N70-37089

NASA TECHNICAL MEMORANDUM

NASA TM X-2058

NASA TM X-2058

A GENERAL TRANSIENT HEAT-TRANSFER COMPUTER PROGRAM FOR THERMALLY THICK WALLS

by L. Bernard Garrett and Joan I. Pitts

Langley Research Center
Hampton, Va. 23365

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • AUGUST 1970

1. Report No. NASA TM X-2058	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle A GENERAL TRANSIENT HEAT-TRANSFER COMPUTER PROGRAM FOR THERMALLY THICK WALLS		5. Report Date August 1970
		6. Performing Organization Code
7. Author(s) L. Bernard Garrett and Joan I. Pitts		8. Performing Organization Report No. L-6850
9. Performing Organization Name and Address NASA Langley Research Center Hampton, Va. 23365		10. Work Unit No. 124-07-18-04
		11. Contract or Grant No.
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546		13. Type of Report and Period Covered Technical Memorandum
		14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

This program is a general heat-transfer program which employs a finite-difference method for the solution of temperature histories of one-dimensional, two-dimensional, or spherical systems. Options are available for heat input given in tabular form, computed from a trajectory, or computed from a temperature history given for a specific location. The types of heat exchange are: (1) conduction; (2) convection – with (a) given heat input, (b) heating due to skin friction with Van Driest equations, (c) stagnation heating with Sibulkin, Detra-Kemp-Riddell, and Cohen equations; (3) radiation-out; (4) air-conduction; and (5) joint conduction. The system configuration is specified by an arbitrary number of discrete elements and their interrelationships.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Heat-transfer computer program Conduction heat transfer Aerodynamic heating computer program Transient heat flow	Unclassified – Unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	152	$3.00

*For sale by the Clearinghouse for Federal Scientific and Technical Information
Springfield, Virginia 22151

CONTENTS

A GENERAL TRANSIENT HEAT-TRANSFER COMPUTER PROGRAM FOR THERMALLY THICK WALLS

By L. Bernard Garrett and Joan I. Pitts
Langley Research Center

SUMMARY

This program is a general heat-transfer program which employs a finite-difference method for the solution of temperature histories of one-dimensional, two-dimensional, or spherical systems. Options are available for heat input given in tabular form, computed from a trajectory, or computed from a temperature history given for a specific location. The types of heat exchange are: (1) conduction; (2) convection – with (a) given heat input, (b) heating due to skin friction with Van Driest equations, (c) stagnation heating with Sibulkin, Detra-Kemp-Riddell, and Cohen equations; (3) radiation-out; (4) air-conduction; and (5) joint conduction. The system configuration is specified by an arbitrary number of discrete elements and their interrelationships.

INTRODUCTION

This report describes a general transient conduction heat-transfer computer program (program D1244) developed at the Langley Research Center and provides the user with the information necessary to use the program. The program is particularly suited for the thermal design of launch vehicles and spacecraft that are subject to aerodynamic heating, and has been used extensively at the Langley Research Center for this purpose.

A finite-difference method is employed for the solution of the temperature histories (ref. 1) of one-dimensional, two-dimensional, or axisymmetric configuration systems. The thermal model is divided into a number of small volumes (blocks). The system can include 100 discrete blocks and 10 different materials with temperature-dependent properties.

Several options are available to consider the heating rates: (1) the heating input may be given in tabular form as a function of time, (2) the heating may be computed by using free-stream environmental conditions or from a velocity-altitude trajectory for entry velocities up to 41 000 ft/sec (12 497 m/sec), which utilizes the 1962 U.S. Standard Atmosphere tables (ref. 2), or (3) the heating rate may be computed from a temperature history given for a specified location within the material.

The types of heat exchange considered are:

(1) Convection
   (a) Given heat input
   (b) Flat-plate or cone heating with Van Driest (refs. 3 and 4) equations
   (c) Stagnation heating with Sibulkin (ref. 5), Detra-Kemp-Riddell (ref. 6), and Cohen (ref. 7) equations

(2) Conduction

(3) Air-conduction (across an air gap)

(4) Internal radiation (across an air gap)

(5) Joint conduction

(6) Radiation-out
   (a) Nonlinearized
   (b) Linearized

This program is written in FORTRAN IV language for the Control Data 6000 Series computer and operates with the scope 3.0 operating system. The equations are programed in U.S. Customary units; some SI units are included in the text for the convenience of the reader.

SYMBOLS

a	speed of sound, ft/sec (m/s)
A	area, ft^2 (m^2)
b	distance across air gap, ft (m)
c_f	local skin-friction coefficient
$c_{f,o}$	reference skin-friction coefficient evaluated at $N_{Re} = 10^6$
c_p	specific heat at constant pressure, Btu/lb-^{o}R (J/kg-^{o}K)
D	diameter, ft (m)
d	width or length of blocks, ft (m)
$G_1(i_d)$	high-enthalpy correlation function (see eqs. (41))
g	acceleration due to gravity, 32.1741 ft/sec^2 (9.80665 m/sec^2)

h	heat-transfer coefficient (Van Driest and Sibulkin equations, Btu/ft^2-sec-$^\circ$R (W/m^2-$^\circ$K); Detra-Kemp-Riddell- and Cohen equations, lbm/ft^2-sec (kg/m^2-sec))
h_j	joint heat-transfer coefficient, Btu/ft^2-sec-$^\circ$R (W/m^2-$^\circ$K)
i	enthalpy, Btu/lb (J/kg)
i_d	dimensionless static enthalpy (see eq. (40))
i_r	recovery enthalpy, Btu/lb (J/kg)
i_w	wall enthalpy, Btu/lb (J/kg)
k	thermal conductivity, Btu/ft-sec-$^\circ$R (W/m-$^\circ$K)
k_{air}	thermal conductivity of air, Btu/ft-sec-$^\circ$R (W/m-$^\circ$K)
k_m	conductivity of material, Btu/ft-sec-$^\circ$R (W/m-$^\circ$K)
L	characteristic length used in Reynolds number, ft (m)
l_i	block length, ft (m)
M	Mach number
N_{Nu}	Nusselt number based on diameter, hD/k
N_{Pr}	Prandtl number, $c_p\mu/k$
N_{Re}	Reynolds number, $\rho VL/\mu$
N_{Re}/ft	Reynolds number per foot (per 0.3048 m), $\rho V/\mu$
N_{St}	Stanton number, $h/(c_p\rho V)$
p	pressure, lb/ft^2 (N/m^2)
$\dot{Q}$	heating rate, Btu/sec (W)

$\dot{q}$	heating rate per unit area, Btu/ft^2-sec (W/m^2)
R	gas constant for air, 0.0686 Btu/lbm-$^\circ$R (287 J/kg-$^\circ$K)
r_n	nose radius, ft (m)
T	temperature, $^\circ$R ($^\circ$K)
t	time, sec
Δt	delta time or computing interval
V	velocity, ft/sec (m/s)
v	volume, ft^3 (m^3)
w_i	block width, ft (m)
Z	compressibility factor
α,δ,ξ,ψ	quantities defined by equations (22)
β	tangential velocity gradient
γ	ratio of specific heats
η	ratio of local to stagnation convective heating rate
θ	temperature ratio in viscosity relation for air $\mu = \dfrac{\left(2.27 \times 10^{-8}\right)T^{1/2}}{1 + \theta}$ used in flat-plate or cone heating section, $\theta = 198\ ^\circ\mathrm{R}/T$ when μ has units of lb-sec/ft^2
ϵ	emissivity
μ	viscosity coefficient, lb-sec/ft^2 (N-sec/m^2)
μ_o	reference value of viscosity defined by equations (38), lb-sec/ft^2 (N-sec/m^2)
ρ	density, slugs/ft^3 (kg/m^3)

ρ_m	material density, lb/ft^3 (kg/m^3)
σ	Stefan-Boltzmann constant, $Btu/ft^2\text{-}sec\text{-}{}^{o}R^4$ $(W/m^2\text{-}{}^{o}K^4)$
Λ	sweep angle measured from normal to free-stream velocity vector

Subscripts:

cond	conduction
conv	convection
e	local flow conditions at edge of boundary layer
i	variable integer which refers to block upon which heat balance is being performed
j	variable integer which refers to any other block that affects the heat balance for block i
o	reference condition
Perfect	properties based on perfect gas relations
r	adiabatic or recovery conditions
rad	radiation
s	stagnation conditions behind a normal shock
w	wall conditions at beginning of computing interval
∞	free-stream conditions upstream of shock

A prime over a symbol indicates the value at the beginning of the time interval.

MATHEMATICAL AND NUMERICAL PROBLEM DESCRIPTION

The prediction of the thermal response of materials to the specified environmental conditions requires the solution of the governing heat-transfer equations subject to the imposed boundary conditions. The basic procedure employed in this program requires that the engineer divide the specific configuration (body) of interest into a system of small volumes (blocks) and describe by the inputs, the size, orientation, material composition, and modes of heat transfer for each individual block and the interrelationships between the blocks. With these inputs the machine selects the appropriate finite-difference heat-balance equations for each block and solves the resulting matrix for the temperatures of the blocks as a function of time. In this section are described the thermal-balance equations for the individual blocks, the convective heat-transfer options, and the finite-difference algorithm for the solution of the governing equations. A specific example of the procedure for the finite-difference solution is included.

General Heat-Balance Equation

The following forms of the general heat-balance equation are given, each term being defined for use:

$$\dot{Q}_{conv} - \dot{Q}_{cond} - \dot{Q}_{air\text{-}cond} - \dot{Q}_{rad\text{-}in} - \dot{Q}_{joint} - \dot{Q}_{rad\text{-}out} - \dot{Q}_{stored} = 0$$

For convection:

where

$$\left(\dot{Q}_{conv}\right)_i = hA_i\left(T_r - T_w\right) \qquad \text{(Van Driest (refs. 3 and 4) and Sibulkin (ref. 5))} \tag{1}$$

or

$$\left(\dot{Q}_{conv}\right)_i = hA_i\left(i_s - i_w\right) \qquad \text{(Detra-Kemp-Riddell (ref. 6) and Cohen (ref. 7))} \tag{2}$$

or

$$\left(\dot{Q}_{conv}\right)_i = \dot{q}_i A_i \qquad \text{(Heating rates given)} \tag{3}$$

or

$$\left(\dot{Q}_{conv}\right)_i = h_i A_i\left(T_r - T_w\right) \qquad \text{(Heat-transfer coefficient and recovery temperature input)} \tag{4}$$

For conduction:

$$\left(\dot{Q}_{cond}\right)_i = \left(\Delta\dot{Q}_{cond}\right)_{i,k} + \left(\Delta\dot{Q}_{cond}\right)_{i,l} + \ldots + \left(\Delta\dot{Q}_{cond}\right)_{i,j} \tag{5}$$

where k, l, and j represent the designated numbers of the blocks which touch block i, and each term on the right-hand side of equation (5) takes the form of the following example:

$$\left(\Delta\dot{Q}_{cond}\right)_{i,j} = \frac{2A_{i,j}\left(T_i - T_j\right)}{\dfrac{d_i}{k_{m,i}} + \dfrac{d_j}{k_{m,j}}} \tag{6}$$

The $\left(\Delta\dot{Q}_{cond}\right)_{i,j}$ is the net rate of conductive heat exchange between block i and block j at a given time, $A_{i,j}$ is the area of block i which touches block j, and d_i and d_j are the dimensions (either length or width) of blocks i and j, respectively, in the direction of the conductive heating.

For air-conduction:

$$\left(\dot{Q}_{air\text{-}cond}\right)_i = \frac{k_{air}A_{i,n}\left(T_i - T_n\right)}{b} \tag{7}$$

where k_{air} is the conductivity of air and n denotes the block number of the block opposite the air gap.

