUS LOCATIONS IN CYLINDER

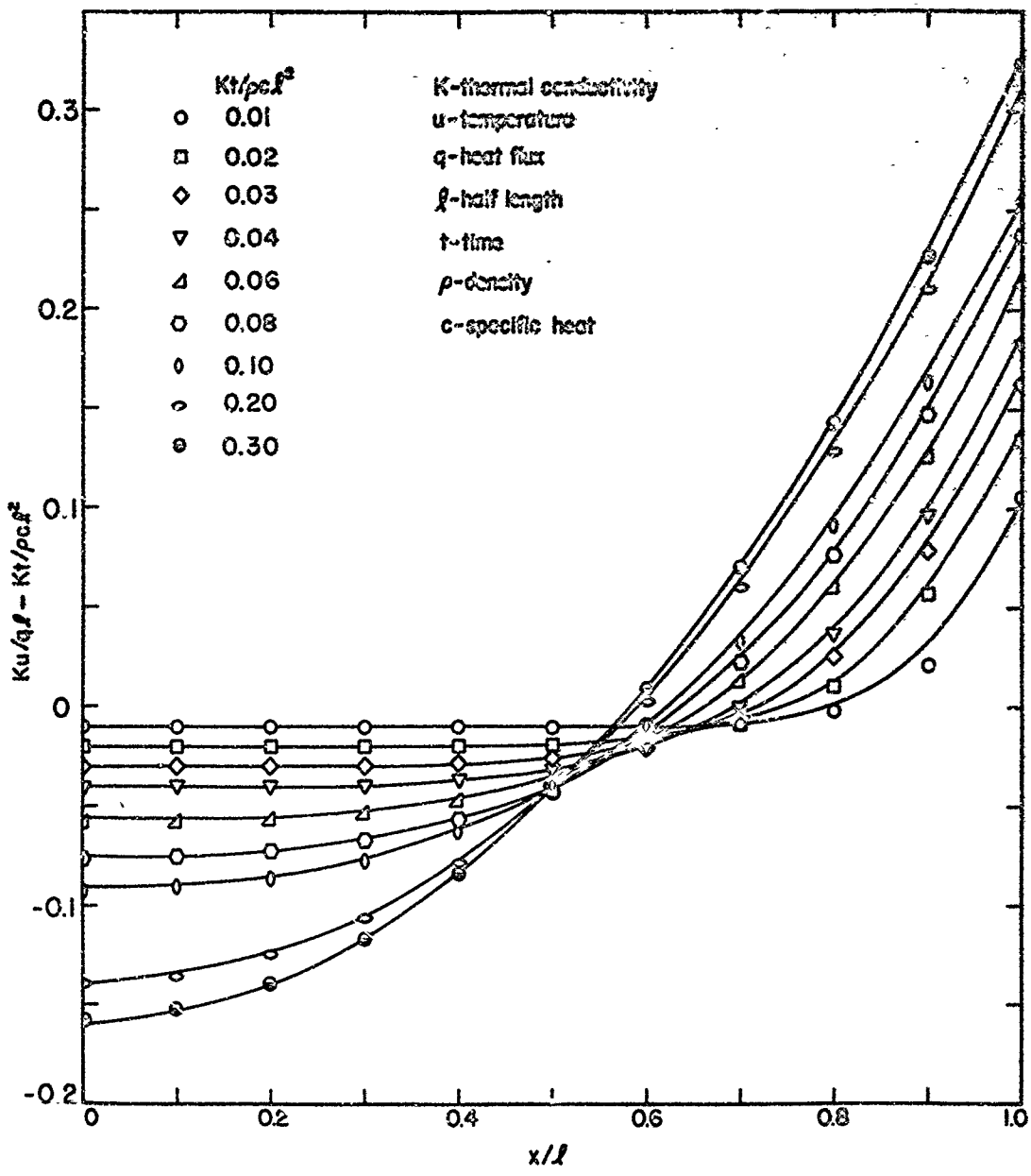


FIG. 9. TEMPERATURE IN A SLAB WITH CONSTANT-FLUX HEAT INPUT AT THE BOUNDARY

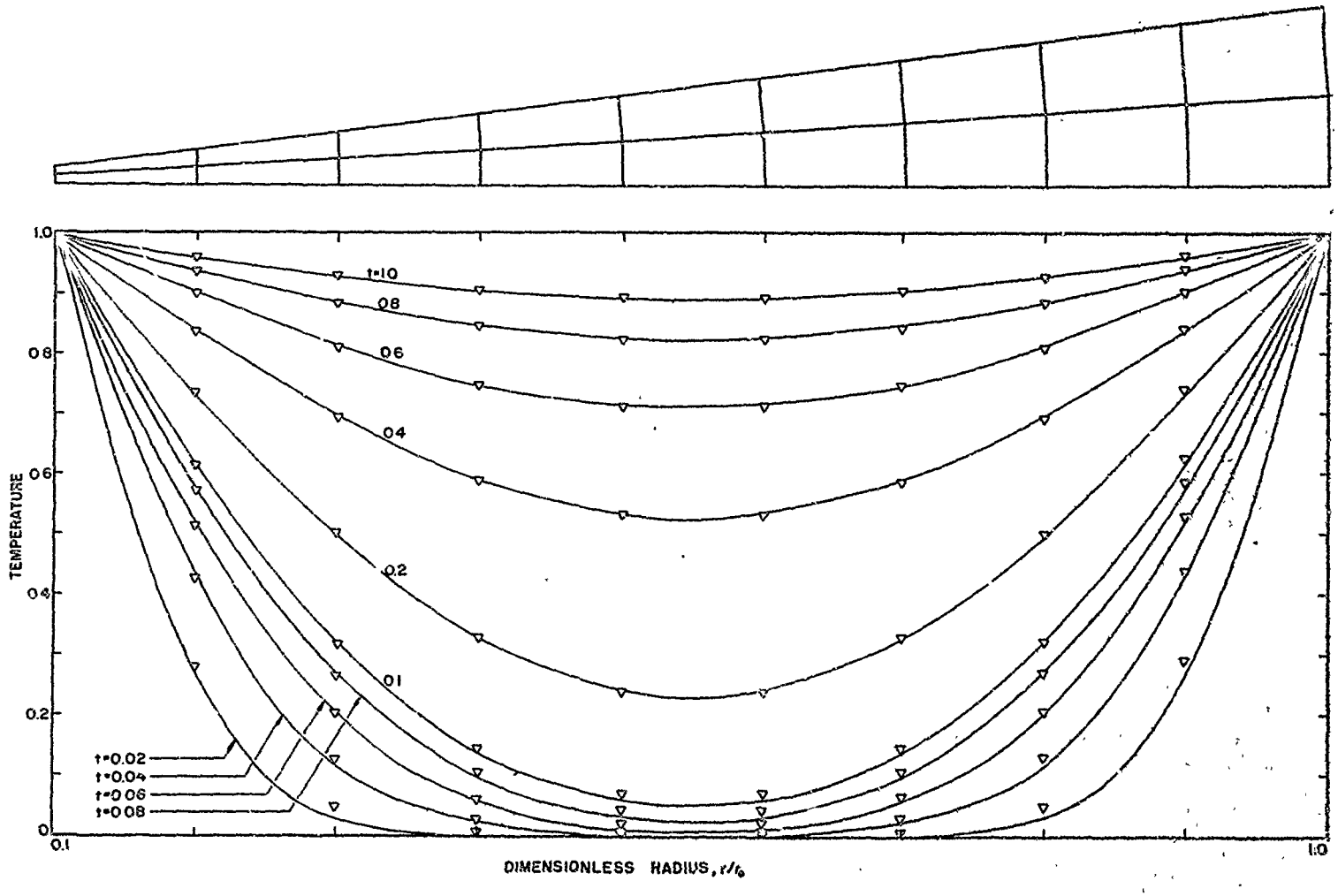


FIG. 10. HEATING OF A CYLINDER WITH NONHOMOGENEOUS PROPERTIES, $K = 1/r$, $\rho c = 5/r$

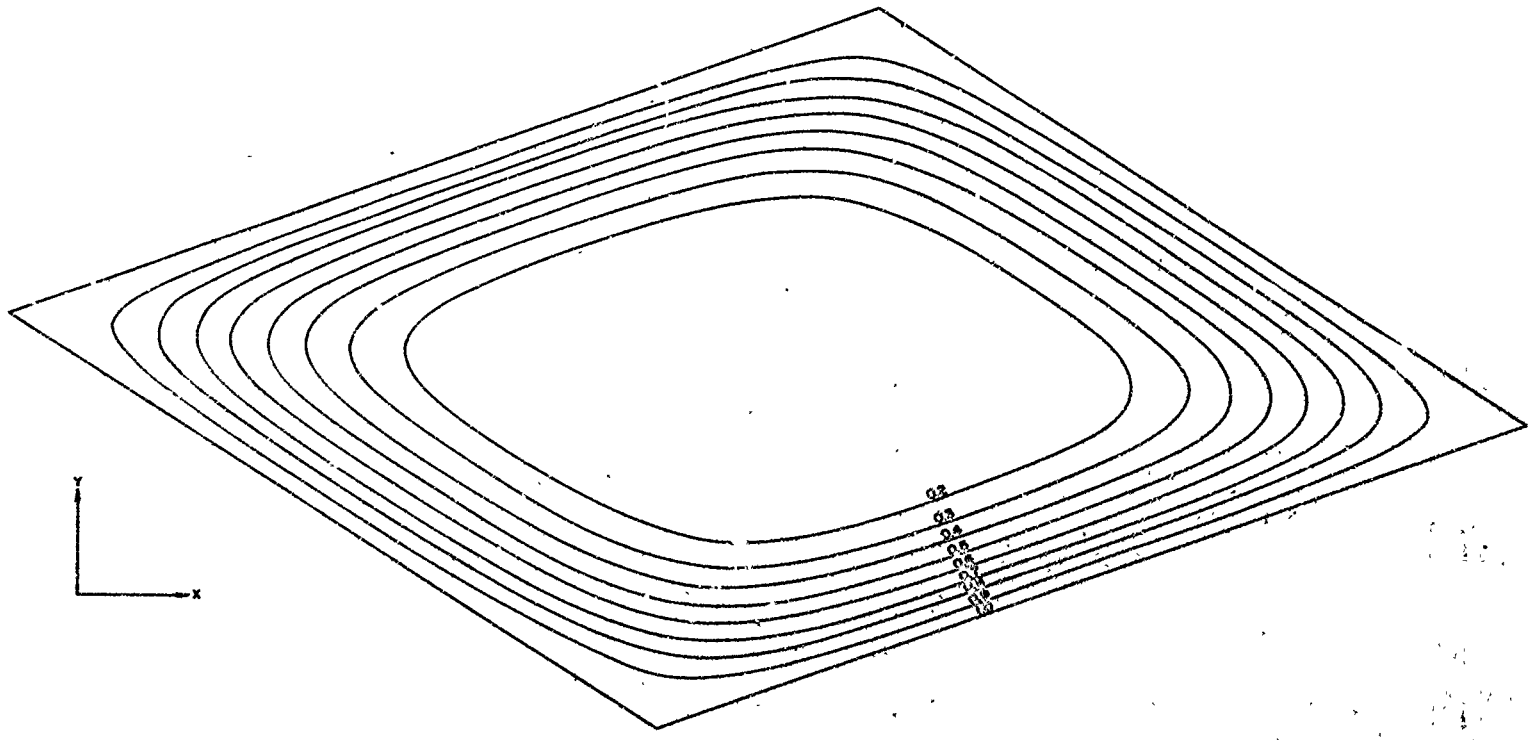


FIG. 11. TEMPERATURE CONTOURS IN AN ANISOTROPIC QUADRILATERAL AT t = 0.05

of the quadrilateral with the condition that $k_x = n^2 k_y$, where n is any positive number. The heat equation

$$k_x \frac{\partial^2 u}{\partial x^2} + k_y \frac{\partial^2 u}{\partial y^2} = \rho c \frac{\partial u}{\partial t}$$

can be rewritten

$$k_y \left[n^2 \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right] = \rho c \frac{\partial u}{\partial t} \quad .$$

If we define a new coordinate $\hat{n}\hat{x} = x$, the equation may be written as

$$k_y \left[\frac{\partial^2 u}{\partial \hat{x}^2} + \frac{\partial^2 u}{\partial y^2} \right] = \rho c \frac{du}{dt} \quad .$$

Thus, the solution of an isotropic problem in $\hat{x}$, y coordinates with isotropic conductivity k_y gives a temperature field similar to an anisotropic problem with $k_x = n^2 k_y$ and $x = \hat{n}x$.

