

UNCLASSIFIED

AD NUMBER
AD809330
NEW LIMITATION CHANGE
TO Approved for public release, distribution unlimited
FROM Distribution authorized to U.S. Gov't. agencies and their contractors; Critical Technology; DEC 1966. Other requests shall be referred to Air Force Flight Dynamics Lab., Attn: FDTR. Wright-Patterson AFB, OH 45433.
AUTHORITY
AFFDL ltr, 12 Mar 1973

THIS PAGE IS UNCLASSIFIED

**TRANSIENT ANALYSIS OF HEAT CONDUCTION
THROUGH A SLAB BY INFINITE SERIES**

**THOMAS N. BERNSTEIN
ROBERT M. ENGLE, JR.**

TECHNICAL REPORT AFFDL-TR-68-109

DECEMBER 1968

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 the Theoretical Mechanics Branch, Structures Division, Air Force Flight Dynamics Laboratory (FDTK), Wright-Patterson Air Force Base, Ohio 45433.

**AIR FORCE FLIGHT DYNAMICS LABORATORY
RESEARCH AND TECHNOLOGY DIVISION
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO**

NOTICES

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Copies of this report should not be returned to the Research and Technology Division unless return is required by security considerations, contractual obligations, or notice on a specific document.

**Best
Available
Copy**

TRANSIENT ANALYSIS OF HEAT CONDUCTION THROUGH A SLAB BY INFINITE SERIES

**THOMAS N. BERNSTEIN
ROBERT M. ENGLE, JR.**

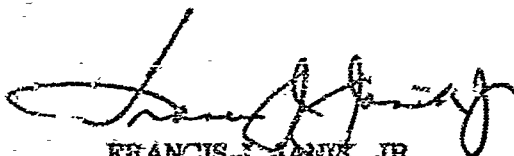
This document is subject to special export controls and each transmittal to foreign governments or foreign nationals may be made only with prior approval of the Theoretical Mechanics Branch, Structures Division, Air Force Flight Dynamics Laboratory (FDTR), Wright-Patterson Air Force Base, Ohio 45433.

FOREWORD

This report was prepared by Thomas N. Bernstein and Robert M. Engle, Jr., of the Theoretical Mechanics Branch, Structures Division, Air Force Flight Dynamics Laboratory. The work was conducted in house under Project No. 1467, "Structural Analysis Methods," Task No. 146702, "Thermoelastic Structural Analysis Methods," and was administered by the Air Force Flight Dynamics Laboratory, Research and Technology Division, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio. Mr. Robert M. Bader is the Project Engineer administering Project No. 1467.

This report covers research conducted from July 1964 to July 1966. The manuscript was released by the authors in September 1966 for publication as a technical report.

This technical report has been reviewed and is approved.



FRANCIS J. JANEK, JR.
Chief, Theoretical Mechanics Branch
Structures Division

ABSTRACT

The exact solution to the problem of conduction of heat through a slab is developed. The solution, formulated in terms of an infinite series, allows arbitrary initial conditions and time-dependent boundary conditions. The solution is programmed in FORTRAN IV for the IBM 7094 II computer. Several check problems were solved and the results were compared with those obtained from a finite difference heat transfer program.

PREVIOUS PAGE WAS BLANK, THEREFOR WAS NOT FILLED.

AFFDL-TR-66-109

CONTENTS

SECTION	PAGE
I. INTRODUCTION	1
II. MATHEMATICAL FORMULATION	1
A. Boundary Conditions	1
B. Steady State Solution	2
C. Transient Solution	3
D. Modification of Solution for Time Dependent Boundary Conditions	6
E. Solution of Initial Condition Problem	7
F. Unsteady State Solution	7
G. Transient Solution	7
H. Complete Solution for the General Problem	9
III. APPLICATIONS	10
IV. COMPUTER PROGRAM FOR SERIES TRANSIENT ANALYSIS OF SLAB HEAT TRANSFER (STASH)	11
A. Description	11
B. Input	13
C. Sample Problem	19
D. Restrictions	21
E. Output	22
V. CONCLUSIONS	26
APPENDIX I COMPUTER PROGRAM SOURCE LISTING	27
APPENDIX II EIGENVALUE SUBROUTINES	41
APPENDIX III RESULTS OF CHECK PROBLEMS	45

ILLUSTRATIONS

FIGURE		PAGE
1.	Geometrical Representation	2
2.	Simplified Flow Chart of Transfer of Information	12
3.	Input Data Format	17
4.	Sample Problem Data Deck	20
5.	Sample Output (Case 4) Showing Display of Input Data	23
6.	Sample Output (Case 4) Showing Eigenvalue Solution	24
7.	Sample Output (Case 4) Showing Intermediate Print	25
8.	Temperature Profiles (Case 1)	50
9.	Temperature Profiles (Case 2)	51
10.	Temperature Profiles (Case 3)	52
11.	Temperature Profiles (Case 4)	53
12.	Temperature Profiles (Case 5)	54

TABLES

TABLE		PAGE
I	Systems of Units Stored Internally	15
II	Comparison of Data for Case 1	46
III	Comparison of Data for Case 2	47
IV	Comparison of Data for Case 3	48
V	Comparison of Data for Case 4	49
VI	Comparison of Data for Case 5	49

SYMBOLS

<u>MATH SYMBOL</u>	<u>FORTTRAN SYMBOL</u>	<u>PHYSICAL DEFINITION</u>
$A, \bar{A}, \bar{A}_0, \bar{A}_L$		Coefficients of Steady State Solution
$B, \bar{B}, \bar{B}_0, \bar{B}_L$		Coefficients of Steady State Solution
C_0, C_1, C_2, C_n		Constants of Integration
C_p	CP	Specific heat
D	DETK	Determinant of K_{ij} 's
F_0, F_L	FO, FL	boundary condition constants
$f(x), \bar{f}_x$	FOFX(X)	Initial conditions
K	K	Thermal conductivity
K_{ij}	KI1, K21, . . .	Boundary condition indicators
L	L	Length
n	NTERMS	Summation index
N		Particular value of n
t	T	Time
T	TEMP	Temperature
$T_s, \bar{T}_s$		Steady state solutions
$T_I, \bar{T}_I, T_C, \bar{T}_C, T_L$		Transient solutions
T_{IC}		Solution of initial condition problem
$\bar{T}$		Complete problem solution
S		Cross-sectional area of slab
x	x	Distance
$\chi(x)$		Assumed solution

SYMBOLS (Cont'd)

<u>MATH SYMBOL</u>	<u>FORTRAN SYMBOL</u>	<u>PHYSICAL DEFINITION</u>
γ_n		Repetitive term in solution
Z, Z_n	ZN	Eigenvalues
∞		Infinity
$\alpha = k/\rho C_p$	ALPHA	Thermal diffusivity
β, β_n	BETAN	Eigenvalues
λ	LAMBDA	Dummy time variable
π	PI	3.1415926
ρ	RHO	Density
$\phi_0(t)$ $\phi_L(t)$	PHI0(T) PHIL(T)	Boundary condition time functions
$\Phi(t)$		Assumed solution
<u>Subscripts</u>		
IC		Initial condition
L, O		Boundary
P		Pressure
S		Steady state
T		Transient
0, 1, 2, i, j, n, N		Counters

Superscripts

Primes denote differentiation

SECTION I

INTRODUCTION

The conduction of heat through a slab is governed by the following partial differential equation.

$$\frac{\partial}{\partial x} \left[K \frac{\partial T}{\partial x} \right] = \rho c_p \frac{\partial T}{\partial t} \quad (1)$$

For constant thermal diffusivity, this equation simplifies to

$$\frac{\partial^2 T}{\partial x^2} = \frac{1}{\alpha} \frac{\partial T}{\partial t} \quad (2)$$

The exact solution to this equation can be formulated in terms of an infinite series. This report develops the exact solution for arbitrary initial conditions and time dependent boundary conditions. The solution has been programmed in FORTRAN for an IBM 7094 computer and the source program listing is contained in Appendix I.

SECTION II

MATHEMATICAL FORMULATION

A. BOUNDARY CONDITIONS

The general solution of Equation (2) must satisfy arbitrary initial and time dependent boundary conditions which can be expressed in the following form:

$$T(x, t) = f(x) \quad t = 0 \quad (3)$$

$$K_{11} \frac{\partial T}{\partial x} + K_{12} T = F_0 \phi_0(\lambda) \quad x = 0 \quad (4)$$

$$K_{21} \frac{\partial T}{\partial x} + K_{22} T = F_L \phi_L(\lambda) \quad x = L \quad (5)$$

where the K_{ij} 's are constants. Selecting different values of these coefficients dictates the mode of heat transfer present at the boundary. By various combinations of constants, the imposition of surface temperature, convection, heat flux or insulation is possible. A more detailed discussion on the interpretation of boundary conditions is contained in Section III, "Applications."

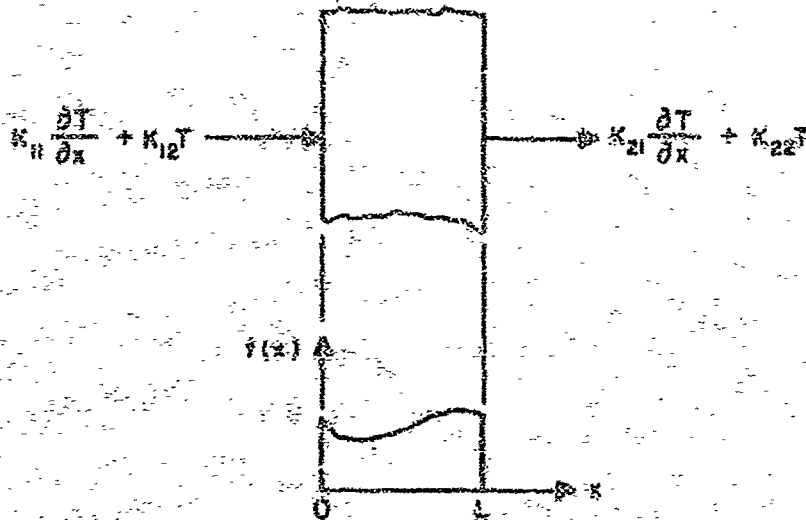


Figure 1. Geometrical Representation

The application of Duhamel's superposition theorem to account for the time dependent boundary conditions necessitated breaking the solution into a transient portion satisfying initial conditions, and a steady state plus transient with zero initial conditions.

Since the restriction has been placed on the boundary conditions that the time dependency can be expressed as a product of a time function and one of the standard boundary conditions, it is therefore possible to solve the equations neglecting the time variation and then modify the solution to account for it.

The problem is first simplified by breaking the solution into two parts: a steady state portion, $T_s(x, \infty)$, satisfying the arbitrary boundary condition, and a transient portion, $T_t(x, t)$, satisfying the initial temperature distribution and homogeneous boundary conditions.

B. STEADY STATE SOLUTION

For steady state conditions we note that $\frac{\partial T}{\partial t} = 0$ and Equation (2) simplifies to

$$\frac{\partial^2 T}{\partial x^2} = 0 \quad (5)$$

The solution of this equation is found directly by integration with the result;

$$T_s = Ax + B \quad (7)$$

We now impose the arbitrary boundary conditions

$$K_{11} \frac{\partial T}{\partial x} + K_{12} T = F_0 \quad x = 0 \quad (8)$$

$$K_{21} \frac{\partial T}{\partial x} + K_{22} T = F_L \quad x = L \quad (9)$$

Substituting Equation (7) into (8) and (9) yields

$$K_{11} A + K_{12} B = F_0 \quad (10)$$

$$K_{21} A + K_{22}(AL + B) = F_L \quad (11)$$

The constants of integration, A and B, can now be evaluated from Equations (10) and (11). In order to keep these expressions in general form, the solution is accomplished by Cramer's rule with the result

$$A = \frac{K_{12} F_L - K_{22} F_0}{K_{12} K_{22} L - (K_{11} K_{22} - K_{12} K_{21})} \quad (12)$$

$$B = \frac{(K_{21} + K_{22} L) F_0 - K_{11} F_L}{K_{12} K_{22} L - (K_{11} K_{22} - K_{12} K_{21})} \quad (13)$$

C. TRANSIENT SOLUTION

A product form of solution is assumed for Equation (3), and designated by $X(x)\Phi(t)$. Substitution into Equation (2) yields

$$X'' \Phi = \frac{1}{\alpha} X \Phi' \quad (14)$$

Rearranging

$$\frac{X''}{X} = \frac{\Phi'}{\alpha \Phi} \quad (15)$$

which requires that each of these functions be equivalent to some, as yet arbitrary constant. Then setting this constant equal to $-\beta^2$ results in two ordinary differential equations of the form

$$\Phi' + \alpha \beta^2 \Phi = 0 \quad (16)$$

$$X'' + \beta^2 X = 0 \quad (17)$$

Equation (8) has the exponential form of solution

$$\Phi = C_0 e^{-\alpha \beta^2 t} \quad (18)$$

whereas Equation (17) is satisfied by

$$X(x) = C_1 \cos \beta x + C_2 \sin \beta x \quad (19)$$

The solution to Equation (3) is then

$$T(x, t) = [C_1 \cos \beta x + C_2 \sin \beta x] [C_0 e^{-a\beta^2 t}] \quad (20)$$

This transient solution must satisfy the initial temperature distribution and homogeneous boundary conditions as follows:

$$T(x, t) = f(x) \quad t = 0 \quad (3)$$

$$K_{11} \frac{\partial T}{\partial x} + K_{12} T = 0 \quad x = 0 \quad (21)$$

$$K_{21} \frac{\partial T}{\partial x} + K_{22} T = 0 \quad x = L \quad (22)$$

Note first that the constant C_0 can be eliminated since its effect can be included in C_1 and C_2 . To evaluate the remaining constants substitute Equation (20) into Equations (3), (21) and (22).

Substitution of Equation (20) into (21) yields

$$K_{12} C_1 + K_{11} \beta C_2 = 0 \quad (23)$$

from which we obtain

$$C_1 = -\frac{K_{11} \beta}{K_{12}} C_2 \quad (24)$$

At this point it becomes necessary to impose the artificial restriction that $K_{12} \neq 0$, in order that calculations performed on the computer remain bounded.

