AEDC-TR-69-121

METHOD FOR SOLVING TRANSIENT HEAT-TRANSFER PROBLEMS INCLUDING COMPUTER PROGRAMS AND AN ANALYSIS OF FORM-SURFACE FACTORS

Donald C. Todd

ARO, Inc.

August 1969

This document has been approved for public release and sale; its distribution is unlimited.

AEROSPACE ENVIRONMENTAL FACILITY
ARNOLD ENGINEERING DEVELOPMENT CENTER
AIR FORCE SYSTEMS COMMAND
ARNOLD AIR FORCE STATION, TENNESSEE

NOTICES

METHOD FOR SOLVING TRANSIENT HEAT-TRANSFER PROBLEMS INCLUDING COMPUTER PROGRAMS AND AN ANALYSIS OF FORM-SURFACE FACTORS

Donald C. Todd
ARO, Inc.

FOREWORD

The work reported herein was sponsored by Arnold Engineering Development Center (AEDC), Air Force Systems Command (AFSC), Arnold Air Force Station, Tennessee, under Program Element 65401F.

The results of the work were obtained by ARO, Inc. (a subsidiary of Sverdrup & Parcel and Associates, Inc.) contract operator of AEDC, under Contract F40600-69-C-0001. The work was performed from October 1964 to December 1968 under ARO Project No. ST8002, and the manuscript was submitted for publication on May 2, 1969.

This technical report has been reviewed and is approved.

Paul L. Landry
Major, USAF
AF Representative, AEF
Directorate of Test

Roy R. Croy, Jr.
Colonel, USAF
Director of Test

ABSTRACT

A method for attacking a wide variety of commonly occurring heat-transfer problems is presented. The method stems from a mathematical model obtained by dividing a system into pieces. The differential equation of the temperature of each piece with respect to time is derived considering heat sources and heat transfer by conduction and radiation. The resulting system of first-order ordinary differential equations is put in a form convenient for programming a numerical solution by defining coefficients in terms of the data of the system. Form-surface factors are a generalization of form factors to include multiple reflections. The factor is defined and discussed. A formula is derived for the factor in terms of the form factors and the reflectivities. The form-surface factor relations for the conservation of energy and the reciprocity law are derived. The conditions necessary and sufficient for its existence are derived. Methods of calculation are discussed. A FORTRAN subroutine is presented that calculates the form-surface factors from the form factors and the reflectivities. A FORTRAN subroutine is presented which performs unit conversions on data tabulated for the mathematical model. Another FORTRAN subroutine is presented which calculates the coefficients from the tabulated data. A numerical method for solving the system of differential equations is developed and a FORTRAN subroutine is presented which effects the solution. Several example problems are worked out, and the results are compared with analytical solutions.

CONTENTS

ILLUSTRATIONS

TABLES

Page

APPENDIXES

NOMENCLATURE

A_i	Constant term in Eq. (10)
a_{ij}	Conducting area between the ith and jth pieces
A_c	Allowable temperature error per unit time
B_{ij}	Coefficient of T_j in Eq. (10)
C_{ij}	Coefficient of $T_j{}^4$ in Eq. (10)
c_i	Specific heat capacity of ith piece
d_i	Density of ith piece
d_{ij}	Conducting distance between the ith and jth pieces
E_i	Temperature error of the ith piece
F	Matrix of form factors
F_{ij}	Form factor from ith piece to jth piece
G	Matrix of form-surface factors

G_{ij}	Form-surface factor from ith piece to jth piece
H_i	Irradiation on the ith piece
n	Number of pieces in a system
P_i	Net rate that heat is gained by the ith piece
P_i^s	Rate that heat is supplied by a source to the ith piece
P_i^k	Rate that heat is gained by conduction by the ith piece
P_i^e	Rate that radiant energy is emitted by the ith piece
P_i^a	Rate that radiant energy is absorbed by the ith piece
Q	Column matrix of the Q_i
Q_i	Flux incident on the ith piece
Q_{ij}	Portion of the flux incident on the jth piece resulting from the flux emitted by the ith piece
R	Diagonal matrix of the ρ_i
S	Diagonal matrix of the S_i
S_i	Radiating area of the ith piece
T_i	Temperature of the ith piece
t	Time
t_1	Base time for the Taylor expansion
t_m	Maximum time for which the Taylor expansions are valid
V_i	Volume of the ith piece
W	Column matrix of the W_i
W_i	Flux emitted by the ith piece
α_i	Absorptivity of the ith piece
δ_{ij}	Kronecker delta
ϵ_i	Emissivity of the ith piece
κ_i	Thermal conductivity of the ith piece
ρ_i	Reflectivity of the ith piece
σ	Stefan-Boltzmann constant

SECTION I
INTRODUCTION

General methods are needed to solve wide varieties of commonly occurring heat-transfer problems to shorten the time from defining the problem to obtaining a solution., This report presents a method for transient, conduction, and radiation heat-transfer problems. The given system is divided into pieces to form a mathematical model. The differential equation of each piece with respect to time is derived considering internal power sources and heat transfer by conduction and radiation. The resulting system of first-order differential equations is put in a form convenient for programming a numerical solution by defining coefficients in terms of the data of the system. FORTRAN subroutines are presented which convert units of the data of the system, calculate the coefficients, and effect a numerical solution to the equations.

The form-surface factor is a generalization of the form factor to include multiple reflections. This factor is defined and discussed, and a formula is derived giving the factor in terms of the form factors and the reflectivities of the system. The form-surface factor relations for conservation of energy and the reciprocity law are derived. Also, conditions necessary and sufficient for the existence of the form-surface factor are derived. Methods of calculation are discussed and a FORTRAN subroutine is presented which calculates the form-surface factor from the form factors and the reflectivities.

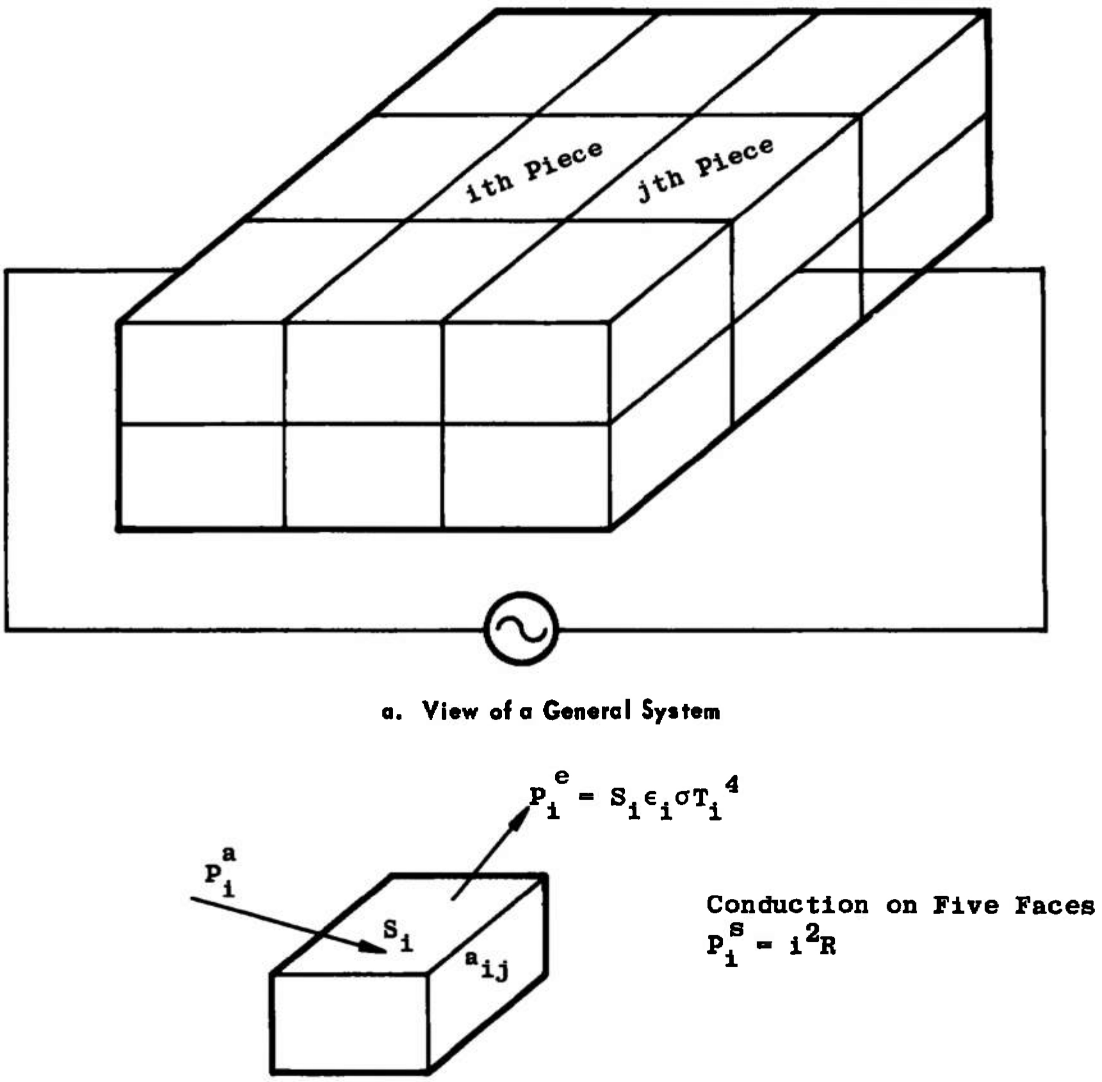
Once familiar with the mathematical model and the subroutines, the following steps are taken: (1) The system is divided into pieces; (2) a FORTRAN program is written to calculate and tabulate data for the mathematical model; (3) a subroutine is called which calculates the form-surface factors from the form factors and the reflectivities; (4) a subroutine is called which performs unit conversions on all the tabulated data; (5) a subroutine is called which calculates the coefficients from the converted data; (6) lastly, a subroutine is called which effects the numerical solution to the system of differential equations.

Example problems illustrating the method are worked out, and the results are presented and compared with analytical solutions. The FORTRAN listings of the programs used in the example problems are given in appendixes to be used as a guide.

SECTION II
MATHEMATICAL MODEL

2.1 DERIVATION OF EQUATIONS

The data input to the computer programs are based on a mathematical model of the system. The model is obtained by dividing the system into pieces, with uniform thermal properties (Fig. 1). The equations for this mathematical model are derived by performing an energy balance on each piece.



a. View of a General System

b. View of ith Piece

Fig. 1 Illustrative System

Assume that the model consists of n pieces. The rate of temperature change of the ith piece depends on the net rate that heat is gained by the piece according to the equation

$$d_i V_i c_i \dot{T}_i = P_i \tag{1}$$

where the dot represents the time derivative. The types of heat transfer which will be considered are heat sources, conduction, and radiant heat transfer in a vacuum; thus, the net rate that heat is gained is

$$P_i = P_i^s - P_i^k + (P_i^a - P_i^e) \tag{2}$$

The first term accounts for heat supplied by a source such as an electrical heater. This quantity may be negative in which case it represents a heat sink.

The rate that heat is gained by the ith piece by conduction is

$$P_i^k = \sum_{j\,on\,i} \left[\kappa_i \int_{a_{ij}} \Delta T \cdot d\vec{a} \right] \tag{3}$$

where the summation is over the pieces j in contact with the ith piece (abbreviated j on i) and where the differential area vector is directed outward. It will be assumed that the temperature gradient is constant over each conducting area, a_{ij}, and that it is proportional to the temperature difference $T_j - T_i$. To derive the proportionality constant consider the linearized case illustrated in Fig. 2.