For the present problem, the quadrilateral is, in fact, a square with the x coordinate doubled, and with $k_x = 4\ k_y$. For initial conditions of T = 0 with the boundaries held at T = 1, the transient-conduction problem was worked for both the isotropic square and the anisotropic quadrilateral. The temperature calculated in the two problems agreed to five significant figures. Temperature contours for the quadrilateral are shown in Fig. 11 for time t = 0.05, $k_x = 4$, $k_y = 1.0$, $\rho c = 0.16$. A mesh of 16 x 31 was used.

6. Complex Geometry

As an example of the utility of the program, an example is given in Figs. 12 through 15 which demonstrates its use on a one-sixth symmetric section of a cylinder with a star-shaped perforation subjected to severe convective cooling conditions. The geometry with the internally generated finite element grid is shown in Fig. 12. The initial temperature of the body was T_i and the environment temperature T_0. Results are presented in dimensionless quantities. Isotherms are demonstrated in Fig. 13, and in Figs. 14 and 15, temperature profiles are compared with those of Willoughby [9]. Willoughby's solution is shown in solid lines. Willoughby used a combination of conformal mapping and finite differences to obtain his solution.

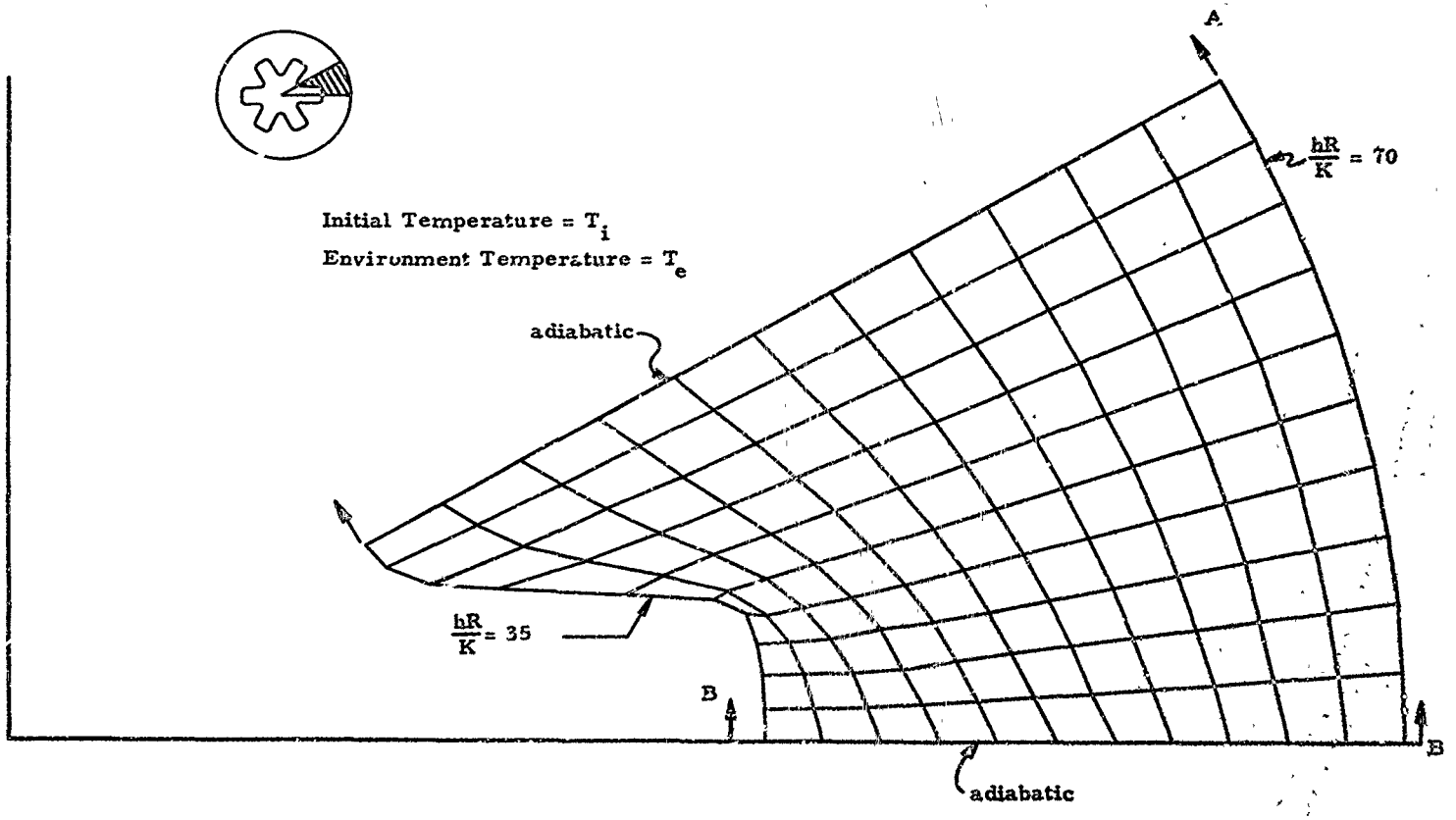


FIG. 12. GRID LAYOUT ON SYMMETRIC SECTION OF SIX-POINT STAR-PERFORATED CYLINDER

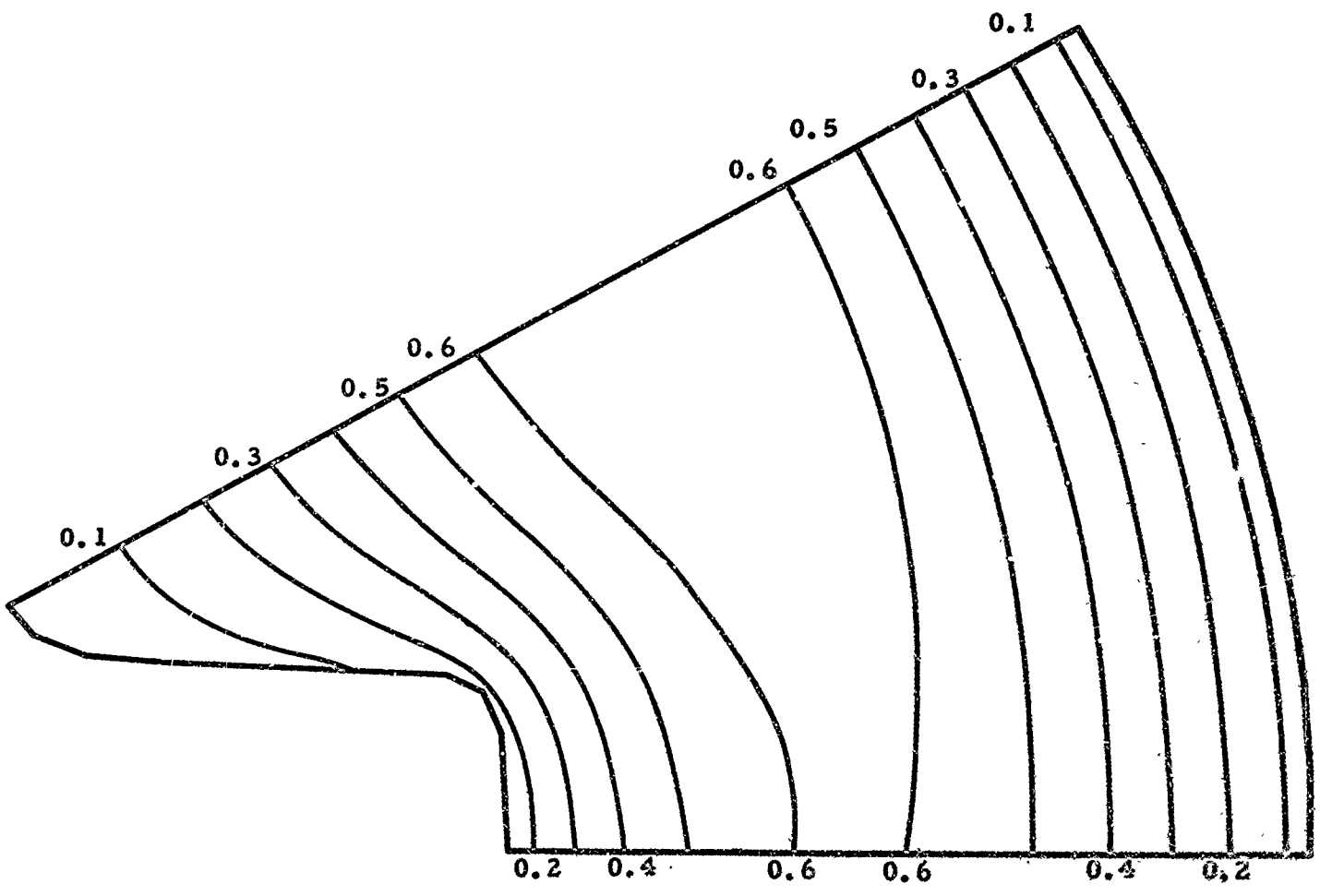


FIG. 13. CONTOURS OF DIMENSIONLESS TEMPERATURE $\theta = (T - T_o/T_i - T_o)$ FOR DIMENSIONLESS TIME $\tau = \alpha t/R^2$ of 0.202