For radiation-in (infinite parallel plates) (ref. 8):

$$\left(\dot{Q}_{rad\text{-}in}\right)_i = \frac{\sigma A_{i,n}\left(T_i'^4 - T_n'^4\right)}{\dfrac{1}{\epsilon_i} + \dfrac{1}{\epsilon_n} - 1.0} \tag{8}$$

where T' is the temperature at the beginning of the computing interval and n denotes the block number of the block opposite the air gap. This expression is strictly correct only for infinite parallel plates. It should be noted that $\dot{Q}_{air\text{-}cond}$ and $\dot{Q}_{rad\text{-}in}$ are always used together.

For the joint (interface between materials) (ref. 1):

$$\left(\dot{Q}_{joint}\right)_i = \frac{2A_{i,n}\left(T_i - T_n\right)}{\dfrac{2}{h_j} + \dfrac{d_i}{k_{m,i}} + \dfrac{d_n}{k_{m,n}}} \tag{9}$$

where n denotes the block number of the block opposite the joint, and h_j is the joint coefficient.

For radiation-out:

$$\left(\dot{Q}_{rad\text{-}out}\right)_i = \sigma\epsilon_i A_i T_i'^4 \tag{10}$$

It should be noted that the radiation-out term generally applies for external surfaces radiating to space whereas the radiation-in term represents a radiation exchange between two surfaces at different temperatures.

The radiation-out term may also be used in the following linear form (see appendix A):

$$\left(\dot{Q}_{rad\text{-}out}\right)_i = \sigma\epsilon_i A_i\left(2T_i'^3 T_i - T_i'^4\right) \tag{11}$$

For heat stored:

$$\left(\dot{Q}_{stored}\right)_i = \frac{c_{p,i}\rho_{m,i}v_i\left(T_i - T_i'\right)}{\Delta t} \tag{12}$$

where $c_{p,i}$ and $\rho_{m,i}$ are the specific heat and density, respectively, of the material of block i, and Δt is the delta time or computing interval.

Convective Heat-Transfer Equations

To compute the free-stream conditions used in the convective heating options of the program, a trajectory consisting of altitude and velocity tables as functions of time are given. By using the altitude table, the values of ρ_∞, p_∞, T_∞, and a_∞ are determined from the 1962 U.S. Standard Atmosphere (ref. 2). The viscosity μ_∞ is computed from the following relation (ref. 9):

$$\mu_\infty = \frac{\left(2.27 \times 10^{-8}\right)T_\infty^{1/2}}{1 + \theta}$$

where $\theta = \dfrac{198.6\ ^{\circ}\mathrm{R}}{T_\infty}$ and μ_∞ has the units lb-sec/ft^2.

The convective heat-transfer equations for the flow over a flat plate or cone and for the stagnation region of blunt bodies are given.

Flat plate or cone.- The convective heating rates to a flat plate or cone are computed from the Van Driest equations (refs. 3 and 4) for either the laminar or turbulent flow of air as follows:

Laminar flow: $N_{Re,e}$ < input transition Reynolds number

$$N_{St} = K\left[\frac{f_1(M_e) + f_2(M_e)\frac{T_w}{T_e}}{\sqrt{N_{Re,e}}}\right] \tag{13}$$

where

$$\left.\begin{aligned} f_1(M_e) &= 0.416594 - 0.00246733M_e - 8.17489\times 10^{-4}\,{M_e}^2 + 2.734033\times 10^{-5}\,{M_e}^3 \\ f_2(M_e) &= -0.0134671 + 2.635807\times 10^{-4}\,M_e + 5.818944\times 10^{-5}\,{M_e}^2 - 2.173257\times 10^{-6}\,{M_e}^3 \end{aligned}\right\} \tag{14}$$

$$\left.\begin{aligned} K &= 1 \quad \text{(Flat plate)} \\ K &= \sqrt{3} \qquad \text{(Cone)} \end{aligned}\right\} \tag{15}$$

A Stanton number N_{St} is computed for each convective block, where the wall temperature T_w is taken to be the temperature of the individual block at the beginning of the computing interval T_i'. The subscript e refers to the local flow conditions at the edge of the boundary layer which are determined from input ratios of local to free-stream conditions.

The Stanton number expressions (eqs. (13) and (14)) were obtained from curve fits to the Van Driest heat-transfer results in air for the flat-plate laminar boundary layer (ref. 3) at $N_{Pr} = 0.75$, $\theta = \frac{198\ {}^{\circ}R}{T_e} = 0.505$. A comparison of the original results and the correlation is shown in figure 1. The correlation is accurate to within 3 percent over the range of $M_e = 0$ to 20 and $T_w/T_e = 0.25$ to $T_w/T_e = 6.0$.

The heat-transfer coefficient is determined from the relation:

$$h = N_{St}\rho_e(c_p)_{Perfect}V_e g \tag{16}$$

For laminar flow the recovery temperature T_r (ref. 10) is

$$T_r = (N_{Pr})^{1/2}(T_s - T_e) + T_e \tag{17}$$

where the laminar recovery factor is approximated by ${N_{Pr}}^{1/2}$.

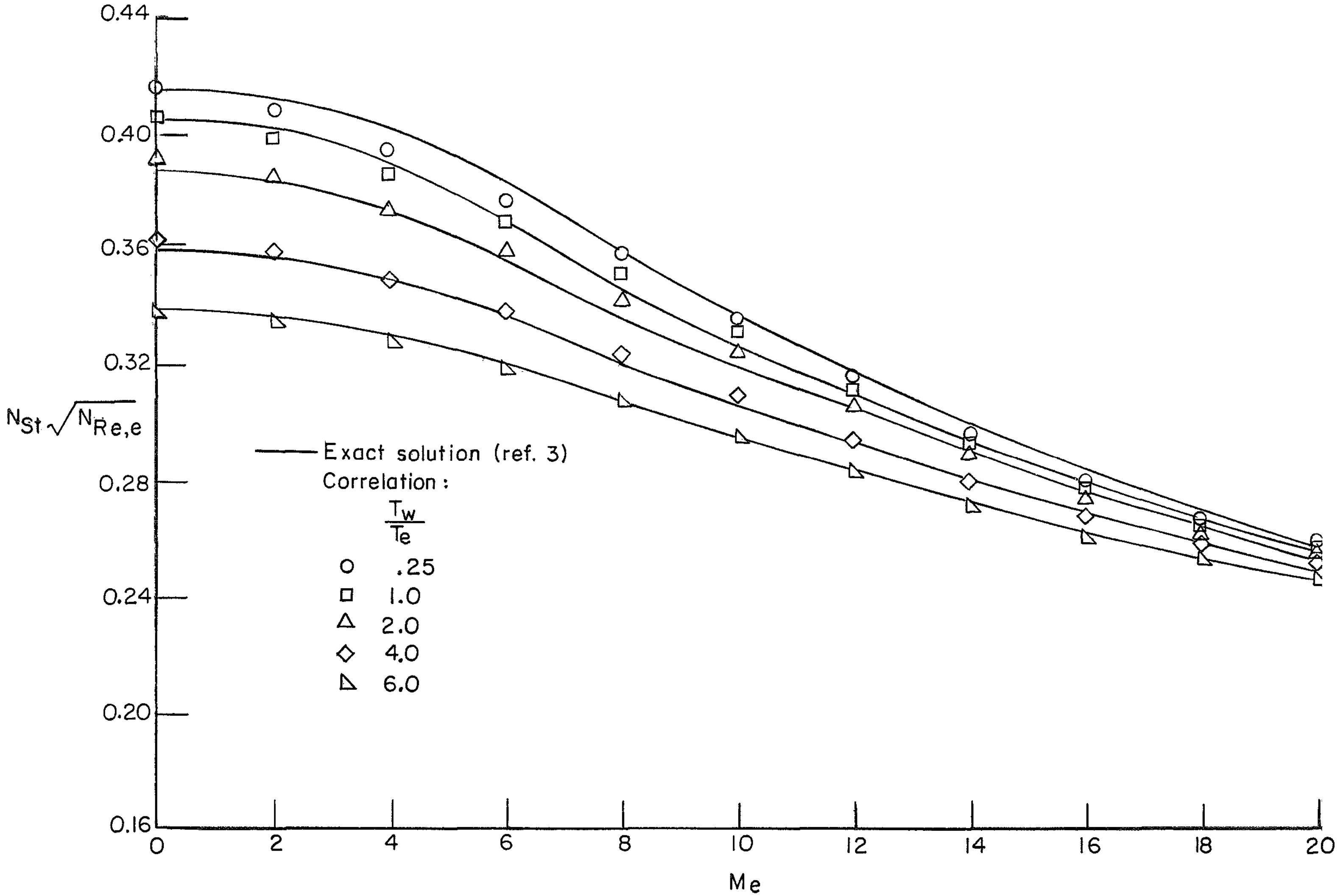


Figure 1.- Comparisons of exact and correlation equation results for the laminar heat-transfer coefficient along a flat plate.

The specific heat c_p is that of low-temperature air ($\gamma = 1.4$); that is,

$$\left(c_p\right)_{Perfect} = 0.24 \text{ Btu/lb-}^{\circ}\text{R} \quad (1004 \text{ J/kg-}^{\circ}\text{K}) \tag{18}$$

and the stagnation temperature for $\gamma = 1.4$ (ref. 10) is

$$T_s = T_\infty\left(1.0 + 0.2M_\infty^2\right) \tag{19}$$

Turbulent flow: $N_{Re,e} \geqq$ Input transition Reynolds number.

The Stanton number is related to the local skin friction by a modified form of Reynolds analogy which for turbulent flow (ref. 4) is

$$N_{St} = 0.6c_f \tag{20}$$

The Van Driest relation for the local skin-friction coefficient c_f (ref. 4) is expressed implicitly as

$$\frac{0.242}{c_f^{1/2}\left(\frac{\gamma - 1}{2}M_e^2\right)^{1/2}}\left(\sin^{-1} \alpha + \sin^{-1} \delta\right) = 0.41 + \log_{10} N_{Re,e}c_f - 0.76 \log_{10} \frac{T_w}{T_e} \tag{21}$$

where

$$\left.\begin{aligned}
\alpha &= \frac{2\xi^2 - \psi}{\left(\psi^2 + 4\xi^2\right)^{1/2}} \\
\delta &= \frac{\psi}{\left(\psi^2 + 4\xi^2\right)^{1/2}} \\
\xi^2 &= \frac{\frac{\gamma - 1}{2}M_e^2}{T_w/T_e} \\
\psi &= \frac{1 + \frac{\gamma - 1}{2}M_e^2}{T_w/T_e} - 1
\end{aligned}\right\} \tag{22}$$

The iteration of c_f is avoided by utilizing the curve fits for c_f developed by Lee and Faget (ref. 11). The Lee and Faget correlation equation is

$$c_f = \frac{c_{f,o}}{\left(K' \frac{N_{Re,e}}{10^6}\right)^N} \tag{23}$$

The reference skin-friction coefficient $c_{f,o}$ is evaluated at $N_{Re,e} = 10^6$ and is obtained from table I as a function of M_e and T_w/T_e.

$$N = C_1 + C_2 M_e \tag{24}$$

where

$$\left.\begin{aligned} C_1 &= \frac{668.12 + \left(-1826.3756 + 9737.6 \frac{T_w}{T_e}\right)^{0.5}}{4868.8} \\ C_2 &= 0.0019 - 0.0001 \frac{T_w}{T_e} \end{aligned}\right\} \tag{25}$$

$$\left.\begin{aligned} K' &= 1.0 \quad \text{(Flat plate)} \\ K' &= 0.5 \quad \text{(Cone)} \end{aligned}\right\} \tag{26}$$

Comparisons of the flat-plate skin-friction results obtained from equation (21) and equation (23) are shown in figure 2 as functions of $N_{Re,e}$ for various values of T_w/T_e and M_e. Over the ranges of $N_{Re,e}$, M_e, and T_w/T_e shown in the figure, the correlation is accurate within 3 percent.