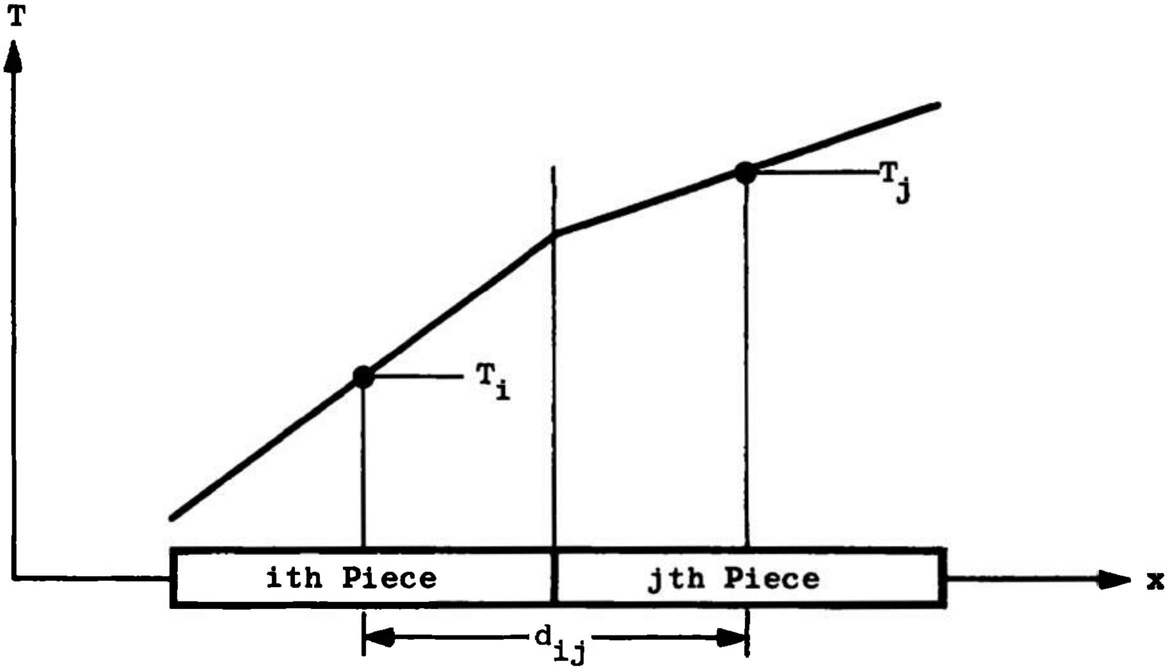


Fig. 2 Linearized Temperature

It can be seen that there will be a discontinuity of the temperature gradient at the interface of the two pieces if their thermal conductivities are different. Conservation of energy requires that at the interface

$$\kappa_i \left.\frac{dT}{dx}\right|_i = \kappa_j \left.\frac{dT}{dx}\right|_j \tag{4}$$

Using this relation and geometrical reasoning one derives

$$\left.\frac{dT}{dx}\right|_j = \frac{2(T_j - T_i)}{d_{ij}\left(1 + \frac{\kappa_i}{\kappa_j}\right)} \tag{5}$$

Using these approximations Eq. (3) becomes

$$P_i^k = \sum_{j\ on\ i} \frac{2\,\kappa_i\,\kappa_j\,a_{ij}\,(T_j - T_i)}{d_{ij}\,(\kappa_i + \kappa_j)} \tag{6}$$

where the summation is over the pieces which are in contact with the ith piece. Note that the conducting area, a_{ij}, is defined as the area perpendicular to the conducting path, thus the cosine factor resulting from the dot product of Eq. (3) is one. The conducting distance, d_{ij}, is the distance along the conducting path.

The rate that radiant energy is emitted by the ith piece is

$$P_i^e = S_i\,\epsilon_i\,\sigma\,T_i^4 \tag{7}$$

The rate that radiant energy is absorbed by the ith piece is

$$P_i^a = \sum_{j=1}^{n} \alpha_i\,G_{ji}\,S_j\,\epsilon_j\,\sigma\,T_j^4 \tag{8}$$

The factor G_{ji} is defined as the ratio of that part of the flux incident on the ith piece, after all reflections, which was originally emitted by the jth piece to the flux emitted by the jth piece. Thus the last four factors of Eq. (8), yielding the rate at which radiant energy is emitted by the jth piece, are multiplied by G_{ji} to obtain the resulting flux on the ith piece. This is multiplied by the absorptivity and summed over all the pieces to get the rate heat is gained by absorption by the ith piece. The explicit dependence of G_{ji} on the geometry and surface properties of the system is derived in the next section along with some important relations it must satisfy. A FORTRAN subroutine is described in Section IV which can be used for its calculation.

After all substitutions, the energy balance equation for the ith piece, Eq. (1), becomes

$$d_i V_i c_i \dot{T}_i = P_i^s + \sum_{j \text{ on } i} \frac{2\kappa_i \kappa_j a_{ij} (T_j - T_i)}{d_{ij} (\kappa_i + \kappa_j)} - S_i \epsilon_i \sigma T_i^4 + \sum_{j=1}^{n} \alpha_i G_{ji} S_j \epsilon_j \sigma T_j^4 \qquad (9)$$

There is one equation like Eq. (9) for each piece in the model.

Equation (9) can be written in a more convenient form for programming a numerical solution by collecting terms and solving for $\dot{T}_i$:

$$\dot{T}_i = A_i + \sum_{j=1}^{n} B_{ij} T_j + \sum_{j=1}^{n} C_{ij} T_j^4 \qquad (10)$$

where

$$A_i = \frac{P_i^s}{d_i V_i c_i} \qquad (11)$$

$$B_{ij} = \begin{cases} 0 & j \text{ not on } i \\ \dfrac{2 \kappa_i \kappa_j a_{ij}}{d_i V_i c_i d_{ij} (\kappa_i \div \kappa_j)} & j \text{ on } i \\ -\dfrac{2\kappa_i}{d_i V_i c_i} \sum\limits_{k \text{ on } i} \dfrac{\kappa_k a_{ik}}{d_{ik} (\kappa_i + \kappa_k)} & j = i \end{cases} \qquad (12)$$

$$C_{ij} = \begin{cases} \dfrac{\alpha_i G_{ji} S_j \epsilon_j \sigma}{d_i V_i c_i} & i \neq j \\ -\dfrac{S_i \epsilon_i \sigma}{d_i V_i c_i} (1 - \alpha_i G_{ii}) & i = j \end{cases} \qquad (13)$$

A FORTRAN subroutine is described in Section V which, when given the values of the quantities on the right side of Eqs. (11) through (13), will evaluate these equations. When evaluating Eq. (12), the subroutine assumes that pieces are in contact if and only if their conducting area is positive. The results of this subroutine can be used directly in another subroutine described in Section VI which numerically calculates the solution of the system of differential equations represented by Eq. (10).

Dividing the system into hypothetical pieces has effectively reduced the partial differential integral equation relating temperature as a function of time and position into a system of ordinary differential equations for which there are standard general numerical methods (Ref. 1) for solution. This method of reducing partial differential equations to ordinary differential equations is discussed on page 110 of Ref. 1 which lists additional references.

2.2 BOUNDARY CONDITIONS

Most heat-transfer problems are subject to boundary conditions, usually in the form of an insulated boundary or a boundary held at a fixed temperature. The insulated boundary requires that no heat flows across the boundary. Since the mathematical model was based on an isolated system, this type of condition is inherent to the model and requires no further analysis. The fixed temperature boundary condition can be imposed by including the boundary as one of the pieces of the model, say the ith piece, and letting its total heat capacity, $d_i V_i c_i$, be infinite. Thus, no amount of heat transferred into or out of the piece will change its temperature. This may or may not be a true representation of the actual physical system, but it gives the desired result of constant temperature. Mathematically, from Eqs. (11) through (13), it is seen that when the total heat capacity of the ith piece is infinite, then

$$\begin{aligned} A_i &= 0 \\ B_{ij} &= 0 \qquad j = 1, 2, \ldots, n \\ C_{ij} &= 0 \qquad j = 1, 2, \ldots, n \end{aligned}$$

thus Eq. (1) becomes

$$\dot{T}_i = 0$$

so the temperature is constant. The FORTRAN subroutine described in Section V was programmed to recognize this case and set the coefficients of the ith equation equal to zero when the input, c_i, is zero.

SECTION III
FORM-SURFACE FACTORS

3.1 DEFINITION

Let Q_{ij} be the portion of the flux incident on the jth piece which was originally emitted by the ith piece. Let W_i be the flux emitted by the ith piece, for radiant flux that is

$$W_i = S_i \epsilon_i \sigma T_i^4 \tag{14}$$

The form-surface factor is defined as

$$G_{ij} = \frac{Q_{ij}}{W_i} \tag{15}$$

This definition warrants some elaboration. It is tacitly assumed that flux can be distinguished as to its origin. Picture, for instance, radiant energy as a stream of photons. Let the units of flux be photons per second and assume that the average time from a photo being emitted to it being absorbed is negligible. Consider a photon which is emitted by the ith piece, is reflected around the system striking the jth piece six times, and finally is absorbed by the kth piece. This photon will contribute one unit to W_i and six units to Q_{ij}. Statistically, as the number of photons becomes large, the ratio, Eq. (15), will become constant. It can be seen that G_{ij} depends only on the geometry and surface properties of the system; thus its name: the form-surface factor. With this idea in mind it is no longer necessary that flux be distinguished as to its origin. It is only necessary to be able to associate with the flux emitted by the ith piece a quantity of flux, Q_{ij}, incident on the jth piece, for which the ratio, Eq. (15), will be constant as W_i is varied. Obviously, the ratio of Eq. (15) will depend on the distribution of W_i over the ith piece and also on the angular emittance at each point. The form-surface factor will be defined by Eq. (15) where W_i is emitted diffusely and uniformly.

3.2 DERIVATIONS

There are several ways to derive an equation for the matrix, G, of the form-surface factors of a system. Three derivations are included here because they lead to a better understanding of the factor and because some interesting observations can be made on the respective approximations.

Consider the integral equation for the irradiation

$$H(\vec{x}) = \int_S \rho(\vec{y})\, H(\vec{y})\, F(\vec{y},\vec{x})\, dS_y - \int_S \epsilon(\vec{y})\, \sigma\, T^4(\vec{y})\, F(\vec{y},\vec{x})\, dS_y \tag{16}$$

where

$$F(\vec{y},\vec{x}) = \lim_{\Delta S_x \to 0} \frac{F_{dS_y - \Delta S_x}}{\Delta S_x} \tag{17}$$

where ΔS_x is a small area around the point x. If the integrals are approximated using the partitioning as determined by the mathematical model as described in Section II, one obtains

$$H_i = \sum_{j=1}^{n} \rho_j H_j \frac{F_{ji}}{S_i} S_j + \sum_{j=1}^{n} \epsilon_j \sigma T_j^4 \frac{F_{ji}}{S_i} S_j \tag{18}$$

Multiplying through by S_i this becomes

$$Q_i = \sum_{j=1}^{n} \rho_j Q_j F_{ji} + \sum_{j=1}^{n} W_j F_{ji} \tag{19}$$

Letting Q and W be column matrices of the Q_i and W_i, respectively, and letting R be the diagonal matrix of the reflectivities, Eq. (19) can be written as the matrix equation

$$Q = F'RQ + F'W \tag{20}$$

Solving for Q one obtains

$$Q = (I - F'R)^{-1} F'W \tag{21}$$

Defining

$$G = F(I - RF)^{-1} \tag{22}$$

Equation (21) becomes

$$Q = G'W \tag{23}$$

or

$$Q_j = \sum_{i=1}^{n} G_{ij} W_i \tag{24}$$

From Eq. (24) it is seen that a quantity of flux, Q_{ij}, incident on the jth piece given by

$$Q_{ij} = G_{ij} W_i \tag{25}$$

can be associated with the flux emitted by the ith piece. From Eq. (22) it is seen that G depends on F and R which are determined by the geometry of the system and its surface properties, respectively. Furthermore, G does not depend on W; thus the elements G_{ij} of the matrix, G, conform to the definition of the form-surface factor.

Equation (22) can also be derived by summing all the reflections. Let Q^k be the column matrix of the incident flux which has been reflected exactly k times. The flux incident directly is

$$Q^0 = F'W \tag{26}$$

If it is assumed that flux is redistributed uniformly over a piece before being reflected, then the flux vector for the next reflection can be obtained from the previous one by the equation

$$Q^k = F'RQ^{k-1} \qquad k > 0 \tag{27}$$

The total incident flux is the sum of all the reflections or

$$\begin{aligned} Q &= \sum_{k=0}^{\infty} Q^k \\ &= F'W + F'RQ^0 + F'RQ^1 + \ldots \\ &= F'W - F'R(Q^0 + Q^1 - \ldots) \end{aligned} \tag{28}$$

or

$$Q = F'W + F'RQ \tag{29}$$

This is the same as Eq. (20); thus Eq. (22) for G follows the same as before.

A different equation for G can be obtained from an equation that is consequent from the definition of G. The flux incident on the jth piece resulting from W_i is the sum of the flux incident directly from the ith piece and the flux from all reflections. If it is assumed that flux incident on a piece directly from the ith piece is redistributed uniformly over the piece before being reflected the first time, then the form-surface factor can be used to obtain the flux incident on the jth piece from all reflections. Thus it is deduced that

$$Q_{ij} = W_i F_{ij} + \sum_{k=1}^{n} W_i F_{ik} \rho_k G_{kj} \tag{30}$$

Thus, from Eq. (15) is obtained

$$G_{ij} = F_{ij} + \sum_{k=1}^{n} F_{ik} \rho_k G_{kj} \tag{31}$$

from which is obtained the matrix equation

$$G = F - FRG \tag{32}$$

Solving for G one obtains

$$G = (I - FR)^{-1} F \tag{33}$$

Equations (22) and (33) are two different equations; however, they give the same matrix, G, as is seen from

$$F(I - RF)^{-1} \overset{?}{=} (I - FR)^{-1}F \tag{34}$$

Multiplying each side of this equation on the left by (I - FR) and on the right by (I - RF) one obtains

$$(I - FR)\, F\, (I - RF)^{-1} (I - RF) \overset{?}{=} (I - FR)\, (I - FR)^{-1}F\, (I - RF)$$

or

$$(I - FR) F \overset{?}{=} F(I - RF) \tag{35}$$

This multiplication does not change the equality since the multipliers are nonsingular. Equation (35) is indeed an equality since by the distributive law it is equivalent to

$$F - FRF = F - FRF \tag{36}$$

which verifies the equivalence of Eqs. (22) and (33).