Substitution of Equation (20) into (22) yields

$$[K_{22} \cos \beta L - K_{21} \sin \beta L] C_1 + [K_{21} \beta \cos \beta L + K_{22} \sin \beta L] C_2 = 0 \quad (25)$$

To obtain a nontrivial solution for C_1 and C_2 , the determinant of their coefficients must be set equal to zero. This yields the following transcendental equation.

$$\tan z = \frac{DLz}{K_{21} K_{11} z^2 + K_{22} K_{12} L^2} \quad (26)$$

where

$$z = \beta L \quad (27)$$

and

$$D = K_{11} K_{22} - K_{12} K_{21} \quad (28)$$

Equation (26) has infinitely many solutions (eigenvalues), and we shall denote these by Z_n , where $n = 0, 1, 2, \dots$. The remaining constant C_2 is evaluated by substituting our solution into Equation (3) in order that the initial temperature distribution be satisfied. It is obvious at this point that, in general, arbitrary functions for the initial temperature distribution can not be satisfied using only one value of Z_n and C_2 . We are thus required to expand our initial condition and our solution in an infinite series. Then the coefficient C_2 becomes C_n and its evaluation proceeds as follows. The total solution at this point can be expressed

$$T(x, t) = Ax + B + \sum_{n=0}^{\infty} C_n \left[\sin \beta_n x - \frac{K_{11} \beta_n}{K_{12}} \cos \beta_n x \right] e^{-\alpha \beta_n^2 t} \quad (29)$$

Imposing the problem initial condition yields

$$T(x, 0) = f(x) = Ax + B + \sum_{n=0}^{\infty} C_n \left[\sin \beta_n x - \frac{K_{11} \beta_n}{K_{12}} \cos \beta_n x \right] \quad (30)$$

Rearranging

$$f(x) - (Ax + B) = \sum_{n=0}^{\infty} C_n \left[\sin \beta_n x - \frac{K_{11} \beta_n}{K_{12}} \cos \beta_n x \right] \quad (31)$$

Since the sines and cosines form a complete set of orthogonal functions, the C_n 's can be evaluated by multiplying both sides of Equation (31) by

$$\left[\sin \beta_N x - \frac{K_{11} \beta_N}{K_{12}} \cos \beta_N x \right] \text{ and integrating from zero to } L.$$

Thus,

$$\begin{aligned} \int_0^L \left[f(x) - (Ax + B) \right] \left[\sin \beta_N x - \frac{K_{11} \beta_N}{K_{12}} \cos \beta_N x \right] dx = \\ \int_0^L \sum_{n=0}^{\infty} C_n \left[\sin \beta_n x - \frac{K_{11} \beta_n}{K_{12}} \cos \beta_n x \right] \left[\sin \beta_N x - \frac{K_{11} \beta_N}{K_{12}} \cos \beta_N x \right] dx \end{aligned} \quad (32)$$

By orthogonality this integration produces nontrivial results only in the case of $n = N$. Therefore

$$C_n = \frac{\int_0^L \left[f(x) - (Ax + B) \right] \left[\sin \beta_n x - \frac{K_{11} \beta_n}{K_{12}} \cos \beta_n x \right] dx}{\int_0^L \left[\sin \beta_n x - \frac{K_{11} \beta_n}{K_{12}} \cos \beta_n x \right]^2 dx} \quad (33)$$

The denominator of C_n can be evaluated directly with the result:

$$\int_0^L \left[\sin \beta_n x - \frac{K_{11} \beta_n}{K_{12}} \cos \beta_n x \right]^2 dx = \frac{1}{2\beta_n} \left\{ z_n \left[\left(\frac{K_{11} \beta_n}{K_{12}} \right)^2 + 1 \right] \right. \\ \left. + \sin z_n \cos z_n \left[\left(\frac{K_{11} \beta_n}{K_{12}} \right)^2 - 1 \right] + 2 \left(\frac{K_{11} \beta_n}{K_{12}} \right) \sin^2 z_n \right\} \quad (34)$$

Collecting the formulations required for problem evaluation leads to the expression of the solution for time independent boundary conditions in the form

$$T(x, t) = Ax + B$$

$$+ \sum_{n=0}^{\infty} \frac{\int_0^L [f(x) - (Ax + B)] [Y_n(x)] dx}{\int_0^L [Y_n(x)]^2 dx} [Y_n(x)] e^{-\alpha \beta_n^2 t} \quad (35)$$

where

$$A = \left[\frac{K_{12} F_L - K_{22} F_0}{K_{11} K_{22} L - D} \right] \quad (36)$$

$$B = \left[\frac{(K_{21} + K_{22} L) F_0 - K_{11} F_L}{K_{11} K_{22} L - D} \right] \quad (37)$$

$$Y_n(x) = \left[\sin \beta_n x - \frac{K_{11} \beta_n}{K_{12}} \cos \beta_n x \right] \quad (38)$$

$$\beta_n = z_n / L \quad (39)$$

and

$$\tan z_n = Z_n = \frac{D L Z_n}{K_{21} K_{11} Z_n^2 + K_{22} K_{12} L^2} \quad (40)$$

D. MODIFICATION OF SOLUTION FOR TIME DEPENDENT BOUNDARY CONDITIONS

Duhamel's superposition integral is now applied to the solution, Equation (35), to account for the time varying boundary conditions. F. B. Hildebrand gives the solution in the form*

$$T(x, t) = T_0(x, \omega) \phi(x) + \left\{ \phi(0) + \int_0^t e^{-\alpha \beta_n^2 \lambda} \phi'(\lambda) d\lambda \right\} T_0(x) \quad (41)$$

This formulation is based on certain limitations, however, which must be eliminated. The first is the assumption of zero initial conditions. This restriction is eliminated by considering a

*Hildebrand, F. B., Introduction to Numerical Analysis, McGraw-Hill Book Company, Inc., New York, 1956.

separate problem, possessing the given initial conditions and homogeneous boundary conditions. The solution of this problem is added to Equation (41). The second simplification utilized to obtain Equation (41) was to hold one boundary at zero and consider the remaining boundary to vary with time. For our problem both boundaries can vary with time so we make use of the superposition principle once again by varying first one boundary condition and then the other, with the remaining boundary held at zero. The two results are then added. Note that $f(x) = 0$ for both these solutions.

E. SOLUTION OF INITIAL CONDITION PROBLEM

For the given initial condition, Equation (3), and the homogeneous boundary conditions, Equations (21) and (22), a "zero" steady state solution is obtained from Equation (7), i.e., $A = 0$, $B = 0$. Employing the given initial condition in Equation (35) then yields the desired result.

$$T_{IC}(x, t) = \sum_{n=1}^{\infty} \frac{\int_0^L f(x) [\gamma_n(x)] dx}{\int_0^L [\gamma_n(x)]^2 dx} [\gamma_n(x)] e^{-\alpha \beta_n^2 t} \quad (42)$$

F. UNSTEADY STATE SOLUTION

The steady state solution, $T_s(x, \infty)$, employed in Equation (35) is modified to $T_s(x, \infty) \phi(t)$ in Equation (41). This result can be viewed as the steady state solution to a problem with our boundary conditions, if those conditions were "frozen" at the instant t , and remained constant as $t \rightarrow \infty$. Since our boundary conditions vary continuously with time, it is not possible to reach a steady state condition. This explains the title employed for this section of the report. $T_s(x, \infty) \phi(t)$ can be obtained immediately from Equation (7), using boundary condition Equations (4) and (5) in place of (3) and (9). The result is

$$T_s \phi(t) = T_c = Ax + B \quad (43)$$

where

$$A = \frac{K_{12} F_L \phi_L(t) - K_{22} F_0 \phi_0(t)}{K_{12} K_{22} L - D} \quad (44)$$

$$B = \frac{(K_{21} + K_{22} L) F_0 \phi_0(t) - K_{11} F_L \phi_L(t)}{K_{12} K_{22} L - D} \quad (45)$$

G. TRANSIENT SOLUTION (TIME VARIABLE BOUNDARY CONDITIONS)

The transient solution for boundary condition Equations (4) and (5) must be evaluated in two parts as indicated in Section II D. First, consider the conditions

$$K_{11} \frac{\partial T}{\partial x} + K_{12} T = F_0 \phi_0(x) \quad (4)$$

and

$$K_{21} \frac{\partial T}{\partial x} + K_{22} T = 0 \quad (22)$$

with

$$f(x) = 0 \quad (46)$$

Equations (44) and (45) for A and B are modified by letting $F_L = 0$, with the result

$$A_0 = \frac{-K_{22} F_0 \phi_0(t)}{K_{12} K_{22} L - D} \quad (47)$$

$$B_0 = \frac{(K_{21} + K_{22} L) F_0 \phi_0(t)}{K_{12} K_{22} L - D} \quad (48)$$

The transient solution is obtained by substituting these results into the transient solution in Equation (35) with the result

$$T_{T_0} = \sum_{n=0}^{\infty} \frac{\int_0^L (-B_0 - A_0 x) Y_n(x) dx}{\int_0^L [Y_n(x)]^2 dx} [Y_n(x)] e^{-\alpha \beta_n^2 t} \quad (49)$$

Similarly for the conditions

$$K_{11} \frac{\partial T}{\partial x} + K_{12} T = 0 \quad (5)$$

$$K_{21} \frac{\partial T}{\partial x} + K_{22} T = F_L \phi_L(t) \quad (21)$$

$$f(x) = 0 \quad (46)$$

and letting $F_0 = 0$ in Equations (41) and (45), with the result

$$A_L = \frac{K_{12} F_L \phi_L(t)}{K_{12} K_{22} L - D} \quad (50)$$

$$B_L = \frac{-K_{11} F_L \phi_L(t)}{K_{12} K_{22} L - D} \quad (51)$$

we obtain from Equation (35) the transient solution

$$T_{T_L} = \sum_{n=0}^{\infty} \frac{\int_0^L (A_L x - B_L) Y_n(x) dx}{\int_0^L [Y_n(x)]^2 dx} [Y_n(x)] e^{-\alpha \beta_n^2 t} \quad (52)$$

H. COMPLETE SOLUTION FOR THE GENERAL PROBLEM

The general form of the complete solutions was expressed by Equation (41). Collecting the solutions obtained in Equations (42), (43), (49) and (52), and substituting into Equation (41), yields the final result,

$$\begin{aligned} \bar{T}(x,t) = & T_0 + \left\{ \phi_0(0) + \int_0^1 e^{-\alpha \beta_n^2 \lambda} \phi_0'(\lambda) d\lambda \right\} T_{T_0}(x,t) \\ & + \left\{ \phi_L(0) + \int_0^1 e^{-\alpha \beta_n^2 \lambda} \phi_L'(\lambda) d\lambda \right\} T_{T_L}(x,t) + T_{IC}(x,t) \end{aligned} \quad (53)$$

SECTION III APPLICATIONS

The generalized boundary conditions utilized in the mathematical formulation can be specialized to handle a number of physical problems. For example, take Equation (5)

$$K_{21} \frac{\partial T}{\partial x} + K_{22} T = F_L \phi_L(\lambda) \quad (5)$$

Appropriate choices of the indicators lead to the following:

- a. Prescribed constant surface temperature.

Let $K_{21} = 0$, $K_{22} = 1$, $\phi_L(\lambda) = 1$, $F_L =$ applied temperature

- b. Prescribed constant heat flux.

Let $K_{21} = KS$, $K_{22} = 0$, $\phi_L(\lambda) = 1$, $F_L =$ applied flux

- c. Insulated boundary.

Let $K_{21} = KS$, $K_{22} = 0$, $\phi_L(\lambda) = 1$, $F_L = 0$

- d. Linear heat transfer at the surface (convection).

Let $K_{21} = KS$, $K_{22} = -hS$, $\phi_L(\lambda) = 1$, $F_L = -hST_c$

This results in a boundary condition equation of the form

$$KS \frac{\partial T}{\partial x} = hS (T - T_c)$$

where h is the usual convective heat transfer coefficient per unit area.

- e. Sign Convention.

The sign convention is such that a positive sign indicates flux into the body.

The boundary conditions described in "a" and "b" above may be arbitrarily varied with time by applying the appropriate time function, $\phi_L(\lambda)$.

SECTION IV

COMPUTER PROGRAM FOR SERIES TRANSIENT
ANALYSIS OF SLAB HEAT TRANSFER (STASH)

A. DESCRIPTION

The program described below was written to solve for the temperature distribution in a one-dimensional rod with arbitrary initial conditions and time-varying boundary conditions. STASH is coded in FORTRAN IV for the IBM 7044-7094 II Direct Coupled-System. Fifteen subprograms make up the program, each of which has a specific task to perform. These subprograms are listed below.

- | | |
|---------|---|
| MAIN | - reads in data, sets up calculations, and prints the results. |
| SOLVE 1 | - solves the eigenvalue equation for positive values of DETK (see Appendix II). |
| SOLVE 2 | - solves the eigenvalue equation for negative values of DETK. |
| SOLVE 3 | - solves the eigenvalue equation for a zero value of DETK. |
| SOLVE 4 | - solves the eigenvalue equation for DETK infinite. |
| FINT | - Simpson's rule integration routine. |
| FUNCX | - sets up the integrand for the x integral |
| FUNCT | - sets up the integrand for the λ integral |
| TABIN | - reads in tabular data, if present |
| INTERP | - performs linear interpolation on tabular data |
| PHIO | - defines the time varying boundary condition at $x = 0$ |
| PHIL | - defines the time varying boundary condition at $x = L$ |
| PHIPRO | - defines the derivative of the time-varying boundary condition at $x = 0$ |
| PHIPRL | - defines the derivative of the time-varying boundary condition at $x = L$ |
| FOFX | - defines the initial conditions in the rod. |

Figure 2 is a simplified flow chart depicting the transfer of information between the subprograms discussed above.

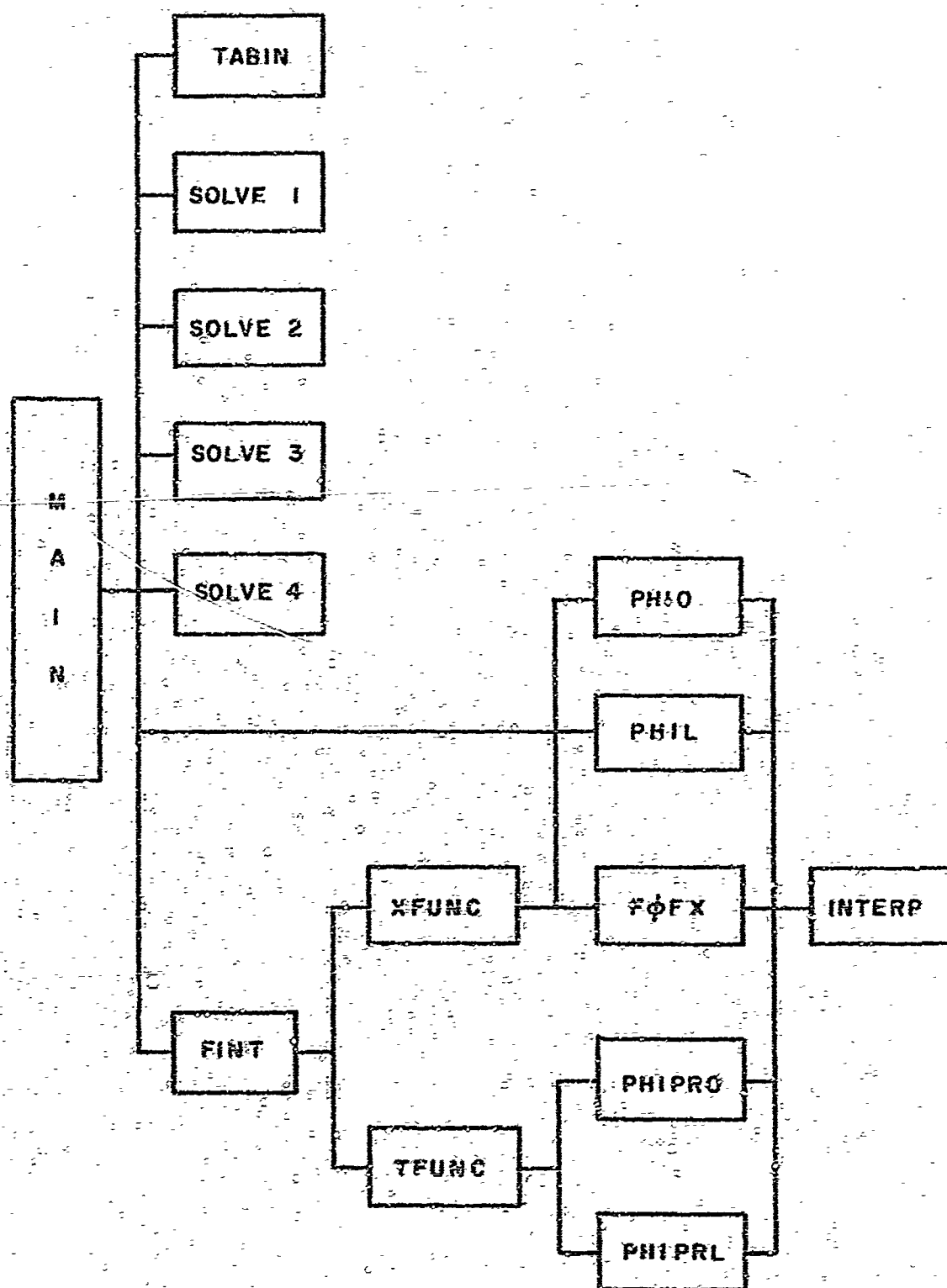


Figure 2. Simplified Flow Chart of Transfer of Information

B. INPUT

There are basically two types of input data to STASH, the physical parameters and the problem parameters. The physical parameters are the characteristics of the rod and the conditions to which it is subjected. The problem parameters are the accuracy parameters, the calculation controls and the print controls.