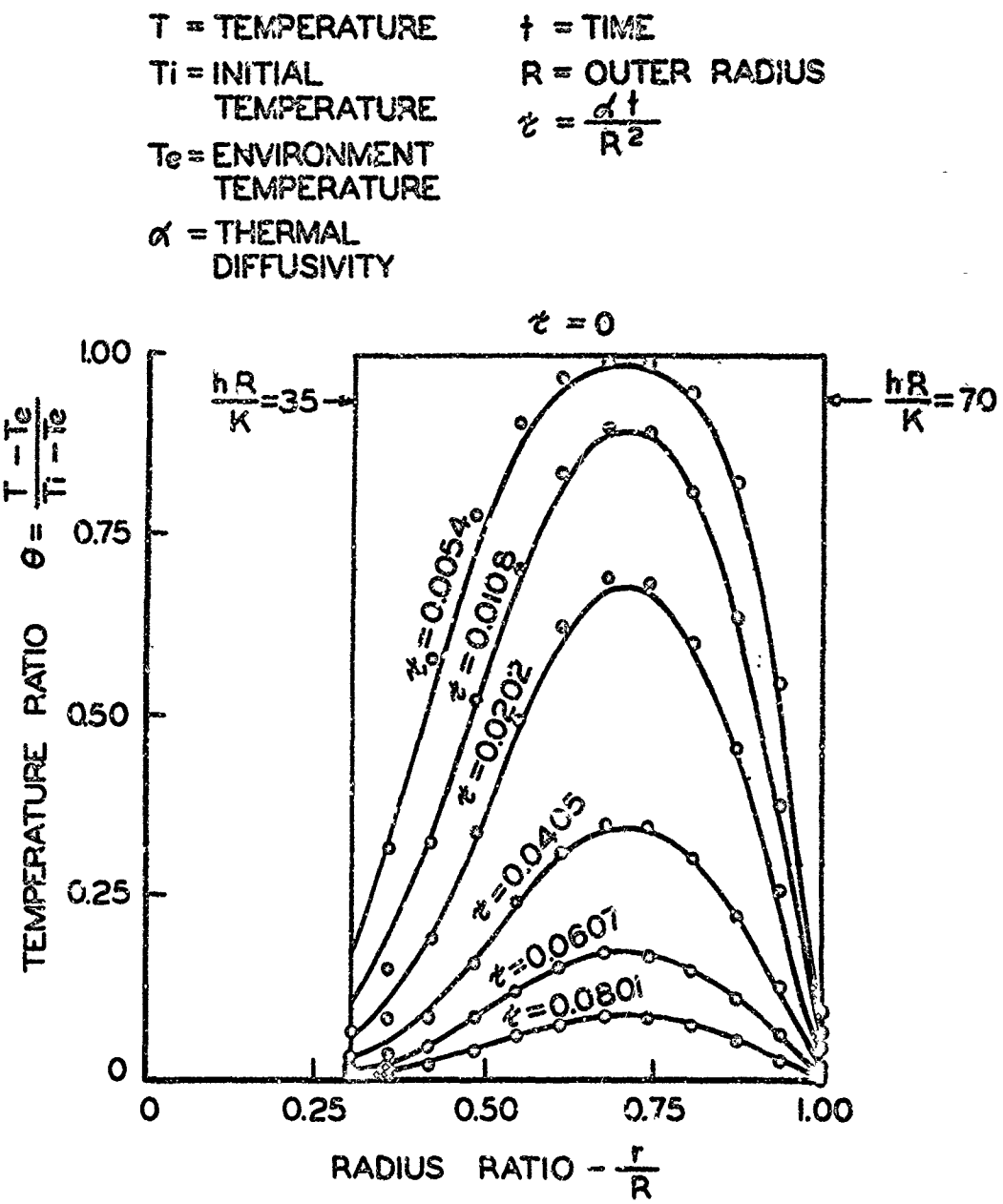


FIG. 14. THERMAL HISTORY COMPARISON AT SECTION A_A

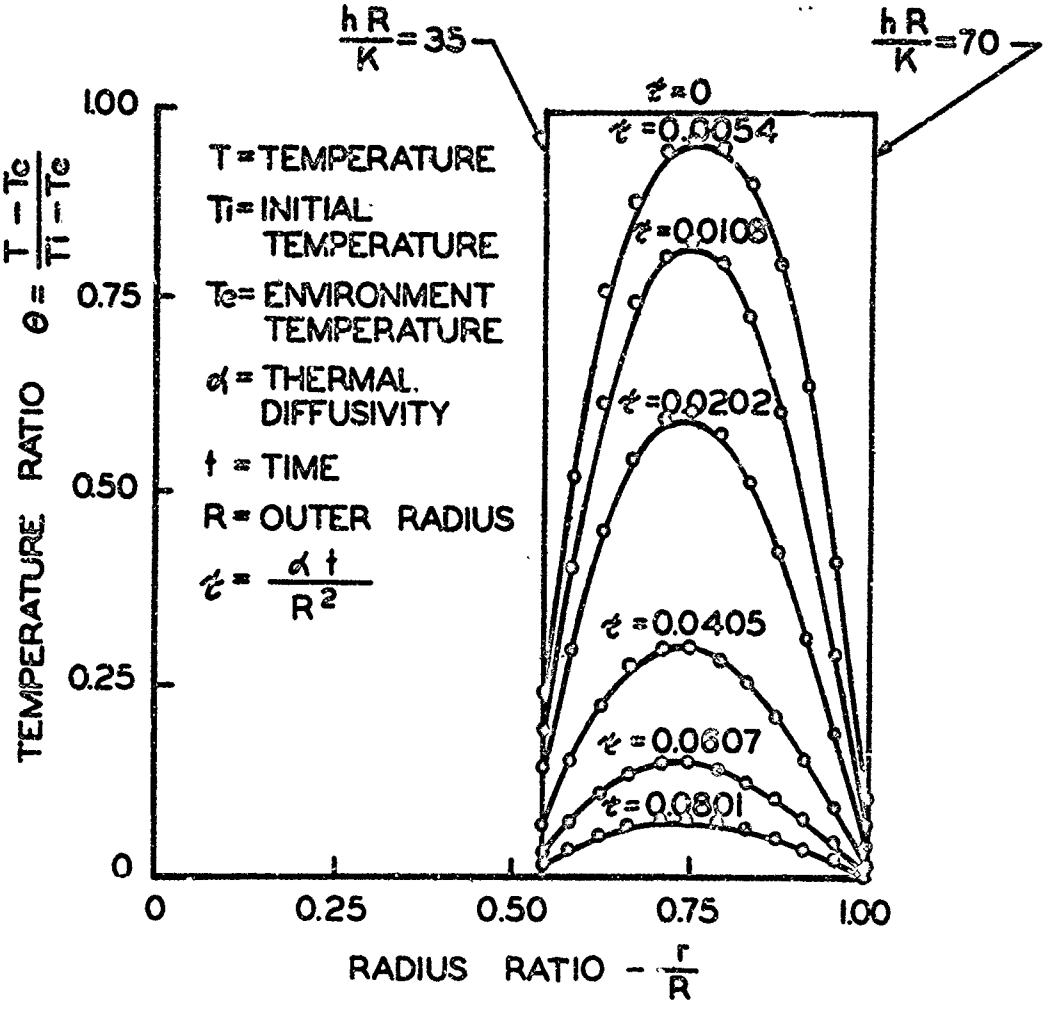


FIG. 15. THERMAL HISTORY COMPARISON AT SECTION B-B

REFERENCES

1. Desinberre, G. M., HEAT TRANSFER CALCULATIONS BY FINITE DIFFERENCES, International Textbook, Scranton, 1961.

2. Donnelly, R. J., Herman, R., and Prigogine, I., NON-EQUILIBRIUM THERMODYNAMICS, VARIATIONAL TECHNIQUES, AND STABILITY, Chicago, University of Chicago Press, 1966.

3. Wilson, E. L. and Nickell, R. E., APPLICATION OF THE FINITE ELEMENT METHOD TO HEAT CONDUCTION ANALYSIS, Nuclear Engineering and Design, 4, 1966, p. 276.

4. Rosen, Philip, ON VARIATIONAL PRINCIPLES FOR IRREVERSIBLE PROCESSES, Journal of Chemical Physics, 21, 1953, p. 1220.

5. Rohm and Haas Company, Huntsville, Alabama, APPLICATION OF THE FINITE ELEMENT METHOD TO STRESS ANALYSIS OF SOLID PROPELLANT ROCKET GRAINS, E. B. Becker and J. J. Brisbane, Report S-76, U. S. Army Missile Command, Redstone Arsenal, Alabama, Contract DA-01-021ORD-12341(Z) and Contract DA-01-021 AMC-11536(Z), Vol. 1, Nov. 18, 1965, AD 474031; Vol. 2, Pt. 1, January 21, 1966, AD 476515; Vol. 2, Pt. 2, January 21, 1966, AD 476735.

6. Wilkinson, J. H., THE ALGEBRAIC EIGENVALUE PROBLEM, Oxford, Clarendon Press, 1965.

7. Carslaw, H. S. and Jaeger, J. C., CONDUCTION OF HEAT IN SOLIDS, Oxford, 1959.

8. Rohm and Haas Company, Huntsville, Alabama, QUARTERLY PROGRESS REPORT ON ENGINEERING RESEARCH, "Fortran Program for the Numerical Solution of Many Coupled Diffusion Processes and Chemical Reactions," W. H. Groetzinger, III, 1963, P-63-6, U. S. Army Missile Command, Redstone Arsenal, Alabama. Contract DA-01-021 ORD-11878(Z) (Confidential).

9. Willoughby, D. A., HEAT CONDUCTION IN STAR PERFORATED SOLID PROPELLANT GRAINS, AIAA Journal, 1965, p. 239.

APPENDIX

PROGRAM OPERATING INSTRUCTIONS

1. Finite Element Heat Conduction Program AMG042

a. Input Data

Initial temperatures must be specified at every point by an appropriately coded subroutine. The present subroutine is only usable for a uniform initial temperature in the body.

Constant temperature, constant flux, convection, or adiabatic conditions may be specified at any of the boundary points. Subroutines BCTEMP (constant temperature), BCCOND (constant flux), BCCONV (convection) for applying a boundary condition on any of four sides of a rectangular nodal point array are listed in this report. The adiabatic condition is imposed by the absence of other boundary conditions.

In the present program, as many as twenty sets of material properties may be specified and assigned to arbitrary elements.

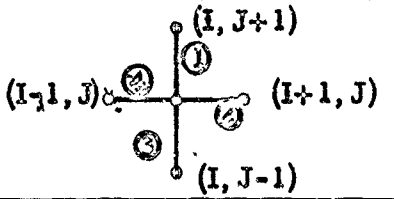
An element is identified by the smallest I and smallest J associated with the nodal points which are its vertices and is said to be associated with the node which is also identified by this I and J. The kind of element associated with a nodal point is specified by a three-digit symbolic word TYPE (see Table I). The last two digits specify the set of thermal properties for the element. The first digit indicates whether or not the coordinates of the nodal point are specified and if an element is associated with that nodal point.

Two additional codes are used for each nodal point to give pertinent information relating to each element. BCCODE (see Table II) branches the program to the correct boundary-condition subroutine. IJCODE (see Table III) indicates the nodal-point line segment to which the boundary condition applies. Only one segment per nodal point may be specified. The words TYPE, BCCODE, and IJCODE are combined internally into a single word, CODE, to conserve storage locations. CODE is output, for checking purposes, with the coordinates of that node. The first three digits of CODE are TYPE, the fourth digit is BCCODE and the fifth digit is IJCODE.

Table I Value and Meaning of Symbolic Word TYPE	
Value	Meaning
X01, X02,...,X20	Identifies the particular set of material properties to be associated with the element. Type X00 is equivalent to X01.
0XX	Coordinates of nodal points are not specified.
1XX	Coordinates of nodal points are specified.
2XX	No element is associated with the corresponding nodal point.

Table II Value and Meaning of Symbolic Word BCCODE	
Value	Meaning
0	Adiabatic (or no boundary condition specified)
1	Temperature specified
2	Flux specified
3	Convection specified

Table III Value and Meaning of Symbolic Word IJCODE	
Value	Meaning
1	Boundary condition is applied on segment
2	
3	
4	



The input data deck is shown in Fig. 16, and the card format is given below.