It is interesting to note that an equation for G has been derived in three different ways using three different simplifying approximations. The approximation in the first derivation was the discretization of the integrals. In the second derivation flux was redistributed uniformly over a piece before each reflection, whereas in the third derivation the flux was redistributed uniformly over each piece before only the first reflection. Nevertheless, the matrices obtained for G in all three derivations are identical.

3.3 CONSERVATION OF ENERGY

From the definition of the form factor one obtains for a closed system the relation that the sum of each row of F must be one, or

$$\sum_{j=1}^{n} F_{ij} = 1 \tag{37}$$

Since the form-surface factor is a generalization of the form factor, one would look for a relation for G generalizing Eq. (37). This generalization is easily obtained from the definition of the form-surface factor and the conservation of energy. Since in a closed system, the energy emitted by a piece must all be absorbed,

$$\sum_{j=1}^{n} W_i G_{ij} \alpha_j = W_i \tag{38}$$

or dividing by W_i

$$\sum_{j=1}^{n} \alpha_j G_{ij} = 1 \tag{39}$$

Thus, from physical laws, the form-surface factors must satisfy Eq. (39). However, it does not necessarily follow that the matrix, G, given by Eq. (22) satisfies Eq. (39), since Eq. (22) was derived under approximate conditions. If Eq. (22) is to be used to calculate G, it is imperative that it be known how well the resulting G satisfies Eq. (39), since conservation of energy in the mathematical model depends on Eq. (39). Fortunately, it can be proved mathematically that the G given by Eq. (22) satisfies Eq. (39) exactly.

Multiplying both sides of Eq. (22) on the right by (I - RF) one obtains

$$G - GRF = F \tag{40}$$

Summing row i of Eq. (40) one obtains

$$\sum_{j=1}^{n} G_{ij} - \sum_{j=1}^{n}\sum_{k=1}^{n} G_{ik}\,\rho_k F_{kj} = \sum_{j=1}^{n} F_{ij} \tag{41}$$

Rearranging this becomes

$$\sum_{j=1}^{n} G_{ij} - \sum_{k=1}^{n} G_{ik}\,\rho_k\left(\sum_{j=1}^{n} F_{kj}\right) = \left(\sum_{j=1}^{n} F_{ij}\right) \tag{42}$$

But, by Eq. (37), the quantities in parentheses are one, so Eq. (42) becomes

$$\sum_{j=1}^{n} G_{ij} - \sum_{k=1}^{n} G_{ik}\,\rho_k = 1 \tag{43}$$

Since k in the second summation is a dummy index it can be changed to j and the two summations can be combined. Factorization of G_{ij} and substitution of α_j for $(1 - \rho_j)$ gives the desired result, Eq. (39).

3.4 RECIPROCITY LAW

It can be proved that form factors obey the reciprocity law, that is that SF is symmetric or

$$SF = (SF)' \tag{44}$$

where S is the diagonal matrix of the S_i. It can be proved from the second law of thermodynamics that the form-surface factor also obeys the reciprocity law. From Eq. (8) the net rate of heat transfer from the ith piece to the jth piece is

$$P_{ij}^{a} = \alpha_j\ G_{ij}\,S_i\,\epsilon_i\,\sigma\,T_i^4 - \alpha_i\,G_{ji}\,S_j\,\epsilon_j\,\sigma\,T_j^4 \tag{45}$$

If $T_i = T_j = T_0$ then the second law of thermodynamics requires that P_{ij}^{a} be zero. Setting the emissivities equal to the absorptivities this condition becomes

$$\alpha_i\,\alpha_j\,\sigma\,T_0^4\,(S_i\,G_{ij} - S_j\,G_{ji}) = 0 \tag{46}$$

In general the factors on the left are not zero, thus the quantities in parentheses must be zero. Thus SG must be symmetric or

$$SG = (SG)' \tag{47}$$

By reasoning parallel to that in Section 3.3, it is desirable to prove mathematically that the matrix, G, given by Eq. (22), satisfies Eq. (47); since this is required if the mathematical model is to obey the second law of thermodynamics.

Multiplying both sides of Eq. (22) on the left by S one obtains

$$SG = SF\,(I - RF)^{-1} \tag{48}$$

Using the substitution, Eq. (44) for SF, and the fact that S is diagonal, thus symmetric, Eq. (48) becomes

$$SG = F'S\,(I - RF)^{-1} \tag{49}$$

Since S is diagonal and has no zero elements on the diagonal, it is nonsingular and one can write

$$\begin{aligned} SG &= F'[(I - RF)\,S^{-1}]^{-1} \\ &= F'[S^{-1} - RFS^{-1}]^{-1} \\ &= F'[S^{-1} - S^{-1}\,SRFS^{-1}]^{-1} \end{aligned} \tag{50}$$

Since S and R are both diagonal, they commute; thus

$$SG = F'[S^{-1} - S^{-1}R(SF)S^{-1}]^{-1} \tag{51}$$

Making the substitution of Eq. (44) for SF again one obtains

$$\begin{aligned} SG &= F'[S^{-1} - S^{-1}\,RF'SS^{-1}]^{-1} \\ &= F'[S^{-1} - S^{-1}\,RF']^{-1} \\ &= F'[S^{-1}\,(I - RF')]^{-1} \\ &= F'[I - RF']^{-1}S \\ &= [S\,(I - FR)^{-1}F]' \end{aligned} \tag{52}$$

Thus by Eq. (33), which has been proven to yield the matrix G identical to Eq. (22), Eq. (52) becomes

$$SG = (SG)'$$

and it has been proved that the G given by Eq. (22) obeys the reciprocity law.

3.5 SINGULARITY AND NONSINGULARITY CONDITIONS

Equation (22) is a formula for G provided the matrix (I - RF) is nonsingular. It is desirable to know the general conditions for the singularity or nonsingularity of this matrix. These conditions can be induced from physical considerations and then proved mathematically. Consider each subsystem of a given system which is by itself a closed system. If for a given subsystem all the reflectivities are one, then any flux inside the subsystem will be reflected continuously and G for the system will not exist. If, on the other hand, every subsystem has at least one surface with a reflectivity less than one, then any flux in the system, eventually after multiple reflections, will be absorbed and G will exist. These conditions can be stated as follows: Group the pieces into sets $I_1, I_2, \ldots I_m$ such that for each set I_k

$$\sum_{j \in I_k} F_{ij} = 1$$

Then by renumbering the pieces the form factor matrix can be partitioned into a diagonal matrix, the elements of which are the subform factor matrices of the groups described above. The matrix (I - RF) for the system is nonsingular if and only if the matrix (I - RF) for each subsystem is nonsingular. From the above physical considerations, a matrix for a subsystem is nonsingular if and only if at least one of its reflectivities is less than one. The property of the subform factor matrices described above is seen to correspond to the mathematical term, irreducible. A general matrix F is said to be reducible if two sets of subscripts I and J exist such that:

1. There exists i such that $i \in I$
2. There exists j such that $j \in J$
3. $k \in I \cup J$ for every k
4. $k \notin I \cap J$ for every k
5. If $i \in I$ and $j \in J$, then $F_{ij} = 0$

Otherwise the matrix is irreducible.

If (I - RF) is singular then there exists a nontrivial solution to

$$(I - RF)\,X = 0 \tag{53}$$

In terms of the elements this becomes

$$x_i - \rho_i \sum_{j=1}^{n} F_{ij}\,x_j = 0$$

or

$$x_i - \rho_i F_{ii} x_i = \rho_i \sum_{j \neq i} F_{ij}\,x_j$$

Let x′ be the value of the largest element in X, then

$$x_i - \rho_i F_{ii} x_i \leq \rho_i x' \sum_{j \neq i} F_{ij}$$

or

$$x_i - \rho_i F_{ii} x_i \leq \rho_i x'(1 - F_{ii}) \tag{54}$$

In particular if $x_k = x'$, then Eq. (54) becomes for i = k

$$x_k - \rho_k F_{kk} x_k \leq \rho_k x_k - \rho_k F_{kk} x_k \tag{55}$$

The only possible way for Eq. (55) to hold is for ρ_k to equal one and for the equality of Eq. (54) to hold. Thus for (I - RF) to be singular the equality of Eq. (54) must hold for i equal to k. That is

$$x_j = x' \quad \text{for} \quad F_{kj} \neq 0$$

Then, however, Eq. (55) must hold for every k such that

$$x_k = x'$$

The only possibility that there could be elements of X not equal to x′ is for there to exist two sets of subscripts I and J such that if $k \in I$ and $j \in J$ then $F_{kj} = 0$, that is for F to be reducible. If $k \in I$, then x_k must equal x′; thus Eq. (55) must hold, which requires ρ_k to be one. Thus a necessary and sufficient condition for (I - RF) to be singular is for the system itself to be irreducible and all of its reflectivities be one, or for there to exist an irreducible subsystem for which all the reflectivities are one. If every irreducible subsystem has at least one surface with a reflectivity less than one, then (I - RF) will be nonsingular.

SECTION IV
CALCULATION OF FORM-SURFACE FACTORS

Just as there were a variety of ways to derive the formula for G, there are a variety of ways to calculate it: namely, matrix inversion, summation, and iteration. The obvious way is directly from the formula, Eq. (22), using a matrix inversion routine. This has the advantage of being simple, and since it is direct, no convergence criterion is needed. Though it was not apparent at the outset, matrix inversion is also the quickest method. However, the other methods will also be given here, since it is sometimes desirable to do a reflection by reflection analysis of a system, in which case the summation and iteration techniques are applicable, whereas the matrix inversion method is not.

4.1 MATRIX INVERSION METHOD

The subroutine GINV was written in FORTRAN IV to calculate G using the matrix inversion method. The routine is called by the statement

```
CALL GINV (N, NDIM, R, F, G, DET)
```

where

N Contains n

NDIM Contains the dimension of F and G (see example below)

R Is a one-dimensional array containing the reflectivities

F Is a two-dimensional array containing the form factors

G Is a two-dimensional array, in which the form-surface factors are returned

DET Returns the value of the determinant of (I-RF) which will be zero if the routine was unable to invert the matrix.

As an example of the purpose of NDIM, suppose the main program has the dimension statement

```
DIMENSION R(60), F(60,60), G(60,60)
```

If G is to be calculated for a system of ten pieces, then the call statement would be

```
CALL GINV (10,60,R,F,G,DET)
```

Thus, it is seen, the dimension of the FORTRAN variables may be greater than or equal to (not less than) the order of the algebraic matrices

The listing of GINV is given in Appendix I. The subroutine GINV calls another subroutine INVERT which is a matrix inversion routine, Appendix II. The timing for GINV is approximately

$$t = t_0 n^2 (3n + 8)$$

where for the IBM System/360 Model 50

$$t_0 = 1.43 \times 10^{-4} \text{ sec}$$

It is to be mentioned that the subroutines set aside 400 bytes of scratch storage in labeled COMMON/SPACE/. This storage is used only during the execution of the subroutines and is free at other times for use by other routines.

4.2 SUMMATION METHOD

Another method of calculating G is to sum the flux from different reflections as was done in the derivation of Eq. (29). From Eq. (15), if W_i is set equal to one, then G_{ij} is equal to Q_{ij}. But if W_j is set equal to zero for j not equal to i, then by definition Q_{ij} is equal to Q_j since the flux incident on the jth piece had to come from the ith piece. Thus, for a given i, if

$$W_j = \delta_{ij} \tag{56}$$

then letting ${}_iG$ be the ith row of G,

$${}_iG = Q' \tag{57}$$

The vector Q can be calculated from Eq. (28). The first term of Eq. (28) is calculated from Eq. (26) where W is given by Eq. (56). Succeeding terms are obtained from Eq. (27). Letting i vary from one to n, G can be calculated a row at a time. A convergence criterion is easily established by noticing that the sum of the elements of Q^k gives the flux remaining after k reflections. Since a unit quantity of flux was started, the summation should be continued until the quantity of flux left is small (to the desired accuracy) when compared to one.

Subroutine GSUM was written to calculate G, employing the method described above. This routine is called by the statement

```
CALL GSUM (N,NDIM,R,F,G,NT,TOL)
```

where N, NDIM, R, F, and G are the same as for GINV. On call, NT contains a limit to the number of terms to be used in the series. On return, NT contains the number of terms actually used. Accuracy is controlled by the argument TOL. Summation is continued until the remaining flux is less than the value contained in TOL. As an example, since unit flux is started; given a value of 0.001 in TOL, summation continues until G is correct to three significant figures. The listing of GSUM is given in Appendix III.

4.3 ITERATION METHOD

Multiplying both sides of Eq. (22) on the right by (I - RF) and transposing the negative term one obtains

$$G = F + GRF \tag{58}$$

from which one can use the method of successive approximations to calculate G. That is, let

$$G^k = \begin{cases} F & \text{for } k = 0 \\ F + G^{k-1}\,RF & \text{for } k > 0 \end{cases} \tag{59}$$

then

$$G = \lim_{k \to \infty} G^k \tag{60}$$

Calculating G in this manner would require one two-dimensional array to store G^{k-1} and another two-dimensional array to store G^k and thus would use more storage than the previous methods. However, storage can be saved by calculating G one row at a time. Writing Eq. (58) in terms of the elements, it becomes

$$G_{ij}^k = F_{ij} + \sum_{\ell=1}^{n} G_{i\ell}^{k-1}\,\rho_\ell F_{\ell k} \tag{61}$$

Letting ${}_iG$ and ${}_iF$ be the ith rows of G and F, respectively, Eq. (61) becomes

$${}_iG^k = {}_iF - {}_iG^{k-1}\,RF \tag{62}$$

Equation (62) can be used to calculate G one row at a time.