The rod is characterized by four basic quantities: length, mass density, thermal conductivity and specific heat. The boundary conditions are identified by the indicators K_{11} , K_{12} , K_{21} , K_{22} . The magnitude of the boundary conditions is characterized by the indicators F_0 and F_L . The magnitude of the initial conditions is similarly characterized by the indicator F_x .

Since the solution may be required at any station on the rod for any given point in time, it is convenient to specify a length increment and a time increment as input parameters. A final time is also specified to terminate calculations. The accuracy of the solution is basically governed by three factors: the accuracy of the eigenvalues, the number of terms in the series portion of the solution and the number of intervals taken in the numerical integration routine. In the interest of maximum flexibility, each of these quantities was made an input parameter.

Information may be input to the program in two basic forms. The first type is the data card. There will always be seven cards in the data deck. If the tabular data option is used there may be many more. The second type of input consists of FORTRAN IV statements which may be inserted into the subprograms defining the initial and boundary conditions in the rod. An example problem using both types of input is given in Section V C. Detailed instructions on inputting the data cards are given below, in the following format:

- (1) Card number and contents
- (2) Program name for contents
- (3) Format of card input referenced to format statement number
- (4) Description of each variable on the card

1. Data Cards

Card 1 Intermediate print options

JPRINT(1), JPRINT(2), JPRINT(3)

5000 FORMAT(3I1)

JPRINT(1) - prints series portion of solution term by term
if a one is entered. If no print is desired enter a zero.

JPRINT(2) - prints unsteady state portion of the solution if a
one is entered. If no print is desired enter a zero.

JPRINT(3) - prints solution for eigenvalues if a one is entered.
If no print is desired enter a zero.

Card 2 Title Card

IUNIT, TITLE

1 FORMAT (I2, 13A6)

IUNIT - an indicator which prints out the system of units to be used in the problem. See Table I for a list of systems presently contained in the program.

TITLE - any alphanumeric information through column 80.

Card 3 Physical Parameters (all must be in consistent units)

L, K, RHO, CP, DELTAX, DELTAT, TIMEF

2 FORMAT (7E10.0)

L length of the rod

K thermal conductivity

RHO mass density

CP specific heat

DELTAX increment of length (100 increments maximum)

DELTAT increment of time

TIMEF final time (initial time is zero)

Card 4 Boundary Condition Indicators

K11, K12, K21, K22

3 FORMAT (4E10.0)

K11 indicator for $\frac{\partial T}{\partial x}$ at $x = 0$ K12 indicator for T at $x = 0$ K21 indicator for $\frac{\partial T}{\partial x}$ at $x = L$ K22 indicator for T at $x = L$ Card 5 Function Multiplying Factors

FO, FL, FX

4 FORMAT (3E10.0)

FO coefficient on function $\phi_0(t)$ FL coefficient on function $\phi_L(t)$ FX coefficient on function $f(x)$

TABLE I
SYSTEMS OF UNITS STORED INTERNALLY

IUNIT	LENGTH	MASS	TIME	WEIGHT	TEMPERATURE
1	INCH	SLUG	SEC	POUND	FAHRENHEIT
2	INCH	SLUG	MIN	POUND	FAHRENHEIT
3	INCH	SLUG	HR	POUND	FAHRENHEIT
4	FOOT	SLUG	SEC	POUND	FAHRENHEIT
5	FOOT	SLUG	MIN	POUND	FAHRENHEIT
6	FOOT	SLUG	HR	POUND	FAHRENHEIT
7	INCH	POUND	SEC	POUND	FAHRENHEIT
8	INCH	POUND	MIN	POUND	FAHRENHEIT
9	INCH	POUND	HR	POUND	FAHRENHEIT
10	FOOT	POUND	SEC	POUND	FAHRENHEIT
11	FOOT	POUND	MIN	POUND	FAHRENHEIT
12	FOOT	POUND	HR	POUND	FAHRENHEIT

Card 6 Calculation Parameters

NTERMS, NSTEPX, NSTEPT, NTAB(1), NTAB(2), NTAB(3), NTAB(4),
NTAB(5)

5 FORMAT (3I5, 5I1)

NTERMS	-	number of terms in the series portion of the solution (100 maximum)	
NSTEPX	-	number of intervals for the x-integration	
NSTEPT	-	number of intervals for the λ -integration	
NTAB(1)	-	flag for table 1	} NTAB(I) = 0 Do not use Table
NTAB(2)	-	flag for table 2	
NTAB(3)	-	flag for table 3	
NTAB(4)	-	flag for table 4	} NTAB(I) = 1 Use Table
NTAB(5)	-	flag for table 5	

If NSTEPX or NSTEPT is zero the program sets the value of the respective integral to zero.

Card 7 Eigenvalue Solution Parameters

LIMIT, ITERMX

6 FORMAT (E10.0, I5)

LIMIT	-	difference between two successive iterations necessary to define convergence to a root.
ITERMX	-	maximum number of iterations to be made in searching for each eigenvalue.

If no tabular data is to be used, this is the last card in the data deck. If tabular data is to be an input, however, the following format will be used.

Card 8 Table Number and Comments

NTABLE, COMMENTS

FORMAT (I5, 20x, 53H)

NTABLE table number

COMMENTS any alphanumeric information in columns 26 through 80.

Card 2 Tabular Data ((2 to 50 data cards per table)

INDVAR, DEPVAR, COMMENTS

FORMAT (5X, 2E10.0, 55H)

INDVAR independent variable

DEPVAR dependent variable

COMMENTS any alphanumeric information in columns 26 through 80

Card 3 End of Table

N

FORMAT (I5)

N negative of table number

There are five tables provided in the program, which are assigned as follows:

Table 1 PHIO

Table 2 PHIL

Table 3 PHIPRO

Table 4 PHIPRL

Table 5 EOFFX

Figure 3. shows a symbolic data deck. A sample problem is generated in detail in Section IV C.

2. Subprogram Input Cards

If the tabular data option is not used STAS^u calculates the required functions internally using FORTRAN statements as loaded in the subprograms at compilation time. The affected subprograms are:

FUNCTION PHIO

FUNCTION PHIL

FUNCTION PHIPRO

FUNCTION PHIPRL

FUNCTION EOFFX

As the initial or boundary conditions change, cards containing the functional statement of the variation must be inserted. Since all the above functions have an associated multiplying factor

FORTRAN STATEMENT

CARD 1. PRINT CONTROLS

J 1 = 1 PRINT SERIES SOLUTION TERM BY TERM
 J 1 = 0 DO NOT PRINT
 J 2 = 1 PRINT STEADY STATE SOLUTION
 J 2 = 0 DO NOT PRINT
 J 3 = 1 PRINT EIGENVALUE SOLUTION
 J 3 = 0 DO NOT PRINT

CARD 2. TITLE CARD

IN ANY ALPHANUMERIC DATA
 I 0 PRINTS IDENTIFICATION FLAG (SEE TABLE I)

CARD 3. PHYSICAL PARAMETERS

L K M C P DELTAX DELTAT TIMEY

CARD 4. BOUNDARY CONDITION INDICATORS

K 11 K 12 K 21 K 22

CARD 5. FUNCTION MULTIPLYING FACTORS

F0 F1 F2 F3

CARD 6. CALCULATION PARAMETERS

NTERM ST FX NST PT 1 2 3 4 5

CARD 7. EIGENVALUE SOLUTION PARAMETERS

LIM IT I1-RMX

TABULAR DATA

NTAB

IND VAR	DEP VAR
IND VAR	DEP VAR

ENDT

Figure 3. Input Data Format

it is convenient to use normalized functional statements in the subprogram. Thus, if we entered the following card in the FUNCTION FOFX

$$\text{FOFX} = 1.0$$

we would obtain the initial condition

$$f(x) = \text{FX}$$

The program (Appendix I) contains subprogram statements corresponding to the following initial and boundary conditions.

$$\phi_0(t) = \text{const}$$

$$\phi_L(t) = \text{const}$$

$$\phi'_0(t) = 0$$

$$\phi'_L(t) = 0$$

$$f(x) = \text{const}$$

C. SAMPLE PROBLEM

Consider a rod, ten inches long, having the following properties:

$$K = 10 \text{ Btu/hr-ft-}^\circ\text{F}$$

$$\rho = 500 \text{ lb}_m/\text{ft}^3$$

$$c_p = 0.1 \text{ Btu/lb}_m$$

We wish to obtain a time history of the temperature - distribution through the rod which is subject to the following boundary conditions

$$T(0, t) = t$$

$$T(L, t) = 0$$

The initial conditions imposed on the rod are:

$$T(x, 0) = 0$$

For purposes of illustration the intermediate print option will be called out on card one.

The first task in setting up the problem is to decide upon a system of units to employ. Since the length of the rod is given in inches, we shall choose the inch as the unit of length. For a transient study a small time unit is desirable, hence, the second becomes the unit of time. The remainder of the system of units is determined by the temperature and mass units. Going now to Table I, we obtain the correct value of IUNIT. This indicator along with a suitable title becomes card two in Figure 4.

FORTRAN CODING FORM

[illegible]

Figure 4. Sample Problem Data Deck

Card three contains the physical parameters of the system in the system of units called for by the indicator, IUNIT. These parameters are:

$$L = 10 \text{ in.} \quad K = 0.002314 \text{ Btu/in.-sec-}^\circ\text{F} \quad \text{RHO} = 0.2895 \text{ lb}_m/\text{in.}^3$$

$$\text{CP} = 0.10 \text{ Btu/lb}_m \quad \text{DELTAX} = 0.5 \text{ in.} \quad \text{DELTAT} = 100 \text{ sec} \quad \text{TIME} = 1000 \text{ sec}$$

Since we have only temperature boundary conditions we set $K11 = K21 = 0$ and $K12 = K22 = 1$. This information is entered on card four.

The functions defining our initial conditions are

$$\phi_0(t) = t, \phi_L(t) = 0, f(x) = 0, \phi_0'(\lambda) = 1, \phi_L'(\lambda) = 0$$

We can take advantage of the functions already stored in the program by using F_L and F_x to zero out the appropriate functions. We then can use $F_0 = 1$ to bring in the other boundary condition. This is shown on card five.

The calculation parameters are entered on card six. A series solution containing twenty-five terms is completely adequate for this problem. Since the initial condition function is zero, we set NSTEPX equal to zero. For a Simpson's rule integration scheme fifty steps should suffice for NSTEPT. Our choice of multiplying factors on card five enabled us to do much of the function calculation internally. However, to eliminate recompiling any portion of the program we used tables to define the functions $\phi_0(t)$ and $\phi_0'(\lambda)$. Thus we set NTAB(1) and NTAB(3) equal to one and the rest equal to zero.

For a solution with temperature boundaries only, the eigenvalues become simply $n\pi$. Thus, the eigenvalue parameters have little meaning. However, for a more complex case they would have a significant effect on the solution so these parameters should be made as stringent as required. Typical values are entered on card seven.

Our choice not to recompile any subprograms leads to the use of tables one and three. The first card in each table is a table designation number. The following cards contain data points. The last card of each table contains the negative of the table designation number and is a flag signalling the end of the table.

This, then is the data deck for the sample problem. The assembled deck is shown in Figure 4.

D. RESTRICTIONS

Certain restrictions must be adhered to in order for the solution to be successful. Violation of these restrictions will usually produce an error message from the computer program.

a. A consistent set of units must be employed. An indicator is provided on the title card which will label the system of units on the output. If this indicator is omitted, the following error message is printed:

SYSTEM OF UNITS NOT SPECIFIED. IUNIT NOT ENTERED OR ZERO.

This message merely informs the user of this omission, execution of the problem is not terminated.

b. K_{11} and K_{12} cannot be zero simultaneously. Similarly, K_{21} and K_{22} cannot be zero simultaneously. These situations lead to an undefined boundary. The error message below results from this case.

BOTH INDICATORS AT ONE BOUNDARY ARE ZERO

c. K_{12} cannot be zero. This is a somewhat artificial restriction imposed by the formulation of the problem. In a physical sense it prevents the possibility of the unsolvable Neuman problem. If K_{12} is zero the following error message is printed.

FORMULATION DOES NOT PERMIT K_{12} TO BE ZERO.

d. The number of integration intervals must be even. This restriction arises from the computer formulation of Simpson's rule. If an uneven number is entered, the following error message results:

NUMBER OF INTEGRATION INTERVALS IS NOT EVEN

e. The computer program generates an error message if the time increment or the length increment is zero or negative. This message reads:

TIME OR LENGTH INCREMENT IS ZERO OR NEGATIVE

f. The initial time for each program is zero.

g. The program uses even increments of time and length. The maximum number of length increments is one hundred.

E. OUTPUT

The output generated by STASH consists of two segments: the input data display and temperature profiles which are always generated; and the intermediate print which is controlled by the first card in the data deck. After reading the data, STASH prints it out along with suitable titles and headings as shown in Figure 5. If the eigenvalue solution is requested a table of the eigenvalues, and iterations, is printed as shown in Figure 6. Intermediate print options giving the values of the series and the unsteady state produce output as shown in Figure 7 for each station along the slab at each time step. Figure 7 also shows the form of the temperature profiles as generated at each time step.

CASE 4 INSULATED HEMISPHERE

PHYSICAL CONSTANTS TO DEFINE THE PROBLEM

SYSTEM OF UNITS

	LENGTH INCH	TEMPERATURE FAHRENHEIT	TIME SECOND	WEIGHT POUND	LENGTH INCREMENT	FINAL TIME
0.09999999E 02	0.23147000E-02	0.28950000E-00	0.09999999E-00	0.09999999E 01	0.09999999E 01	0.35000000E 04

BOUNDARY CONDITION INDICATORS FROM THE DIFFERENTIAL EQUATION

K22

K21

0.

0.09999999E 01

0.09999999E 01

0.

MULTIPLYING FACTORS
FOR

BOUNDARY AND INITIAL CONDITION FUNCTIONS

F(1)

F(1)

0.

0.

0.09999999E 04

0.

0.

0.09999999E 03

0.

0.

0.09999999E 03

0.

0.

0.

CALCULATION PARAMETERS

NUMBER OF TERMS
IN THE SUMMATIONNUMBER OF INTERVALS
FOR THE X INTEGRATIONNUMBER OF INTERVALS
FOR THE Y INTEGRATION

100

100

100

100

100

100

100

100

100

100

100

100

100

100

EIGENVALUE SOLUTION PARAMETERS

ACCURACY
LIMITNUMBER OF
ITERATIONS

0.99999999E-08

200

ALPHA

0.77730715E-01

DEPK

-0.09999999E 01

Figure 8. Sample Output (Case 4) Showing Display of Input

SOLUTION FOR EIGENVALUES

ROOT NO

1
2
3
4
5
6
7
8
9
10

Z

0.15707962E 01
9.47123838E 01
0.78539814E 01
0.10995574E 02
0.14137166E 02
0.17278759E 02
0.20420352E 02
0.23531944E 02
0.26703537E 02
0.29845123E 02

Figure 5. Sample Output (Case 4) Showing Eigenvalue Solution

```

X= 0.0000000E 03
X= 0.09999999E 02

SERIES PORTION OF SOLUTION

TERM          SUMMATION
-0.11827815E 03
-0.13002856E-03
-0.84859587E-15
-0.81741985E-31
-0.
-0.
-0.
-0.
-0.

UNSTEADY STATE PORTION OF SOLUTION

X          TEMP
0.09999999E 02  0.09999999E 04
TEMPERATURE DISTRIBUTION ALONG THE ROD AT 1.000 INTERVALS
0.09999999E 04  0.96299432E 03  0.92689987E 03  0.89260542E 03  0.86095545E 03
0.93272925E 03  0.90062185E 03  0.78922683E 03  0.77502171E 03  0.76635632E 03
0.76344399E 03

```

Figure 7. Sample Output (Case 4) Showing Intermediate Print

SECTION V

CONCLUSIONS

Several classes of problems were run to check out the program. The results were compared with a finite-difference heat transfer program (LTA) which was developed by Lockheed Aircraft Corporation. These results were examined for accuracy and speed of convergence.