The recovery temperature for turbulent flow (ref. 10) is

$$T_r = N_{Pr}{}^{1/3}\left(T_s - T_e\right) + T_e \tag{27}$$

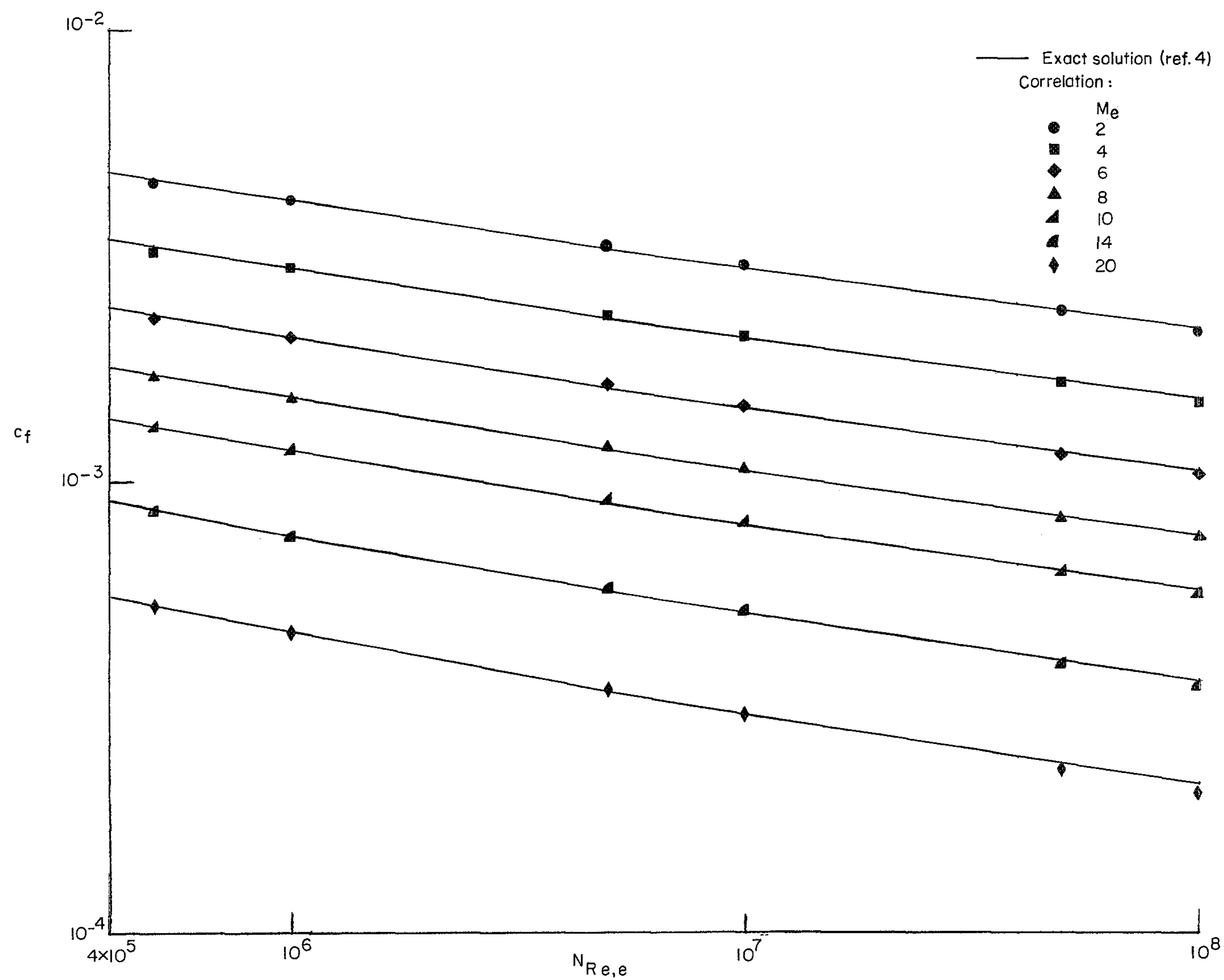
where the turbulent recovery factor is approximated by $N_{Pr}{}^{1/3}$. The heat-transfer coefficient, specific heat, and stagnation temperature expressions given in the laminar-flow section (eqs. (16), (18), and (19)) are the same for turbulent flow.

TABLE I.- REFERENCE SKIN-FRICTION COEFFICIENTS

$\left[N_{Re,e} = 10^6\right]$

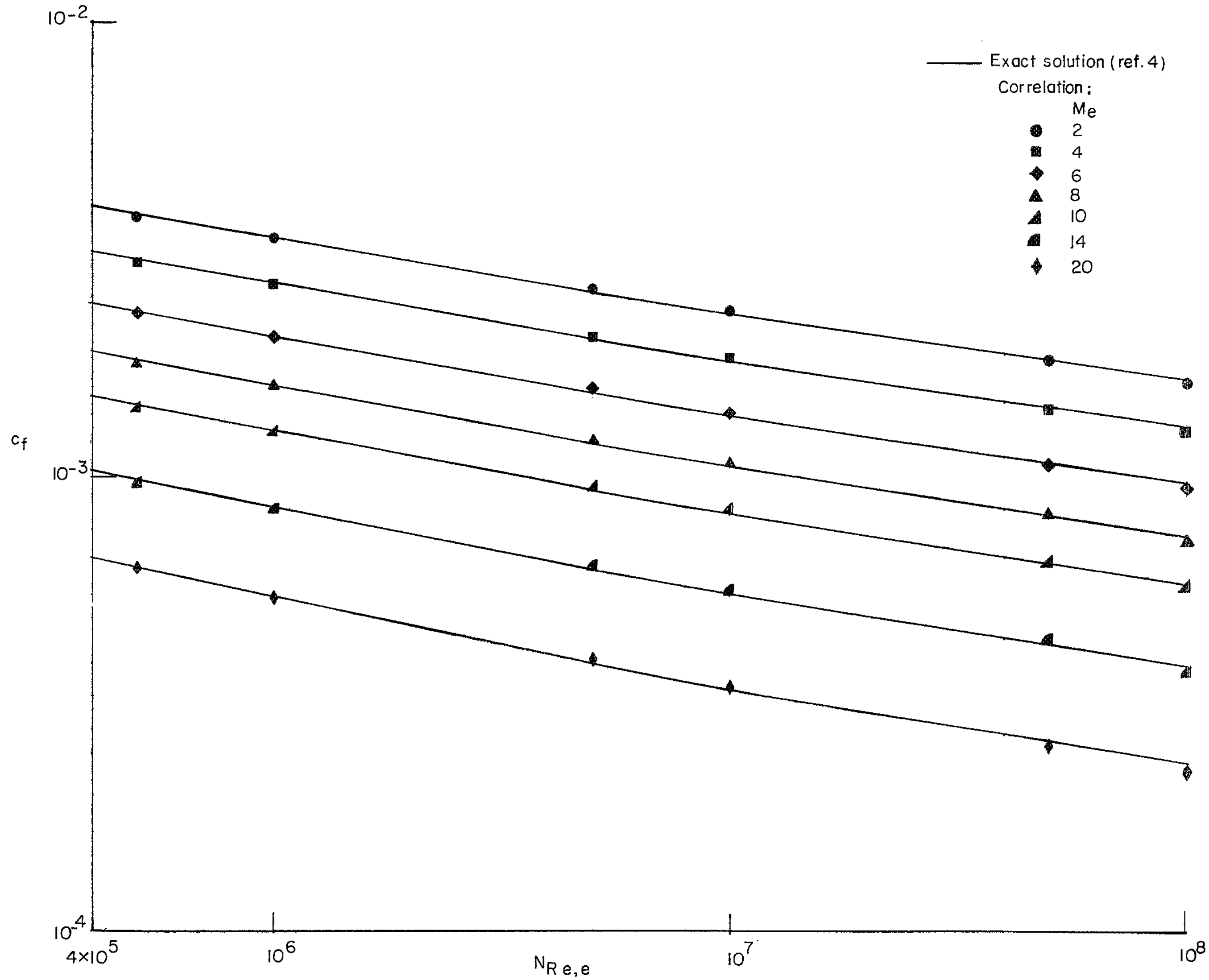
	Reference skin-friction coefficient $c_{f,o}$ for T_w/T_e of –						
M_e	0.2	0.6	1.0	2.0	3.0	4.0	6.0
0.5	0.005100	0.004170	0.003673	0.002990	0.002610	0.002350	0.002005
1.0	.004904	.004070	.003603	.002948	.002578	.002328	.002000
2.0	.004290	.003709	.003345	.002800	.002475	.002251	.001950
3.0	.003596	.003258	.003005	.002593	.002328	.002137	.001875
4.0	.002978	.002812	.002652	.002357	.002153	.002000	.001779
5.0	.002475	.002417	.002325	.002126	.001974	.001854	.001675
6.0	.002078	.002084	.002036	.001907	.001798	.001708	.001566
7.0	.001764	.001809	.001788	.001711	.001635	.001568	.001460
8.0	.001516	.001581	.001582	.001539	.001489	.001441	.001357
9.0	.001317	.001394	.001407	.001389	.001357	.001323	.001262
10.0	.001156	.001237	.001258	.001259	.001241	.001218	.001173
12.0	.000913	.000995	.001025	.001047	.001046	.001039	.001017
13.0	.0008208	.0009016	.0009322	.0009598	.0009652	.0009630	.0009500
14.0	.0007425	.0008209	.0008523	.0008838	.0008934	.0008950	.0008887
15.0	.0006755	.0007512	.0007828	.0008168	.0008295	.0008341	.0008331
16.0	.0006177	.0006905	.0007219	.0007576	.0007724	.0007793	.0007825
18.0	.0005236	.0005905	.0006208	.0006578	.0006755	.0006853	.0006944
20.0	.0004507	.0005121	.0005409	.0005777	.0005967	.0006082	.0006210

Some general comments are in order concerning the Van Driest equations as programed. The final form of the programed convective heat-transfer equations are restricted to air $\left(\text{that is, } \gamma = 1.4, \ \mu = f(\theta), \text{ where } \theta = \frac{198\ {}^{\circ}R}{T_e} = 0.505, \text{ and } c_p = 0.24\ \text{Btu/lbm-}{}^{\circ}R\right)$. Further, the local flow conditions must be within or near the region of the correlations and tables where $M_e \leqq 20$ and $0.2 \leqq T_w/T_e \leqq 6.0$. For turbulent flow, the additional restriction is $10^5 \leqq N_{Re,e} \leqq 10^8$. Most of the flight aerodynamic heating cases will be within these local flow regions. However, the basic assumptions of Van Driest, namely, a perfect gas which has c_p = Constant, and $\gamma = 1.4$ introduce fundamental restrictions in the intermediate to high free-stream Mach number ranges (above about $M_\infty = 10$) where the real-gas effects become significant.