Card 1 TITLE (12A6)

Col	1-72	Any alphanumeric statement

Card 2 Initial-Temperature Card (F10.5)

Col	1-10	Initial Temperature

Card 3 Boundary-Temperature Card (4F10.5)

Col	1-10	Temperature for Side I = IMIN	Input for specified temperature boundary condition. Zero otherwise.
	11-20	Temperature for Side I = IMAX	
	21-30	Temperature for Side J = 1	
	31-40	Temperature for Side J = JMAX	

Card 4 Boundary-Flux Card (4F10.5)

Col	1-10	Flux for Side I = IMIN	Input for specified boundary flux. Zero otherwise.
	11-20	Flux for Side I = IMAX	
	21-30	Flux for Side J = 1	
	31-40	Flux for Side J = JMAX	

Card 5 Boundary-Convection Card (8F10.5)

Col	1-10	Film coefficient	Side I=IMIN	Input for convective boundary condition. Zero otherwise.
	11-20	Environment temperature		
	21-30	Film coefficient	Side I=IMAX	
	31-40	Environment temperature		
	41-50	Film coefficient	Side J=1	
	51-60	Environment temperature		
	61-70	Film coefficient	Side J=JMAX	
	71-80	Environment temperature		

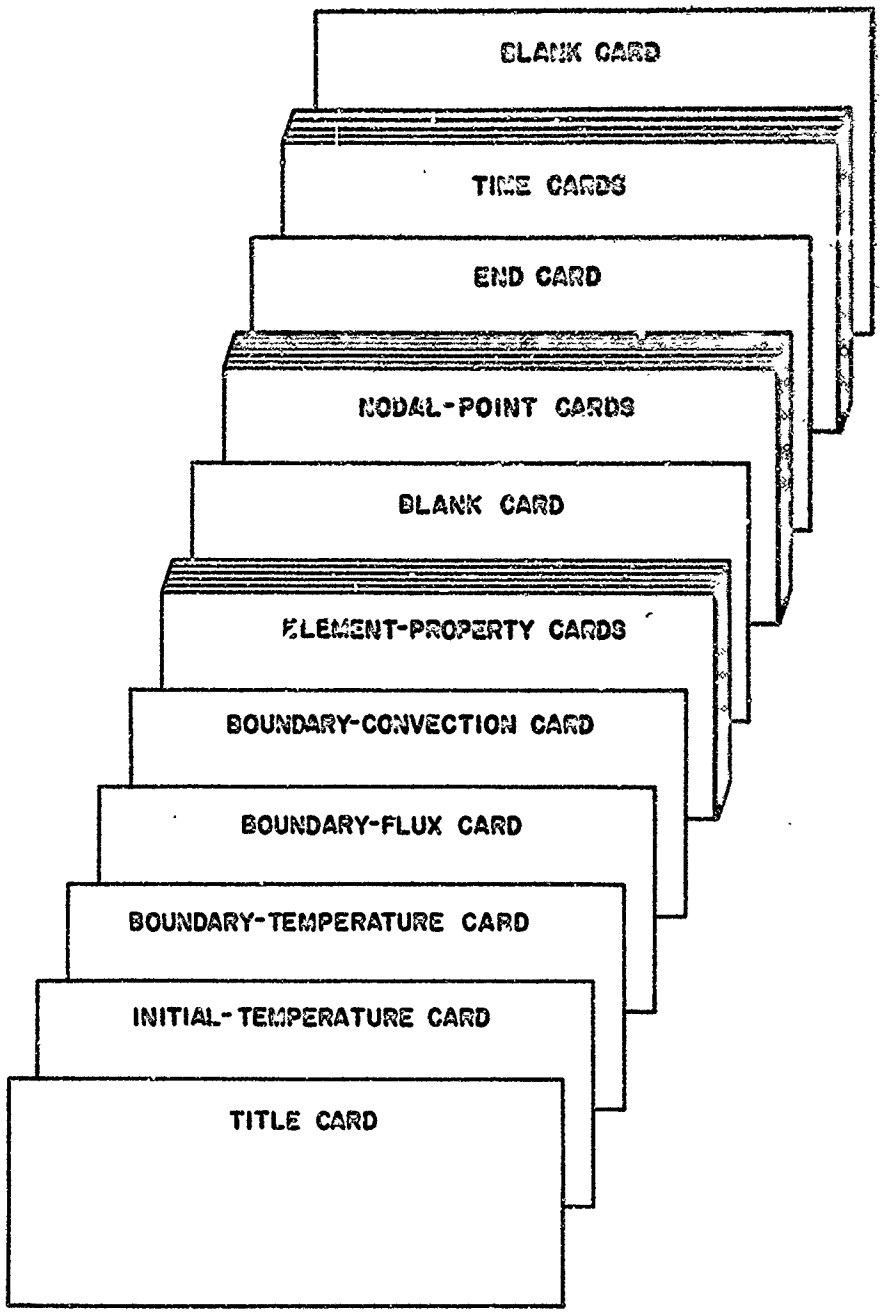


FIG. 16. DATA DECK FOR AMG042

Cards 6 Element Property Cards (5X, I5, 3F10.5)

Col		
	1-5	Blank
	6-10	Identifying number for data set. Range is from 0 to 20. 0 is interpreted internally as 1.
	11-20	Conductivity in x direction.
	21-30	Conductivity in y direction.
	31-40	Product of density and specific heat.

Card 7 Blank Card

Cards 8 Nodal-Point Cards (A5, I2, I3, I2, I3, 4F10.5, 3I5)

Col		
	1-5	Word - POLAR for polar coordinates, blank otherwise.
	6-7	I1 } lowest I (or J) for line segment.
	8-10	J1 }
	11-12	I2 } largest I (or J) for line segment. Zero if only a point is input.
	13-15	J2 }
	16-25	X1 (or R1) } coordinates for lowest I (or J).
	26-35	Y1 (or θ1) }
	36-45	X2 (or R2) } coordinates for largest I (or J).
	46-55	Y2 (or θ2) }
	56 60	TYPE (see Table I).
	61-65	BCCODE (see Table II).
	65-70	IJCODE (see Table III).

Card 9 End Card (I5)

Col		
	1-3	END

Cards 10 Time Cards (F10.3, F5.0, F5.0)

Col		
	1-10	TMAX.
	10-15	Number of time steps from T to TMAX.
	15-20	TOUT (Prints temperatures of T $\geq$ TOUT).

Card 11 Blank Card.

b. Output

Program AMG042 outputs in printed form the coordinates of each nodal point, the word CODE for each nodal point, the value of the boundary condition and nodal points for each type of boundary condition, element property data, and the temperature at each nodal point for each time requested. Tape Unit 6 prepares a tape which can be used as input to Program AMG042P.