The intermediate print feature of the program was used to determine the convergence. Examination showed that convergence was quite rapid, usually less than ten terms, for constant boundary conditions, away from time zero. More terms were needed in the vicinity of zero time to produce convergence. For time-varying boundary conditions, the series does not converge very quickly. However, a study of the solution convergence showed that a twenty term series using one hundred integration steps obtained results within one percent of the LTA solution for times exceeding one hundred seconds. At smaller times a fifty term series with one hundred integration steps was required.

Appendix III contains the results of five check problems compared with the results from the LTA finite difference program.

No comparison is made at zero time since the program obtains these values from the initial conditions rather than from a series calculation. The curves are plotted from data generated by the program. Case 1 is the sample problem detailed in Section IV C. All other cases used the same physical parameters.

APPENDIX I
COMPUTER PROGRAM SOURCE LISTING

```

000JCB STASH MAP STASH001
$IDFTC MAIN P94/2,XR7 STASH002
C STASH003
C A GENERAL SOLUTION TO THE ONE-DIMENSIONAL HEAT TRANSFER PROBLEM STASH004
C WITH TIME-DEPENDENT BOUNDARY CONDITIONS AND ARBITRARY INITIAL STASH005
C CONDITION STASH006
C VERSION 2 STASH007
C VERSION 2 CALCULATES ZERO TIME USING BOUNDARY AND INITIAL CONDITIONS STASH008
C VERSION 2 INCORPORATES A MULTIPLIER ON THE INITIAL CONDITION FUNCTION STASH009
C AND BOUNDARY CONDITION FUNCTIONS TO PERMIT THE USE OF NORMALIZED STASH010
C FUNCTIONS STASH011
C STASH012
C EXTERNAL TFUNC,XFUNC STASH013
C DIMENSION TEMP(100),TITLE(13),EIGEN(100) STASH014
C DIMENSION JPRINT(3) STASH015
C DIMENSION NTAB(5) STASH016
C REAL K11BN,K11BN2 STASH017
C REAL K11K21,K12K22 STASH018
C REAL LTHIT,L,K,LAMBDA,NUMXSS,NUMSS,KFPHI,KFPHIO,KFPHIL,K11,K12 STASH019
C I,K21,K22,N,KTERM1,KTERM2 STASH020
C COMMON T,X,LAMBDA,KTERM1,KTERM2,ALPHA,BETAN,EXPML,CBNX,SNX,DELTA STASH021
C TX,DELTAT,K11,K12,K21,K22,FX,NTAB STASH022
C COMMON/ROCTS/NTERMS,L,ITERMX,K11K21,K12K22,LIMIT STASH023
C COMMON/PRINT/JPRINT STASH024
C STASH025
C FORMAT STATEMENTS STASH026
C STASH027
1 FORMAT(12,13A6) STASH028
2 FORMAT(7E10.0) STASH029
3 FORMAT(4E10.0) STASH030
4 FORMAT(3E10.0) STASH031
5 FORMAT(315.511) STASH032
6 FORMAT(E10.0,15) STASH033
11 FORMAT(1H1,29X,13A6) STASH034
20 FORMAT(1H0,8X,6HLENGTH,12X,7HTHERMAL,11X,7HDEENSITY,10X,8HSPECIFIC, STASH035
111X,6HLENGTH,13X,4HTIME,14X,5HFINAL/24X,12HCONDUCTIVITY,28X,4HHEAT STASH036
2,11X,9HINCREMENT,9X,9HINCREMENT,12X,4HTIME) STASH037
22 FORMAT(1H0,3X,E15.8,3X,E15.8,4X,E15.8,2X,E15.8,3(3X,E15.8)) STASH038
30 FORMAT(1HA,35X,60HBOUNDARY CONDITION INDICATORS FROM THE DIFFEREN STASH039
1IAL EQUATION//36X,3HK11,16X,3HK12,16X,3HK21,16X,3HK22) STASH040
31 FORMAT(1HA,55X,19HMULTIPLYING FACTORS/64X,3HFOR/45X,40HBOUNDARY AN STASH041
10 INITIAL CONDITION FUNCTIONS//30X,4HF(0),30X,4HF(L),30X,4HF(X)//2 STASH042
24X,2(E15.8,19X),E15.8) STASH043
33 FORMAT(1H0,29X,4(E15.8,4X)) STASH044
66 FORMAT(1HA,51X,30HEIGENVALUE SOLUTION PARAMETERS//42X,8HACCURACY,3 STASH045
*0X,9HNUMBER OF/44X,5HLIMIT,31X,10HITERATIONS//39X,E15.8,26X,15) STASH046
120 FORMAT(28X,4HINCH,14X,4HSLUG,14X,3HMIN,16X,5HPOUND,13X,10HFAHRENHE STASH047
117) STASH048
130 FORMAT(28X,4HINCH,14X,4HSLUG,14X,3HMIN,16X,5HPOUND,13X,10HFAHRENHE STASH049
117) STASH050
140 FORMAT(28X,4HINCH,14X,4HSLUG,15X,2HHR,16X,5HPOUND,13X,10HFAHRENHE STASH051
17) STASH052
150 FORMAT(28X,4HFCUT,14X,4HSLUG,14X,3HSEC,16X,5HPOUND,13X,10HFAHRENHE STASH053
117) STASH054
160 FORMAT(28X,4HFCOT,14X,4HSLUG,14X,3HMIN,16X,5HPOUND,13X,10HFAHRENHE STASH055
117) STASH056
170 FORMAT(28X,4HFOOT,14X,4HSLUG,15X,2HHR,16X,5HPOUND,13X,10HFAHRENHE STASH057
17) STASH058
180 FORMAT(28X,4HINCH,14X,5HPOUND,13X,3HSEC,16X,5HPOUND,13X,10HFAHRENHE STASH059
1E17) STASH060
190 FORMAT(28X,4HINCH,14X,5HPOUND,13X,3HMIN,16X,5HPOUND,13X,10HFAHRENHE STASH061
1E17) STASH062

```

```

303  FORMAT(1HA,54X,22H)CALCULATION PARAMETERS//26X,15HNUMBER OF TERMS,1STASH063
      15X,19HNUMBER OF INTERVALS,13X,19HNUMBER OF INTERVALS/25X,16HIN THESTASH064
      2 SUMMATION,14X,21HFOR THE X INTEGRATION,11X,21HFOR THE Y INTEGRATIONSTASH065
      3CN//30X,15,28X,15,26X,15) STASH066
333  FORMAT(1HA,63X,5HALPHA) STASH067
1002 FORMAT(26X,5E15.8) STASH068
1003 FORMAT(1X,E15.8) STASH069
1100 FORMAT(28X,4HINCH,14X,5HPOUND,14X,2HHR,16X,5HPOUND,13X,10HFAHRENHSTASH070
      1IT) STASH071
1110 FORMAT(28X,4HFCOT,14X,5HPOUND,13X,3HSEC,16X,5HPOUND,13X,10HFAHRENHSTASH072
      1EIT) STASH073
1120 FORMAT(28X,4HFCOT,14X,5HPOUND,13X,3HMIN,16X,5HPOUND,13X,10HFAHRENHSTASH074
      1EIT) STASH075
1130 FORMAT(28X,4HFCOT,14X,5HPOUND,14X,2HHR,16X,5HPOUND,13X,10HFAHRENHSTASH076
      1IT) STASH077
1140 FORMAT(1HA,45X,40H)PHYSICAL CONSTANTS TO DEFINE THE PROBLEM//59X,15STASH078
      1HSYSTEM OF UNITS//27X,6HLENGTH,13X,4HMASS,14X,4HTIME,14X,6HWEIGHT,STASH079
      213X,11HTEMPERATURE) STASH080
2002 FORMAT(7X,4HTIME,22X,42HTEMPERATURE DISTRIBUTION ALONG THE ROD AT STASH081
      1 ,F3.3,10H INTERVALS) STASH082
3003 FORMAT(1H) STASH083
3333 FORMAT(1H0,58X,E15.8) STASH084
4004 FORMAT(1HA,63X,4HCEYK) STASH085
4444 FORMAT(1H0,58X,E15.8) STASH086
5000 FORMAT(5I1) STASH087
6100 FORMAT(1H1,25X,2HT=,E15.8,20X,2HX=,E15.8//45X,26H)SERIES PORTION OFSTASH088
      * SOLUTION//14X,1HN,16X,4HTERM,19X,9HSUMMATION//) STASH089
6110 FORMAT(10X,15,10X,E15.8,10X,E15.8) STASH090
6200 FORMAT(1H0,40X,34H)UNSTEADY STATE PORTION OF SOLUTION//35X,1HX,25X,STASH091
      *4HTEMP//) STASH092
6210 FORMAT(28X,F15.8,20X,E15.8) STASH093
      IERROR=0 STASH094
C STASH095
C INTERMEDIATE PRINT OPTIONS STASH096
C STASH097
C READ(5,5000) JPRINT STASH098
C STASH099
C JPRINT=1,PRINT INTERMEDIATE CALCULATIONS. JPRINT=0,DO NOT PRINT. STASH100
C JPRINT(1)-SERIES PORTION OF THE SOLUTION TERM BY TERM STASH101
C JPRINT(2)-UNSTEADY STATE PORTION OF THE SOLUTION STASH102
C JPRINT(3)-SOLUTION FOR THE EIGENVALUES STASH103
C STASH104
C READ (5,1)IUNIT,(TITLE(I),I=1,13) STASH105
C READ(5,2)L,K,RHO,CP,DELTA X,DELTA T,TIMEF STASH106
C READ(5,3)K11,K12,K21,K22 STASH107
C READ(5,4)FO,FL,FY STASH108
C READ(5,5) NTERMS,NSTEPX,NSTEPT,NTAB STASH109
C READ(5,6) LIMIT,ITERMX STASH110
C STASH111
C STASH112
C PRINT INPUT DATA STASH113
C STASH114
C TITLE STASH115
C STASH116
C WRITE (6,21)(TITLE(I),I=1,13) STASH117
C STASH118
C SYSTEM OF UNITS STASH119
C STASH120
C WRITE(6,1140) STASH121
C IF(IUNIT.EQ.0) GO TO 5500 STASH122
C GO TO (102,103,104,105,106,107,108,109,110,111,112,113),IUNIT STASH123
102 WRITE(6,120) STASH124

```

103	GO TO 9999	STASH125
	WRITE(6,130)	STASH126
	GO TO 9999	STASH127
104	WRITE(6,140)	STASH128
	GO TO 9999	STASH129
105	WRITE(6,150)	STASH130
	GO TO 9999	STASH131
106	WRITE(6,160)	STASH132
	GO TO 9999	STASH133
107	WRITE(6,170)	STASH134
	GO TO 9999	STASH135
108	WRITE(6,180)	STASH136
	GO TO 9999	STASH137
109	WRITE(6,190)	STASH138
	GO TO 9999	STASH139
110	WRITE(6,1100)	STASH140
	GO TO 9999	STASH141
111	WRITE(6,1110)	STASH142
	GO TO 9999	STASH143
112	WRITE(6,1120)	STASH144
	GO TO 9999	STASH145
113	WRITE(6,1130)	STASH146
	GO TO 9999	STASH147
5500	WRITE(6,5550)	STASH148
C		STASH149
C	PHYSICAL CONSTANTS	STASH150
C		STASH151
9999	WRITE(6,20)	STASH152
	WRITE(6,22) L,K,RHO,CP,DELTA X,DELTA T,TIME F	STASH153
C		STASH154
C	BOUNDARY CONDITIONS	STASH155
C		STASH156
	WRITE(6,30)	STASH157
	WRITE(6,33) K11,K12,K21,K22	STASH158
C		STASH159
C	MULTIPLYING FACTORS	STASH160
C		STASH161
	WRITE(6,31) F0,FL,FX	STASH162
C		STASH163
C	CALCULATION PARAMETERS	STASH164
C		STASH165
	WRITE(6,303) INTERMS,NSTEPX,NSTEPT	STASH166
C		STASH167
C	EIGENVALUE SOLUTION PARAMETERS	STASH168
C		STASH169
	WRITE(6,36) LIMIT,IYERMX	STASH170
C		STASH171
C	TEST FOR TABULAR DATA	STASH172
C		STASH173
	DO 50 I=1,5	STASH174
	IF (INTAB(I).NE.0) CALL YASIN(I)	STASH175
50	CONTINUE	STASH176
C		STASH177
C	IF TABLES ARE USED, THEY ARE ASSIGNED AS FOLLOWS	STASH178
C		STASH179
C	NO.1 FUNCTION PHIC(T)	STASH180
C	NO.2 FUNCTION PHILE(T)	STASH181
C	NO.3 FUNCTION PHIPRO(LAMBDA)	STASH182
C	NO.4 FUNCTION PHIPRL(LAMBDA)	STASH183
C	NO.5 FUNCTION FOFX(X)	STASH184
C		STASH185
		STASH186

C		STASH187
C	ERROR CHECKS ON INPUT DATA	STASH188
C		STASH189
C		STASH190
C	NUMBER OF INTEGRATION STEPS MUST BE EVEN OR ZERO.	STASH191
C	IF ZERO THE PROGRAM SETS THE INTEGRAL EQUAL TO ZERO.	STASH192
C		STASH193
	IF(NSTEPX.EQ.0) GO TO 248	STASH194
	IF(MOD(NSTEPX,2).EQ.0) GO TO 248	STASH195
	WRITE (6,5553)	STASH196
	IERROR=IERROR+1	STASH197
248	IF(NSTEPT.EQ.0) GO TO 249	STASH198
	IF(MOD(NSTEPT,2).EQ.0) GO TO 249	STASH199
	WRITE(6,5553)	STASH200
	IERROR=IERROR+1	STASH201
249	IF((DELTA X.GT.0.)	STASH202
	* .AND.	STASH203
	* (DELTA T.GT.0.))	STASH204
	* GO TO 250	STASH205
	WRITE(6,5551)	STASH206
	IERROR=IERROR+1	STASH207
250	IF(TIMEF.GT.0.) GO TO 251	STASH208
	WRITE(6,5552)	STASH209
	IERROR=IERROR+1	STASH210
C		STASH211
C	K12 CANNOT BE ZERO DUE TO A RESTRICTION IN THE FORMULATION.	STASH212
C		STASH213
251	IF (K12.NE.0.) GO TO 252	STASH214
	WRITE(6,5555)	STASH215
	IERROR=IERROR+1	STASH216
C		STASH217
C	CHECK FOR UNDEFINED BOUNDARY CONDITIONS.	STASH218
C		STASH219
252	IF(((K11.NE.0.).OR.(K12.NE.0.))	STASH220
	* .AND.	STASH221
	* ((K21.NE.0.).OR.(K22.NE.0.)))	STASH222
	* GO TO 253	STASH223
	WRITE(6,441	STASH224
	IERROR=IERROR+1	STASH225
C		STASH226
253	IF(IERROR.GT.0) STOP	STASH227
C		STASH228
	WRITE(6,333)	STASH229
	PI=3.1415926	STASH230
	ALPHA=K/(RHO*CP)	STASH231
	WRITE(6,3333) ALPHA	STASH232
	DETK=K11*K22-K12*K21	STASH233
	WRITE(6,4004)	STASH234
	WRITE(6,4444) DETK	STASH235
C		STASH236
C	DETERMINE CONSTANTS FOR USE IN DO LOOPS	STASH237
C		STASH238
	CON01=K12*K22*L-DETK	STASH239
	CON01=K11*K12*L	STASH240
	CON02=K11**2	STASH241
	CON03=(K12*L)**2	STASH242
	CON04=FL	STASH243
	CON05=K11*K21	STASH244
	CON06=(K12*K22*(L**2))	STASH245
	CON07=DETK*L	STASH246
	CON08=F0	STASH247
	CON09=K12*FL	STASH248