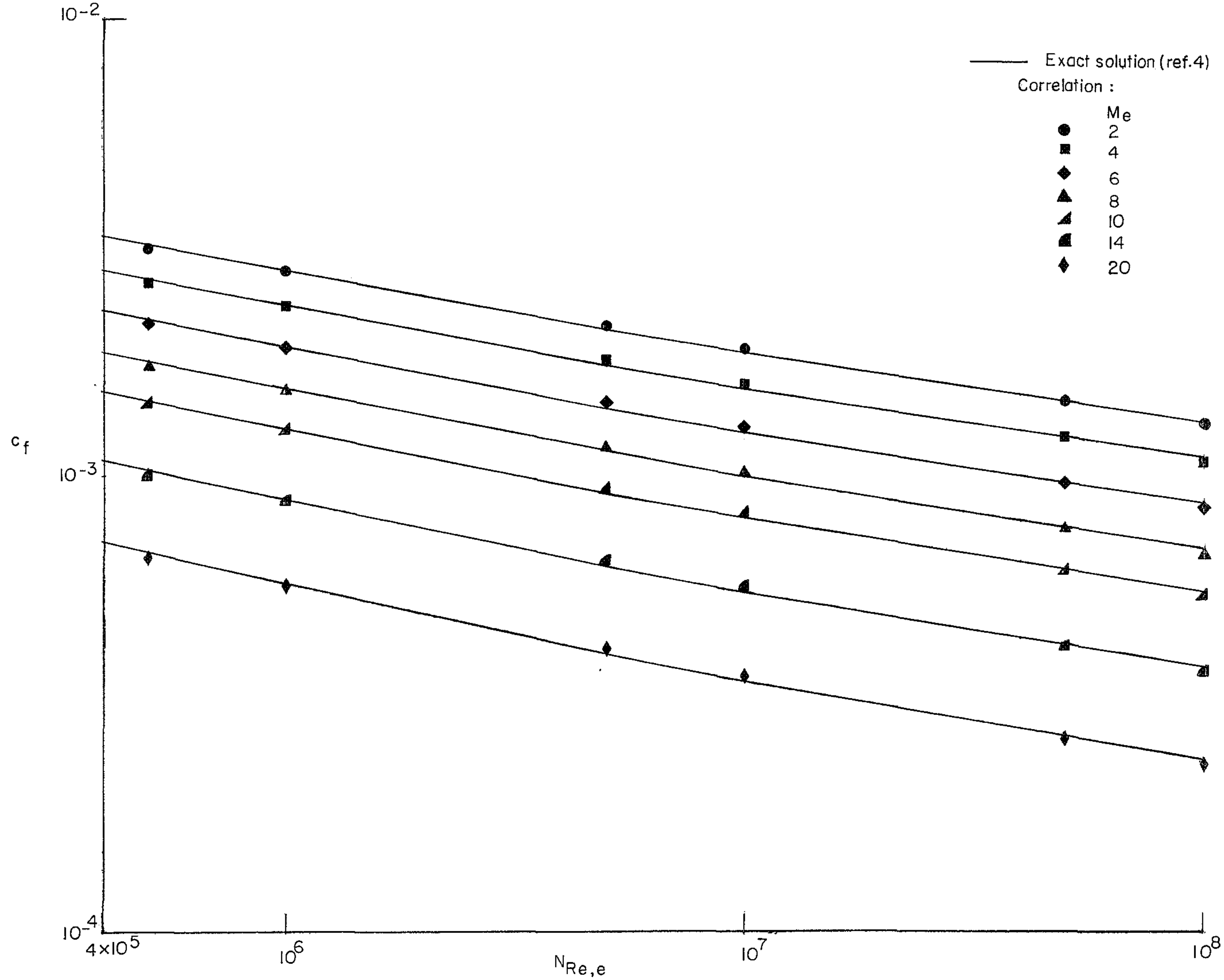
(a) $\frac{T_w}{T_e} = 0.25.$

Figure 2.- Comparisons of exact and correlation equation results for the turbulent skin-friction coefficient along a flat plate.



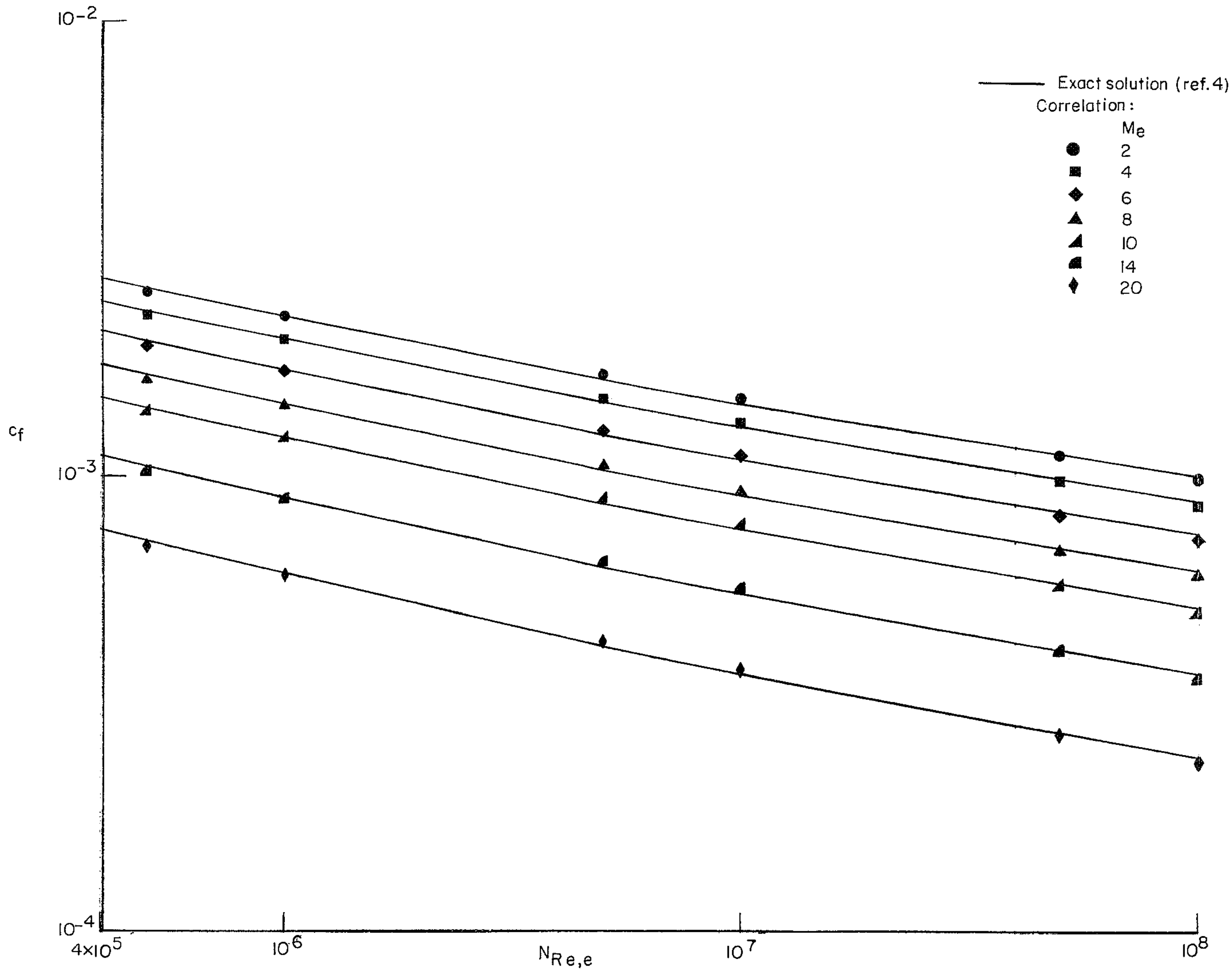
(b) $\frac{T_w}{T_e} = 1.0.$

Figure 2.- Continued.



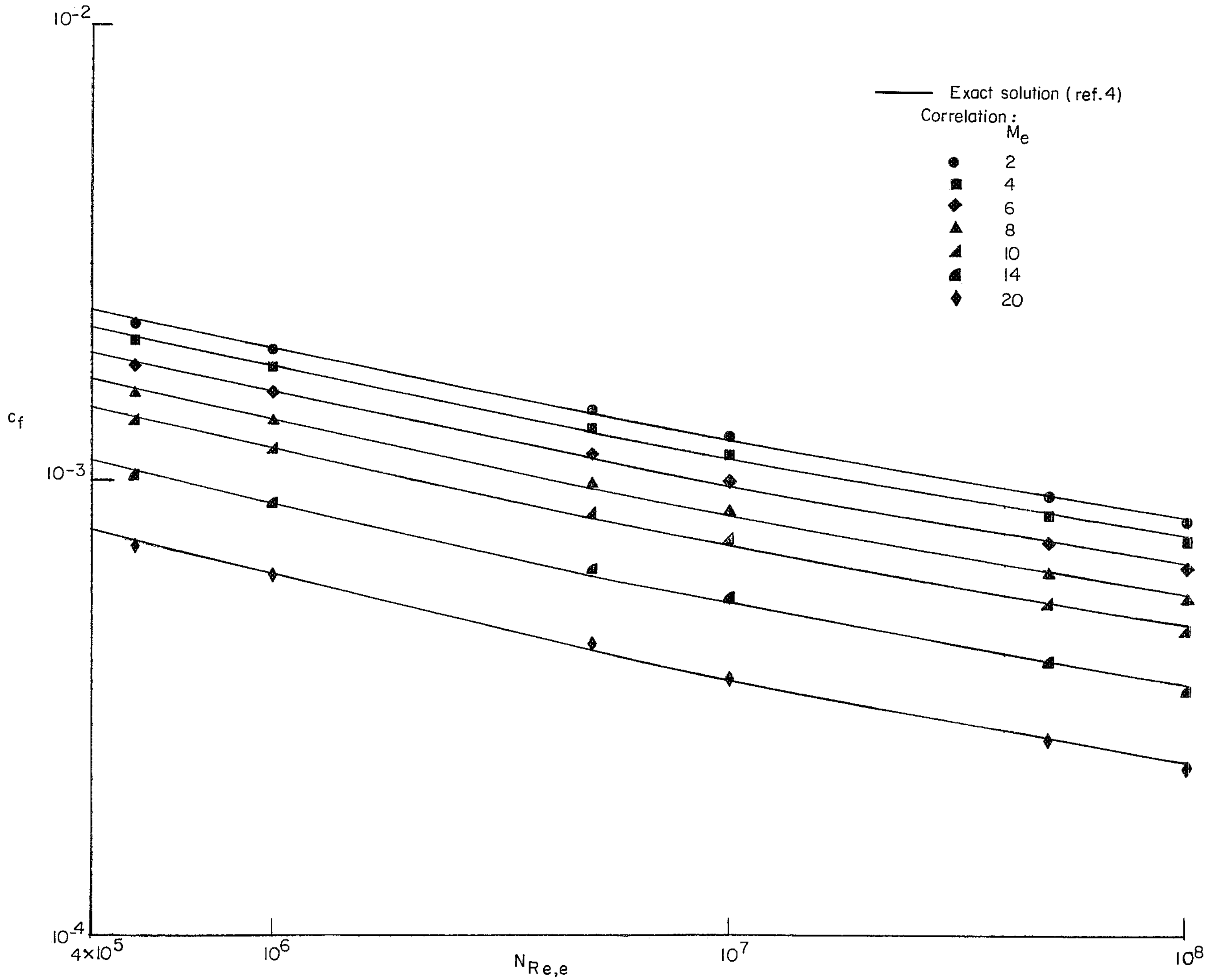
(c) $\frac{T_w}{T_e} = 2.0.$

Figure 2.- Continued.



(d) $\frac{T_w}{T_e} = 4.0.$

Figure 2.- Continued.



(e) $\frac{T_w}{T_e} = 6.0.$

Figure 2.- Concluded.

The reference enthalpy methods of Eckhert (ref. 12) for laminar and turbulent flow are among the more widely accepted methods for computing heating rates in high temperature air. Since it is not advantageous to incorporate heating theories based on real-gas properties within this program because of the amount of time involved to compute heating rates at intervals of 0.1 or 0.2 second, it is recommended that for the high-temperature real-gas environment problem, the heating rates be generated from the appropriate theory by separate computations and then these rates input to this program to determine the thermal response of the skin.