Using Eq. (58) to derive an expression for G, one obtains

$$G = F + FRF + FRFRF + \ldots \tag{63}$$

When calculating G a row at a time, one observes that each term of Eq. (63) is the contribution attributable to the corresponding reflection. Thus the iteration and summation methods are equivalent except that the iteration method calculates a partial sum at each iteration, whereas the summation method calculates a term each time and adds it to the total. Since the elements of ${}_iG^k$ are monotonically increasing with k and have upper limits, convergence is ensured. Equation (39) can be used as a convergence criterion to test for any desired accuracy.

It is interesting to note that a formula for calculating G by columns can be obtained in an analogous manner except using Eq. (33) instead of Eq. (22). Letting G_j and F_j be the jth colums of G and F, respectively, the corresponding formula is

$$G_j^k = F_j^k + FRG_j^{k-1} \tag{64}$$

Numerically, calculating G by columns should have the same attributes as calculating it by rows; however, it lacks the convenient convergence criterion, Eq. (39).

Subroutine GROW was written to calculate G by iteration a row at a time. This routine is called by the statement

```
CALL GROW(N,NDIM,R,F,G,NT,TOL)
```

where N, NDIM, R, F, and G are the same as for GINV. On call NT contains a limit to the number of iterations to be taken. On return it contains the number of iterations actually taken. Accuracy is controlled by the argument TOL. Iteration is continued until Eq. (39) is satisfied within the given tolerance. Thus, given a value of 0.001 in TOL, iteration will continue until G is correct to three significant figures. The listing of GROW is given in Appendix IV.

SECTION V
CALCULATION OF THE COEFFICIENTS

5.1 DATA MANAGEMENT

In the process of taking the data of the mathematical model and calculating the coefficients of Eqs. (11) through (13), three tasks are amenable to computer programming for a general system. These are calculation of G, conversion of units, and the evaluation of the equations. The first of these has already been discussed. The subroutines UNIT and COEF were written to perform unit conversions and to evaluate the equations. Before describing these subroutines, however, the data management will be viewed.

Table I tabulates the algebraic variables, the FORTRAN variables, and the dimensional formulas of the data of the mathematical model. For the moment consider only the storage of data. It can be seen that the first eight parameters of the table are stored in the same array P. This was done solely for the purpose of decreasing the length of the argument list of the subroutines. The first eight parameters of Table I are called local parameters, since they give the properties of individual pieces. Once G has been calculated, the form factors are no longer needed. Thus, if this calculation is done before the conducting areas and distances are stored, then the FORTRAN array F can be equivalenced to the variable B. Since a_{ij} and d_{ij} are symmetric, they both can be stored in the same array by storing a_{ij} above the diagonal and d_{ij} below the diagonal, replacing the form factors. Since the products $d_iV_ic_i$ occur in all three of the Eqs. (11) through (13), the subroutine COEF calculates these products, replacing the c_i. The coefficients, A_i, replace the $P_i{}^s$, and the coefficients B_{ij} replace the a_{ij} and d_{ij}. Also, the coefficients C_{ij} replace the G_{ij}; therefore the FORTRAN array G should be equivalenced to the array C. With this arrangement of data, considerable storage economy is realized.

TABLE I
DATA REPRESENTATION AND DIMENSIONAL FORMULAS

Algebraic	FORTRAN	Dimensional Formula
d_i	P(I, 1)	$M_1L_2^{-3}$
V_i	P(I, 2)	L_3^3
c_i	P(I, 3)	$E_4M_5^{-1}\Theta_6^{-1}$
κ_i	P(I, 4)	$E_7T_8^{-1}L_9^{-2}(\Theta_{10}L_{11}^{-1})^{-1}$
α_i	P(I, 5)	---
ϵ_i	P(I, 6)	---
S_i	P(I, 7)	L_{12}^2
P_i^s	P(I, 8)	$E_{13}T_{14}^{-1}$
a_{ij}	B(I, J), J > 1	L_{15}^2
d_{ij}	B(I, J), J < I	L_{16}
σ	SIG	$E_{17}L_{18}^{-2}\Theta_{19}^{-4}$
$d_iV_ic_i$	P(I, 3)	$E\Theta^{-1}$
F_{ij}	F(I, J) ≡ B(I, J)	---
G_{ij}	G(I, J) ≡ C(I, J)	---
A_i	P(I, 8)	ΘT^{-1}
B_{ij}	B(I, J)	T^{-1}
C_{ij}	C(I, J)	$\Theta^{-3}T^{-1}$

5.2 UNIT CONVERSIONS

Subroutine UNIT was written to perform the unit conversions. The subroutine is called by the statement

```
CALL UNIT (N,NDIM,SIG,P,B,FROM,TO)
```

where briefly:

N	Contains n
NDIM	Contains the dimension of the arrays P and B
SIG	Returns σ in desired units
P	Array containing local parameters (See Table I)
B	Two-dimensional array containing conducting areas above the diagonal and the conducting distances below
FROM	Sixteen element integer array indicating units of input data
TO	Nineteen element integer array indicating units of returned data

If the main program contains the statement

```
DIMENSION P (100,8), B(100,100)
```

then in the call statement above, NDIM would contain 100. The dimension of the arrays can be greater than or equal to (not less than) n. The value of σ in the International System of Units (SI) (Ref. 2),

$$\sigma = 5.6697 \times 10^{-8} \frac{\text{J}}{\text{s m}^2 \, {}^\circ\text{K}^4}$$

is stored in the subroutine. The value of σ as indicated in the array TO is returned in the FORTRAN variable SIG. When UNIT is called, P and B contain data in the units indicated in FROM. The subroutine replaces these data with data in units indicated in the array TO.

The dimensional formulas are given in Table I, where the dimensions are indicated as follows:

Symbol	Dimension
L	Length
M	Mass
T	Time
Θ	Temperature
E	Energy

In Table I the input data have subscripts on the dimensional symbols. These subscripts give the position of the indicators for that dimension in the arrays FROM and TO. Thus the fourth elements of FROM and TO contain the indicators of the units of the energy dimension of the specific heat capacity. It is noticed that there are two length dimensions in the dimensional formula for the thermal conductivity. The first derives from the conducting area and the second from the temperature gradient. The formula was left in this form rather than simplified since it is given this way in some tables. The units for σ are not included in the array FROM since its value is not input to the subroutine, but rather is stored internally.

The indicators of the various units and the conversion factors (obtained from Ref. 2) are tabulated in Table II. For example, values of 4 and 2 in the fourth elements of FROM and TO, respectively, indicate that the unit of the energy dimension of the specific heat capacity data input to UNIT is in Btu and it is desired to convert it to ergs. It is to be noticed that a one in each element of TO converts all units to SI. The FORTRAN listing of UNIT is given in Appendix V.

TABLE II
UNIT INDICATORS AND CONVERSION FACTORS

Dimension	Indicator	Unit	Conversion Factor
Length	1	m	1 m/m
	2	cm	0.01 m/cm
	3	in.	0.0254 m/in.
	4	ft	0.3048 m/ft
Mass	1	kg	1 kg/kg
	2	g	0.001 kg/g
	3	lb	0.4535933 kg/lb
	4	slug	14.5939 kg/slug
Time	1	s	1 s/s
	2	min	60 s/min
	3	hr	3600 s/hr
Temperature	1	°K	1 °K/°K
	2	°R	0.5555556 °K/°R
Energy	1	J	1 J/J
	2	erg	1 x 10^{-7} J/erg
	3	cal	4.184 J/cal
	4	Btu	1054.35 J/Btu

Note: s = seconds
J = joules

5.3 EVALUATION OF EQUATIONS

Subroutine COEF was written to evaluate Eqs. (11) through (13). It is assumed that all data are in consistent units. The subroutine is called by the statement

CALL COEF (N,NDIM,SIG,P,B,C)

where:

N Contains n

NDIM Contains dimension of P, B, and C

SIG Inputs the value of σ in units consistent with the rest of the data

P Array which inputs the local parameters, returns the product d_i V_i c_i in place of the c_i, and returns the coefficients A_i in place of the P_is

B Two-dimensional array which inputs the conducting areas above the diagonal and the conducting distances below the diagonal; returns the B coefficients

C Two-dimensional array which inputs G and returns the C coefficients

The argument, NDIM, is the same as for UNIT except that it applies also to the array C. When evaluating Eq. (12), it is assumed that pieces are in contact if and only if their conducting areas are positive. If c_i is zero, then A_i, the ith row of B, and the ith row of C are set to zero (see Section 2. 2). The evaluation of the equations is straightforward, and the argument descriptions above are self-explanatory. A FORTRAN listing of COEF is given in Appendix VI.

SECTION VI
SOLUTION OF THE DIFFERENTIAL EQUATIONS

6.1 NUMERICAL METHOD

The numerical method used to obtain solutions of Eq. (10) is the Taylor expansion of order four (Ref. 1). Leaving off the summation signs and letting it be understood that there is a summation over j from one to n, Eq. (10) and its derivatives are:

$$\dot{T}_i = A_i + B_{ij} T_j + C_{ij} T_j^4 \tag{65}$$

$$\ddot{T}_i = \dot{T}_j (B_{ij} + 4 C_{ij} T_j^3) \tag{66}$$

$$\dddot{T}_i = \ddot{T}_j (B_{ij} + 4 C_{ij} T_j^3) + 12 C_{ij} T_j^2 \dot{T}_j^2 \tag{67}$$

$$\ddddot{T}_i = \dddot{T}_j (B_{ij} + 4 C_{ij} T_j^3) + 24 C_{ij} T_j \dot{T}_j (\dot{T}_j^2 + 1.5 T_j \ddot{T}_j) \tag{68}$$

By the Taylor formula with a remainder (Ref. 3) the temperature can be written

$$T_i(t) = T_i(t_1) + \dot{T}_i (t_1) (t - t_1) + \frac{1}{2} \ddot{T}_i (t_1) (t - t_1)^2 + \frac{1}{6} \dddot{T}_i(t_1) (t - t_1)^3 + \frac{1}{24} \ddddot{T}_i (t^*) (t - t_1)^4 \tag{69}$$

where

$$t_1 < t^* < t \tag{70}$$

The errors obtained by approximating Eq. (69) by

$$T_i(t) = T_i (t_1) + \dot{T}_i (t_1) (t - t_1) + \frac{1}{2} \ddot{T}_i (t_1) (t - t_1)^2 + \frac{1}{6} \dddot{T}_i (t_1) (t - t_1)^3 + \frac{1}{24} T_i (t_1) (t - t_1)^4 \tag{71}$$

are

$$E_i = \frac{1}{24} | \ddddot{T}_i (t_1) - \ddddot{T}_i (t^*)| (t - t_1)^4 \tag{72}$$

If the temperature error per unit time is to be less than A_C, then for the time interval from t_1 to t, it is necessary to ensure the inequality

$$E_i < A_c (t - t_1) \tag{73}$$

for all i. It is known that the derivative of the temperatures with respect to time of the physical system described by Eq. (10) approach constants. This indicates that the absolute value of the derivatives of higher order decreases monotonically with t to zero. Therefore, from Eq. (72)

$$E_i < \frac{1}{24} | \ddddot{T}_i (t_1) | (t - t_1)^4 \tag{74}$$

Therefore, to guarantee Eq. (74), it is sufficient to ensure for all i

$$\frac{1}{24} | \ddddot{T}_i (t_1) | (t - t_1)^4 < A_c (t - t_1) \tag{75}$$

or

$$(t - t_1) < \left(\frac{24 A_c}{|\ddddot{T}(t_1)|} \right)^{1/3} \tag{76}$$

To find the maximum time, t_m, for which the Taylor expansions for all n temperatures are valid one must use the maximum value over i for the denominator of Eq. (76), so

$$t_m = t_1 + \left(\frac{24A_c}{\max_i \left| \ddddot{T}_i(t_1) \right|} \right)^{1/3} \qquad (77)$$

Given the initial temperatures and time, the Taylor expansion and t_m are calculated. If the time, t, at which the temperatures are desired is less than t_m, then the Taylor expansions are evaluated at t. Otherwise the Taylor expansions are evaluated at t_m and then t_1 is set equal to t_m and new expansions and a new t_m are calculated. This is repeated until t is less than t_m. The flow chart of this method is shown in Fig. 3.