CON10=K22*F0	STASH249
CON11=F0/K12	STASH250
CON12=K12*DENOM	STASH251
CON088=K12**2	STASH252
K11K21=K11*K21	STASH253
K12K22=K12*K22	STASH254
C	STASH255
C DETERMINE EIGENVALUES FOR SERIES SOLUTION	STASH256
C	STASH257
IF((K11K21.EQ.0.).AND.(K12K22.EQ.0.))GO TO 400	STASH258
IF(DELTA/20,300,100	STASH259
100 CALL SOLVE1(DETK,EIGEN,NTERMS)	STASH260
GO TO 900	STASH261
200 CALL SOLVE2(DETK,EIGEN,NTERMS)	STASH262
GO TO 900	STASH263
300 CALL SOLVE3(DETK,EIGEN,NTERMS)	STASH264
GO TO 900	STASH265
400 CALL SOLVE4(DETK,EIGEN,NTERMS)	STASH266
C	STASH267
C SET UP INDICES FOR DO LOOPS	STASH268
C	STASH269
900 NTEMP=(TIMEFF/DELTA)+1.5	STASH270
NTEMPX=(L/DELTA)+1.5	STASH271
WRITE(6,3003)	STASH272
T=-DELTA	STASH273
DO 199 N7=1,NTEMP	STASH274
T=T+DELTA	STASH275
X=-DELTA	STASH276
DO99NX=1,NTEMPX	STASH277
IF(T.EQ.0.) GO TO 299	STASH278
X=X+DELTA	STASH279
SERIES=0.	STASH280
IF(JPRINT(1).NE.0) WRITE(6,6100) T,X	STASH281
C	STASH282
C SET UP LOOP TO GENERATE SERIES SUMMATION FOR TRANSIENT SOLUTION	STASH283
C	STASH284
CO999I=1,NTERMS	STASH285
ZN=EIGEN(I)	STASH286
9 BETAN=ZN/L	STASH287
ZN2=ZN**2	STASH288
ZDENOM=ZN2*DENOM	STASH289
SBNX=SIN(BETAN*X)	STASH290
CBNX=COS(BETAN*X)	STASH291
SZN=SIN(ZN)	STASH292
CZN=COS(ZN)	STASH293
K11BN=K11*BETAN	STASH294
K11BN2=K11BN**2	STASH295
ZN2SZN=ZN2*SZN	STASH296
EXPUL=EXP(-(ALPHA*(BETAN)**2)*T)	STASH297
SUMCON=BETAN*(K12*SBNX-K11BN*CBNX)/(ZN*(K11BN2+CON088)+(K11BN2-	STASH298
CON088)*SZN+CZN-2.*K11BN*K12*(SZN**2))	STASH299
KTERM1=((CON01*ZN2SZN-CON02*ZN2SZN+CON03*(ZN*CZN-SZN))/ZDENOM)+	STASH300
CON04	STASH301
KTERM2=((CON05*ZN2SZN+CON06*(SZN-ZN)+CON07*ZN*(1.-CZN))/ZDENOM)+	STASH302
CON08	STASH303
KFPHIL=KTERM1*PHIL(0.)	STASH304
KFPHIO=KTERM2*PHIO(0.)	STASH305
KFPHI=EXPUL*(KFPHIO+KFPHIL)	STASH306
IF(NSTEPT.EQ.0) GOTO 399	STASH307
TIMEINT=FIAT(0.,T,NSTEPT,TFUNC)	STASH308
GO TO 599	STASH309
399 TIMEINT=0.	STASH310

DATA PI/3.1415926/	STASH373
SLOPE =DEK/(K12K22*L)	STASH374
IF(JPRINT(3).NE.0) WRITE(6,6300)	STASH375
DO999 I=1,NTERMS	STASH376
NN=I	STASH377
IF(K11K21.EQ.0.) GO TO 801	STASH378
GO TO 799	STASH379
801 IF(ABS(SLOPE).LT.1.0) GO TO 800	STASH380
799 Z0=(2.*NN-1.)*(PI/2.)	STASH381
GO TO 850	STASH382
800 Z0=(2.*NN+1.)*(PI/2.)	STASH383
850 BOUND=0.0	STASH384
DO 99 J=1,ITERMX	STASH385
U=DEK*L*Z0/(K11K21*Z0**2+K12K22*L**2)	STASH386
IF(ABS(SLOPE).LT.1.0) GO TO 860	STASH387
Z1=(NN-1.)*PI+ATAN(U)	STASH388
GO TO 870	STASH389
860 Z1=NN*PI+ATAN(U)	STASH390
870 IF(ABS(Z1-Z0).LT.LIMIT) GO TO 9	STASH391
IF(BOUND.EQ.(Z1-Z0)) GO TO 90	STASH392
BOUND=Z1-Z0	STASH393
Z0=Z1	STASH394
IF(JPRINT(3).NE.0) WRITE(6,6310) I,J,U,Z1	STASH395
CONTINUE	STASH396
WRITE(6,11) I,ITERMX	STASH397
STOP	STASH398
9 EIGEN(I)=Z1	STASH399
GO TO 999	STASH400
90 EIGEN(I)=(Z1+Z0)/2.0	STASH401
999 CONTINUE	STASH402
RETURN	STASH403
C	STASH404
C	STASH405
C	STASH406
11 FORMAT(10X,12HROOT NUMBER ,I3,22H010 NOT CONVERGE AFTER,I5,11H ITERATIONS)	STASH407
END	STASH408
86	STASH409
*IBFTC SOLV2 #94/2,XRT	STASH410
SUBROUTINE SOLVE2(DEK,EIGEN,NTERMS)	STASH411
C	STASH412
C	STASH413
C	STASH414
SOLVES TAN(Z)=-D*L*Z/(K11*K21*Z**2+K22*K12*L**2)	STASH415
C	STASH416
DIMENSION EIGEN(NTERMS)	STASH417
DIMENSION JPRINT(3)	STASH418
DIMENSION NTAB(5)	STASH419
REAL K11K21,K12K22,L,LIMIT,NN	STASH420
REAL LIMIT,L,K,LAMBDA,NUMXS,NUMSS,KFPHI,KFPHIO,KFPHIL,K11,K12	STASH421
I,K21,K22,N,KTERM1,KTERM2	STASH422
COMMON T,X,LAMBDA,KTERM1,KTERM2,ALPHA,BETAK,EXPUL,CBNX,SBNX,DELTA	STASH423
IX,DELTAT,K11,K12,K21,K22,IX,NTAB	STASH424
COMMON/ROOTS/NTERMS,L,ITERMX,K11K21,K12K22,LIMIT	STASH425
COMMON/PRINT/JPRINT	STASH426
6300 FORMAT(1H1,46X,24HSOLUTION FOR EIGENVALUES//26X,7HROOT NO,5X,9HITERATIONS,10X,1HU,24X,1HZ//)	STASH427
6310 FORMAT(28X,I3,10X,I3,6X,E15.8,10X,E15.8)	STASH428
DATA PI/3.1415926/	STASH429
IF(JPRINT(3).NE.0) WRITE(6,6300)	STASH430
DO999 I=1,NTERMS	STASH431
NN=I	STASH432
Z0=(NN+1.)*PI	STASH433
850 BOUND=0.0	STASH434

```

DATA PI/3.1415926/
SLOPE =DETK/(K12K22*L)
IF(JPRINT(3).NE.0) WRITE(6,6300)
CO999 I=1,NTERMS
NN=I
IF(K11K21.EQ.0.) GO TO 801
GO TO 799
801 IF(ABS(SLOPE).LT.1.0) GO TO 800
799 Z0=(2.*NN-1.)*(PI/2.)
GO TO 850
800 Z0=(2.*NN+1.)*(PI/2.)
850 BOUND=0.0
CO 99 I=1,ITERMX
U=DETK*L*Z0/(K11K21*Z0**2+K12K22*L**2)
IF(ABS(SLOPE).LT.1.0) GO TO 860
Z1=(NN-1.)*PI+ATAN(U)
GO TO 870
860 Z1=NN*PI+ATAN(U)
870 IF(ABS(Z1-Z0).LT.LIMIT) GO TO 9
IF(BOUND.EQ.(Z1-Z0)) GO TO 90
BOUND=Z1-Z0
Z0=Z1
IF(JPRINT(3).NE.0) WRITE(6,6310) I,J,U,Z1
99 CONTINUE
WRITE(6,11) I,ITERMX
STOP
9 EIGEN(I)=Z1
GO TO 999
90 EIGEN(I)=(Z1+Z0)/2.0
999 CONTINUE
RETURN
C
C          ERRCR MESSAGES
C
11 FORMAT(10X,12HROOT NUMBER ,71,23HDO NOT CONVERGE WITHIN 11H ITERATIONS)
END
66
$IOFTC SOLV2 #94/2,XR7
SUBROUTINE SOLVE2(DETK,EIGEN,NTERMS)
C
C   SOLVES TAN(Z)=-D*L*Z/(K11*Z**2+K22*K12*L**2)
C
C   DIMENSION EIGEN(NTERMS)
C   DIMENSION JPRINT(3)
C   DIMENSION NYAB(5)
C   REAL K11K21,K12K22,L,K11,K12,K21,K22,N,KTERM1,KTERM2
C   REAL LIMIT,L,K,LAMBDA,ALPHA,BETA,EXOMUL,CORR,SNMX,DELTA
C   COMMON T,X,LAMBDA,KTERM1,KTERM2,ALPHA,BETA,EXOMUL,CORR,SNMX,DELTA
C   IX,DELTAT,K11,K12,K21,K22,FE,FIAB
C   COMMON/ROOTS/NTERMS,L,ITERMX,K11K21,K12K22,LIMIT
C   COMMON/PRINT/JPRINT
6300 FORMAT(1H1,46X,71HROOT NUMBER FOR EIGENVALUES//26X,71HDO NOT CONVERGE WITHIN 11H ITERATIONS)
6310 FORMAT(26X,13,12X,71HDO NOT CONVERGE WITHIN 11H ITERATIONS)
DATA PI/3.1415926/
IF(JPRINT(3).NE.0) WRITE(6,6300)
CO999 I=1,NTERMS
NN=I
Z0=(NN+1.)*PI
850 BOUND=0.0

```

STASH373
 STASH374
 STASH375
 STASH376
 STASH377
 STASH378
 STASH379
 STASH380
 STASH381
 STASH382
 STASH383
 STASH384
 STASH385
 STASH386
 STASH387
 STASH388
 STASH389
 STASH390
 STASH391
 STASH392
 STASH393
 STASH394
 STASH395
 STASH396
 STASH397
 STASH398
 STASH399
 STASH400
 STASH401
 STASH402
 STASH403
 STASH404
 STASH405
 STASH406
 STASH407
 STASH408
 STASH409
 STASH410
 STASH411
 STASH412
 STASH413
 STASH414
 STASH415
 STASH416
 STASH417
 STASH418
 STASH419
 STASH420
 STASH421
 STASH422
 STASH423
 STASH424
 STASH425
 STASH426
 STASH427
 STASH428
 STASH429
 STASH430
 STASH431
 STASH432
 STASH433
 STASH434

```

DO 99 J=1,ITERMX
U=DETK*L*Z0/(K11K21*Z0**2+K12K22*L**2)
Z1=(NN*PI)+ATAN(U)
IF(ABS(Z1-Z0).LT.LIMIT) GO TO 9
IF(BOUND.EQ.(Z1-Z0)) GO TO 90
BOUND=Z1-Z0
Z0=Z1
IF(JPRINT(3).NE.0) WRITE(6,6310) I,J,U,Z1
99 CONTINUE
WRITE(6,11) I,ITERMX
STOP
9 EIGEN(I)=Z1
GO TO 999
90 EIGEN(I)=(Z1+Z0)/2.0
999 CCNTINUE
RETURN
C
C ERROR MESSAGES
C
11 FORMAT(10X,12HROOT NUMBER ,I3,22HDID NOT CONVERGE AFTER,I5,11H ITERATIONS)
END
$*
$IBFYC SOLV3 P94/2,XR7
SUBROUTINE SOLVE3(DETK,EIGEN,NTERMS)
C
C SOLVES TAN(Z)=0.0
C
DIMENSION EIGEN(NTERMS)
DIMENSION JPRINT(3)
DIMENSION NTAB(5)
REAL K11K21,K12K22,L,LIMIT,NN
REAL LIMIT,L,K,LAMBDA,NUMXSS,NUMSS,KFPHI,KFPHIO,KFPHIL,K11,K12
I,K2C,K22,N,KTERM1,KTERM2
COMMON T,X,LAMBDA,KTERM1,KTERM2,ALPHA,BETAN,EXPUL,CBNX,SENX,DELTA
IX,DELTAT,K11,K12,K21,K22,FX,NTAB
COMMON/ROOTS/NTERMS,L,ITERMX,K11K21,K12K22,LIMIT
COMMON/PRINT/JPRINT
6300 FORMAT(1H1,46X,24HSOLUTION FOR EIGENVALUES//26X,7HROOT NO,49X,1H2)
6320 FORMAT(28X,I3,44X,E15.8)
DATA PI/3.1415926/
IF(JPRINT(3).NE.0) WRITE(6,6300)
DO 999 I=1,NTERMS
NN=I
Z1=NN*PI
EIGEN(I)=Z1
IF(JPRINT(3).NE.0) WRITE(6,6320) I,Z1
999 CONTINUE
RETURN
$*
$IBFYC SOLV4 P94/2,XR7
SUBROUTINE SOLVE4(DETK,EIGEN,NTERMS)
C
C SOLVES TAN(Z)= INFINITY
C
DIMENSION EIGEN(NTERMS)
DIMENSION NTAB(5)
DIMENSION JPRINT(3)
REAL K11K21,K12K22,L,LIMIT,NN
REAL LIMIT,L,K,LAMBDA,NUMXSS,NUMSS,KFPHI,KFPHIO,KFPHIL,K11,K12
I,K21,K22,N,KTERM1,KTERM2