Spherical or blunt-nose bodies stagnation region.- For spherical bodies several options are available:

(1) Sibulkin (ref. 5) option where $V_\infty < 5000$ ft/sec (1524 m/sec) and

$$\dot{Q} = hA\left(T_s - T_w\right)$$

Sibulkin's expression for the stagnation-point heat-transfer coefficient is

$$h = 0.763\, N_{Pr}^{\;0.4} k_{air,s}\left(\frac{\beta\rho}{\mu}\right)_s^{1/2} \tag{28}$$

where β is the tangential velocity gradient at the stagnation point. This relation is correlated in terms of free-stream conditions to yield a more tractable expression for machine computations. The resulting expression for local heating, in terms of the stagnation-point heating equation developed in appendix B, becomes

$$h = \frac{Bg\rho_\infty V_\infty \left(c_p\right)_{Perfect}}{\left(N_{Re,\infty}/\text{ft}\ r_n\right)^{1/2}}\eta \tag{29}$$

$$B = 0.80 + 0.541M_\infty - 0.00574M_\infty^{\;2} \tag{30}$$

$$\left(c_p\right)_{Perfect} = 0.24\ \text{Btu/lb-}^{\circ}\text{R}$$

where h is in Btu/ft^2-sec, r_n is the nose radius in feet, and ρ_∞ is in slugs/ft^3. The ratio of the local heating to the stagnation heating is denoted by η. This relation is a required input for each convective element (block) when the stagnation-heating option is used. These ratios may be determined from Lees (ref. 13). The stagnation temperature is

$$T_s = T_\infty\left(1.0 + 0.2M_\infty^{\;2}\right) \tag{31}$$

(2) Detra-Kemp-Riddell (ref. 6) option where $5000 \leqq V_\infty \leqq 20\,000$ ft/sec (1525 m/sec $\leqq V_\infty \leqq 6100$ m/sec) and

$$\dot{Q} = hA(i_s - i_w)$$

$$h = \frac{(0.447 \times 10^{-8})\rho_\infty^{0.5} V_\infty^{3.15}}{r_n^{1/2}(i_s - 130)}\eta \tag{32}$$

$$i_s = \frac{V_\infty^2}{50\,000} + 0.24T_\infty \tag{33}$$

$$i_w = 0.0686\left(Z\frac{i}{RT}\right)_w T_w \tag{34}$$

where h is in lbm/ft^2-sec and i_s and i_w are in Btu/lbm. The temperature, T_w, is the temperature at the beginning of the computing interval and the term in parentheses $\left(Z\frac{i}{RT}\right)_w = f(p_s, T_w)$ is determined from Hansen's tables for air (ref. 14). The stagnation pressure p_s is determined from the AEDC tables (ref. 15); that is,

$$p_s = p_\infty\left(\frac{p_s}{p_\infty}\right)$$

where

$$\frac{p_s}{p_\infty} = f(\text{alt}, V_\infty)$$

(3) Cohen (ref. 7) option where $V_\infty \geqq 20\,000$ ft/sec (6100 m/sec) (The Cohen option may be used not only for velocities above 20 000 ft/sec (6100 m/sec) but also for the entire range if specified) and

$$\dot{Q} = hA(i_s - i_w)$$

$$h = 0.767g\left(N_{Pr,w}\right)_s^{-0.6}\left(\rho_e\mu_e\right)_s^{0.43}\left(\rho_w\mu_w\right)_s^{0.07}\left[\frac{2(p_s - p_\infty)}{\rho_{e,s}}\right]^{0.25}\frac{G_1(i_d)}{r_n^{1/2}}\eta \tag{35}$$

where the double subscripts e,s and w,s refer to the conditions at the edge of the boundary layer along the stagnation line and the condition at the wall at the stagnation point, respectively,

$$i_s = \frac{V_\infty^{\ 2}}{50\,000} + 0.24T_\infty \tag{36}$$

$$i_w = 0.0686\left(Z\,\frac{i}{RT}\right)_w T_w \tag{37}$$

where h is in lbm/ft^2-sec and i_s and i_w are in Btu/lbm. The quantities $\frac{T_{e,s}}{T_\infty}$, $\frac{p_s}{p_\infty}$, and $\frac{\rho_{e,s}}{\rho_\infty}$ are obtained from reference 15 as $f\left(\text{alt}, V_\infty\right)$ and

$$p_s = p_\infty \frac{p_s}{p_\infty}$$

$$T_{e,s} = T_\infty \frac{T_{e,s}}{T_\infty}$$

$$\rho_{e,s} = \rho_\infty \frac{\rho_{e,s}}{\rho_\infty}$$

$$\frac{\mu_{e,s}}{\mu_{o,s}} = f\left(T_{e,s}, p_s\right)$$

$$\frac{\mu_{w,s}}{\mu_{o,w}} = f\left(T_w, p_s\right)$$

The viscosity ratios are obtained from Hansen's tables (ref. 14) with linear interpolation on temperature and logarithmic on pressure. Then

$$\mu_{e,s} = \mu_{o,s} \frac{\mu_{e,s}}{\mu_{o,s}}$$

and

$$\mu_{w,s} = \mu_{o,w} \frac{\mu_{w,s}}{\mu_{o,w}}$$

where

and

$$\left.\begin{aligned}\mu_{o,s} &= \frac{\left(2.28 \times 10^{-8}\right)T_{e,s}^{3/2}}{T_{e,s} + 201.6}\ \text{lb-sec/ft}^2 \\ \mu_{o,w} &= \frac{\left(2.28 \times 10^{-8}\right)T_w^{3/2}}{T_w' + 201.6}\ \text{lb-sec/ft}^2\end{aligned}\right\} \tag{38}$$

$$\rho_{w,s} = \left(5.825 \times 10^{-4}\right)\frac{p_s}{Z_w T_w}\ \text{slugs/ft}^3 \tag{39}$$

where

$$\left.\begin{aligned}Z_w &= f\left(T_w,p_s\right) \\ \left(N_{Pr,w}\right)_s &= f\left(T_w,p_s\right)\end{aligned}\right\} \text{ from Hansen's tables (ref. 14)}$$

$$i_d = \frac{8465}{\frac{V_\infty^2}{50\,000}} \tag{40}$$

where

$$\left.\begin{aligned}G_1\left(i_d\right) &= 1.0 & &\left(i_d \geqq 0.5\right) \\ G_1\left(i_d\right) &= 1.0 + 0.075\left(\frac{1}{i_d} - 2\right)^2 & &\left(i_d < 0.5\right)\end{aligned}\right\} \tag{41}$$

(4) Cohen option over entire velocity range: Below a velocity of 5000 ft/sec (1525 m/sec), the following perfect gas relations apply for γ of 1.4 (ref. 9):

$$\frac{p_s}{p_\infty} = \left(\frac{6M_\infty^2}{5}\right)^{3.5}\left(\frac{6}{7M_\infty^2 - 1}\right)^{2.5} \tag{42}$$

$$\frac{T_{e,s}}{T_\infty} = 1 + 0.2M_\infty^2 \tag{43}$$

$$\frac{\rho_{e,s}}{\rho_\infty} = \frac{p_s}{p_\infty}\bigg/\frac{T_{e,s}}{T_\infty} \tag{44}$$

Comparisons of the heat-transfer coefficients predicted by the theories of Cohen, Sibulkin, and Detra, Kemp, and Riddell, are shown in figure 3 as a function of altitude for a range of velocities. It is noted that there is little difference between Cohen and Sibulkin theories at the lower velocities and between Cohen theory and the Detra, Kemp, and Riddell correlation of experimental data at the intermediate velocities.

Nonspherical blunt bodies will now be considered. The foregoing stagnation heating equations are applicable only for a sphere; however, the inputs can generally be manipulated to compute accurate heating rates to arbitrary axisymmetric bodies and to cylinders.

For instance, the proper stagnation heating rates to ellipsoids of body bluntness ratios greater than unity and to Apollo-type configurations can be obtained by replacing the body radius of curvature at the stagnation point with an "effective radius" input. Zoby and Sullivan (ref. 16) provided a method for obtaining the effective nose radius values for these configurations. However, if one is also interested in the heating-rate distributions around the blunt forebody away from the stagnation point, further flow-field details are required to generate the necessary inputs.

The stagnation heat flux to a cylinder can also be related to the stagnation heat flux on a sphere by changing the nose radius input. For instance, at low supersonic free-stream Mach numbers, the heating rate to the cylinder is approximately 0.92 times that of the sphere. (See ref. 13.) Since $\dot{q}_{conv} \propto r_n^{-1/2}$, the nose radius input to the program should be $(0.92)^{-2}$ or 1.18 times greater than the actual radius of the cylinder being studied. In the supersonic-hypersonic regimes, $M > 5$, the proper factor is more nearly

$$\dot{q}_{cyl} = \frac{0.57}{0.763}\dot{q}_{sphere} = 0.747\dot{q}_{sphere} \tag{45}$$

(See Cohen (ref. 7) and Truitt (ref. 10).) Thus the nose radius input should be 1.79 times greater than the actual radius of the cylinder. No provision is included in the program for a change in the nose radius within a given computer run. However, one can generally establish within the constraints of the problem which heating-rate factor applies. For instance, for low velocity flights if M_∞ is about 2 to 5, then the 1.18 factor should be applied, particularly if the problem is one of thermal design. Whereas for hypersonic entry speeds, the primary heat pulse generally occurs at the higher velocities and thus the 1.79 factor should be used.