c. Program Listing - AMG042

```
$IBFTC MAIN     LIST
CAM6042    FINITE ELEMENT HEAT CONDUCTION PROGRAM
C          PROPERTIES ARE TEMPERATURE INDEPENDENT
C          BOUNDARY CONDITIONS ARE TIME INDEPENDENT
C
C      WRITTEN BY E.B. BECKER
C      ROHM AND HAAS COMPANY
C      DECEMBER 1966
C
C
       COMMON  X(16,31),Y(16,31)
       COMMON  A(496,18),MULT(17,496),B(496)
       COMMON  U(496),UOUT(496),F(496),CODE(16,31)
       COMMON  IMAX(31),IMIN(31),LOC(32),NN(4),JMAX
       COMMON /STIFF/ S3(3,3),S4(5,5),C4(5),XI,XJ,XK,YI,YJ,YK
       EQUIVALENCE    (X,ARRAY)
       INTEGER CODE,BCCODE,TYPE
       LOGICAL EL1,EL2
       REAL MULT
       DIMENSION TITLE(12),ARRAY(1),C(496)
       DATA   END,POLAR/6HEND    ,6HPOLAR /
       DATA XXX/1HX/
C*****     INITIALIZE
       REWIND 9
     1 DO 2  N=1,20832
     2 ARRAY(N)=0
       REWIND 10
       DO 300 N=1,496
   300 C(N) = 0.
       T = 0.
       JMAX = 0
       DO 3  N=1,31
       IMAX(N)=0
       IMIN(N)=16
       DO 3 M=1,31
     3 CODE(M,N)=0
```

```
      READ (5,100)  TITLE
      F1=   UO(0,0)
      F1=   BCTEMP(0,0)
      CALL BCCOND(0,0,0,F1,Q1)
      CALL BCCONV(0,0,0,F1,F2,A1,A2,A3,H1,T2)
      CALL PROP(0,0,XX,XY,CP)
C<<<<<<     READ GRID DATA
   10 READ (5,101) WORD,I1,J1,I2,J2,X1,Y1,X2,Y2,TYPE,BCCODE,IJCODE
      IF (WORD.EQ.END) GO TO 15
      I=I1
      J=J1
      XX=X1
      YY=Y1
      DD= AMAXO(I2-I1,J2-J1)
      JMAX = MAXO(JMAX,J2,J1)
      DX=(X2-X1)/DD
      DY=(Y2-Y1)/DD
   11 IF (COORD.NE.POLAR) GO TO 12
      X(I,J)=XX*COS(YY/57.2957795)
      Y(I,J)=XX*SIN(YY/57.2957795)
      GO TO 13
   12 X(I,J)=XX
      Y(I,J)=YY
   13 CODE(I,J)=100*TYPE+10*BCCODE+IJCODE
      IMIN(J) = MINO(IMIN(J),I)
      IMAX(J) = MAXO(IMAX(J),I)
      IF (I.GE.I2)GO TO 14
      I=I+1
      XX=XX+DX
      YY=YY+DY
      GO TO 11
   14 IF (J.GE.J2) GO TO 10
      J=J+1
      XX=XX+DX
      YY=YY+DY
      GO TO 11
```

```
   15 DO 41 J = 1,JMAX
      I1 = IMIN(J)
      I2 = IMAX(J)
      DO 41 I= I1,I2
      TYPE = CODE(I,J)/10000
      IF (ABS(X(I,J)) + ABS(Y(I,J)) .GT. 0. .AND. TYPE .EQ. 0)
     1 CODE(I,J) = CODE(I,J) + 10000
   41 IF (IMAX(J+1) .LE. I .OR. IMIN(J+1) .GT. I .OR. J .EQ. JMAX
     1 .OR. I .EQ. IMAX) CODE(I,J) = MOD(CODE(I,J),10000) + 20000
C*****    GENERATE NODAL POINTS WHICH WERE NOT INPUT
      IF (JMAX.LE.2) GO TO 18
      J2=JMAX-1
      N = 0
   16 RESID=0.
      N = N + 1
      DO 17 J=2,J2
      I1=IMIN(J)+1
      I2=IMAX(J)-1
      DO 17 I=I1,I2
      IF (CODE(I,J).GE.   10000) GO TO 17
      DX=(X(I+1,J)+X(I-1,J)+X(I,J+1)+X(I,J-1))/4.-X(I,J)
      DY=(Y(I+1,J)+Y(I-1,J)+Y(I,J+1)+Y(I,J-1))/4.-Y(I,J)
      RESID=RESID+ABS(DX)+ABS(DY)
      X(I,J)=X(I,J)+1.86*DX
      Y(I,J)=Y(I,J)+1.86*DY
   17 CONTINUE
      IF (N.EQ.1) RES=RESID
      IF (RESID/RES.GT.1.E-5) GO TO 16
C*****    OUTPUT NODAL POINT COORDINATES
   18 WRITE (6,200) TITLE
      WRITE (9) TITLE,XXX,JMAX,(IMIN(J),IMAX(J),J=1,JMAX)
      DO 19 J=1,JMAX
      I1=IMIN(J)
      I2=IMAX(J)
      DO 19 I=I1,I2
      EL1 = .FALSE.
```

```
      EL2 = .FALSE.
      IF (I+1 .LE. IMAX(J)) EL2 = .TRUE.
      IF (I .GE. IMIN(J+1) .AND. I.LE.IMAX(J+1).AND.J.LT.JMAX)EL1=.TRUE.
      IF (.NOT. EL1 .AND. .NOT. EL2) N=1
      IF (.NOT. EL1 . AND. EL2) N=2
      IF (EL1 .AND. .NOT. EL2) N=3
      IF (EL1 .AND. EL2) N=4
      WRITE (9) X(I,J),Y(I,J),N
   19 WRITE (6,201)  I,J,X(I,J),Y(I,J),CODE(I,J)
      DO 60 M=2,4
      GO TO (67,61,62,63),M
   61 WRITE(6,64)
   64 FORMAT(1H1 20X31HTEMPERATURE BOUNDARY CONDITIONS/3X1HI 4X1HJ 3X20HT
     1EMPERATURE      SIDE )
      GO TO 67
   62 WRITE(6,65)
   65 FORMAT(///20X 30HCONDUCTIVE BOUNDARY CONDITIONS/3X1HI 4X1HJ10X13HG
     1LUX      SIDE )
      GO TO 67
   63 WRITE(6,66)
   66 FORMAT(///20X 30HCONVECTIVE BOUNDARY CONDITIONS/3X1HI 4X1HJ2X39HF
     1LM COEFF    ENVIRONMENT TEMP      SIDE )
   67 DO 60 J=1,JMAX
      I1=IMIN(J)
      I2=IMAX(J)
      DO 60 I=I1,I2
      BCCODE=MOD(CODE(I,J),100)/10+1
      IJCODE=MOD(CODE(I,J),10)
      IF(BCCODE.NE.M) GO TO 60
      GO TO (60,68,69,70),BCCODE
   68 TEMP=BCTEMP(I,J)
      WRITE(6,71)I,J,TEMP,IJCODE
   71 FORMAT(2I5,F13.5,I8)
      GO TO 60
   69 CALL BCCOND(I,J,IJCODE,F1,Q1)
      WRITE(6,71) I,J,Q1,IJCODE
```

```
      GO TO 60
70    CALL BCCONV(I,J,IJCODE,F1,F2,A1,A2,A3,H1,T2)
      WRITE(6,72) I,J,H1,T2,IJCODE
72    FORMAT(2I5,F11.5,F20.5,I7)
60    CONTINUE
C     FORM MATRIX EQUATION
      WRITE (6,202)
      LOC(1)=1
      J1=JMAX+1
      DO 20  J=2,J1
   20 LOC(J)=LOC(J-1)+IMAX(J-1)-IMIN(J-1)+1
      N=0
      DO 21  J=1,J2
      I1=IMIN(J)
      I2=IMAX(J)-1
      DO 21  I=I1,I2
      IF (CODE(I,J).GE.20000)      GO TO 21
      CALL STIFFQ(I,J)
      NN(1)=LOC(J)+I-I1
      NN(2)=NN(1)+1
      NN(3)=LOC(J+1)+I-IMIN(J+1)
      NN(4)=NN(3)+1
      N=MAXO(N,NN(4)-NN(1)+1)
      DO  211 L=1,4
      LN=NN(L)
      C(LN)=C(LN)+CQ(L)
      A(LN,1)=A(LN,1)+SQ(L,L)
      IF (L.EQ.4) GO TO 211
      DO 210  K=L,3
      KN=NN(K+1)-NN(L)+1
      A(LN,KN)=A(LN,KN)+SQ(L,K+1)
  210 CONTINUE
  211 CONTINUE
   21 CONTINUE
   22 DO 23 J = 1,JMAX
      I1 = IMIN(J)
```

```
      I2 = IMAX(J)
      DO 23 I = I1,I2
      L=LOC(J)+I-I1
   23 U(L)=U0(I,J)
      NMAX=LOC(J1)-1
C*****    MODIFICATION OF EQUATIONS FOR BOUNDARY CONDITIONS
      DO 50 J=1,JMAX
      I1=IMIN(J)
      I2=IMAX(J)
C          TEMPERATURE BOUNDARY CONDITION
      DO 50  I=I1,I2
      N=LOC(J)+I-I1
      BCCODE= MOD( CODE(I,J),100)/10  +1
      GO TO   (30,26,28,29                 ),BCCODE
   26 F(N)=BCTEMP(I,J)
      A(N,1) = 1.
      C(N)=0.
      U(N)=F(N)
      DO  27  K=2,M
      N1=N-K+1
      IF(N1.GT.0)F(N1)=F(N1)-A(N1,K)*F(N)
      N2=N+K-1
      IF(N2.LE.NMAX)F(N2)=F(N2)-A(N ,K)*F(N)
      IF (N1.GT.0) A(N1,K)=0.
   27 A(N,K)=0.
      GO TO  30
   28 IJCODE= MOD(CODE(I,J),10)
      CALL BCCOND(I,J,IJCODE,F1,Q1)
      N1=N+1
      IF (IJCODE.EQ.1)  N1=LOC(J+1)+I-IMIN(J+1)
      IF (IJCODE.EQ.3) N1=LOC(J-1)+I-IMIN(J-1)
      IF (IJCODE.EQ.4) N1=N-1
      F(N)=F(N)+F1
      F(N1)=F(N1)+F1
      GO TO  30
   29 IJCODE=MOD(CODE(I,J),10)
```

```
      CALL BCCONV(I,J,IJCODE,F1,F2,A1,A2,A3,H1,T2)
      N1=N+1
      IF (IJCODE.EQ.1)  N1=LOC(J+1)+I-IMIN(J+1)
      IF (IJCODE.EQ.3) N1=LOC(J-1)+I-IMIN(J-1)
      IF (IJCODE.EQ.4) N1=N-1
      F(N)=F(N)+F1
      F(N1)=F(N1)+F2
      A(N,1)=A(N,1)+A1
      A(N1,1)=A(N1,1)+A3
      IF (IJCODE .EQ. 3 .OR. IJCODE .EQ. 4) GO TO 5
      K=1+N1-N
      A(N,K)=A(N,K)+A2
      GO TO 30
    5 K = N - N1 + 1
      A(N1,K) = A(N1,K) +A2
   30 CONTINUE
   50 CONTINUE
      WRITE (10) A,C
      REWIND 10
      WRITE (6,203) T
      WRITE (6,205)
      DO 40 J = 1,JMAX
      I1 = IMIN(J)
      I2 = IMAX(J)
      DO 40 I = I1,I2
      L = LOC(J) + I - I1
   40 WRITE (6,204) I,J,U(L)
C*****    READ DATA FOR TIME INTEGRATION
   31 READ (5,102)  TMAX, STEPS,  TOUT
      IF (TMAX .EQ. 0.) GO TO 1000
      DT=(TMAX-T)/STEPS
      T1 = T + TOUT - .5*DT
C***      TRIANGULARIZE  MATRIX   A+2C/DT
      DT2=2./DT
      READ (10) A,C
      REWIND 10
```

```
      DO 32  N=1,NMAX
   32 A(N,1)=A(N,1)+DT2*C(N)
      CALL  TRIANG(NMAX,M)
C*****    STEP FORWARD IN TIME
 3201 DO 33 N=1,NMAX
   33 B(N)=F(N)+C(N)*(DT2*U(N))
      CALL  TRISOL(NMAX,M)
      DO 34  N=1,NMAX
   34 U(N) = 2.*B(N) - U(N)
      T=T+DT
      IF (T .LT. T1) GO TO 3501
      WRITE (6,203)  T
      K=0
      T1 = T1+TOUT
      WRITE (9) T
      WRITE(6,73)DT,TOUT
      WRITE (6,205)
73    FORMAT(40H TEMPERATURES CALCULATED AT INTERVALS OFF10.5/
     137H TEMPERATURES PRINTED AT INTERVALS OFF13.5/)
      DO  35  J=1,JMAX
      I1=IMIN(J)
      I2=IMAX(J)
      DO 35   I=I1,I2
      K=K+1
   35 WRITE (6,204)  I,J,U(K)
      WRITE (9) U
 3501 IF  (T .LT. TMAX-.5*DT) GO TO 3201
      GO TO 31
 1000 T = -T
      WRITE (9) T
      GO TO 1
  100 FORMAT (12A6)
  101 FORMAT(A5,I2,I3,I2,I3,4F10.5,3I5)
  102 FORMAT(E10.3,F5.0,F5.0)
  200 FORMAT(1H1,12A6///20X31HN O D A L   P O I N T   D A T A //
     14X1HI,4X1HJ,10X1HX,14X1HY,14X4HCODE//)
```

```
201 FORMAT(2I5,2F15.5,I12)
202 FORMAT(1H120X23HE L E M E N T   D A T A  //
   14X1HI4X1HJ7X14HCONDUCTIVITY X7X14HCONDUCTIVITY Y7X13HSPECIFIC HEAD
   2     )
203 FORMAT(35H1 TEMPERATURE DISTRIBUTION AT TIME=F10.5//)
205 FORMAT(4X1HI4X1HJ15X11HTEMPERATURE // )
204 FORMAT(2I5,1PE30.5)
206 FORMAT(1H1,12A6)
207 FORMAT (3I10)
    END
```

```
$IBFTC UO
C          INITIAL CONDITIONS
      FUNCTION   UO(I,J)
      IF (I.EQ.0) READ (5,1) T
      UO=T
    1 FORMAT (F10.5)
      RETURN
      END
```

```
$IBFTC BCTEM
      FUNCTION  BCTEMP(I,J)
      COMMON  DUM(20832),IMAX(31),IMIN(31),B(36),JMAX
      DIMENSION T(4)
      IT=1
      IF (I.EQ.0) GO TO  1
      IF (J.EQ.1) IT=3
      IF (J.EQ.JMAX) IT=4
      IF (I.EQ.IMAX(J)) IT=2
      BCTEMP=T(IT)
      RETURN
    1 READ (5,2) T
    2 FORMAT (4F10.5)
      RETURN
      END
```

```
$IBFTC BCCOND
C          CONDUCTIVE BOUNDARY CONDITION
      SUBROUTINE BCCOND(I,J,IJ,F1,Q1)
      COMMON  X(16,31),Y(16,31),A(1984C),IMAX(31),IMIN(31),B(36),JMAX
      DIMENSION  Q(4),IJ1(4),JI1(4)
      DATA  IJ1/ 0,1,0,-1/, JI1/ 1,0,-1,0/
      K=1
      IF (I .EQ.0) GO TO 1
      IF (J.EQ.1)  K=3
      IF (J.EQ.JMAX) K=4
      IF (J.EQ.IMAX(J)) K=2
      I1=I+IJ1(IJ)
      J1=J+JI1(IJ)
      F1=Q(K)*SQRT((X(I,J)-X(I1,J1))**2+(Y(I,J)-Y(I1,J1))**2)/2.
      Q1=Q(K)
      RETURN
    1 READ (5,2) Q
    2 FORMAT (4F10.5)
      RETURN
      END
```

```
$IBFTC BCCONV
C          CONVECTIVE BOUNDARY CONDITIONS
      SUBROUTINE BCCONV(I,J,IJ,F1,F2,A1,A2,A3,H1,T2)
      COMMON  X(16,31),Y(16,31),A(19840),IMAX(31),IMIN(31),B(36),BMAX
      DIMENSION  H(4),T(4),IJ1(4),JI1(4)
      DATA  IJ1/0,1,0,-1/,JI1/ 1,0,-1,0/
      K=1
      IF (I.EQ.0) GO TO 1
      IF (J.EQ.1) K=3
      IF (J.EQ.JMAX) K=4
      IF (I.EQ.IMAX(J)) K=2
      I1=I+IJ1(IJ)
      J1=J+JI1(IJ)
      H1=H(K)
      T2=T(K)
      HL=H(K)*SQRT((X(I,J)-X(I1,J1))**2+(Y(I,J)-Y(I1,J1))**2)
      F1=T(K)*HL/2.
      F2=F1
      A1=HL/3.
      A2=HL/6.
      A3=A1
      RETURN
    1 READ (5,2)  (H(N),T(N),N=1,4)
    2 FORMAT (8F10.5)
      RETURN
      END
```

```
$IBFTC PROP
C          ELEMENT PROPERTIES
      SUBROUTINE   PROP(I,J,KX,KY,CP)
      COMMON  X(20336),CODE(16,31)
      REAL KX,KY
      INTEGER  CODE
      DIMENSION            XK(20),YK(20),PC(20)
      DATA  XK,YK,PC/ 60*0.0/
      IF  (I.EQ.0) GO TO 1
      N = MOD(CODE(I,J),10000)/100
      IF (N .EQ. 0) N = 1
      KX=XK(N)
      KY=YK(N)
      CP=PC(N)
      RETURN
    1 READ (5,2) N,KX,KY,CP
      IF (N.EQ.0)  RETURN
      XK(N)=KX
      YK(N)=KY
      PC(N)=CP
      GO TO 1
    2 FORMAT (5XI5,3F10.5)
      END
```

```
$IBFTC TRISO
      SUBROUTINE  TRISOL(NMAX,M)
      COMMON  X(992),A(496,18),MULT(17,496),B(496)
      REAL  MULT
      DO  1  N=1,NMAX
      B(N)=B(N)/A(N,1)
      I1=N+1
      I2=MINO(N+M-1,NMAX)
      DO  1 I=I1,I2
      NI=I-N
    1 B(I)=B(I)-MULT(NI,N)*B(N)
      N1=NMAX-1
      DO 2 N=1,N1
      NI=NMAX-N
      I2=MINO(N+1,M)
      DO 2 I=2,I2
      IN=NI+I-1
    2 B(NI)=B(NI)-A(NI,I)*B(IN)
      RETURN
      END
```

```
$IBFTC TRIAN
      SUBROUTINE TRIANG(NMAX,M)
      COMMON  X(992),A(496,18),MULT(17,496),B(496)
      REAL MULT
      N1=NMAX-1
      DO 3  N=1,N1
      I1=N+1
      I2=MINO(M,NMAX-N+1)
      DO 1  I=2,I2
    1 MULT(I-1,N)=A(N,I)
      DO 2  I=2,I2
    2 A(N,I)=A(N,I)/A(N,1)
      DO 3  I=2,I2
      DO 3  J=2,I
      NI= N+J-1
      NJ=I-J+1
    3 A(NI,NJ)=A(NI,NJ)-A(N,I)*MULT(J -1,N)
      RETURN
      END
```

```
$IBFTC STIFQ
C          PLANE,FOUR-TRIANGLE,TAPEZIUM ELEMENT STIFFNESS
      SUBROUTINE  STIFFQ(I,J)
      COMMON  X(16,31),Y(16,31)
      COMMON /STIFF/ S3(3,3),S4(5,5),C4(5),XI,XJ,XK,YI,YJ,YK
      DIMENSION  XX(5),YY(5),IT(3,4)
      REAL KX,KY
      DATA     IT/ 1,2,5,2,4,5,4,3,5,3,1,5/
      DO 3    N=1,5
      DO 4    M=1,5
    4 S4(N,M)=0.
    3 C4(N)=0.
      CALL PROP(I,J,KX,KY,CP)
      XX(1)=X(I,J)
      XX(2)=X(I+1,J)
      XX(3)=X(I+1,J+1)
      XX(4)=X(I,J+1)
      XX(5)=XX(1)
      YY(1)=Y(I,J)
      YY(2)=Y(I+1,J)
      YY(3)=Y(I+1,J+1)
      YY(4)=Y(I,J+1)
      YY(5)=YY(1)
      XK=(XX(1)+XX(2)+XX(3)+XX(4))/4.
      YK=(YY(1)+YY(2)+YY(3)+YY(4))/4.
      DO  2 N=1,4
      XI=XX(N)
      XJ=XX(N+1)
      YI=YY(N)
      YJ=YY(N+1)
      CALL  STIFF3(KX,KY,CP,A)
      DO  1  M=1,3
      M1=IT(M,N)
      DO  1   L=1,3
      L1=IT(L,N)
    1 S4(M1,L1)=S4(M1,L1)+S3(M,L)
```

```
      M1=I (1,N)
      L1=J (2,N)
      C4(M1)=C4(M1)+CP*A
    2 C4(L1)=C4(L1)+CP*A
      DO 6 N = 1,4
      DO 5 M = 1,4
    5 S4(N,M) = S4(N,M)-S4(N,5)*S4(N,5)/S4(5,5)
    6 C4(N) = C4(N)-S4(N,5)*C4(5)/S4(5,5)
      WRITE (6,100)  I,J,KX,KY,CP
  100 FORMAT (2I5,1P3E20.7)
      RETURN
      END
```

```
$IBFTC STIF3
C          PLANE TRIANGULAR ELEMENT STIFFNESS
      SUBROUTINE   STIFF3(KX,KY,CP,A)
      COMMON  /STIFF/ S3(3,3),S4(5,5),C4(15),XI,XJ,XK,YI,YJ,YK
      REAL KX,KY
      AJ=XJ-XI
      AK=XK-XI
      BJ=YJ-YI
      BK=YK-YI
      AA=AJ*BK-AK*BJ
      IF (AA .LE. 0.) WRITE (6,100)
      A=AA/4.
      AA=AA+AA
      S3(1,1)=(KX*(BJ-BK)**2+KY*(AK-AJ)**2)/AA
      S3(1,2)=(KX*BK*(BJ-BK)-KY*AK*(AK-AJ))/AA
      S3(1,3)=(-KX*BJ*(BJ-BK)+KY*AJ*(AK-AJ))/AA
      S3(2,2)=(KX*BK*BK +KY*AK*AK)/AA
      S3(2,3)=(-KX*BJ*BK-KY*AK*AJ)/AA
      S3(3,3)=(KX*BJ*BJ+KY*AJ*AJ)/AA
      DO  1  I=2,3
      DO  1  J=1,2
    1 S3(I,J)=S3(J,I)
      RETURN
  100 FORMAT (45HOTRIANGULAR ELEMENT HAS ZERO OR NEGATIVE AREA)
      END
```

```
$IBMAP UNITS
       ENTRY    .UN09.
       ENTRY    .UN10.
.UN09. PZE      UNIT09
.UN10. PZE      UNIT10
UNIT09 FILE     ,B(2),READY,OUTPUT,BIN,HOLD          B-6
UNIT10 FILE     ,UT4,READY,INOUT,BIN                 B-3
       END
```