```

STASH435
 STASH436
 STASH437
 STASH438
 STASH439
 STASH440
 STASH441
 STASH442
 STASH443
 STASH444
 STASH445
 STASH446
 STASH447
 STASH448
 STASH449
 STASH450
 STASH451
 STASH452
 STASH453
 STASH454
 STASH455
 STASH456
 STASH457
 STASH458
 STASH459
 STASH460
 STASH461
 STASH462
 STASH463
 STASH464
 STASH465
 STASH466
 STASH467
 STASH468
 STASH469
 STASH470
 STASH471
 STASH472
 STASH473
 STASH474
 STASH475
 STASH476
 STASH477
 STASH478
 STASH479
 STASH480
 STASH481
 STASH482
 STASH483
 STASH484
 STASH485
 STASH486
 STASH487
 STASH488
 STASH489
 STASH490
 STASH491
 STASH492
 STASH493
 STASH494
 STASH495
 STASH496

```

COMMON T,X,LAMBDA,KTERM1,KTERM2,ALPHA,BETAN,EXPUL,CBNX,SBNX,DELTA STASH497
IX,DELTAT,K11,K12,K21,K22,FX,NTAB STASH498
COMMON/ROOTS/INTERPS,L,ITERMX,K11K21,K12K22,LIMIT STASH499
COMMON/PRINT/JPRINT STASH500
0300 FORMAT(1H1,46X,24F,SOLUTION FOR EIGENVALUES//26X,7HROOT NO,49X,1HZ) STASH501
6320 FORMAT(28X,13,44X,E15.8) STASH502
DATA PI/3.1415926/ STASH503
IF(JPRINT(3).NE.0) WRITE(6,6300) STASH504
DO 999 I=1,NTERMS STASH505
  NN=I STASH506
  Z1=((2.*NN-1.)*PI)/2.0 STASH507
  EIGEN(I)=Z1 STASH508
  IF(JPRINT(3).NE.0) WRITE(6,6320)I,Z1 STASH509
999 CONTINUE STASH510
  RETURN STASH511
  END STASH512
$S
$IBFTC SIMPS  M94/2,XR7 STASH513
  REAL FUNCTION FINT(A,B,NN,F) STASH514
C STASH515
C SIMPSON METHOD STASH516
C A=LOWER BOUND STASH517
C B=UPPER BOUND STASH518
C NN=NUMBER OF INTERVALS, MUST BE EVEN STASH519
C F=FUNCTION TO BE INTEGRATED, FUNCTION NAME MUST BE DECLARED STASH520
C EXTERNAL STASH521
C STASH522
  REAL LIMIT,L,K,LAMBDA,NUMXSS,NUMSS,XFPHI,KFPHIO,KFPHIL,K11,K12
  I,K21,K22,N,KTERM1,KTERM2 STASH523
  COMMON T,X,LAMBDA,KTERM1,KTERM2,ALPHA,BETAN,EXPUL,CBNX,SBNX,DELTA STASH524
  IX,DELTAT,K11,K12,K21,K22,FX,NTAB STASH525
  FN=NN STASH526
  H=(B-A)/FN STASH527
  SUMA=0.0 STASH528
  SUMB=0.0 STASH529
  F=FN-1 STASH530
  COIOJ=1,H,2 STASH531
  FJ=F STASH532
  XX=FJ*H STASH533
10 SUMA=SUMA+F(XX) STASH534
  F=FN-2 STASH535
  CO2OKK=2,H,2 STASH536
  FK=K STASH537
  XX=FK*H STASH538
  SUMB=SUMB+F(XX) STASH539
20 CONTINUE STASH540
  FINT=(H/3.)*((F(A)+F(B)+4.*SUMA+2.*SUMB)) STASH541
  RETURN STASH542
  END STASH543
$S
$IBFTC FUNCX  M94/2,XR7 STASH544
  FUNCTION XFUNC(XX) STASH545
C STASH546
C THIS FUNCTION SETS UP THE INTEGRAND FOR THE / INTEGRAL STASH547
C STASH548
C STASH549
  DIMENSION NTAB(5) STASH550
  REAL LIMIT,L,K,LAMBDA,NUMXSS,NUMSS,XFPHI,KFPHIO,KFPHIL,K11,K12 STASH551
  I,K21,K22,N,KTERM1,KTERM2 STASH552
  COMMON T,X,LAMBDA,KTERM1,KTERM2,ALPHA,BETAN,EXPUL,CBNX,SBNX,DELTA STASH553
  IX,DELTAT,K11,K12,K21,K22,FX,NTAB STASH554
  BNXX=BETAN*XX STASH555
  XFUNC=EXPUL*(K12*SIN(BNXX))-K11*BETAN*COS(BNXX)+FX*FOPX(XX) STASH556
  STASH557
  STASH558

```

```

      RETURN
      END
**
$1BFTC FUNCT  #94/2,XRT
      FUNCTION TFUNC(LAMBDA)
C
C   THIS FUNCTION SETS UP THE INTEGRAND FOR THE LAMBDA INTEGRAL
C
      REAL LIMIT,L,K,LAMBDA,NUNXS,NUNXS,KFPHI,KFPHIO,KFPHIL,K1,K12
      1,K21,K22,N,KYERN1,KYERN2
      COMMON T,X,LAMBDA,KYERN1,KYERN2,ALPHA,BETAN,EXPANL,CONX,SBNX,DELTA
      1X,DELTA1,K11,K12,K21,K22,FX,NTAB
      TERM1=KYERN1*PHIPR1(LAMBDA)
      TERM2=KYERN2*PHIPR2(LAMBDA)
      TFUNC=(TERM1+TERM2)*EXP((ALPHA*(BETAN**2)*(LAMBDA-1))
      RETURN
      END
**
$1BFTC TABLE  #94/2,XRT
      SUBROUTINE TABIN(I)
C
C   READS IN TABULAR DATA
C
      1  FORMAT(15,2F10.0,55H
      2  FORMAT(1HA,61X,9HTABLE NO.,1Z)
      3  FORMAT(1HD,50X,34HINDEPENDENT
      4  FORMAT(48X,E15.8,16X,E15.8)
      5  FORMAT(1HA,41X,9HTABLE NO.,1Z,29H CONTAINS MORE THAN 50 POINTS)
      6  FORMAT(1HA,48X,27HERROR (N TABULAR INPUT DATA)
      7  END
**
$1BFTC ENTERP  #94/2,XRT
      REAL FUNCTION INTERP(TX,I)
C
C   LINEAR INTERPOLATION FOR TABULAR DATA
C
      DIMENSION NTAB(5)
      DIMENSION INDVAR(5,50),DEPVAR(5,50)
      COMMON/PTCAT/ INDVAR,DEPVAR
      REAL INDVAR
      DO 10 J=1,50

```

```

      IF (TX-INDVAR(L,J)) 20,30,10
10    CONTINUE
      WRITE(6,99) I
      CALL EXIT
20    INTERP=DEPVAR(I,J-1)+(DEPVAR(I,J)-DEPVAR(I,J-1))*(TX-INDVAR(I,J-1)
      *)/(INDVAR(L,J)-INDVAR(L,J-1))
      GO TO 100
30    INTERP=DEPVAR(I,J)
100  RETURN
C
C          ERROR MESSAGES
C
99    FORMAT(10X,35HARGUMENT EXCEEDS EXTENT OF TABLE NO.,12)
      END
84
$IBFTC PHIOT  M94/2,XR7
      FUNCTION PHIOT(TT)
C
C    THIS FUNCTION CALCULATES THE INSTANTANEOUS VALUE OF THE
C    TIME-VARYING BOUNDARY CONDITION AT X=0.
C    THE FUNCTION PHIOT(TT) MAY BE LOADED INTO THE PROGRAM
C    AS AN ANALYTICAL EXPRESSION OR AS POINT DATA IN
C    TABULAR FORM
C
      REAL INTERP
      REAL LIMIT,L,K,LAMBDA,NUMXSS,NUMSS,KFPHI,KFPHIO,KFPHIL,K11,K12
      I,K21,K22,N,KTERM1,KTERM2
      COMMON T,X,LAMBDA,KTERM1,KTERM2,ALPHA,BETAN,EXPMUL,CBNX,SBNX,DELTA
      IX,DELTAT,K11,K12,K21,K22,FX,NTAB
      DIMENSION NTAB(5)
      IF (NTAB(1).NE.0) GO TO 100
C
C    PHIOT=ANY FUNCTION OF TIME
C
      PHIOT=1.0
      RETURN
100  PHIOT=INTERP(TT,1)
      RETURN
      END
89
$IBFTC PHILT  M94/2,XR7
      FUNCTION PHILT(TT)
C
C    THIS FUNCTION CALCULATES THE INSTANTANEOUS VALUE OF THE
C    TIME-VARYING BOUNDARY CONDITION AT X=L.
C    THE FUNCTION PHILT(TT) MAY BE LOADED INTO THE PROGRAM
C    AS AN ANALYTICAL EXPRESSION OR AS POINT DATA IN
C    TABULAR FORM.
C
      REAL INTERP
      REAL LIMIT,L,K,LAMBDA,NUMXSS,NUMSS,KFPHI,KFPHIO,KFPHIL,K11,K12
      I,K21,K22,N,KTERM1,KTERM2
      COMMON T,X,LAMBDA,KTERM1,KTERM2,ALPHA,BETAN,EXPMUL,CBNX,SBNX,DELTA
      IX,DELTAT,K11,K12,K21,K22,FX,NTAB
      DIMENSION NTAB(5)
      IF (NTAB(2).NE.0) GO TO 100
C
C    PHILT=ANY FUNCTION OF
C
      PHILT=1.
      RETURN
100  PHILT=INTERP(TT,2)

```

```

      RETURN
      END
**
$IBFTC DERIVO M94/2,XR7
      FUNCTIONPHIPRO(LAMBDA)
C
C      THIS FUNCTION CALCULATES THE INSTANTANEOUS VALUE OF THE
C      DERIVATIVE OF THE TIME-VARYING BOUNDARY CONDITION AT X=0.
C      THE FUNCTION MAY BE LOADED ANALYTICALLY OR AS POINT
C      DATA IN TABULAR FORM.
C
      REAL INTERP
      REAL LIMIT,L,K,LAMBDA,NUMXSS,NUMSS,KFPHI,KFPHIO,KFPHIL,K11,K12
      I,K21,K22,N,KTERM1,KTERM2
      COMMON T,X,LAMBDA,KTERM1,KTERM2,ALPHA,BETAN,EXPMUL,CBNX,SBNX,DELTA
      IX,DELTAT,K11,K12,K21,K22,FX,NTAB
      DIMENSION NTAB(5)
      IF(NTAB(3).NE.0) GO TO 100
C
C      PHIPRO=ANY FUNCTION OF TIME
C
      PHIPRO=0.
      RETURN
100 PHIPRO=INTERP(LAMBDA,3)
      RETURN
      END
**
$IBFTC DERIVL M94/2,XR7
      FUNCTIONPHIPRL(LAMBDA)
C
C      THIS FUNCTION CALCULATES THE INSTANTANEOUS VALUE OF THE
C      DERIVATIVE OF THE TIME-VARYING BOUNDARY CONDITIONS AT X=L.
C      THE FUNCTION MAY BE LOADED ANALYTICALLY OR AS POINT DATA
C      IN TABULAR FORM
C
      REAL INTERP
      REAL LIMIT,L,K,LAMBDA,NUMXSS,NUMSS,KFPHI,KFPHIO,KFPHIL,K11,K12
      I,K21,K22,N,KTERM1,KTERM2
      COMMON T,X,LAMBDA,KTERM1,KTERM2,ALPHA,BETAN,EXPMUL,CBNX,SBNX,DELTA
      IX,DELTAT,K11,K12,K21,K22,FX,NTAB
      DIMENSION NTAB(5)
      IF(NTAB(4).NE.0) GO TO 100
C
C      PHIPRL=ANY FUNCTION OF TIME
C
      PHIPRL=0.
      RETURN
100 PHIPRL=INTERP(LAMBDA,4)
      RETURN
      END
**
$IBFTC FX M94/2,XR7
      FUNCTIONFCFX(L)
C
C      THIS FUNCTION COMPUTES THE INITIAL CONDITIONS OF THE RND.
C      THESE INITIAL CONDITIONS MAY BE LOADED INTO THE PROGRAM
C      ANALYTICALLY OR AS POINT DATA IN TABULAR FORM.
C
      REAL INTERP
      REAL LIMIT,L,K,LAMBDA,NUMXSS,NUMSS,KFPHI,KFPHIO,KFPHIL,K11,K12
      I,K21,K22,N,KTERM1,KTERM2
      COMMON T,X,LAMBDA,KTERM1,K11,K12,ALPHA,BETAN,EXPMUL,CBNX,SBNX,DELTA
      STASH683
      STASH684
      STASH685
      STASH686
      STASH687
      STASH688
      STASH689
      STASH690
      STASH691
      STASH692
      STASH693
      STASH694
      STASH695
      STASH696
      STASH697
      STASH698
      STASH699
      STASH700
      STASH701
      STASH702
      STASH703
      STASH704
      STASH705
      STASH706
      STASH707
      STASH708
      STASH709
      STASH710
      STASH711
      STASH712
      STASH713
      STASH714
      STASH715
      STASH716
      STASH717
      STASH718
      STASH719
      STASH720
      STASH721
      STASH722
      STASH723
      STASH724
      STASH725
      STASH726
      STASH727
      STASH728
      STASH729
      STASH730
      STASH731
      STASH732
      STASH733
      STASH734
      STASH735
      STASH736
      STASH737
      STASH738
      STASH739
      STASH740
      STASH741
      STASH742
      STASH743
      STASH744

```

AFFDL-TR-66-109

```

      1X,DELTA1,N11,K12,K21,K22,FX,NTAB
      DIMENSION NTAB(5)
      IF(NTAB(5).NE.0) GO TO 100
C
C      FOFX= ANY FUNCTION OF X
C
      FOFX=1.
55    RETURN
100   FOFX=INTERP(XX,5)
      RETURN
      END
```

STASH745
STASH746
STASH747
STASH748
STASH749
STASH750
STASH751
STASH752
STASH753
STASH754
STASH755

APPENDIX II
EIGENVALUE SUBROUTINES

APPENDIX II

EIGENVALUE SUBROUTINES

The solution of equation 53 depends on values of β_n and which are derived from the positive eigenvalues of Equation 40.

$$\tan z_n = \frac{D L Z_n}{K_{21} K_{11} Z_n^2 + K_{22} K_{12} L^2} \quad (40)$$

Since we are seeking positive values of z_n the sign of the left-hand side of the equation may be associated with the parameter, D. Thus three formulations are possible corresponding to D being positive, negative or zero. A fourth possibility is that of the denominator going to zero. The last two solutions are trivial. If D is zero we have

$$\tan z_n = 0$$

The solution to this equation is merely

$$z_n = n\pi \quad (54)$$

If the denominator of equation 40 goes to zero we have

$$\tan z_n = \infty$$

The solution to the equation is

$$z_n = \frac{(2n-1)\pi}{2} \quad (55)$$

However, if D has a value other than zero the equations are solved by an iterative process. In these cases the eigenvalue subroutine has been programmed to ignore the root at (0,0) since this produces a trivial solution. The procedure then is outlined below for a positive value of D.

We know that the solution lies between

$$\left[(n-1)\pi, \frac{(2n-1)\pi}{2} \right], \quad n = 1, 2, 3, \dots$$

For a first approximation to the root, $z_{n,0}$, we shall choose

$$z_{n,0} = \frac{(2n-1)\pi}{2} \quad (56)$$

We then write two equations

$$u_{n,m} = \frac{D L Z_{n,m-1}}{K_{21} K_{11} Z_{n,m-1}^2 + K_{22} K_{12} L^2} \quad (57)$$

$$Z_{n,m} = (n-1)\pi + \tan^{-1}(u_{n,m}) \quad (58)$$

Where the subscripts n and m refer to the mth iteration toward the nth root. The root is then determined to any desired LIMIT of accuracy by writing

$$Z_{n,m} - Z_{n,m-1} < \text{LIMIT} \quad (59)$$

Provision is made in the program to print out the iteration steps should any trouble occur. The basic difference in the solution for a negative value of D arises in the first approximation, $Z_{n,0}$. For negative values of D Equation 56 becomes

$$Z_{n,0} = (n + 1) \pi \quad (60)$$

and the solution proceeds as before with Equation 58 becoming

$$Z_{n,m} = n\pi + \tan^{-1}(u_{n,m}) \quad (61)$$

AFFDL-TR-66-109

PREVIOUS PAGE WAS BLANK, THEREFORE WAS NOT FILLED.