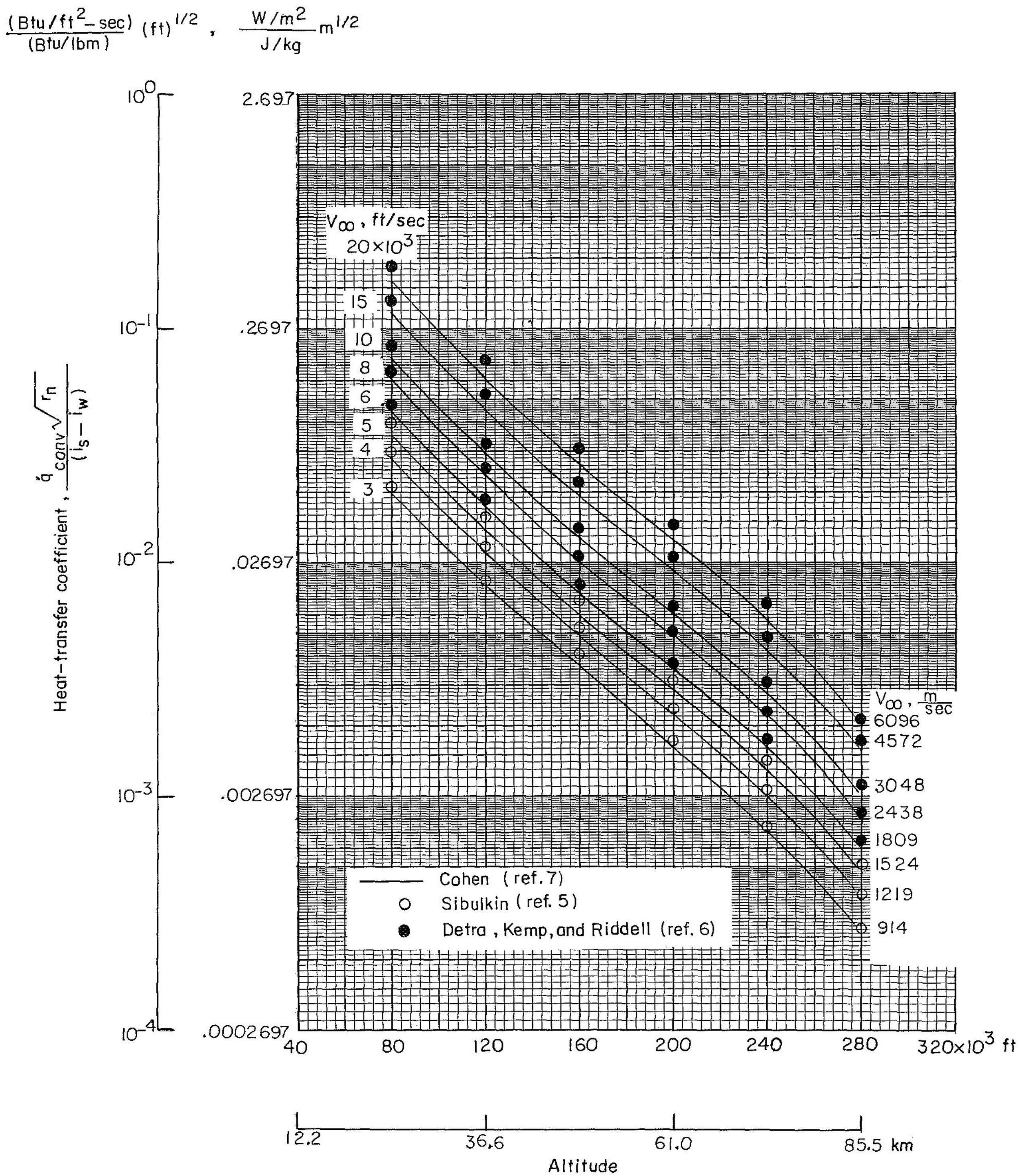


Figure 3.- Comparison of the stagnation-point heat-transfer coefficients from Cohen, Sibulkin, and Detra, Kemp, and Riddell. i_w = 200 Btu/lbm (465 kJ/kg).

For swept cylinders, it is recommended in reference 17 that the heating factor on the stagnation line be used to compute the appropriate radius input as follows:

$$\dot{q}_{cyl,\Lambda\neq 0} = \dot{q}_{cyl,\Lambda=0}(\cos \Lambda)^{1.5} \tag{46}$$

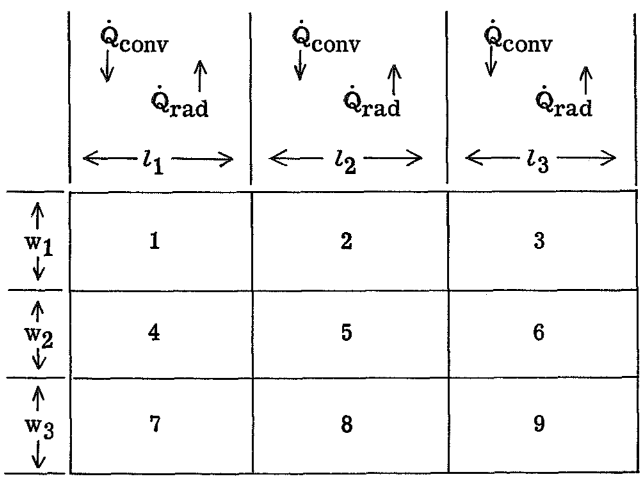
Geometry Layout for the Physical Problem

The configuration is divided into small volumes or blocks at the discretion of the user. (Present restrictions of storage areas allow a maximum of 100 divisions and is limited to only 30 for the case of calculating heating rates from known temperatures.) The geometry of these divisions and their heat relationships are then given as input to the program. The program computes the temperatures of these divisions or blocks at their centers. The program is also restricted to the use of only ten materials in one configuration at any one time.

Finite-Difference Solution

The implicit form (backward difference) of the heat-balance equations is used in this program, the finite-difference network coming from the division of the configuration being analyzed into small volumes called blocks. The mass of each block is considered to be concentrated at its center of gravity, and heat is assumed to flow from block to block over a distance equal to that between the centers of gravity and through an area equal to the touching surfaces.

The following example illustrates the procedure for the finite-difference solution. Consider the simplified case of a thermally thick skin in which the exterior surface is subjected to convective heating and the back wall is insulated. The exterior surface is permitted to radiate, and the remaining heat-transfer mode is the exchange of heat between blocks by conduction. The configuration is shown in sketch (a).



Sketch (a)

The heat-balance equation for each block is

$$\dot{Q}_{conv} - \dot{Q}_{rad\text{-}out} - \dot{Q}_{cond} = \dot{Q}_{stored} \tag{47}$$

By using this form of the general heat-balance equation and choosing block 2 as an example, the following equations arise. By considering each term separately,

$$\left(\dot{Q}_{conv}\right)_2 = hA_2(T_r - T_2) \tag{48}$$

where A_2 is surface area of block 2

$$\left(\dot{Q}_{rad\text{-}out}\right)_2 = \sigma\epsilon_2 A_2 \left(T_2'\right)^4 \tag{49}$$

and the conduction terms are

$$\left(\Delta\dot{Q}_{cond}\right)_{2,1} = \frac{2A_{2,1}\left(T_2 - T_1\right)}{\dfrac{l_2}{k_{m,2}} + \dfrac{l_1}{k_{m,1}}} \tag{50}$$

$$\left(\Delta\dot{Q}_{cond}\right)_{2,3} = \frac{2A_{2,3}\left(T_2 - T_3\right)}{\dfrac{l_2}{k_{m,2}} + \dfrac{l_3}{k_{m,3}}} \tag{51}$$

$$\left(\Delta\dot{Q}_{cond}\right)_{2,5} = \frac{2A_{2,5}\left(T_2 - T_5\right)}{\dfrac{w_2}{k_{m,2}} + \dfrac{w_5}{k_{m,5}}} \tag{52}$$

where k_m is the conductivity of the material of the specified blocks.

$$\left(\dot{Q}_{cond}\right)_2 = \left(\Delta\dot{Q}_{cond}\right)_{2,1} + \left(\Delta\dot{Q}_{cond}\right)_{2,3} + \left(\Delta\dot{Q}_{cond}\right)_{2,5} \tag{53}$$

and finally

$$\left(\dot{Q}_{stored}\right)_2 = \left(\rho_{m,2}\right)v_2 c_{p,2} \frac{T_2 - T_2'}{\Delta t} \tag{54}$$

where T_2' is the previous temperature of block 2 and Δt is the specified time increment. The thermal properties c_p, k_m, and ϵ are based on T' for the specific block. Summation of these terms yields

$$\left(\rho_{m,2}\right) v_2 c_{p,2} \frac{T_2 - T_2'}{\Delta t} = hA_2\left(T_r - T_2\right) - \sigma\epsilon_2 A_2 T_2'^4 - \frac{2A_{2,1}\left(T_2 - T_1\right)}{\frac{l_2}{k_{m,2}} + \frac{l_1}{k_{m,1}}}$$

$$- \frac{2A_{2,3}\left(T_2 - T_3\right)}{\frac{l_2}{k_{m,2}} + \frac{l_3}{k_{m,3}}} - \frac{2A_{2,5}\left(T_2 - T_5\right)}{\frac{w_2}{k_{m,2}} + \frac{w_5}{k_{m,5}}} \tag{55}$$

Let

$$\left.\begin{aligned} C_1 &\equiv \frac{2A_{2,1}}{\frac{l_2}{k_{m,2}} + \frac{l_1}{k_{m,1}}} \\ C_3 &\equiv \frac{2A_{2,3}}{\frac{l_2}{k_{m,2}} + \frac{l_3}{k_{m,3}}} \\ C_5 &\equiv \frac{2A_{2,5}}{\frac{w_2}{k_{m,2}} + \frac{w_5}{k_{m,5}}} \end{aligned}\right\} \tag{56}$$

and

$$\left.\begin{aligned} C_{hs} &\equiv \frac{\left(\rho_{m,2}\right) v c_{p,2}}{\Delta t} \\ C_{cv} &\equiv hA_2 \\ C_{ro} &\equiv \sigma\epsilon_2 A_2\left(T_2'\right)^4 \end{aligned}\right\} \tag{57}$$

Substituting equations (56) and (57) into equation (55) yields

$$C_{cv}\left(T_r - T_2\right) - C_{ro} - C_1\left(T_2 - T_1\right) - C_3\left(T_2 - T_3\right) - C_5\left(T_2 - T_5\right) = C_{hs}\left(T_2 - T_2'\right) \quad (58)$$

By collecting like terms, equation (58) becomes

$$T_2\left(-C_{cv} - C_1 - C_3 - C_5 - C_{hs}\right) + C_1T_1 + C_3T_3 + C_5T_5 = C_{ro} - C_{cv}T_r - C_{hs}T_2' \quad (59)$$

At any given time increment, the right-hand side of equation (59) may be evaluated as a constant with all parameters known. Thus, equation (59) now becomes

$$\left(-C_{cv} - C_1 - C_3 - C_5 - C_{hs}\right)T_2 + C_1T_1 + C_3T_3 + C_5T_5 = \text{Constant} \quad (60)$$

Therefore, energy balance equations can be written for each block in a configuration resulting in a set of simultaneous equations which are solved by a matrix subroutine at each time interval giving a temperature distribution through the blocks.

Thin Wall Option

A special option, subroutine THINWAL, is included in the program to compute the heating rate and temperature time histories for a thermally thin skin. The thickness of the single block of unit surface area is specified by the user. The inputs of width, length, volume, and the conduction, convection, and radiation areas are not required for this special case.

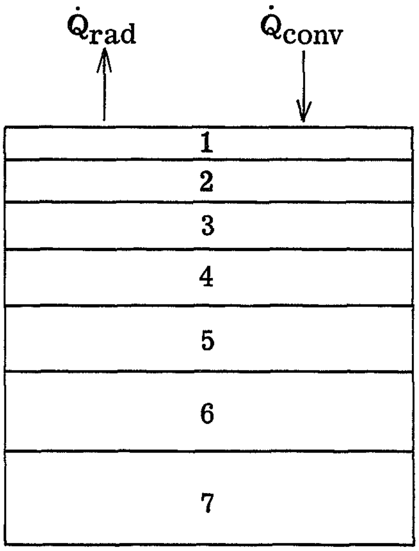
Inverse Solution Option

When the temperature of one of the block divisions is known and the heating rate is unknown, the inverse option may be used to calculate heating rates from the known temperature time history of the block as input. (See ref. 18.) The solution is restricted to one-dimensional cases.

This option uses the equation in which the convective heating term appears in the heat flux equation; and by a process of reordering the equations for each of the unknown temperatures, a new system of simultaneous linear equations is set up. This system is solved by a library subroutine SIMEQ (see appendix C) which solves the matrix equation $AX = B$ (where X denotes the unknown variables, A is a square coefficient matrix, and B is a vector of constants). The heating rate, rather than the known temperature, is evaluated by the subroutine.

If there are more temperature monitoring locations in the skin, the additional information can be incorporated into the solution for the convective heating-rate time history.

The program is written to accommodate as many as four known temperature time histories of the blocks. Basically, the procedure is to sum the temperatures recorded at the various locations (blocks) at a given time and then to replace one of the known temperatures in the finite-difference equations with the numerical sum of all the known temperatures minus the functional form for the monitored temperatures of the remaining blocks. Consider, for example, sketch (b) which illustrates the inverse solution procedure.



Sketch (b)

If the temperature histories of blocks 2, 4, and 6 are known, and if block 2 is chosen as the "pivotal" element, which means that everywhere T_2 appears in the governing equations, it is replaced by the numerical sum of $T_2 + T_4 + T_6 \equiv T$ less T_4 and T_6 (where T_4 and T_6 are now taken to be unknown a priori), then the finite-difference equations become

For block 1,

$$\left(\dot{Q}_{conv}\right)_1 - \sigma\epsilon_1 A_1\left(T_1'\right)^4 + \frac{2A_{1,2}\left[(T - T_4 - T_6) - T_1\right]}{\frac{l_1}{k_{m,1}} + \frac{l_2}{k_{m,2}}} = \rho_{m,1}v_1c_{p,1}\frac{T_1 - T_1'}{\Delta t}$$

where $k_{m,2}$ is evaluated at the previous temperature $T_2' = T' - T_4' - T_6'$.