2. Plot Program AMG042P

To assist in interpreting the output from program AMG042, a subsidiary program AMG042P was devised. This plot program used a magnetic tape prepared by AMG042, along with conventional card data input, to perform the following functions.

(1) The array of nodal point temperatures is scanned, and the coordinates of points on selected temperature contours are determined by linear inverse interpolation. The coordinates are printed and also put on tape for use with the plotter.

(2) The values of temperature along any specified coordinate line are calculated and printed and also put on a tape for use with the plotter.

The user has the option of obtaining all of the above information in printed form and/or having these plotted by the Electronic Associates, Inc., 3440 Dataplotter. The title is written at the bottom of the plot sheet (30-in. x 30-in.) beginning four letter heights from the bottom. Allowance must be made for this in specifying the board and data offsets on the plotter control card (Card 2). Each plot requires a separate sheet.

There are some items on the plotter control card (Card 2), such as board offset and data offset, which are difficult to explain briefly. Those who use this plotter will find sufficient explanation in the plotter literature. Those who have another plotter available can adapt this program to their particular needs. If no plotter is available, the tape-writing instructions should be removed from the program and the printed output can be used for manual plotting. Only IPRINT need be specified on Card 2 if a plotter is unavailable.

a. Input Data

The following input data must be included along with the tape created by logical unit 6 in the temperature-calculation program AMG042. AMG042P uses logical unit 11 to read the input tape and writes an output tape on logical unit 12.

The input data deck setup is shown in Fig. 17 and the card format is given below.

Card 1	TITLE (12A6)		
Col	1-72	Any alphameric statement	
Card 2	Plotter Control Card (4F5.2, 3F10.5, I5)		
Col	1-5	SCLX	Size factor in x direction.
	6-10	SCLY	Size factor in y direction.
	11-15	BOFFX	Board offset in x direction.
	16-20	BOFFY	Board offset in y direction.
	21-30	DOFFX	Data offset in x direction.
	31-40	DOFFY	Data offset in y direction.
	41-50	SL	Letter height in inches (0 if no lettering is desired).
	51-55	IPRNT	Print control. 0 = print. 1 = no print.
Card 3	Time Control Card (1A5, 9E8.1)		
Col	1-5	W1	"TIME="
	6-13	TIME(1)	Time values. Only TIME(1) may be zero. A large positive value of time will skip to the next problem. A negative value of TIME(1) will end the program and rewind tapes.
	14-21	TIME(2)	
	.....		
	70-77	TIME(9)	
Card 4	Temperature Contour and Coordinate Card (1A5, 9E8.1)		
Col	1-5	W2	
	6-13	TEMP(1) or COORDINATE(1)	TEMP=, X=, Y=, R=, Z=, or T=. (TEMP= for temperature contours. X=, Y=, R=, Z=, or T= for temperature on constant coordinate.) Must have a Card 5 if coordinate is given.
	14-21	TEMP(2) or COORDINATE(2)	
	.....		
	70-77	TEMP(9) or COORDINATE(9)	

Card 5 Coordinate Scale Card (2F10.5)
(Use this card only when W2 is X=, Y=, R=, Z=, or T=.)

Col 1-10 TNORM Normalizing value for temperatures
(Height of temperature scale in inches) =