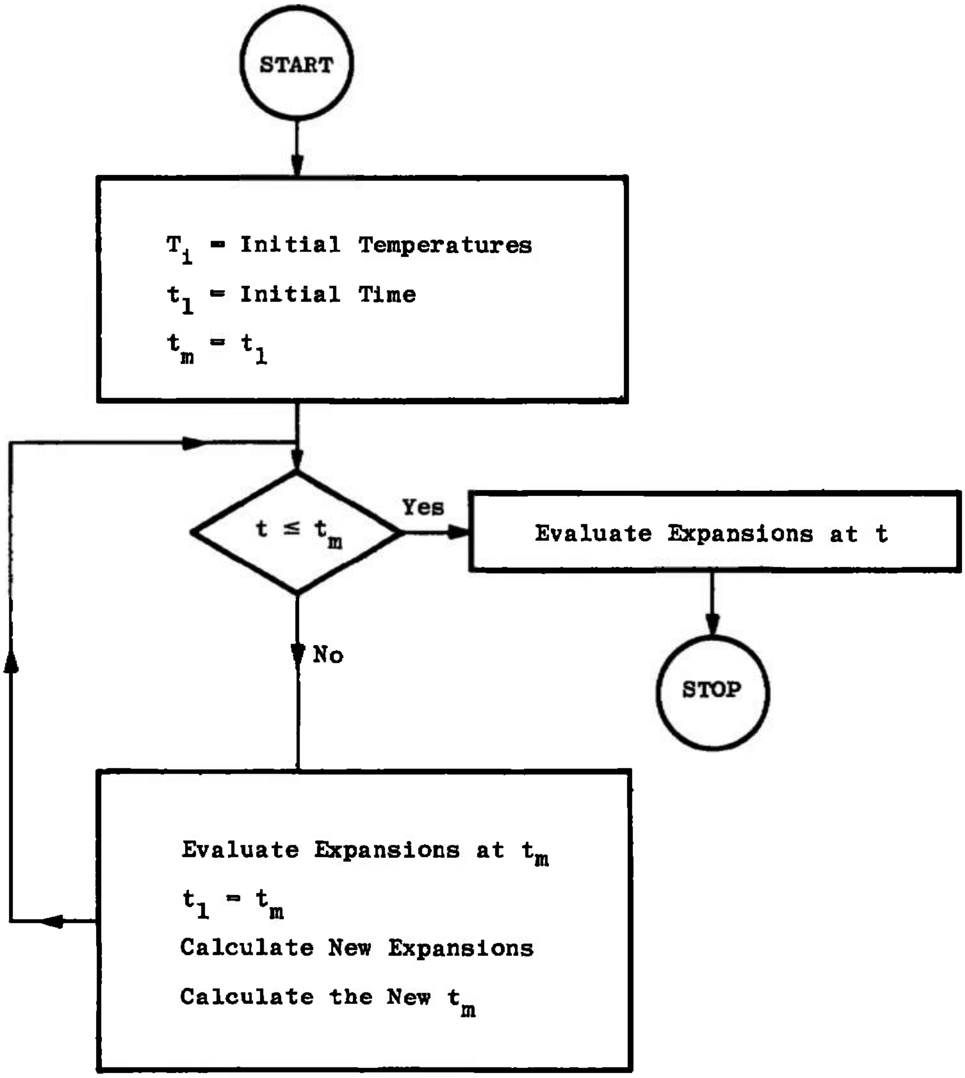


Fig. 3 Flow Chart of Numerical Method for Obtaining the Solutions to Eq. (10)

6.2 SUBROUTINE SOLU

Subroutine SOLU was written employing the numerical method described. The subroutine is called by the statement

```
CALL SOLU (N,NDIM,AC,T1,T,TM,TEMP,Q,A,B,C)
```

where

N Contains n

NDIM Contains the dimension of Q, A, B, and C

AC Contains the allowable temperature error per unit time

T1 On first call contains the initial time: returns with the base time for the Taylor expansions and must be left unchanged for subsequent calls.

T Contains the time at which the temperatures are desired.

TM On the first call contains the initial time; returns with t_m and must be left unchanged for subsequent calls

TEMP Returns the temperatures at the time contained in T

Q On first call Q(1,I) must contain the initial temperature; on return Q contains the Taylor expansions of the temperatures and must be left unchanged for subsequent calls

A One-dimensional array containing the A coefficients

B and C Two-dimensional arrays containing the B and C coefficients, respectively.

In the main program, if the arrays are dimensioned

```
DIMENSION TEMP (70),Q(5,70),A(70),B(70,70),C(70,70)
```

then in the call statement above, NDIM would contain 70. The dimension of the arrays can be greater than or equal (not less than) n. Note that if the local parameters are no longer needed, one can save storage by the statement (assuming that NDIM is the same for COEF and SOLU)

```
EQUIVALENCE (P(1),Q(1)), (P(1,7),TEMP (1)), (P(1,8),A(1))
```

The FORTRAN listing of SOLU is given in Appendix VII.

SECTION VII
EXAMPLE PROBLEMS

The following example problems are to serve as a guide for users of the computer programs. The problems were chosen such that solutions were available so the results could be checked; thus they serve as check problems for the programs. Although the examples are simple, the real usefulness of the programs is for much more complicated problems.

7.1 ISOTHERMAL RADIATING OBJECT WITH A HEAT SOURCE

The differential equation for an isothermal radiating object with a heat source is

$$dVc\dot{T} = P^{s} - S\epsilon\sigma T^{4} \tag{78}$$

which can be put in the form

$$\dot{T} = A + CT^{4} \tag{79}$$

It is seen that the steady-state temperature, T_∞, is

$$T_\infty = \left(\frac{-A}{C}\right)^{1/4} \tag{80}$$

and the initial time derivative of the temperature is

$$\dot{T}_0 = A + CT_0^{\,4} \tag{81}$$

In these terms the solution is

$$t = \begin{cases} \dfrac{T_\infty^{\,4} - T_0^{\,4}}{4\dot{T}_0\, T_\infty^{\,3}} \left[\ln \left|\dfrac{T + T_\infty}{T - T_\infty}\right| + 2 \tan^{-1} \dfrac{T}{T_\infty} - \ln \left|\dfrac{T_0 + T_\infty}{T_0 - T_\infty}\right| - 2 \tan^{-1} \dfrac{T_0}{T_\infty} \right] & \text{for } T_\infty \neq 0 \\[2ex] \dfrac{T_0}{3\dot{T}_0} \left[1 - \dfrac{T_0^{\,3}}{T^{3}}\right] & \text{for } T_\infty = 0 \end{cases} \tag{82}$$

A program was written using SOLU to solve Eq. (79) and compare the solution with Eq. (82). Some of these solutions are shown in Fig. 3. The accuracy was better than that specified by the FORTRAN variable, AC, and the error was much too small to be seen in Fig. 4. The FORTRAN listing of this program is given in Appendix VIII.

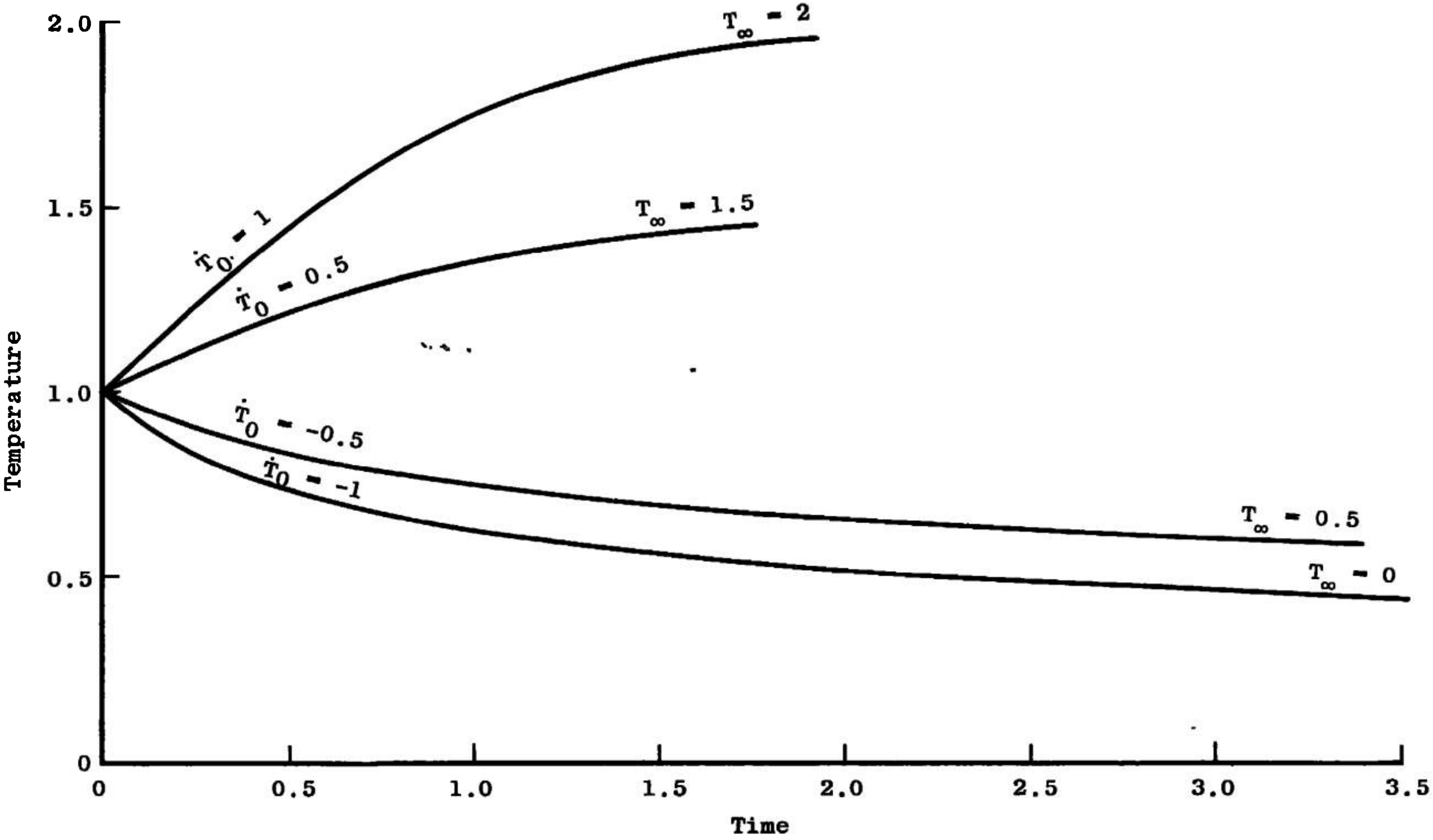


Fig. 4 Solutions to Isothermal Radiating Object with a Heat Source

7.2 CONDUCTING RADIATING SPHERICAL SHELL

Consider a spherical shell of thickness b and radius a shown in Fig. 5. The emissivity of the inside surface is ϵ^{i} and the outside surface is ϵ^{0}. The sphere is exposed to parallel radiation of uniform cross section of intensity q_s for which the absorptivity of the surface is α_s. There is vacuum inside and outside the sphere and it is assumed that b << a. The heat-transfer equations for this system have been derived in Ref. 4, and a numerical method of solution for the steady-state equations is given. The numerical method described in this report was applied to this system to obtain the steady-state solution as an illustration of the application of the method.

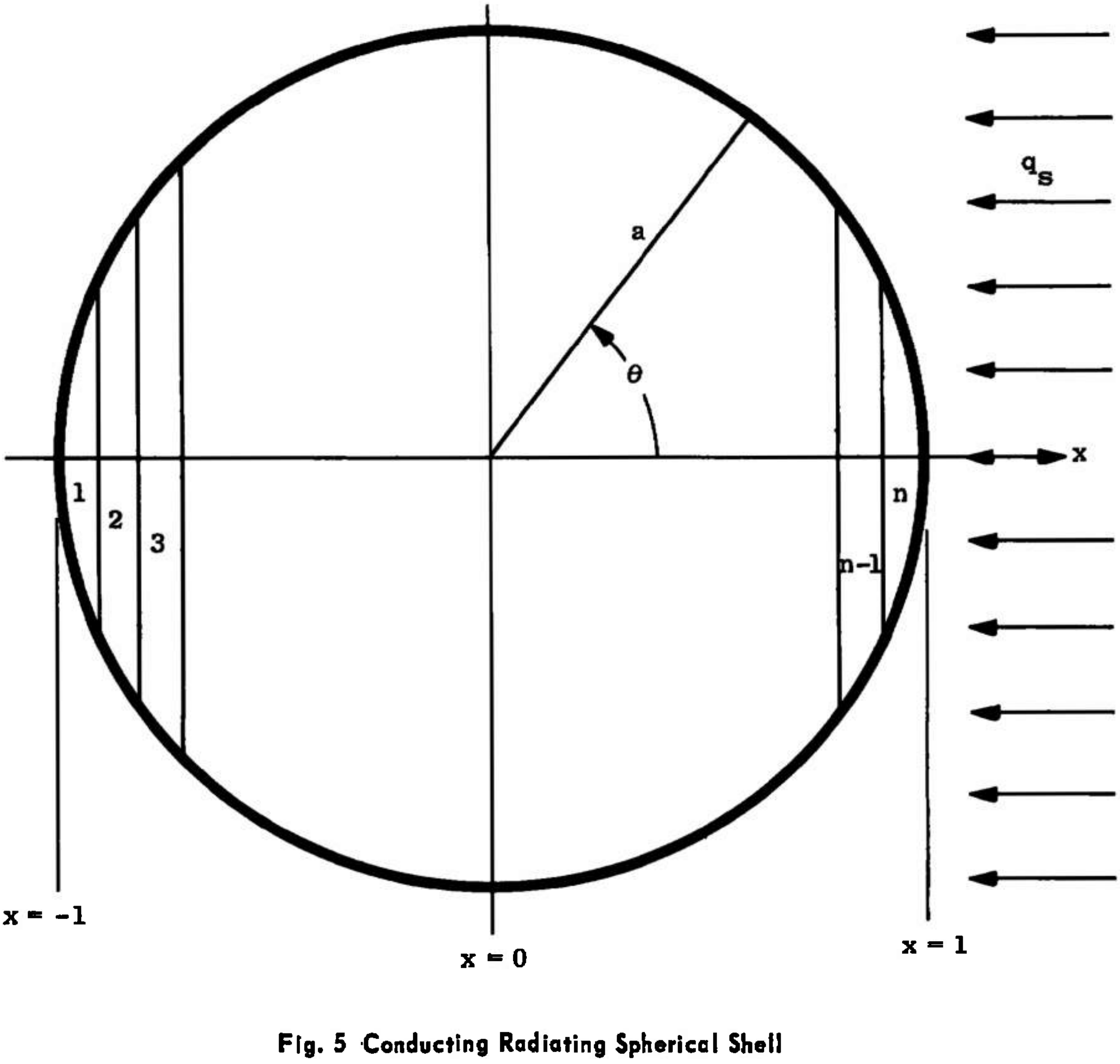


Fig. 5 Conducting Radiating Spherical Shell

Dividing the system into n pieces one obtains the following equations:

$$x = \cos \theta$$

$$\Delta x = \frac{2}{n} \approx -\sin \theta \, \Delta \theta$$

$$\Delta\theta \approx \frac{-\Delta x}{\sqrt{1 - x^2}}$$

$$d_{ij} = -a \Delta \theta \approx \frac{a \Delta x}{\sqrt{1 - x^2}}$$

$$a_{ij} = 2\pi ab \sin \theta = 2\pi ab \sqrt{1 - x^2}$$

$$P_i^s = (\alpha_s q_s)(2\pi a \sin \theta)(- \Delta a \sin \theta)$$

$$\approx 2\pi a^2 \alpha_s q_s x \Delta x$$

$$V_i = 2\pi a^2 b \Delta x$$

It will be assumed that $\alpha^i = \epsilon^i$ and $\alpha^0 = \epsilon^0$, but note that the interior radiating surface of a piece is different than the exterior surface, which is contrary to the assumptions of the mathematical method. One could divide the sphere into two spherical shells, but this would require twice as many pieces. This can be avoided by defining effective emissivities and form-surface factors in the following manner. The radiating area will be the sum of the internal and external areas, so

$$S_i = 4\pi a^2 \Delta x$$

The effective emissivity, ϵ', will be defined so that the rate radiant energy is emitted remains unchanged, so

$$S_i \epsilon' \sigma T_i^4 = \frac{S_i}{2} \epsilon^i \sigma T_i^4 + \frac{S_i}{2} \epsilon^0 \sigma T_i^4$$

or

$$\epsilon' = \frac{\epsilon^i + \epsilon^0}{2}$$

The effective form-surface factor, G'_{ij}, will be defined so the heat transfer by radiation between two pieces remains unchanged, so

$$\alpha' S_i \epsilon' \sigma T_i^4 G'_{ij} = \alpha^i S_i \epsilon^i \sigma T_i^4 G_{ij}$$

or

$$G'_{ij} = \left(\frac{\epsilon^i}{\epsilon'}\right)^2 G_{ij}$$

The effective form-surface factor is thus

$$G'_{ij} = \left(\frac{\epsilon^i}{\epsilon'}\right)^2 \frac{\Delta x}{4\alpha^i}$$

In Ref. 4 it is proved that the steady-state temperature distribution depends only on the two parameters

$$N_R = \frac{a^2 \epsilon^0 \sigma}{b\,k} \left(\frac{\alpha_s\, q_s}{4 \epsilon^0 \sigma}\right)^{3/4}$$

and

$$E = \frac{\epsilon^i}{\epsilon^0}$$

So that solutions of the two different methods could be compared, the following numerical values were assigned to the system data to obtain the specified N_R and E:

$$a = 1$$

$$\epsilon^0 = 1$$

$$\alpha_s q_s = 4\sigma$$

$$b = \sigma$$

$$k = \frac{1}{N_R}$$

$$\epsilon^i = E$$

A program was written using subroutines COEF and SOLU to obtain the solutions. A FORTRAN listing of the program is given in Appendix IX. Three solutions are shown in Fig. 6. Letting n = 20, the solutions by the two different methods are the same to better than three significant figures:

7.3 CLAUSING FACTOR CALCULATIONS

The formula for the form-surface factor was based on the assumption of diffuse reflection. Thus the use of the formula is not confined to radiation, but can be used to calculate flux transfer of any type provided the diffuse reflection assumption is valid. In fact, the method has been used for molecular flux as reported in Ref. 5. To illustrate the method, subroutine GINV was used to calculate Clausing factors. The Clausing factor is used in vacuum technology to calculate the molecular conductance of tubes and is defined as the ratio of the rate at which gas leaves the outlet of a tube to the rate at which it is incident on the inlet. To employ the method the tube was divided into n-2 equal pieces and the inlet and outlet were each counted as a piece. The form factors for the system for various ratios of length to the radius of the tube, L/r, were calculated. Each piece of the tube was assigned a reflectivity of one, and the inlet and outlet were assigned a reflectivity of zero. It can be seen that the Clausing factor is just G_{1n}. The results obtained were compared with a table given in Ref. 6 and are shown in Fig. 7. The FORTRAN listing of the program used is given in Appendix X.

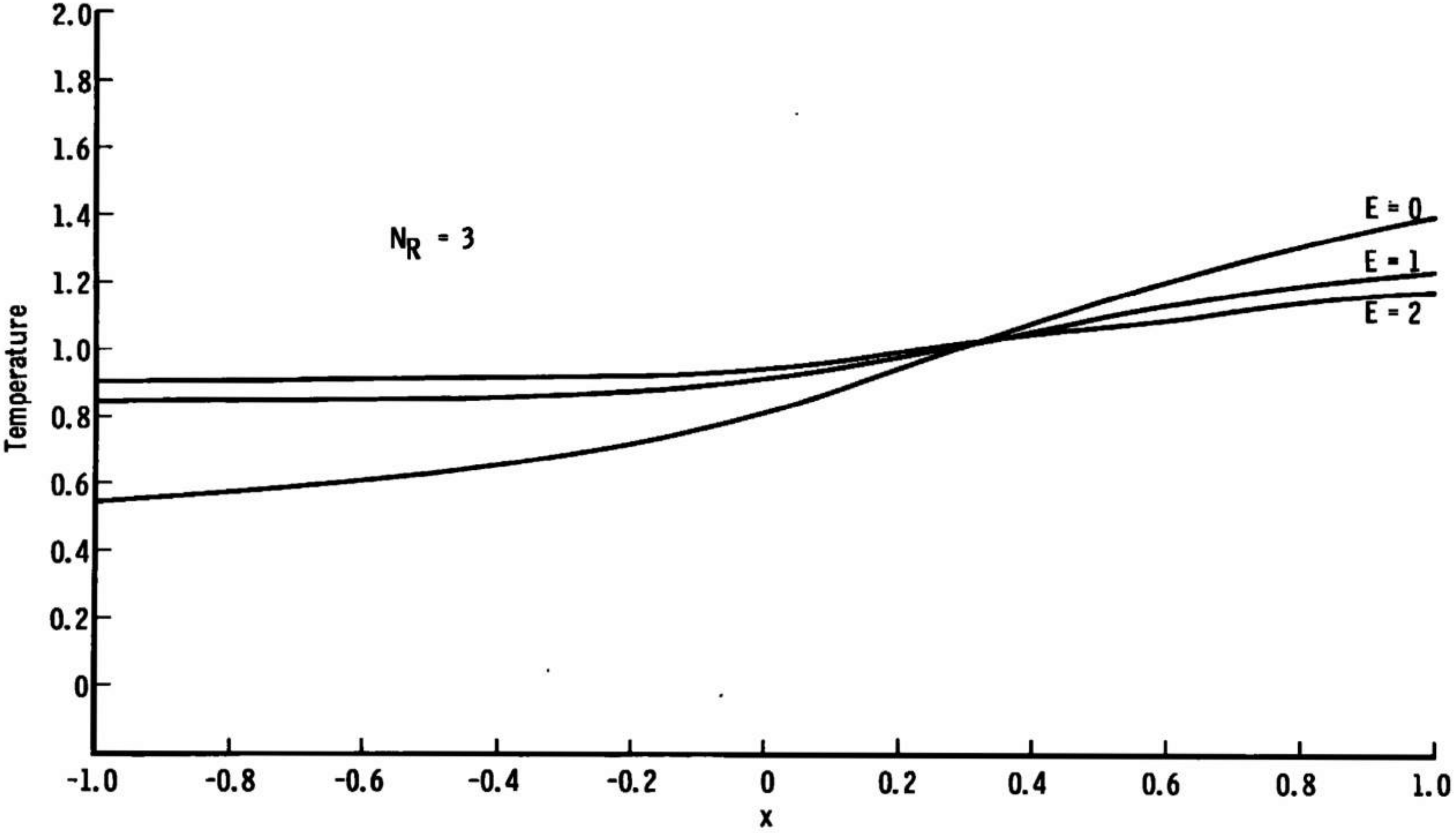


Fig. 6 Solutions to Conducting Radiating Spherical Shell

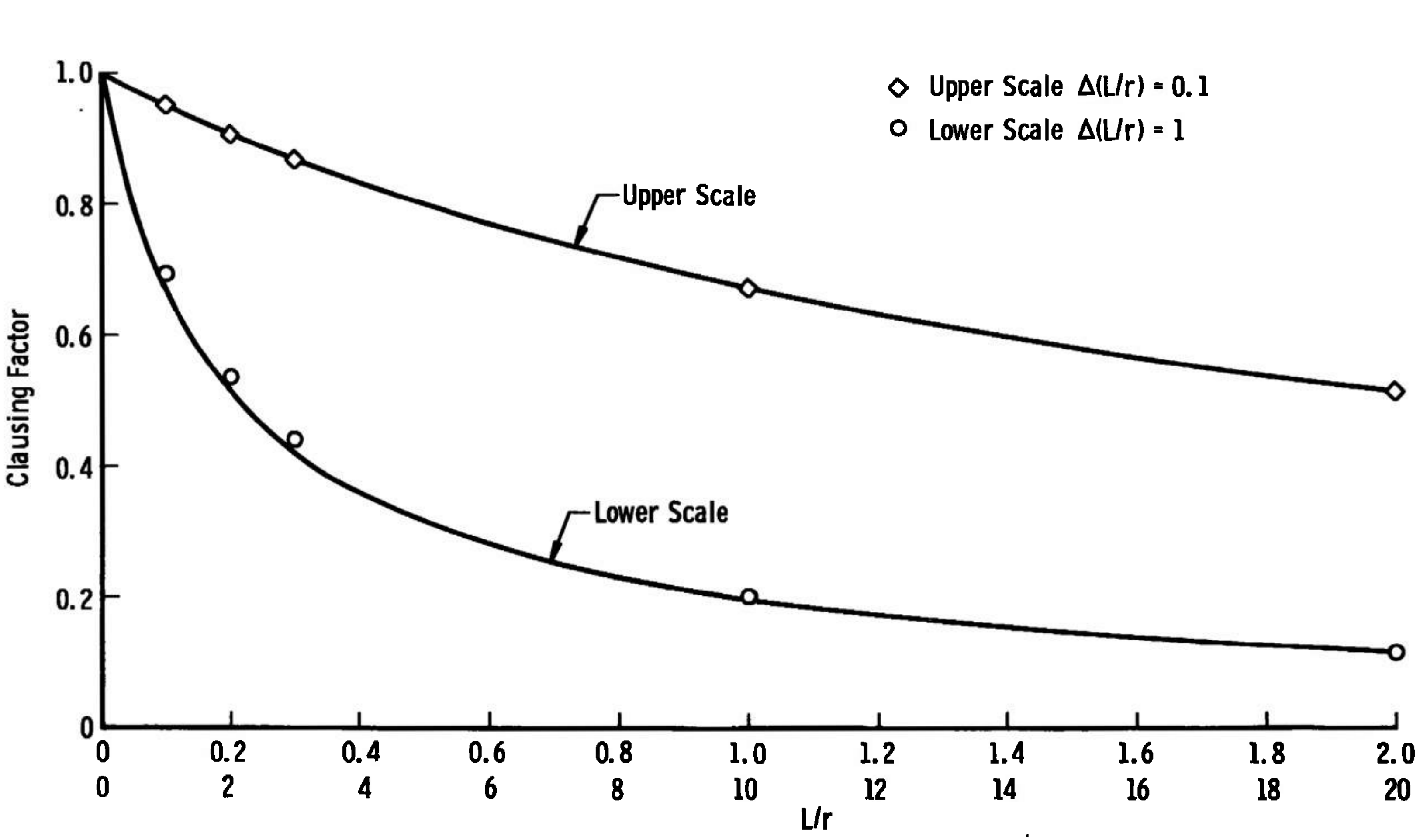


Fig. 7 Clausing Factor Calculation

REFERENCES

1. Henrici, Peter. Discrete Variable Methods in Ordinary Differential Equations. John Wiley and Sons, Inc., New York, 1962.