For block 2,

$$\frac{2A_{2,1}\left[T_1 - (T - T_4 - T_6)\right]}{\frac{l_2}{k_{m,2}} + \frac{l_1}{k_{m,1}}} + \frac{2A_{2,3}\left[T_3 - (T - T_4 - T_6)\right]}{\frac{l_2}{k_{m,2}} + \frac{l_3}{k_{m,3}}} = \rho_{m,2}v_2c_{p,2}\frac{(T - T_4 - T_6) - T_2'}{\Delta t}$$

For block 3,

$$\frac{2A_{3,2}\left[(T - T_4 - T_6) - T_3\right]}{\dfrac{l_3}{k_{m,3}} + \dfrac{l_2}{k_{m,2}}} + \frac{2A_{3,4}(T_4 - T_3)}{\dfrac{l_3}{k_{m,3}} + \dfrac{l_4}{k_{m,4}}} = \rho_{m,3} v_3 c_{p,3} \frac{T_3 - T_3'}{\Delta t}$$

et cetera.

In this example there are seven equations and seven unknowns, $\dot{Q}_{conv}$, T_1, T_3, T_4, T_5, T_6, and T_7. The solution is obtained from the SIMEQ subroutine. This method is based on the assumption that random errors in thermocouple readings will tend to balance out when summed, and although the sum of the known temperatures remains the same, the solution may not give temperatures of the known blocks which are equal to their individual input temperature-time histories. However, for reasonably accurate thermocouple data and material properties, the computed individual temperatures should be fairly close to the measured data.

Note that for multiple thermocouple data, the known block nearest the heat source (namely, the lowest numbered block for which the temperature is known) is used as the pivotal element for the heating-rate computation.

PROGRAM DESCRIPTION, FLOW CHARTS, AND LISTINGS

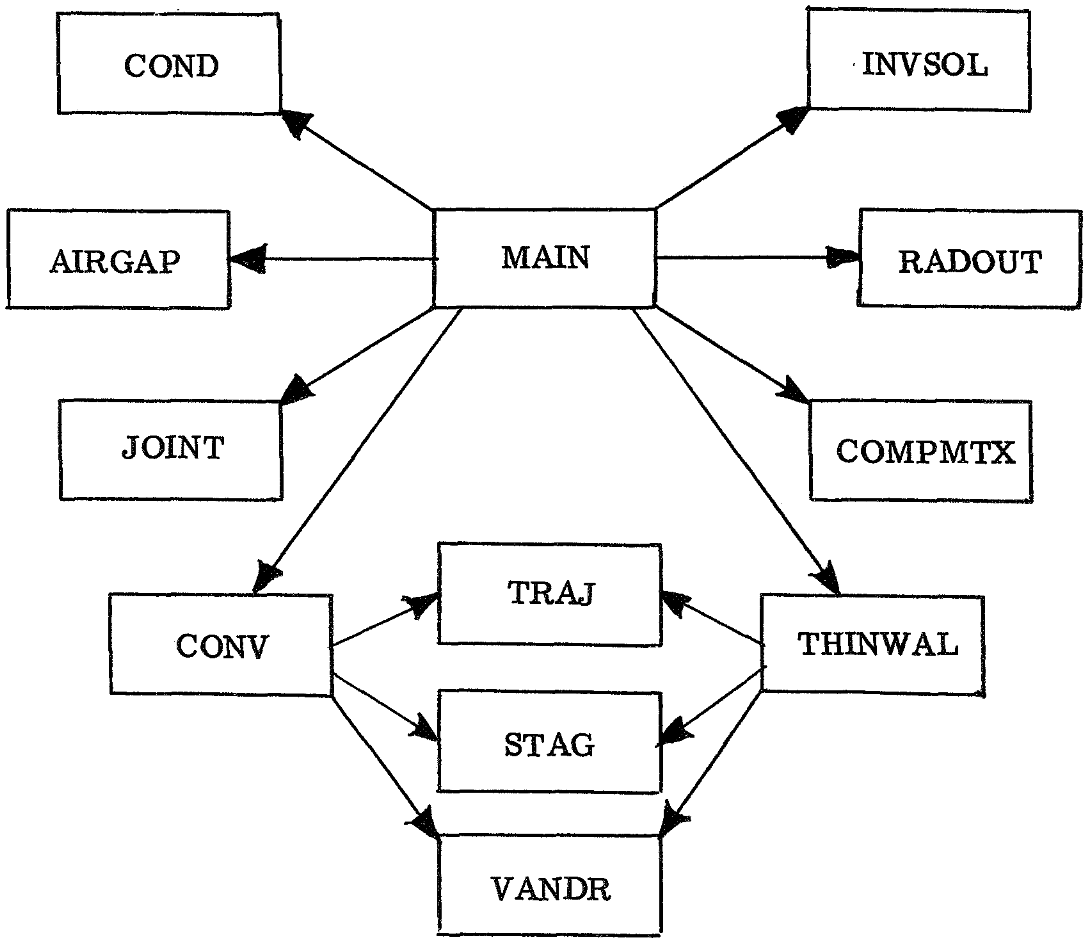
Program Description

In the following pages the program, its subroutines, and their functions are given. A graphical representation of the communication between the program and its subroutines follows. A brief description of each subroutine is followed by its flow diagram and listing.

Program	Description
MAIN	Reads and writes NAMELIST input, initializes program variables, and controls program flow
VANDR	Computes the heat-transfer coefficient using Van Driest equations
STAG	Computes the heat-transfer coefficient using Sibulkin, Detra-Kemp-Riddell, and Cohen equations
TRAJ	Computes local and free-stream conditions
COND	Computes conduction coefficient

RADOUT	Computes radiation-out coefficient
AIRGAP	Computes the coefficients for conduction and radiation across an air gap
JOINT	Computes the joint coefficient
COMPMTX	Solves the packed coefficient matrix for temperatures
THINWAL	Solves a simplified thin-wall case
CONV	Computes the final heating-rate coefficient
INVSOL	Solves the coefficient matrix when an inverse solution is desired

The directed flow diagram of program D1244 follows:



A detailed listing of FORTRAN variables used in the program and their descriptions is contained in the "FORTRAN Variable Description" section and within the listing of the MAIN program. These variables appear at the beginning of the listing as comments following the initial reference of each variable. Also, pertinent details as to internal control for specific options are included with these variables.

FORTRAN Variable Description

ACOND	conduction areas
ACRADI	storage for air-gap information
AIRK	average value for conductivity of air
AJOINT	storage for joint information
AKMTAB	table values for conductivity of air
AK1	conductivity of air for operating block
AK2	conductivity of air for block opposite air gap
ALPHA	$f_1(M_e)$ (eq. (14))
ALT	value of altitude from table values
ALTTAB	table values of altitude from input trajectory
ARRAY1	POP/P1 tables (pitot to free-stream pressure ratio)
ARRAY2	ROP/R1 tables (normal shock stagnation to free-stream density ratio)
ARRAY3	TOP/T1 tables (normal shock stagnation to free-stream temperature ratio)
ATAB	altitude table for normal shock-wave parameters
BETA	$f_2(M_e)$ (eq. (14))
CFMTAB	Mach number table for skin-friction coefficient
CFR	$c_{f,o}$
CFS	c_f
CFTAB	reference skin-friction-coefficient table

CFTTAB	temperature ratio table for skin-friction coefficients
CH	N_{St}
COEF	array of heat coefficients for matrix solution
COHEN	control for option to use Cohen equations
CONST	storage for constant part of block equations
CONVRO	storage for convection and radiation-out information
CPIDEAL	$\left(c_p\right)_{Perfect}$
CPTAB	tables of specific-heat values for materials
CPVAL	value of specific heat from table
CROAREA	convection and radiation-out areas
D1	width or length of operating block
D2	width or length of touching block
DELTAT	computing interval (delta time, Δt)
DELTEMP	change in temperature over time interval
EMS	storage for emissivity information
EMSS	value of emissivity in subroutine RADOUT
EMSVAL	value of emissivity in subroutine THINWAL
EM1	emissivity of operating block
EM2	emissivity of block across air gap
FPCONE	control for flat plate or cone
G	acceleration due to gravity
G1TE	$G_1\left(i_d\right)$ (eq. (41))
HIPTS	number of points in heat input tables
HITIME	table of times for given heat input

HSUBDIF	$i_s - i_w$
HSUBR	i_s
HSUBW	i_w
HTAB	table of values for h given as inputs
HVAL	h
ICONVQ	control for heating options
ICROBLK	convection and radiation-out block numbers
IDIFF	control for different heat input tables
IDIM	configuration control
IHORQ	control for program flow
IMAT	material number currently in use
INVBLK	inverse block number currently in use
INVBLKS	inverse block numbers
INVERSE	control for inverse option
INVRT	control for program flow
IORDER	order of interpolation
IPFCT	counter for print frequency
IPTS	storage for number of table values
IQORH	control for heating rates given as input
IROUTE	control for program flow
ITBLK	block number of touching block
JTBLK	block number of joint block
KMTAB	tables of conductivity values for materials
LEN	lengths of blocks

LINEAR	control for linearizing T^4
LOCH	location of h
LOCQ	location of $\dot{Q}$
MACTAB	Mach number ratio table
MAT	material numbers of blocks
MFS	free-stream Mach numbers, M_∞
MFSTAB	table of Mach numbers for ratio evaluations
MLOCL	local Mach number, M_e
MUFS	μ_∞
NACBLK	number of air conduction block
NAIRGAP	number of air gaps
NAKMPT	number of values in k_{AIR} tables
NAVPTS	number of values in trajectory input
NBLOCK	current block number
NCDCT	counter for conduction terms
NCPPTS	number of c_p table values
NHOR	number to determine matrix storage
NJOINT	number of joints
NKMPTS	number of k_m table values
NO	number of blocks
NOCONV	number of convection blocks
NOINV	number of inverse blocks
NOINVT	number of values in inverse temperature table
NPRS	number of values in table

NRADO	number of radiation-out terms
NRATAB	number of ratio table
NRATIO	control for ratio table number
NRATPT	number of points in ratio tables
NTOUCH	numbers of touching blocks
NUMCD	number of conduction terms for current block
PELOG	LOG(p_s)
PERCRNL	η or Reynolds number lengths
PESTAG	p_s
PFS	free-stream pressure, p_∞
POPP1	POP/P1 (pitot to free-stream pressure ratio)
PRESTB	pressure table (log)
PRFREQ	print frequency
PRNO	value of Prandtl number
PRNOTB	table of Prandtl number values
PRWSTG	$(N_{Pr,w})_s$
PR2TAB	table of $N_{Pr}{}^{1/2}$
PR3TAB	table of $N_{Pr}{}^{1/3}$
QCONV	value of $\dot{q}$
QTAB	table of $\dot{q}$ values when given as input
QVAL	array in which computer values of heat-transfer coefficients are stored
RADIAT	radiation-out term in subroutine THINWAL
RADNOSE	nose radius, r_n
REYTAB	Reynolds number ratio table