$$10 \times (\text{SIZE T}) \times \frac{(\text{max. temp.})}{(\text{TNORM})}$$

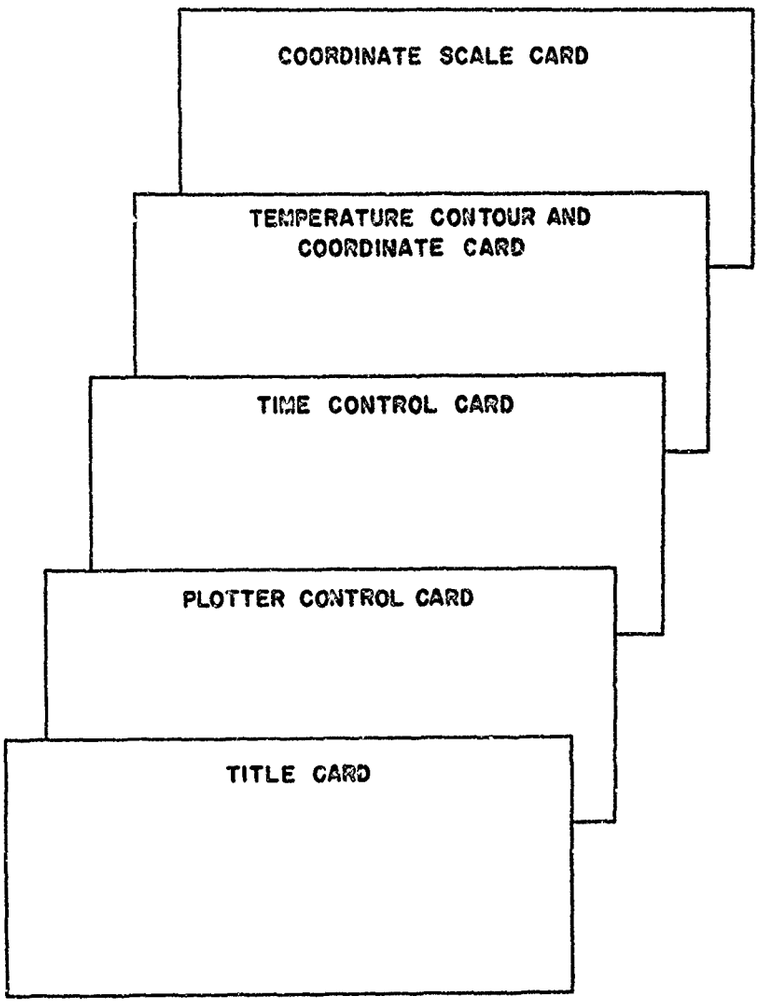


FIG. 17. DATA DECK FOR AMG042P

b. Program Listing AMG042P

```
$IBFTC MAIN     LIST
C AMG042PLOT PROGRAM
C       PLOTS ARE PUT ON TAPE 12 ,INPUT TAPE IS 11 , TAPE 8 IS SCRATCH
        DIMENSION T(30,50),X(30,50),Y(30,50),COORD(30,50,4),IMIN(50),
       1IMAX(50),N(30,50),RX(2,3),AUC(8),TITLE(12),CTITLE(12)
       2,TIME(9),TEMP(9),U(496),CNST1(8),CNST2(12),R(30,50),TH(30,50)
        EQUIVALENCE (X,COORD),(Y,COORD(1501))
       1,(R,COORD(3001)),(TH,COORD(4501))
        REAL M1,M2
        DATA RX/2HR=,2HZ=,2HX=,2HY=,2HR=,2HT=/,
       1ALC/6H*TEMPE,6HRATURE,6H PLOT ,6HAT TIN,6HE =    ,1H ,1H ,1H$/,
       2END/6HEND OF/,RR,U2P,U2M,D1,D2/1HR,1H+,1H-,6HTIME= ,6HTEMP= /,
       3POLAR,THETA,BLK/6HPOLAR ,5HTHTA=,6H      /
        REWIND 11
        REWIND 12
        REWIND 8
 1000   READ(5,100)CTITLE,SCLX,SCLY,BOFFX,BOFFY,DOFFX,DOFFY,SL,IPRNT
        P1=U2P
        P2=U2P
        P3=U2P
        P4=U2P
        RAD=57.2957795
        IF(BOFFX.LT.0.)P1=U2M
        IF(BOFFY.LT.0.)P2=U2M
        IF(DOFFX.LT.0.)P3=U2M
        IF(DOFFY.LT.0.)P4=U2M
        READ(11)TITLE,TCORD,JMAX,(IMIN(J),IMAX(J),J=1,JMAX)
        IF(TITLE(1).EQ.END)GO TO 500
        L=2
        IF(TCORD.EQ.RR)L=1
        XBIG=0.
        YBIG=0.
        RBIG=0.
        TBIG=0.
        DO 1 J=1,JMAX
        IB=IMIN(J)
```

```
      IE=IMAX(J)
      DO 1 I=IB,IE
      READ(11)X(I,J),Y(I,J),N(I,J)
      IF(TCORD.NE.POLAR)GO TO 111
      R(I,J)=SQRT(X(I,J)**2+Y(I,J)**2)
      TH(I,J)=ATAN(Y(I,J)/X(I,J))
      IF(R(I,J).GT.RBIG)RBIG=R(I,J)
      IF(TH(I,J).GT.TBIG)TBIG=TH(I,J)
C     CALCULATE SCALE FACTORS
111   IF(ABS(X(I,J)).GT.XBIG)XBIG=ABS(X(I,J))
      IF(ABS(Y(I,J)).GT.YBIG)YBIG=ABS(Y(I,J))
1     CONTINUE
      NXBIG=XBIG
      NYBIG=YBIG
      NRBIG=RBIG
      NTBIG=TBIG
      NFCTR=1
      NFCTT=1
      NFCTX=1
      NFCTY=1
      DO 6 I=1,4
      IF(NXBIG.GE.NFCTX)NFCTX=NFCTX*10
      IF(NYBIG.GE.NFCTY)NFCTY=NFCTY*10
      IF(NRBIG.GE.NFCTR)NFCTR=NFCTR*10
      IF(NTBIG.GE.NFCTT)NFCTT=NFCTT*10
6     CONTINUE
      YFCT=10000/NFCTY
      XFCT=10000/NFCTX
700   XSCL=(50./SCLX)*XFCT/10.
      YSCL=(50./SCLY)*YFCT/10.
      IF(XSCL.LT.999.9.AND.YSCL.LT.999.9) GO TO 5
      IF(XSCL.GT.999.9)XFCT=XFCT/10.
      IF(YSCL.GT.999.9)YFCT=YFCT/10.
      GO TO 700
5     NT=0
      NTT=1
```

```
      READ(5,1011)W1,TIME
      IF(TIME(1) .LT.0.)GO TO 500
      READ(5,1011)W2,TEMP
101   NT=NT+1
      NTT=1
      IF((NT .GT. 1 .AND. TIME(NT) .EQ  0.) .OR. NT .GT. 9) GO TO 5
12    READ(11)V2
      IF(V2.LT.0.)GO TO 1000
      TOL=V2/TIME(NT)
11    READ (11) U
      IF(NT .EQ. 1 .AND. TIME(NT) .EQ. 0.) GO TO 121
      IF(.999.LT.TOL.AND.TOL.LT.1.001)GO TO 121
      GO TO 12
121   K=0
      BACKSPACE 11
      BACKSPACE 11
      DO 122 J=1,JMAX
      IB=IMIN(J)
      IE=IMAX(J)
      DO 122 I=IB,IE
      K=K+1
122   T(I,J)=U(K)
      IF(IPRNT.EQ.0)WRITE(6,200)TITLE,CTITLE,V2
      WRITE(8,99)V2
      WRITE(8,96)ALC(1),ALC(1),W2,TEMP,ALC(8),ALC(8)
   96 FORMAT(2A1,1A6,9F6.2,2A5)
      REWIND 8
      READ(8,98)ALC(6),ALC(7)
      READ(8,95) CNST2
95    FORMAT(12A6)
      REWIND 8
      NCC=0
      DO 951 I=2,9
  951 IF(TEMP(I) .EQ. 0.) CNST2(I+2)=BLK
      IF(W2.NE.D2)GO TO 20
15    NCC=NCC+1
```

```
      IF(NCC.GT.1)GO TO 16
      END FILE 12
C     WRITE LETTERING AND GRID ON TAPE FOR PLOTTING
      CALL PLOT(TITLE,0,SL)
      CALL PLOT(ALC,0,SL)
      CALL PLOT(CNST2,1,SL)
      WRITE(12,210)P1,BOFFX,P2,BOFFY,P3,DOFFX,P4,DOFFY
      WRITE(12,211)XSCL,YSCL
      DO 40 J=1,JMAX
      IB=IMIN(J)
      IE=IMAX(J)
      DO 40 I=IB,IE
      XX=X(I,J)*XFCT
      YY=Y(I,J)*YFCT
      M1=U2P
      M2=U2P
      IF(XX.LT.0.)M1=U2M
      IF(YY.LT.0.)M2=U2M
      WRITE(12,103)M1,XX,M2,YY
40    CONTINUE
      WRITE(12,104)
16    IF(IPRNT.EQ.0)WRITE(6,202)TEMP(NTT),RX(1,L),RX(2,L)
      WRITE(12,105)NCC
C     INTERPOLATE TEMPERATURES FOR PLOTTING
      DO 25 J=1,JMAX
      IB=IMIN(J)
      IE=IMAX(J)
      DO 25 I=IB,IE
      N2=N(I,J)
      K=0
      GO TO(25,22,23,24,25),N2
22    JX=J
      IX=I+1
      GO TO 30
23    JX=J+1
      IX=I
```

```
      K=0
      GO TO 30
24    K=2
      GO TO 22
30    IF((T(I,J)-TEMP(NTT))*(TEMP(NTT)-T(IX,JX)))27,31,31
31    ALP=(TEMP(NTT)-T(I,J))/(T(IX,JX)-T(I,J))
      XX=X(I,J)+(X(IX,JX)-X(I,J))*ALP
      YY=Y(I,J)+(Y(IX,JX)-Y(I,J))*ALP
      XX1=XX*XFCT
      YY1=YY*YFCT
      IF(IPRNT.EQ.0)WRITE(6,204) XX,YY
      M1=U2P
      M2=U2P
      IF(XX1.LT.0.)M1=U2M
      IF(YY1.LT.0.)M2=U2M
      WRITE(12,103)M1,XX1,M2,YY1
27    IF(K.EQ.2)GO TO 23
25    CONTINUE
      NTT=NTT+1
      IF(TEMP(NTT).EQ.0..OR.NTT.GT.9)GO TO 101
      GO TO 15
C     INTERPOLATE FOR TEMPERATURES ALONG COORDINATE LINE
20    NC=2
      IF(W2.EQ.RX(1,L))NC=1
      NC1=MOD(NC,2)+1
      ND=0
      IF(TCORD.EQ.POLAR.AND.W2.EQ.RX(1,1))ND=1
      IF(TCORD.EQ.POLAR.AND.W2.EQ.RX(2,3))ND=2
      NCX=NC+ND
      KK=L
      IF(NCX.EQ.3.OR.NCX.EQ.4)KK=3
      IF(NT.EQ.1)
     1READ(5,1012)TNORM,SIZET,SCLR,SCLT
      END FILE 12
      IF(TCORD.EQ.POLAR)GO TO 1111
      XYSCL=YSCL
```

```
      XYFCT=YFCT
      IF(W2.EQ.RX(1,L))GO TO 1112
      XYSCL=XSCL
      XYFCT=XFCT
      GO TO 1112
1111  RFCT=10000/NFCTR
      IF(W2 .EQ. RX(1,3)) NC = 1
      NC1 = MOD(NC,2) + 1
      TFCTT=10000/NFCTT
      RSCL=(50./SCLR)*RFCT/10.
      TSCLT=(50./SCLT)*TFCTT/10.
      XYSCL=TSCLT
      XYFCT=TFCTT
      IF(W2.EQ.RX(1,KK))GO TO 1112
      XYSCL=RSCL
      XYFCT=RFCT
1112  TFCT=1000./TNORM
      TSCL=(50./SIZET)*10.
      WRITE(8,94)RX(NC1,KK),ALC(6),ALC(7),ALC(8),ALC(9)
94    FORMAT(16H*TEMPERATURE VS 1A1,9H AT TIME=2A6,4H FOR 2A3)
      REWIND 8
      READ(8,93)CNST1
93    FORMAT(9A6)
      REWIND 8
      CALL PLOT(TITLE,5,SL)
      CALL PLOT(CNST1,0,SL)
      CALL PLOT(CNST2,1,SL)
      NCC=0
      WRITE(12,210)P1,BOFFX,P2,BOFFY,P3,DOFFX,P4,DOFFY
      WRITE(12,211)XYSCL,TSCL
      WRITE(12,104)
21    IF(IPRNT.EQ.0)WRITE(6,203)RX(NC1,KK),RX(NC,KK),TEMP(NPT),
     1RX(NC1,KK)
      NCC=NCC+1
      WRITE(12,105)NCC
      DO 50 J=1,JMAX
```

```
      IB=IMIN(J)
      IE=IMAX(J)
      DO 50 I=IB,IE
      N2=N(I,J)
      GO TO(50,2,3,2,50),N2
2     IX=I+1
      JX=J
      GO TO 52
3     N2=1
      IX=I
      JX=J+1
52    IF((COORD(I,J,NCX)-TEMP(NTT))*(TEMP(NTT)-COORD(IX,JX,NCX)).LT.0.)
     1GO TO 60
      IF(NCX.EQ.4)A=(Y(IX,JX)-X(IX,JX)*TAN(TEMP(NTT)/RAD))/(Y(I,J)-
     1Y(IX,JX)+(X(IX,JX)-X(I,J))*TAN(TEMP(NTT)/RAD))
      IF(NCX.EQ.3)A=(Y(IX,JX)*(Y(IX,JX)-Y(I,J))-X(IX,JX)*(X(IX,JX)-
     1X(I,J))+SQRT(TEMP(NTT)**2*((X(I,J)-X(IX,JX))**2+(Y(IX,JX)-
     2Y(I,J))**2)-(X(I,J)*Y(IX,JX)-Y(I,J)*X(IX,JX))**2))/((X(I,J)-
     3X(IX,JX))**2+(Y(IX,JX)-Y(I,J))**2)
      IF(NCX.EQ.1.OR.NCX.EQ.2)A=(TEMP(NTT)-COORD(I,J,NCX))/(COORD(IX,JX,
     1NCX)-COORD(I,J,NCX))
      XY=COORD(I,J,NC1)+(COORD(IX,JX,NC1)-COORD(I,J,NC1))*A
      SIG=T(I,J)+(T(IX,JX)-T(I,J))*A
      IF(NCX.EQ.4)XY=XY/COS(TEMP(NTT)/RAD)
      IF(NCX.EQ.3)XY=ARSIN(XY/TEMP(NTT))*RAD
      XYT=XY*XYFCT
      SIGT=(SIG*TFCT)/1000.
      IF(IPRNT.EQ.0)WRITE(6,201)XY,SIG,SIGT
      SIGT=SIGT*1000.
      WRITE (12,103) M1,XYT,M2,SIGT
60    IF(N2.EQ.4)GO TO 3
50    CONTINUE
      NTT=NTT+1
      IF(TEMP(NTT).EQ.0..OR.NTT.GT.9)GO TO 101
      GO TO 21
500   END FILE 12
```

```
      REWIND 11
      REWIND 12
      STOP
98    FORMAT(2A6)
99    FORMAT(E12.4)
100   FORMAT(12A6/4F5.2,3F10.5,I5)
1011  FORMAT(1A5,9E8.1)
1012  FORMAT(4F10.5)
103   FORMAT(2(1H1,1A1,F5.0))
104   FORMAT(14H0000000*000000)
105   FORMAT(12H000000050000I1,1H0)
200   FORMAT(19H1AMG042PLOT PROGRAM /1X12A6//1X12A6//34H THE FOLLOWING D
     1LOTS ARE FOR TIME= F10.5)
201   FORMAT(1H F12.4,1PE15.5,10X1PE15.5)
202   FORMAT(14HOPLOT OF TEMP=F10.5 //5X1A1,10X1A1)
203   FORMAT(33HOTEMPERATURE DISTRIBUTION VERSUS 1A1,23H ALONG COORDINAT
     1E LINE 1A1,1H= F8.5/1HO10XA1,5X11HTEMPERATURE 10X22HNORMALIZED TEM
     2PERATURE )
204   FORMAT(2F10.5)
210   FORMAT(1H41A1,F5.0,1H41A1,F5.0/2(1H31A1,F5.0))
211   FORMAT(14H0000000=000000/2(1H2 F6.0))
      END
```

```
$IBFTC PLOTT
CAMGO   SUBROUTINE PLOT FOR PLOTTING ALPHAMBRIC CHARACTERS
      SUBROUTINE PLOT(ALIST,NEND,SCALE)
      DIMENSION ACODE(41),          XTABLE(261),YTABLE(261),N(2,39),
     1ALIST(12),AMASK(6)
      EQUIVALENCE (BLIST,IDLIST)
      DATA MASK1,AMASK/000000000040,077000000000,000770000000,
     1000007700000,000000770000,000000007700,000000000077/
      DATA ACODE /6H00000A,6H00000B,6H00000C,6H00000D,6H00000E,6H00000F,
     1    6H00000G,6H00000H,6H00000I,6H00000J,6H00000K,6H00000L,6H00000M,
     2    6H00000N,6H00000O,6H00000P,6H00000Q,6H00000R,6H00000S,6H00000T,
     3    6H00000U,6H00000V,6H00000W,6H00000X,6H00000Y,6H00000Z,6H000001,
     4    6H000002,6H000003,6H000004,6H000005,6H000006,6H000007,6H000008,
     5    6H000009,6H000000,6H00000/,6H00000.,6H00000-,6H00000*,6H00000$/
      DATA XTABLE/0.,0.,1.,6.,7.,7.,
     37.,0.,0.,6.,7.,7.,6.,0.,0.,6.,7.,7.,6.,7.,5.,2.,0.,0.,2.,5.,7.,0.,
     40.,5.,7.,7.,5.,0.,7.,0.,0.,5.,0.,0.,7.,0.,0.,5.,0.,0.,7.,5.,7.,7.,
     55.,2.,0.,0.,2.,5.,7.,0.,0.,0.,0.,7.,7.,7.,0.,6.,3.,3.,0.,6.,0.,0.,
     61.,4.,5.,5.,3.,7.,0.,0.,0.,5.,1.,7.,7.,0.,0.,0.,0.,4.,8.,8.,0.,0.,
     77.,7.,2.,0.,0.,2.,5.,7.,7.,5.,2.,0.,0.,0.,6.,7.,7.,6.,0.,6.,5.,2.,
     80.,0.,2.,5.,7.,7.,6.,5.,7.,0.,0.,6.,7.,7.,6.,0.,4.,7.,0.,1.,6.,7.,
     97.,6.,1.,0.,0.,1.,6.,7.,4.,4.,0.,8.,0.,0.,2.,5.,7.,7.,0.,4.,8.,0.,
     10.,4.,8.,8.,0.,8.,4.,0.,8.,0.,4.,4.,4.,8.,0.,7.,0.,7.,1.,3.,3.,0.,
     26.,0.,2.,5.,7.,7.,0.,0.,7.,0.,1.,6.,7.,7.,6.,1.,6.,7.,7.,6.,1.,0.,
     35.,5.,0.,7.,0.,1.,5.,7.,7.,5.,1.,0.,0.,7.,7.,6.,1.,0.,0.,1.,6.,8.,
     47.,6.,1.,0.,0.,7.,4.,1.,0.,0.,1.,6.,7.,7.,6.,1.,0.,0.,1.,6.,7.,7.,
     56.,7.,6.,1.,0.,0.,1.,6.,7.,7.,6.,1.,0. /
      DATA YTABLE,N/0.,8.,10.,10.,8.,0.,4.,4.,6.,6.,7.,9.,10.,10.,0.,0.,
     11.,5.,6.,2.,0.,0.,2.,8.,10.,10.,8.,0.,10.,10.,8.,2.,0.,0.,0.,0.,
     25.,5.,5.,10.,10.,0.,5.,5.,5.,10.,10.,4.,4.,2.,0.,0.,2.,8.,10.,10.,
     38.,0.,5.,10.,5.,5.,10.,0.,0.,0.,0.,10.,10.,10.,2.,1.,0.,0.,1.,10.,
     410.,10.,0.,10.,5.,10.,6.,0.,0.,0.,0.,10.,0.,10.,5.,10.,0.,0.,10.,0.,
     510.,0.,2.,8.,10.,10.,8.,2.,0.,0.,0.,5.,10.,10.,9.,6.,5.,5.,1.,0.,
     60.,2.,8.,10.,10.,8.,2.,1.,2.,0.,0.,10.,10.,9.,6.,5.,5.,5.,0.,1.,
     70.,0.,1.,4.,5.,5.,6.,9.,10.,10.,9.,0.,10.,10.,10.,10.,2.,0.,0.,2.,
     810.,10.,0.,10.,10.,0.,5.,0.,10.,10.,0.,5.,0.,10.,10.,0.,0.,6.,10.,
```

```
910.,10.,0.,0.,8.,10.,0.,0.,0.,8.,10.,10.,8.,6.,2.,0.,0.,9.,10.,10
1,9.,6.,5.,5.,5.,4.,1.,0.,0.,1.,0.,10.,4.,4.,1.,0.,0.,2.,4.,8.,6.,
25.,10.,10.,9.,10.,10.,9.,1.,0.,0.,1.,5.,6.,6.,5.,10.,10.,0.,5.,6.,
39.,10.,10.,9.,6.,5.,5.,4.,1.,0.,0.,1.,4.,5.,6.,5.,5.,8.,9.,10.,10
4,9.,1.,0.,0.,1.,1,8,9,19,20,27,28,34,35,41,42,47,48,55,56,64,65,70
5,71,78,79,84,85,87,88,92,93,96,97,105,106,113,114,125,126,134,135
6146,147,150,151,156,157,159,160,164,165,169,170,174,175,178,179,
7183,184,191,192,204,205,208,209,218,219,230,231,233,234,249,250,
8261,97,105,176,177,177,177,61,62/
 IF(SCALE.EQ.0.) GO TO 22
 SCAL =100./(2.*SCALE)
 WRITE(12,200)SCAL ,SCAL
 CX=0.
 CY=45.
 DO 20 I = 1,12
 DO 20 K = 1,6
 BLIST = AND(ALIST(I),AMASK(K))
 IBLIST= IBLIST/2**((6-K)*6)
 IF (IBLIST.LT.0) IBLIST = IABS(IBLIST)+MASK1
 DO 1 J=1,41
 IF(BLIST.EQ.ACODE(J)) GO TO 3
1 CONTINUE
 CX=CX+7.
 GO TO 20
3 IF(J.NE.40) GO TO 5
 CY=CY-15.
 CX=0.
 GO TO 20
5 IF(J.EQ.41) GO TO 21
 KK=N(1,J)
 L=N(2,J)
 YY=CY+YTABLE(KK)
 XX=CX+XTABLE(KK)
 WRITE (12,202)XX,YY
 WRITE(12,201)
 DO 4 LL=KK,L
```

```
      YY=CY+YTABLE(LL)
      XX=CX+XTABLE(LL)
    4 WRITE(12,202) XX,YY
      WRITE (12,202)XX,YY
      WRITE(12,203)
      CX=CX+10.
   20 CONTINUE
   21 IF(NEND.EQ.1)END FILE 12
   22 RETURN
  200 FORMAT(14H0000000=000000/14H3000000300000O/14X4-002604-00280/
     12(1H2F6.0))
  201 FORMAT(14H000000006000000)
  202 FORMAT(2(1H1F6.0))
  203 FORMAT(14H000000007000000)
      END
```

```
$IBMAP UNITS
       ENTRY   .UN08.
       ENTRY   .UN11.
       ENTRY   .UN12.
.UN08. PZE     UNIT08
.UN11. PZE     UNIT11
.UN12. PZE     UNIT12
UNIT08 FILE    ,UT4,READY,BIN,INOUT              8-3
UNIT11 FILE    ,B(2),READY,BIN,HOLD              8-6
UNIT12 FILE    ,B(4),READY,OUTPUT,HOLD           8-8
       END
```