2. Mechtly, E. A. "The International System of Units, Physical Constants, and Conversion Factors." NASA SP-7012, 1964.

3. Kaplan, Wilfred. Advanced Calculus. Addison-Wesley Press. Cambridge, Massachusetts, 1952, p. 357.

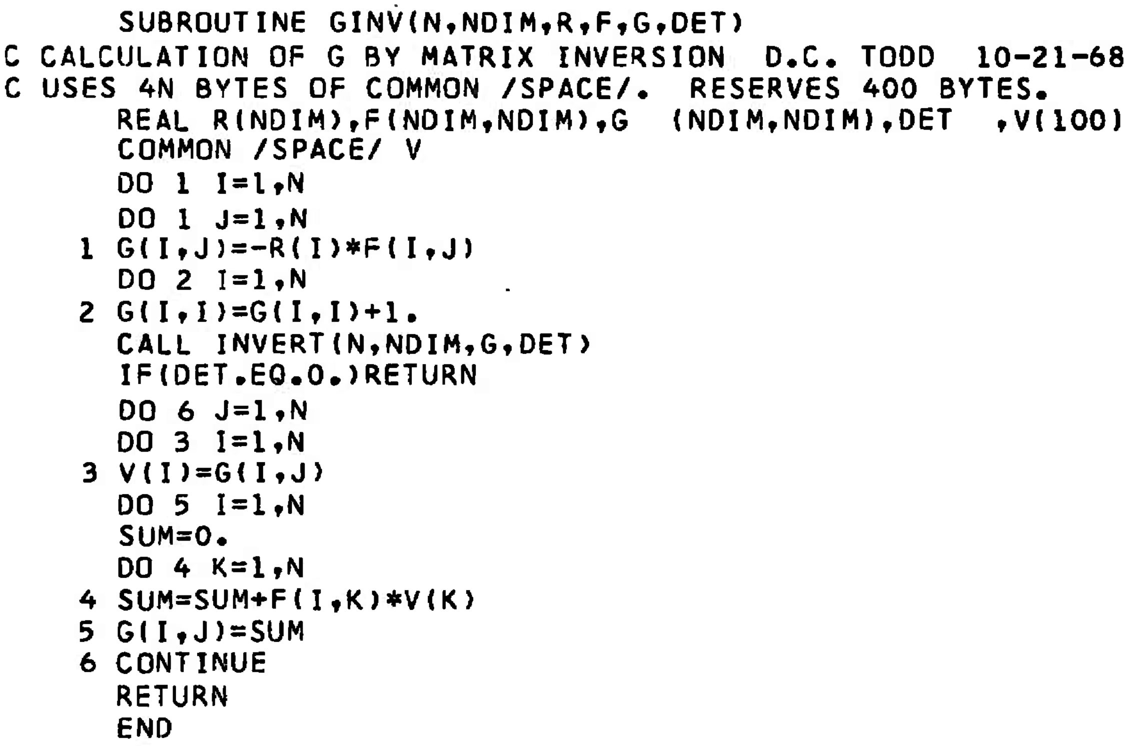
4. Todd, Donald C., van der Bliek, Jan A., and Hsia, Han M. "Effect of Internal Radiation on the Temperature Distribution of a Spherical Shell." AEDC-TR-67-254 (AD664081), January 1968.

5. Link, Cord H. "Molecular Kinetics Studies." AEDC-TDR-63-120 (AD414530), August 1963.

6. Dushman, Saul. Scientific Foundations of Vacuum Technique. Second edition, John Wiley and Sons, Inc., New York, 1962, p. 94.

APPENDIXES

APPENDIX I
FORTRAN LISTING OF GINV



```
      SUBROUTINE GINV(N,NDIM,R,F,G,DET)
C CALCULATION OF G BY MATRIX INVERSION  D.C. TODD  10-21-68
C USES 4N BYTES OF COMMON /SPACE/.  RESERVES 400 BYTES.
      REAL R(NDIM),F(NDIM,NDIM),G  (NDIM,NDIM),DET  ,V(100)
      COMMON /SPACE/ V
      DO 1 I=1,N
      DO 1 J=1,N
    1 G(I,J)=-R(I)*F(I,J)
      DO 2 I=1,N
    2 G(I,I)=G(I,I)+1.
      CALL INVERT(N,NDIM,G,DET)
      IF(DET.EQ.0.)RETURN
      DO 6 J=1,N
      DO 3 I=1,N
    3 V(I)=G(I,J)
      DO 5 I=1,N
      SUM=0.
      DO 4 K=1,N
    4 SUM=SUM+F(I,K)*V(K)
    5 G(I,J)=SUM
    6 CONTINUE
      RETURN
      END
```

APPENDIX II
FORTRAN LISTING OF INVERT

```
      SUBROUTINE INVERT(N,IDIM,A,DET)
C MATRIX INVERSION  D.C. TODD  11-9-67
C USES 4N BYTES OF COMMON /SPACE/.  RESERVES 400 BYTES.
      INTEGER*2 IRC(2,100)
      COMMON /SPACE/ IRC
      DIMENSION A(IDIM,IDIM)
C
      DET=1.
      DO 9 K=1,N
C FIND PIVOT
      ABSP=0.
      DO 1  I=K,N
      DO 1  J=K,N
      ABSA= ABS(A(I,J))
      IF(ABSA.LE.ABSP)GO TO 1
      IRC(1,K)=I
      IRC(2,K)=J
      ABSP=ABSA
      PIVOT=A(I,J)
    1 CONTINUE
C DET IS PRODUCT OF PIVOTS.  IF PIVOT IS ZERO, THEN A IS SINGULAR
      DET=DET*PIVOT
      IF(PIVOT.EQ.0.)GO TO 15
C INTERCHANGE ROWS K AND I
      I=IRC(1,K)
      IF(I.EQ.K)GO TO 3
      DET=-DET
      DO 2 J=1,N
      SAVE=A(K,J)
      A(K,J)=A(I,J)
    2 A(I,J)=SAVE
C INTERCHANGE COLUMNS K AND J
    3 J=IRC(2,K)
      IF(J.EQ.K)GO TO 5
      DET=-DET
      DO 4 I=1,N
      SAVE=A(I,K)
      A(I,K)=A(I,J)
    4 A(I,J)=SAVE
C DIVIDE COLUMN K BY MINUS PIVOT
    5 R=1./PIVOT
      DO 6 I=1,N
    6 A(I,K)=-R*A(I,K)
C REDUCE MATRIX
      DO 20 I=1,N
      IF(I.EQ.K)GO TO 20
      DO 7 J=1,N
      IF(J.EQ.K)GO TO 7
      A(I,J)=A(I,J)+A(I,K)*A(K,J)
    7 CONTINUE
   20 CONTINUE
C DIVIDE ROW K BY PIVOT.  A(K,K)=R


      DO 8 J=1,N
    8 A(K,J)=R*A(K,J)
      A(K,K)=R
    9 CONTINUE
C REARRANGE MATRIX
      K=N+1
   10 K=K-1
      J=IRC(1,K)
      IF(J.EQ.K)GO TO 12
      DO 11 I=1,N
      SAVE=A(I,K)
      A(I,K)=A(I,J)
   11 A(I,J)=SAVE
   12 I=IRC(2,K)
      IF(I.EQ.K)GO TO 14
      DO 13 J=1,N
      SAVE=A(K,J)
      A(K,J)=A(I,J)
   13 A(I,J)=SAVE
   14 IF(K.NE.1)GO TO 10
C
   15 CONTINUE
      RETURN
      END
```

APPENDIX III
FORTRAN LISTING OF GSUM

```
      SUBROUTINE GSUM(N,NDIM,R,F,G,NT,TOL)
C CALCULATION OF G BY SUMMATION  D.C. TODD  10-21-68
C USES 8N BYTES OF COMMON /SPACE/. RESERVES 800 BYTES.
      REAL R(NDIM),F(NDIM,NDIM),G(NDIM,NDIM),V(100,2)
      COMMON /SPACE/ V
      MM=0
      DO 15 I=1,N
      DO 10 J=1,N
      G(I,J)=F(I,J)
   10 V(J,1)=R(J)*F(I,J)
      L1=1
      L2=2
      DO 13 M=1,NT
      ERR=0.
      DO 12 J=1,N
      SUM=0.
      DO 11 K=1,N
   11 SUM=SUM+V(K,L1)*F(K,J)
      G(I,J)=G(I,J)+SUM
      V(J,L2)=R(J)*SUM
   12 ERR=ERR+V(J,L2)
      IF(ERR.LE.TOL)GO TO 14
      L3=L2
      L2=L1
   13 L1=L3
      M=NT
   14 IF(MM.LT.M)MM=M
   15 CONTINUE
      NT=MM
      RETURN
      END
```

APPENDIX IV
FORTRAN LISTING OF GROW

```
C CALCULATION OF G BY ROW ITERATION  D.C. TODD  10-21-68
C USES 8N BYTES OF COMMON /SPACE/. RESERVES 800 BYTES.
      SUBROUTINE GROW(N,NDIM,R,F,G,NT,TOL)
      REAL R(NDIM),F(NDIM,NDIM),G(NDIM,NDIM),V(100,2)
      COMMON /SPACE/ V
      MM=0
      DO 6 I=1,N
      DO 1 J=1,N
    1 V(J,1)=F(I,J)
      L1=1
      L2=2
      DO 4 M=1,NT
      ERR=0.
      DO 3 J=1,N
      SUM=0.
      DO 2 K=1,N
    2 SUM=SUM+V(K,L1)*R(K)*F(K,J)
      V(J,L2)=F(I,J)+SUM
    3 ERR=ERR+V(J,L2)*(1.-R(J))
      L3=L2
      L2=L1
      L1=L3
      IF(1.-ERR.LE.TOL)GO TO 5
    4 CONTINUE
      M=NT
    5 IF(MM.LT.M)MM=M
      DO 6 J=1,N
    6 G(I,J)=V(J,L1)
      NT=MM
      RETURN
      END
```

APPENDIX V
FORTRAN LISTING OF UNIT

```
      SUBROUTINE UNIT(N,NDIM,SIG,P,B,FROM,TO)
C CONVERT UNITS OF SYSTEM DATA  D.C. TODD  10-24-68
      REAL P(NDIM,8),B(NDIM,NDIM),
     L    L(4)/1.,.01,.0254,.3048/,
     M    M(4)/1.,.001,.4535933,14.5939/,
     S    S(3)/1.,60.,3600./,
     T    T(2)/1.,.5555556/,
     E    E(4)/1.,1.E-7,4.184,1054.35/
      INTEGER FROM(16),TO(19)
      SIG=5.6697E-8
      FD=(M(FROM(1))/M(TO(1)))*(L(TO(2))/L(FROM(2)))**3
      FV=(L(FROM(3))/L(TO(3)))**3
      FC=E(FROM(4))*M(TO(5))*T(TO(6))/(E(TO(4))*M(FROM(5))*T(FROM(6)))
      FK=E(FROM(7))*L(FROM(11))*S(TO(8))*T(TO(10))*(L(TO(9))/L(FROM(9)))
     K**2/(E(TO(7))*L(TO(11))*S(FROM(8))*T(FROM(10)))
      FS=(L(FROM(12))/L(TO(12)))**2
      FP=E(FROM(13))*S(TO(14))/(E(TO(13))*S(FROM(14)))
      FA=(L(FROM(15))/L(TO(15)))**2
      FN=L(FROM(16))/L(TO(16))
      FG=(E(FROM(17))/E(TO(17)))*(L(TO(18))/L(FROM(18)))**2
     G*(T(TO(19))/T(FROM(19)))**4
      DO 1 I=1,N
      P(I,1)=FD*P(I,1)
      P(I,2)=FV*P(I,2)
      P(I,3)=FC*P(I,3)
      P(I,4)=FK*P(I,4)
      P(I,7)=FS*P(I,7)
    1 P(I,8)=FP*P(I,8)
      NM1=N-1
      DO 2 I=1,NM1
      IP1=I+1
      DO 2 J=IP1,N
      B(I,J)=FA*B(I,J)
    2 B(J,I)=FN*B(J,I)
      SIG=FG*SIG
      RETURN
      END
```