APPENDIX III

RESULTS OF CHECK PROBLEMS

TABLE II
COMPARISON OF DATA FOR CASE I

$T(0,t)=t; T(L,t)=0; T(x,0)=0$							
L	TEMP @	TEMP @ t=50		TEMP @ t=500		TEMP @ t=1000	
	t=0	STASH	LTA	STASH	LTA	STASH	LTA
0.0	0.00	50.00	50.00	500.00	500.00	1000.00	1000.00
0.5	0.00	37.39	37.38	455.95	455.93	930.71	930.69
1.0		27.45	27.46	414.85	414.83	864.38	864.35
1.5		19.79	19.80	376.56	376.53	800.87	800.83
2.0		13.99	14.00	340.92	340.87	740.02	739.97
2.5		9.76	9.70	307.74	307.69	681.66	681.60
3.0		6.58	6.59	276.88	276.83	625.65	625.59
3.5		4.37	4.38	248.18	248.12	571.82	571.76
4.0		2.84	2.85	221.48	221.42	520.03	519.96
4.5		1.80	1.81	196.61	196.56	470.11	470.04
5.0		1.12	1.13	173.43	173.37	421.90	421.84
5.5		0.68	0.68	151.77	151.72	375.26	375.20
6.0		0.40	0.41	131.43	131.43	330.03	329.97
6.5		0.23	0.24	112.40	112.36	286.04	285.99
7.0		0.13	0.13	94.38	94.34	243.15	243.10
7.5		0.07	0.07	77.27	77.24	201.19	201.15
8.0		0.04	0.04	60.91	60.89	160.02	159.98
8.5		0.02	0.02	45.16	45.14	119.47	119.44
9.0		0.01	0.01	29.86	29.84	79.38	79.37
9.5		0.00	0.00	14.85	14.84	39.61	39.60
10.0	0.00	0.00	0.00	0.00	0.00	0.00	0.00

TABLE III

COMPARISON OF DATA FOR CASE 2

$T(0,t)=100+t; T(L,t)=10^2; T(x,0)=10^2$							
L	TEMP @	TEMP @ t=50		TEMP @ t=500		TEMP @ t=1000	
	t=0	STASH	LTA	STASH	LTA	STASH	LTA
0.0	100.00	150.00	150.00	600.00	600.00	1100.00	1100.00
0.5	100.00	137.38	137.38	555.95	555.93	1030.71	1030.69
1.0		127.45	127.46	514.85	514.83	964.38	964.35
1.5		119.79	119.80	476.56	476.53	900.87	900.83
2.0		113.99	114.00	440.92	440.87	840.02	839.97
2.5		109.70	109.70	407.74	407.69	781.66	781.60
3.0		106.58	106.59	376.88	376.83	725.55	725.50
3.5		104.37	104.38	348.18	348.12	671.82	671.76
4.0		102.84	102.85	321.48	321.42	620.03	619.96
4.5		101.80	101.81	296.61	296.56	570.11	570.04
5.0		101.11	101.13	273.43	273.37	521.40	521.34
5.5		100.68	100.68	251.77	251.72	475.76	475.70
6.0		100.40	100.40	231.48	231.43	430.03	429.97
6.5		100.23	100.24	212.40	212.36	386.04	385.99
7.0		100.13	100.13	194.38	194.34	343.15	343.10
7.5		100.07	100.07	177.27	177.24	301.19	301.15
8.0		100.04	100.04	160.91	160.88	260.02	259.98
8.5		100.02	100.02	145.16	145.14	219.47	219.44
9.0		100.01	100.01	129.36	129.34	179.38	179.37
9.5		100.00	100.00	114.85	114.85	139.61	139.60
10.0	100.00	100.00	100.00	100.00	100.00	100.00	100.00

TABLE IV
COMPARISON OF DATA FOR CASE 3

$T(0,t)=t; T(L,t)=10^2; T(x,0)=\frac{100x}{L}$							
L	TEMP @ t=0	TEMP @ t=50		TEMP @ t=500		TEMP @ t=1000	
		STASH	LTA	STASH	LTA	STASH	LTA
0.0	0.00	50.00	50.00	500.00	500.00	1000.00	1000.00
0.5	5.00	42.38	42.38	460.95	460.93	935.71	935.69
1.0	10.00	37.46	37.46	424.85	424.83	874.38	874.35
1.5	15.00	34.79	34.80	391.56	391.53	815.87	815.83
2.0	20.00	33.99	34.00	360.92	360.87	760.02	759.97
2.5	25.00	34.70	34.70	332.74	332.69	706.66	706.60
3.0	30.00	36.58	36.59	306.88	306.83	655.65	655.59
3.5	35.00	39.37	39.38	283.18	283.16	606.82	606.76
4.0	40.00	42.84	42.85	261.48	261.42	560.03	559.96
4.5	45.00	46.80	46.81	241.61	241.56	515.11	515.04
5.0	50.00	51.11	51.13	223.43	223.37	471.90	471.84
5.5	55.00	55.68	55.68	206.77	206.72	430.26	430.20
6.0	60.00	60.40	60.41	191.48	191.43	390.03	389.97
6.5	65.00	65.23	65.24	177.40	177.36	351.04	350.99
7.0	70.00	70.13	70.13	164.38	164.34	313.15	313.10
7.5	75.00	75.07	75.07	152.27	152.24	276.19	276.15
8.0	80.00	80.04	80.04	140.91	140.89	240.02	239.98
8.5	85.00	85.02	85.02	130.16	130.14	204.47	204.44
9.0	90.00	90.01	90.01	119.86	119.84	169.38	169.37
9.5	95.00	95.00	95.00	109.85	109.84	134.61	134.60
10.0	100.00	100.00	100.00	100.00	100.00	100.00	100.00

COMPARISON OF DATA FOR CASE 1

$T(0,t)=t; T(L,t)=100; T(x,0)=\frac{100x}{L}$							
L	TEMP @	TEMP @ t=50		TEMP @ t=100		TEMP @ t=150	
	t=0	STASH	LTA	STASH	LTA	STASH	LTA
0.0	0.00	50.00	50.00	500.00	500.00	1000.00	1000.00
0.5	5.00	42.38	42.38	460.95	460.95	935.71	935.69
1.0	10.00	37.46	37.46	424.85	424.83	811.38	811.34
1.5	15.00	34.79	34.80	391.56	391.53	715.37	715.33
2.0	20.00	33.99	34.00	360.92	360.87	660.02	659.97
2.5	25.00	34.70	34.70	332.74	332.69	606.66	606.60
3.0	30.00	36.58	36.59	306.88	306.83	555.65	555.59
3.5	35.00	39.37	39.38	283.18	283.16	506.82	506.76
4.0	40.00	42.84	42.85	261.48	261.42	460.03	459.96
4.5	45.00	46.80	46.81	241.61	241.56	415.11	415.04
5.0	50.00	51.11	51.13	223.43	223.37	371.90	371.84
5.5	55.00	55.68	55.68	206.77	206.72	330.26	330.20
6.0	60.00	60.40	60.41	191.48	191.43	290.03	289.97
6.5	65.00	65.23	65.24	177.40	177.36	251.04	250.99
7.0	70.00	70.12	70.13	164.38	164.34	213.15	213.10
7.5	75.00	75.07	75.07	152.27	152.24	176.19	176.15
8.0	80.00	80.04	80.04	140.91	140.89	140.02	139.98
8.5	85.00	85.02	85.02	130.16	130.14	104.47	104.44
9.0	90.00	90.01	90.01	119.86	119.84	69.38	69.37
9.5	95.00	95.00	95.00	109.85	109.84	34.61	34.60
10.0	100.00	100.00	100.00	100.00	100.00	0.00	0.00

TABLE V

COMPARISON OF DATA FOR CASE 4

$T(0,t)=1000; q(L,t)=0; T(x,0)=100$							
L	TEMP @ $t=0$	TEMP @ $t=300$		TEMP @ $t=1800$		TEMP @ $t=3600$	
		STASH	LTA	STASH	LTA	STASH	LTA
0.0	100.00	1000.00	1000.00	1000.00	1000.00	1000.00	1000.00
1.0	100.00	899.95	899.90	994.85	994.40	999.85	999.80
2.0	100.00	802.53	802.50	989.83	989.45	999.71	999.65
3.0	100.00	710.26	709.99	985.06	984.72	999.57	999.53
4.0	100.00	625.48	625.30	980.06	978.98	999.44	999.41
5.0	100.00	550.27	550.05	976.72	976.13	999.33	999.27
6.0	100.00	486.38	486.14	973.37	972.66	999.24	999.19
7.0	100.00	435.25	434.97	970.67	970.04	999.16	999.12
8.0	100.00	397.92	397.52	968.69	968.08	999.10	999.06
9.0	100.00	375.51	375.02	967.49	967.10	999.07	999.01
10.0	100.00	367.70	367.56	967.08	966.99	999.05	998.99

TABLE VI

COMPARISON OF DATA FOR CASE 5

$T(0,t)=0; q(L,t)=\frac{1 \text{ Btu}}{\text{in}^2 \text{ sec}}; T(x,0)=0$							
L	TEMP @ $t=0$	TEMP @ $t=300$		TEMP @ $t=1800$		TEMP @ $t=3600$	
		STASH	LTA	STASH	LTA	STASH	LTA
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1.0	0.0	129.76	129.08	416.41	416.34	431.70	431.69
2.0	0.0	266.80	265.45	833.21	833.08	863.41	863.39
3.0	0.0	418.26	416.26	1250.77	1250.58	1295.14	1295.12
4.0	0.0	590.98	588.36	1669.46	1669.22	1726.91	1726.88
5.0	0.0	791.36	788.17	2089.61	2089.32	2128.72	2128.68
6.0	0.0	1025.20	1021.51	2511.51	2511.17	2592.51	2592.54
7.0	0.0	1297.54	1293.43	2935.41	2935.04	3022.49	3022.45
8.0	0.0	1612.46	1608.04	3361.52	3361.12	3454.47	3454.47
9.0	0.0	1973.03	1968.41	3789.99	3789.57	3883.51	3883.47
10.0	0.0	2381.11	2376.43	4220.90	4220.48	4318.69	4318.58

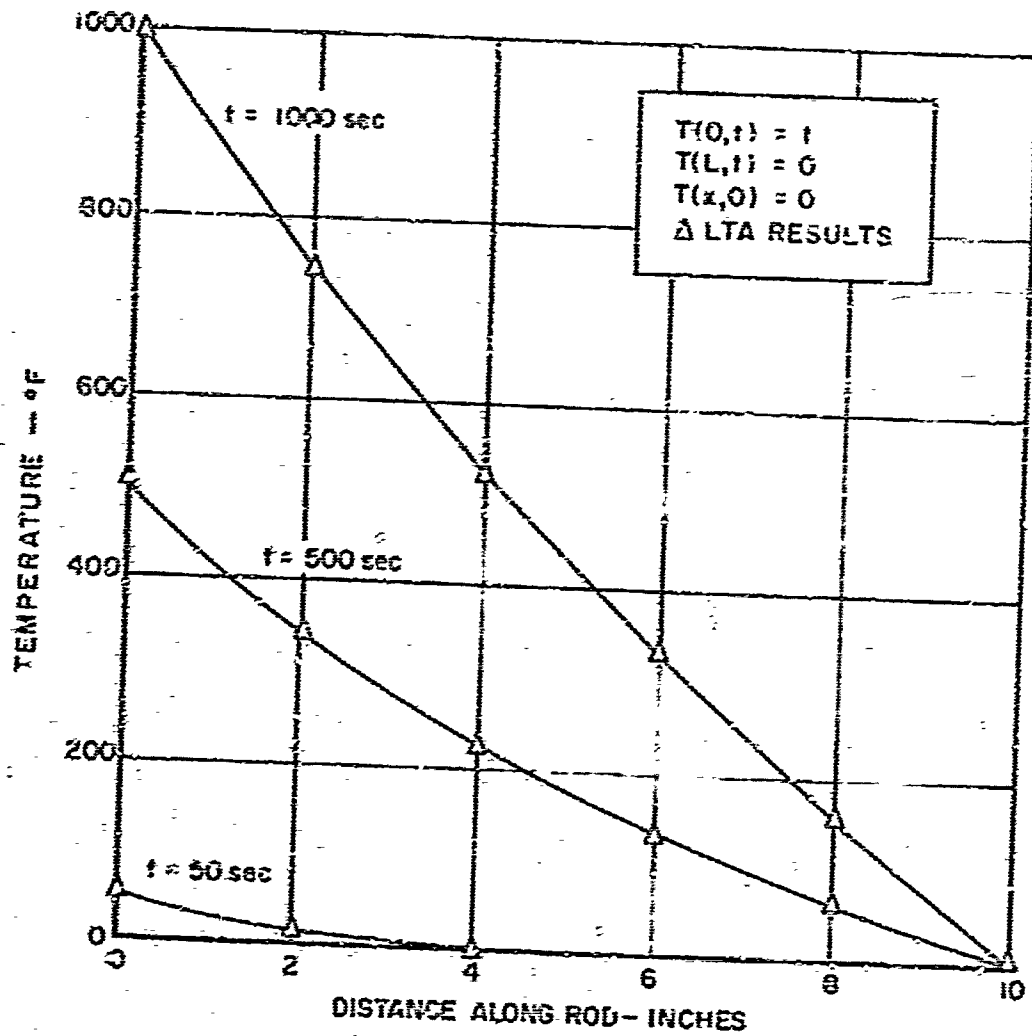


Figure 8. Temperature Profiles (Case 1)

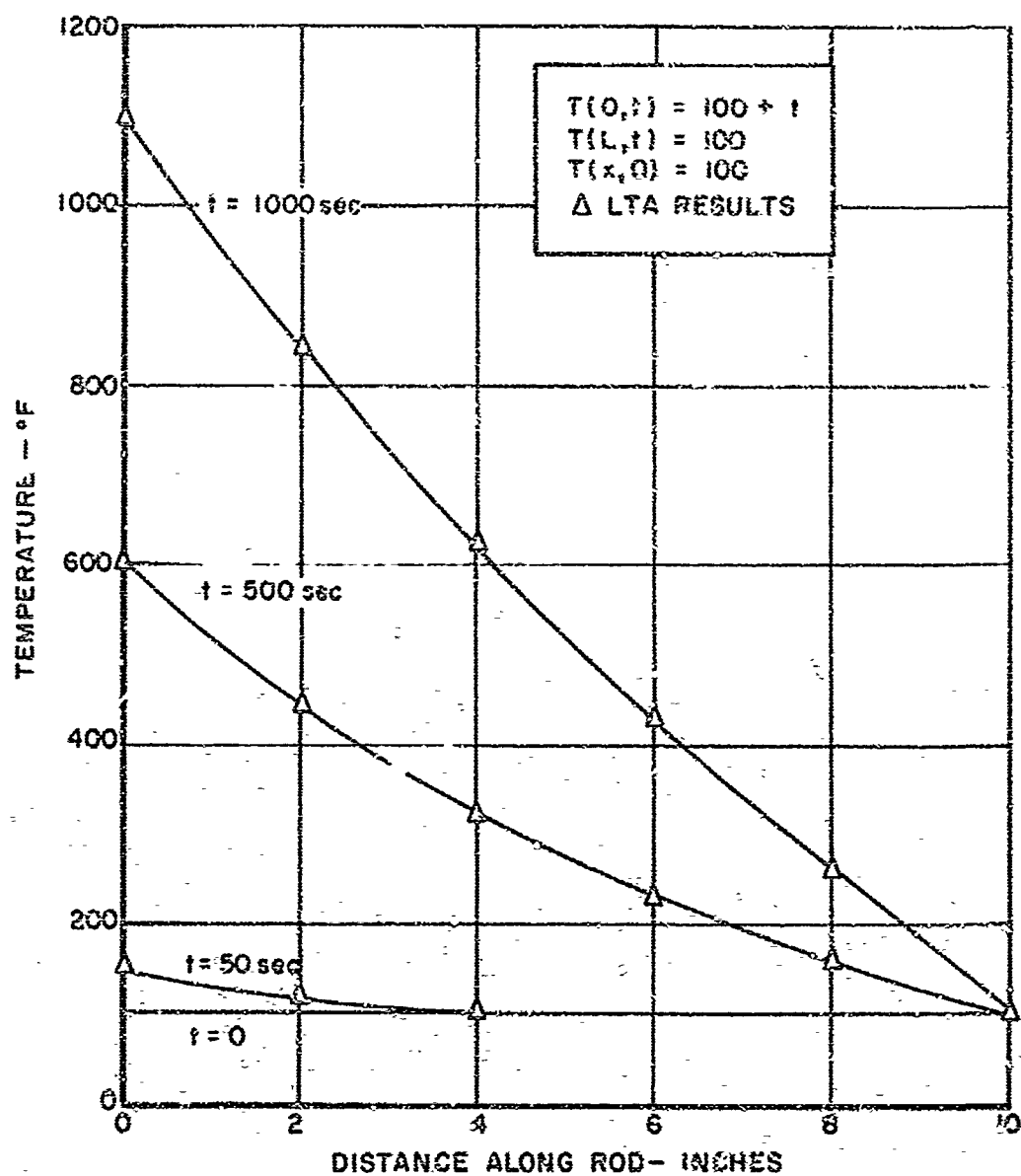


Figure 2. Temperature Profiles (Case 2)