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Rohm and Haas Company Redstone Research Laboratories Huntsville, Alabama 35807	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

APPLICATION OF THE FINITE ELEMENT METHOD TO HEAT CONDUCTION IN SOLIDS

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

5. AUTHOR(S) *(First name, middle initial, last name)*

Eric B. Becker
Charles H. Parr

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
November 1967	111	9

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
DA-01-021 AMC-11536(Z) DAAH01-67-C-0947 b. PROJECT NO.	S-117
c.	9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*
d.	

10. DISTRIBUTION STATEMENT

This document is subject to special report controls and each transmittal to foreign governments or foreign nationals may be made only with specific prior approval of AMSMI-RK.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	U. S. Army Missile Command Redstone Arsenal, Alabama 35809

13. ABSTRACT

A new numerical method for the solution of heat conduction problems in thermally anisotropic, nonhomogeneous bodies of complex geometry was devised which is based on a discretization concept developed in the matrix analysis of structures. This discretization method, commonly referred to as the finite element method, reduces the problem formulation to the solution of a matrix equation for the nodal point temperatures of the assembly of finite elements. The resulting matrix equation is stable for any time step. The method is extremely flexible and easy to apply. The method was applied by writing a computer program for the solution of heat conduction problems in plane, thermally anisotropic, nonhomogeneous bodies.

DD FORM 1 NOV 65 1473 REPLACES DD FORM 1473, 1 JAN 64, WHICH IS OBSOLETE FOR ARMY USE.