APPENDIX VI
FORTRAN LISTING OF COEF

```
      SUBROUTINE COEF(N,NOIM,SIG,P,B,C)
C COEFICIENTS FOR HEAT TRANSFER DIFFERENTIAL EQUATIONS
C D.C. TODD  10-23-6B  ST8002-Y00
C P INPUTS D,V,C,K,ALPHA,EPSILON,S, AND P
C   RETURNS DVC IN C AND A IN P
C B INPUTS AREAS ABOVE THE DIAGONAL ANO DISTANCES BELOW
C   RETURNS WITH B COEFICIENTS
C C INPUTS G AND RETURNS WITH C COEFFICIENTS
      REAL P(NDIM.B),B(NDIM,NDIM),C(NDIM,NDIM)
      DO 2 I=1,N
      D=P(I,1)*P(I,2)*P(I,3)
      P(I,3)=D
      IF(D.NE.0.)GO TO 1
      P(I,8)=0.
      GO TO 2
    1 P(I,8)=P(I,8)/D
    2 CONTINUE
      DO 7 I=1,N
      D=P(I,3)
      IF(D.NE.0.)GO TO 24
      B(I,I)=0.
      C(I,I)=0.
      GO TO 7
   24 SUM=0.
      DO 6 K=1,N
      IF(K-I)4,6,3
    3 II=I
      JJ=K
      GO TO 5
    4 II=K
      JJ=I
    5 IF(B(II,JJ).EQ.0.)GO TO 6
      SUM=SUM+P(K,4)*B(II,JJ)/(B(JJ,II)*(P(I,4)+P(K,4)))
    6 CONTINUE
      B(I,I)=-2.*P(I,4)*SUM/D
      C(I,I)=SIG*P(I,6)*P(I,7)*(P(I,5)*C(I,I)-1.)/D
    7 CONTINUE
      NM1=N-1
      DO 13 I=1,NM1
      DI=P(I,3)
      IP1=I+1
      DO 12 J=IP1,N
      DJ=P(J,3)
      IF(B(I,J).EQ.0.)B(J,I)=1.
   11 E=2.*P(I,4)*P(J,4)*B(I,J)/(B(J,I)*(P(I,4)+P(J,4)))
      IF(DI.NE.0.)GO TO 20
      B(I,J)=0.
      CIJ=0.
      GO TO 21
   20 B(I,J)=E/DI
      CIJ=SIG*P(I,5)*C(J,I)*P(J,7)*P(J,6)/DI
   21 IF(DJ.NE.0.)GO TO 22


      B(J,I)=0.
      C(J,I)=0.
      GO TO 23
   22 B(J,I)=E/DJ
      C(J,I)=SIG*P(J.5)*C(I,J)*P(I,7)*P(I,6)/DJ
   23 C(I,J)=CIJ
   12 CONTINUE
   13 CONTINUE
      RETURN
      END
```

APPENDIX VII
FORTRAN LISTING OF SOLU

```
      SUBROUTINE SOLU(N,NDIM,AC,T1,T,TM,TEMP,P,A,B,C)
C SOLUTION TO HEAT TRANSFER DIFFERENTIAL EQUATION
C D.C. TODD  10-25-68  ST8002-Y00
C AT FIRST CALL P(1,I) INPUTS INITIAL TEMPERATURES AND TM=T1
C CONTENTS OF P, T1, AND TM MUST BE LEFT UNCHANGED FOR SUBSEQUENT CALLS
      REAL TEMP(1),P(5,1),A(1),B(NDIM,1),C(NDIM,1)
    1 IF(TM.GE.T)GO TO 2
      X=TM
      K=0
      GO TO 3
    2 X=T
      K=1
    3 DX=X-T1
      DO 4 I=1,N
    4 TEMP(I)=P(1,I)+DX*(P(2,I)+DX*(P(3,I)+DX*(P(4,I)+DX*P(5,I))))
      IF(K.EQ.1)RETURN
      T1=TM
      P5M=0.
      DO 5 I=1,N
      P(1,I)=TEMP(I)
      P(2,I)=A(I)
      P(3,I)=0.
      P(4,I)=0.
    5 P(5,I)=0.
      DO 6 I=1,N
      DO 6 J=1,N
    6 P(2,I)=P(2,I)+B(I,J)*P(1,J)+C(I,J)*P(1,J)**4
      DO 7 I=1,N
      DO 7 J=1,N
    7 P(3,I)=P(3,I)+P(2,J)*(B(I,J)+4.*C(I,J)*P(1,J)**3)
      DO 8 I=1,N
      DO 8 J=1,N
    8 P(4,I)=P(4,I)+P(3,J)*(B(I,J)+4.*C(I,J)*P(1,J)**3)
     1         +12.*C(I,J)*P(1,J)**2*P(2,J)**2
      DO 10 I=1,N
      DO 9  J=1,N
    9 P(5,I)=P(5,I)+P(4,J)*(B(I,J)+4.*C(I,J)*P(1,J)**3)
     1         +24.*C(I,J)*P(1,J)*P(2,J)*(P(2,J)**2+1.5*P(1,J)*P(3,J))
      AP5=ABS(P(5,I))
      IF(AP5.GT.P5M)P5M=AP5
   10 CONTINUE
      P5M=P5M/24.
      DO 11 I=1,N
      P(3,I)=.5*P(3,I)
      P(4,I)=.1666667*P(4,I)
   11 P(5,I)=.04166667*P(5,I)
      IF(P5M.GT.1.E-66)GO TO 12
      TM=1.E16
      GO TO 1
   12 TM=T1+(AC/P5M)**.3333333
      GO TO 1
      END
```

APPENDIX VIII
FORTRAN LISTING OF PROGRAM TO COMPARE SOLUTIONS OF RADIATING OBJECT

```
C COMPARISON OF ANALYTICAL AND NUMERICAL SOLUTIONS OF ISOTHERMAL
C RADIATING OBJECT WITH HEAT SOURCE  D.C. TODD  12-16-68  AEA00071
      REAL P(5),TI*8,CONST*8
    1 READ(5,1001,END=900)N,TO,TI,TOD,AC
      IF(TI.NE.0.)GO TO 2
      A=0.
      C=TOD/TO**4
      D=1./(3.*C)
      CONST=1.DO/TO**3
      GO TO 3
    2 A=TOD/(1.-(TO/TI)**4)
      C=-A/TI**4
      D=TI/(4.*A)
      CONST=DLOG((TO+TI)/DABS(TO-TI))+2.*DATAN(TO/TI)
    3 S1=0.
      SM=0.
      P(1)=TO
      DT=(TI-TO)/(N+1)
      T=TO
      WRITE(6,1000)
      WRITE(6,1005)
      WRITE(6,1003)TO,TI,TOD,AC,A,C,D,CONST
      WRITE(6,1006)
      DO 6 I=1,N
      T=T+DT
      IF(TI.NE.0.)GO TO 4
      S=D*(CONST-1.00/T**3)
      GO TO 5
    4 S=D*(DLOG((T+TI)/DABS(T-TI))+2.*DATAN(T/TI)-CONST)
    5 CALL SOLU(1,1,AC,S1,S,SM,TEMP,P,A,0.,C)
      ERR=T-TEMP
    6 WRITE(6,1003)S,T,TEMP,ERR
      GO TO 1
  900 WRITE(6,1004)
      STOP
 1000 FORMAT('1D.C. TODD  AEA00071')
 1001 FORMAT(I2,E10.0,5E12.0)
 1003 FORMAT(1P11E12.4)
 1004 FORMAT('0THE END'/1H1)
 1005 FORMAT(1H04X2HT010X2HTI10X3HTOD9X2HAC11X1HA11X1HC11X1HD9X5HCONST/)
 1006 FORMAT(1H05X1HS11X1HT9X4HTEMP9X3HERR/)
      END
```

APPENDIX IX

FORTRAN LISTING OF PROGRAM TO OBTAIN SOLUTION TO CONDUCTING RADIATING SPHERICAL SHELL

```
C CONOUCTING RAOIATING SPHERICAL SHELL  O.C. TODO  12-16-6B  AEA00072
      REAL P(60,B),O(5,60),A(6D),B(6D,6D),C(60,60),TEMP(60),NR,
     * PI/3.141593/,SIG/5.6697E-B/,TURE(6D)
      EOUIVALENCE (P(1),O(1)),(P(1,7),TEMP(1)),(P(1,B),A(1))
    1 READ(5,1001,ENO=90D)NH,E,NR
      WRITE(6,1000)
      WRITE(6,1005)E,NR
      N=2*NH
      DX=2./N
      DS=4.*PI*DX
      DV=2.*PI*SIG*DX
      TK=1./NR
      EP=.5*(E+1.)
      AO=B.*PI*SIG*DX
      X=-1.-.5*DX
      DD 2 I=1,N
      X=X+OX
      P(1,1)=1.
      P(I,2)=DV
      P(I,3)=TK
      P(I,4)=TK
      P(I,5)=EP
      P(I,6)=EP
      P(I,7)=DS
    2 P(I,8)=AO*X
      DO 3 I=1,NH
    3 P(I,B)=0.
      G=.25*E*OX/EP**2
      DD 4 I=1,N
      OO 4 J=1,N
      B(I,J)=0.
    4 C(I,J)=G
      X=-1.
      DO 5 J=2,N
      X=X+DX
      I=J-1
      ST=(1.-X*X)**.5
      B(I,J)=2.*PI*SIG*ST
    5 B(J,I)=DX/ST
      CALL COEF(N,60,SIG,P,B,C)
      T1=0.
      T=1.E-10
      TM=0.
      DO 6 I=1,N
      TURE(I)=1.
    6 O(1,I)=1.
      WRITE(6,1006)
      LAST=0
      IPRT=20
      DO 9 IT=1,1DO
      CALL SOLU(N,60,.2,T1,T,TM,TEMP,O,A,B,C)
      DMAX=0.


      DO 7 I=1,N
      OIFF=TEMP(I)-TURE(I)
      IF(ABS(DIFF).GT.ABS(OMAX))OMAX=DIFF
    7 TURE(I)=TEMP(I)
      WRITE(6,1002)IT,T,OMAX
      IF(T.LT.1.E15)GO TO B
      IPRT=IT
      LAST=1
    8 IF(IPRT.NE.IT)GO TO 9
      IPRT=IPRT+20
      WRITE(6,1DD3)(TEMP(I),I=1,N)
      IF(LAST.NE.O)GO TO 10
    9 T=1.D01*TM
   1D GO TO 1
  90D WRITE(6,1004)
      STOP
 10D0 FORMAT('1D.C. TODD  AEA00D71')
 10D1 FORMAT(I2,E1D.D,5E12.D)
 1002 FORMAT(I18,6X1P2E12.4)
 1003 FORMAT(1P11E12.4)
 1004 FORMAT('OTHE ENO'/1H1)
 1005 FORMAT('OE ='1PE12.4,6X'NR ='E12.4)
 1D06 FORMAT(1H )
      ENO
```

APPENDIX X
FORTRAN LISTING OF PROGRAM TO CALCULATE CLAUSING FACTORS

```
C CALCULATION OF CLAUSING FACTORS  D.C. TODD  1-3-69
       REAL F(60,60),A(2),G(60,60),R(60)
      WRITE(6,1000)
    1 READ(5,1001,END=900)M,H
      CALL TUBE(M,1.,H,A,F)
      N=M+2
      DO 2 I=2,N
    2 R(I)=1.
      R(1)=0.
      R(N)=0.
      CALL GINV(N,60,R,F,G,DET)
      IF(DET.EQ.0.)G(1,N)=2.
      WRITE(6,1003)M,H,G(1,N)
      GO TO 1
  900 WRITE(6,1004)
      STOP
 1000 FORMAT('1AEA00075  D.C. TODD'/)
 1001 FORMAT(I2,E22.0)
 1003 FORMAT(I6,6X,1P2E12.4)
 1004 FORMAT('0THE END'/1H1)
      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)*
Arnold Engineering Development Center
ARO, Inc., Operating Contractor
Arnold Air Force Station, Tennessee 37389

2a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

2b. GROUP
N/A

3. REPORT TITLE
METHOD FOR SOLVING TRANSIENT HEAT-TRANSFER PROBLEMS INCLUDING COMPUTER PROGRAMS AND AN ANALYSIS OF FORM-SURFACE FACTORS

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*
Final Report October 1964 to December 1968

5. AUTHOR(S) *(First name, middle initial, last name)*
Donald C. Todd, ARO, Inc.

6. REPORT DATE
August 1969

7a. TOTAL NO. OF PAGES
53

7b. NO. OF REFS
6

8a. CONTRACT OR GRANT NO.
F40600-69-C-0001

b.
Program Element 65401F

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)
AEDC-TR-69-121

9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*
N/A

10. DISTRIBUTION STATEMENT
This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES
Available in DDC.

12. SPONSORING MILITARY ACTIVITY
Arnold Engineering Development Center, Air Force Systems Command, Arnold AFS, Tennessee 37389

13. ABSTRACT

A method for attacking a wide variety of commonly occurring heat-transfer problems is presented. The method stems from a mathematical model obtained by dividing a system into pieces. The differential equation of the temperature of each piece with respect to time is derived considering heat sources and heat transfer by conduction and radiation. The resulting system of first-order ordinary differential equations is put in a form convenient for programming a numerical solution by defining coefficients in terms of the data of the system. Form-surface factors are a generalization of form factors to include multiple reflections. The factor is defined and discussed. A formula is derived for the factor in terms of the form factors and the reflectivities. The form-surface factor relations for the conservation of energy and the reciprocity law are derived. The conditions necessary and sufficient for its existence are derived. Methods of calculation are discussed. A FORTRAN subroutine is presented that calculates the form-surface factors from the form factors and the reflectivities.

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
heat transfer						
form-surface factors						
computer programs						
Euclidian geometry						
surface temperatures						