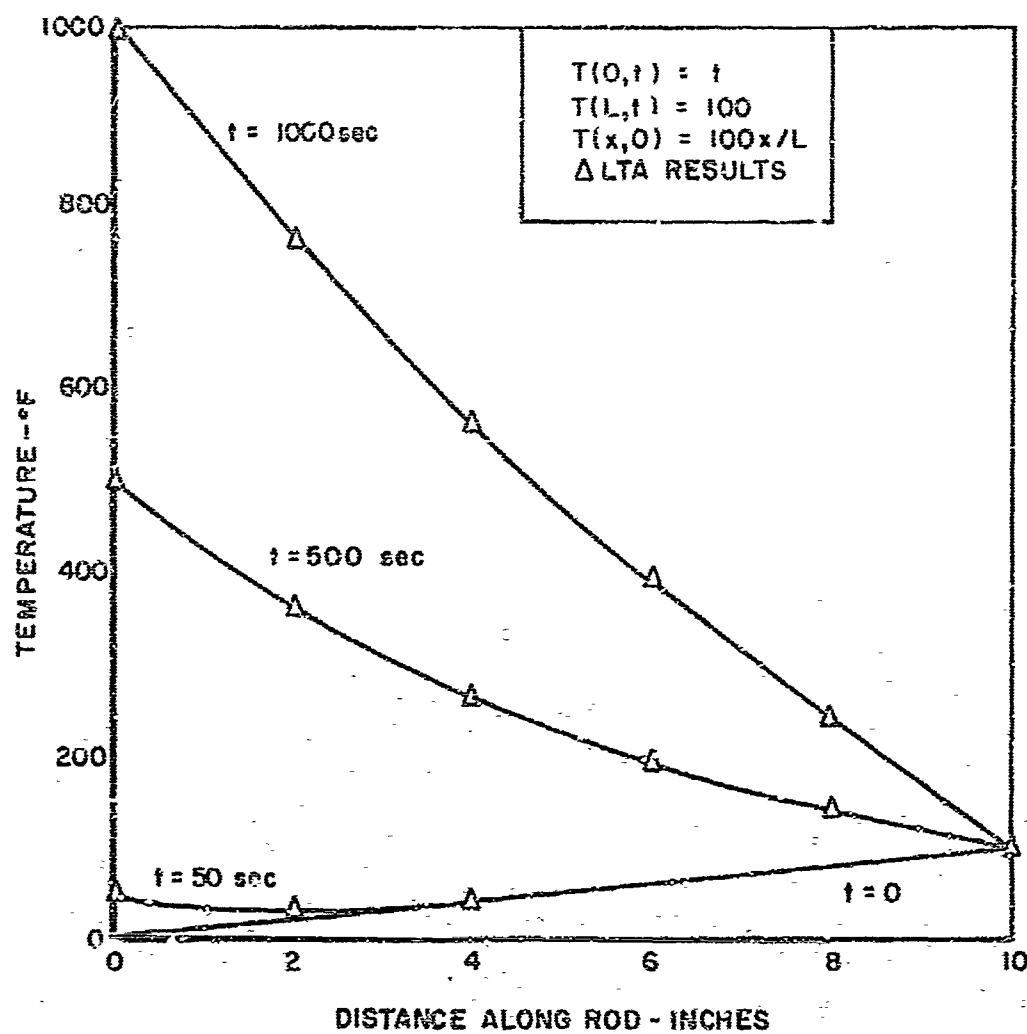


Figure 10. Temperature Profiles (Case 3)

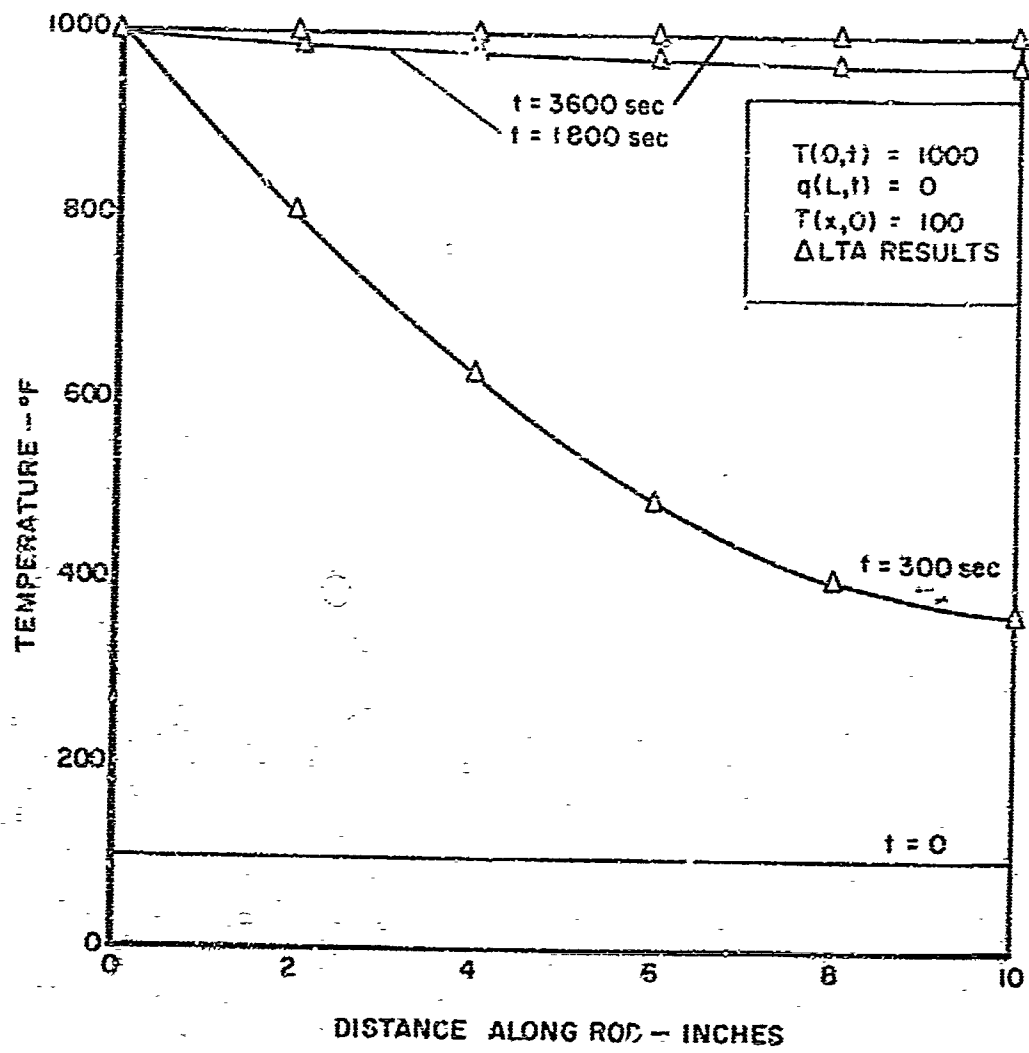


Figure 11. Temperature Profiles (Case 4)

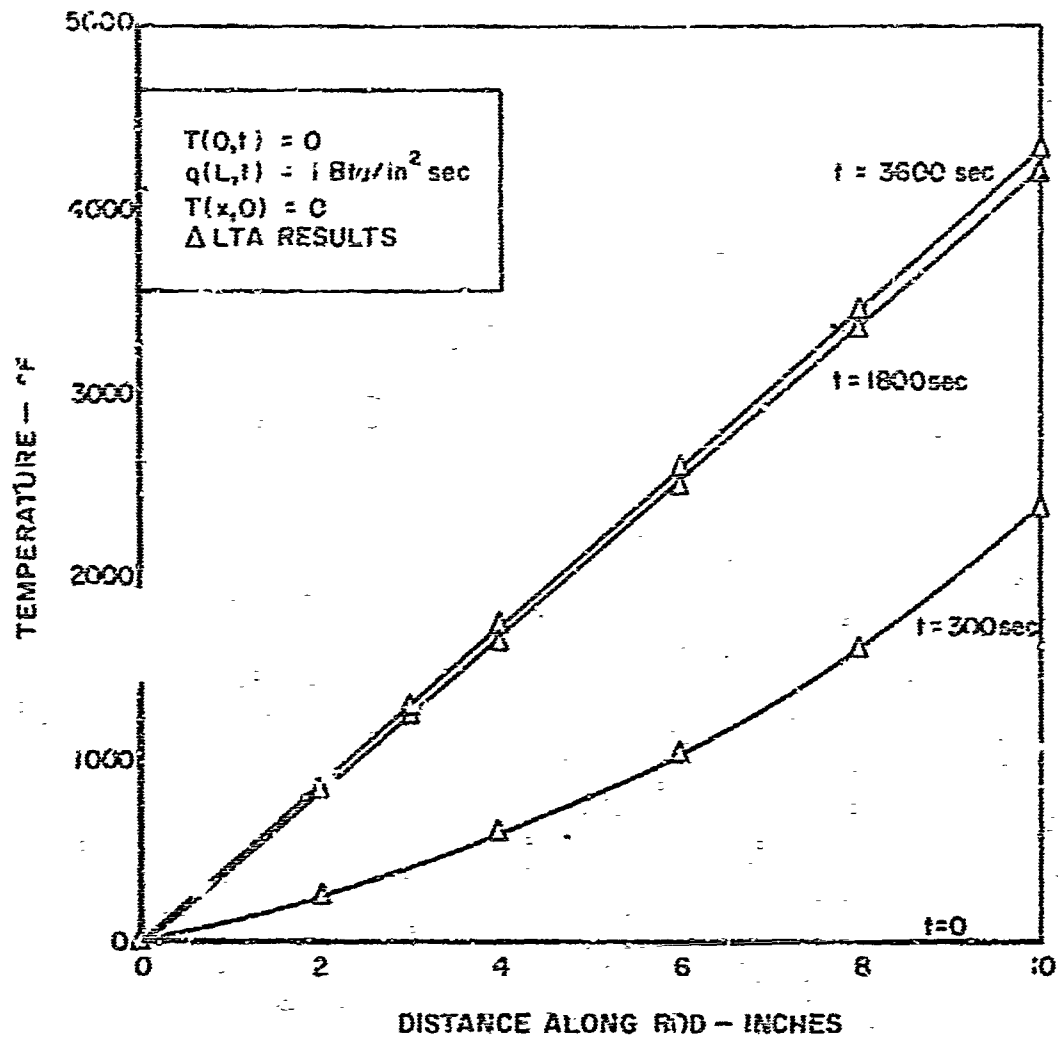


Figure 12. Temperature Profiles (Case 5)

~~Unclassified~~

Security Classification

DOCUMENT CONTROL DATA - E2D

(Security classification of title, body of abstract and indexing annotation must be derived from the overall report to which it is attached)

1. ORIGINATING ACTIVITY (Corporate author)		2. REPORT SECURITY CLASSIFICATION	
Air Force Flight Dynamics Laboratory, Research and Technology Division, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio 45433		2. GROUP	
3. REPORT TITLE			
TRANSIENT ANALYSIS OF HEAT CONDUCTION THROUGH A SLAB BY INFINITE SERIES			
4. DESCRIPTIVE NOTES (Type of report and inclusive dates)			
5. AUTHOR(S) (Last name, first name, initial)			
Bernstein, Thomas N., Engle, Robert M., Jr.			
6. REPORT DATE		7. TOTAL NO. OF PAGES	7. NO. OF REFS
December 1966		64	
8. CONTRACT OR GRANT NO.		9. ORIGINATOR'S REPORT NUMBER(S)	
A. PROJECT NO. 1467		AFFDL-TR-66-109	
C. Task No. 146702		10. OTHER REPORT NO(S) (List other numbers that may be assigned this report)	
11. AVAILABILITY/LIMITATION NOTES 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 the Theoretical Mechanics Branch, Structures Division, Air Force Flight Dynamics Laboratory (EDTR), Wright-Patterson Air Force Base, Ohio 45433.			
12. SUPPLEMENTARY NOTES		13. SPONSORING/MILITARY ACTIVITY	
		Air Force Flight Dynamics Laboratory, Research and Technology Division, Air Force Systems Command, Wright-Patterson AFB, Ohio	
14. ABSTRACT			
The exact solution to the problem of conduction of heat through a slab is developed. The solution, formulated in terms of an infinite series, allows arbitrary initial conditions and time-dependent boundary conditions. The solution is programmed in FORTRAN IV for the IBM 7094 II computer. Several check problems were solved and the results were compared with those obtained from a finite difference heat transfer program.			

DD FORM 1 JAN 64 1473

~~Unclassified~~
Security Classification

Unclassified

Security Classification

KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Thermal Analysis Mathematical Formulation Digital Computer Analysis Arbitrary Boundary Conditions Infinite Series Computer Program One-Dimensional Transient Conduction FORTRAN						

INSTRUCTIONS

1. **ORIGINATING ACTIVITY** Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization (corporate author) issuing the report.

2a. **REPORT SECURITY CLASSIFICATION** Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2b. **GROUP** Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. **REPORT TITLE** Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parentheses immediately following the title.

4. **DESCRIPTIVE NOTES** If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. **AUTHOR(S)** Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. **REPORT DATE** Enter the date of the report as day, month, year, or month, year. If more than one date appears on the report, use date of publication.

7a. **TOTAL NUMBER OF PAGES** The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. **NUMBER OF REFERENCES** Enter the total number of references cited in the report.

8a. **CONTRACT OR GRANT NUMBER** If appropriate, enter the applicable number of the contract or grant under which the report was written.

8b, c, & 8d. **PROJECT NUMBER** Enter the appropriate (a) key department identification number, such as project number, or project number, system number, task number, etc.

9a. **ORIGINATOR'S REPORT NUMBER(S)** Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9b. **OTHER REPORT NUMBER(S)** If the report has been assigned any other report numbers (either by the originator or by the sponsor), also enter this number(s).

10. **AVAILABILITY/LIMITATION NOTICES** Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

- (1) "Qualified requesters may obtain copies of this report from DDC."
- (2) "Foreign announcement and dissemination of this report by DDC is not authorized."
- (3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through _____."
- (4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through _____."
- (5) "All distribution of this report is controlled. Qualified DDC users shall request through _____."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. **SUPPLEMENTARY NOTES** Use for additional explanatory notes.

12. **SPONSORING MILITARY ACTIVITY** Enter the name of the departmental project office or laboratory sponsoring (paying for) the research and development. Include address.

13. **ABSTRACT** Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as (TS), (S), (C), or (U).

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. **KEY WORDS** Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, roles, and weights is optional.

Unclassified

Security Classification