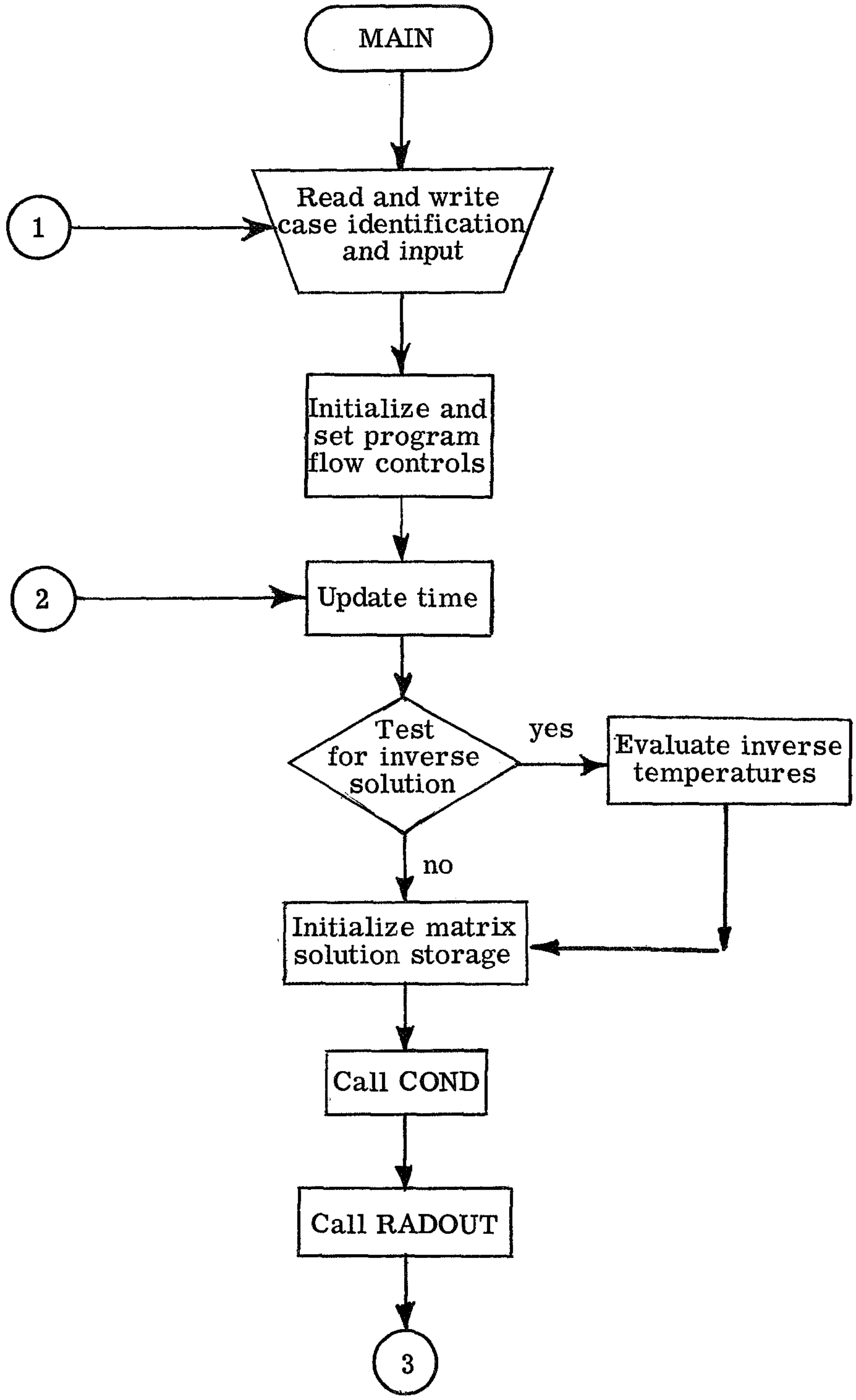
RHO	density of material, $\rho_{m,i}$
RHOLOCL	local density, ρ_e
RHOSF3	ρ, slugs/ft^3
RHOSTAG	ρ_s
RHOTAB	density ratio table ρ_e/ρ_∞
RNEXP	N (eq. (24))
RNFS	free-stream Reynolds number, $N_{Re,\infty}$
RNLOCL	local Reynolds number, $N_{Re,e}$
RNPFL	local Reynolds number per foot, $N_{Re,e}/ft$
ROPR1	ROP/R1 (normal-shock stagnation to free-stream density ratio)
ROWSTAG	$\rho_{w,s}$
TAW	value of adiabatic wall temperature, T_r
TAWTAB	table of TAW values when H and TAW are given inputs
TCPTAB	table of temperatures for all c_p tables
TCUBED	T^3
TEMINV	inverse temperature table
TEMP	temperatures of blocks
TEMPINV	temperature of inverse block
TEMTAB	temperature ratio table
TESTAG	$T_{e,s}$
TFS	free-stream temperature, T_∞
TIME	current time
TIMEAV	time table for trajectory input
TIMSTOP	stopping time

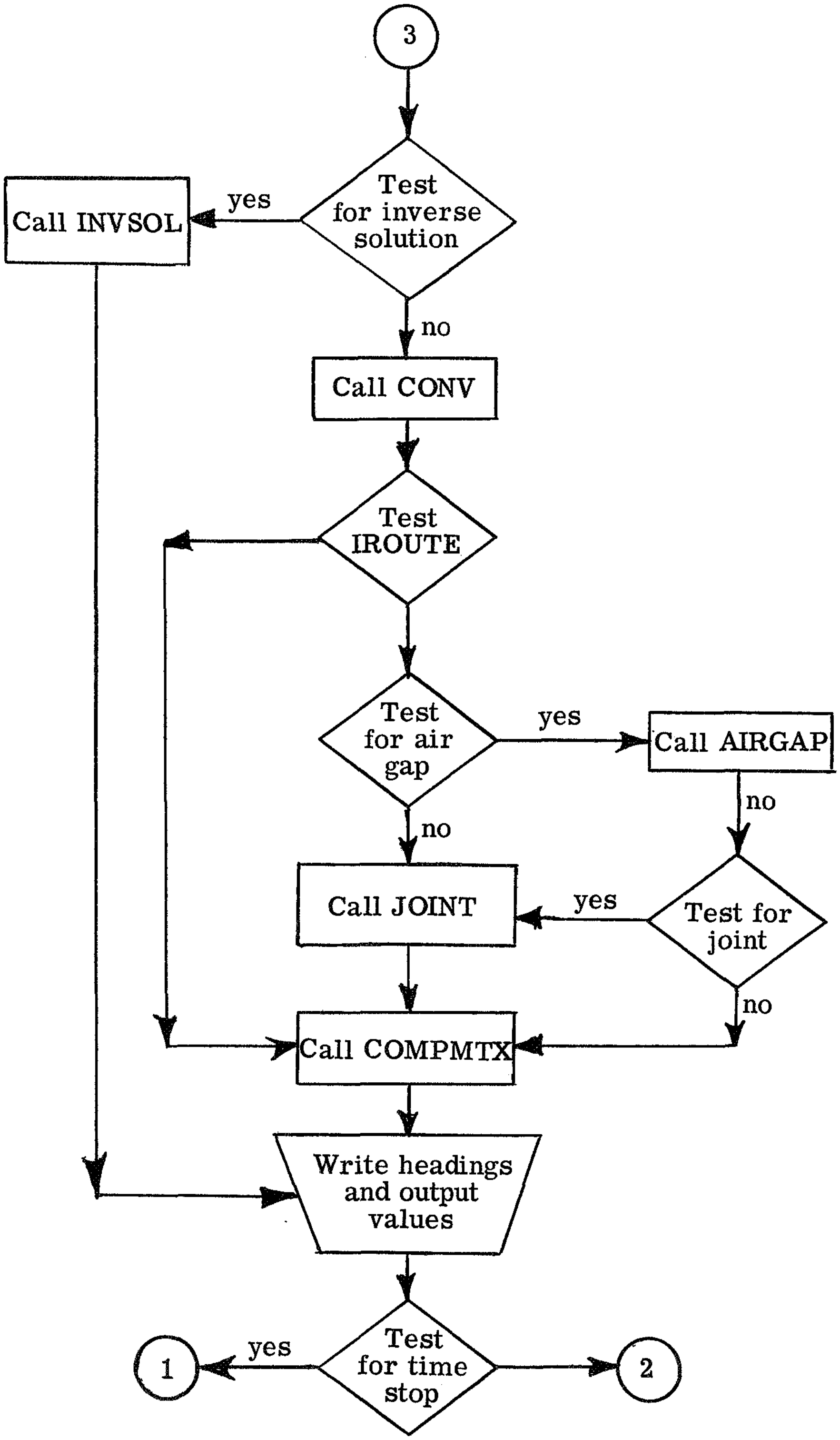
TINV	table of times for given temperature inputs
TKMTAB	temperature tables for all k_m values
TLOCL	local temperature, T_e
TMPTAB	temperature table for viscosity and enthalpy
TOPT1	TOP/T1 (normal-shock stagnation to free-stream temperature ratio)
TPRTAB	temperature table for $N_{Pr}^{1/2}$ and $N_{Pr}^{1/3}$
TRANRN	transition Reynolds number
TRAT	temperature ratio, T_w/T_e
TREC	T_r
TSTAG	T_s
TSUBE	i_d
TTKTAB	table of temperatures for k_m values
TWDEPTH	thin-wall depth
T1P4	T_1^4
T2P4	T_2^4
UESTAG	$\mu_{e,s}$
UEUREF	μ_e/μ_o
UEURTAB	viscosity ratio table
UREFS	$\mu_{o,s}$
UREFW	$\mu_{o,w}$
UWSTAG	$\mu_{w,s}$
UWUREF	μ_w/μ_o
VALCP	value of c_p of material
VALK1	value of k_m of operating block

VALK2	value of k_m of touching block
VELTAB	table of velocities for input trajectory
VERTAB	velocity ratio table, V_e/V_∞
VFS	free-stream velocity, V_∞
VLOCL	local velocity, V_e
VOL	volume of blocks
VTAB	velocity table for normal shock-wave parameters
WD	widths of blocks
WVCP	$\rho v c_p$
ZHRT	dimensionless enthalpy, $Z\frac{i}{RT}$
ZHRTAB	table for dimensionless enthalpy
ZPRTEM	temperature tables for compressibility and Prandtl number
ZTAB	table for compressibility
ZW	value of compressibility from table, Z_w

MAIN

Program MAIN is the control program. It reads and writes NAMELIST input, initializes, and sets up controls for flow of program as designated by input values. This control program then calls the designated subroutines for solution, writes the output, and either updates and continues solution or terminates according to input specifications. The flow diagram of program MAIN follows:


MAIN
1
Read and write case identification and input
Initialize and set program flow controls
2
Update time
Test for inverse solution
yes
Evaluate inverse temperatures
no
Initialize matrix solution storage
Call COND
Call RADOUT
3


3
Test for inverse solution
yes
Call INVSOL
no
Call CONV
Test IROUTE
Test for air gap
yes
Call AIRGAP
no
no
Call JOINT
yes
Test for joint
no
Call COMPMTX
Write headings and output values
Test for time stop
yes
1
2

The MAIN program listing follows:

```
      PROGRAM D1244(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
      COMMON  TIME,DELTAT,TIMSTOP,PRFREQ,NO,IDIM,ICONVQ,G,
     1TEMP(100),CONST(100)
*             TIME   =STARTING TIME
*             DELTAT =DELTA TIME
*             TIMSTOP=STOPPING TIME
*             PRFREQ =PRINT FREQUENCY
*             NO     =NO OF BLOCKS
*             IDIM, 0=THIN WALL
*                   1=ONE-DIMENSIONAL
*                   2=TWO-DIMENSIONAL  (RECTANGULAR)
*             ICONVQ, 0=INVERSE SOLUTION
*                     1=HEATING RATES ARE GIVEN
*                     2=STAGNATION OPTION
*                     3=VAN DRIEST OPTION
*                     4=STAGNATION AND VAN DRIEST COMBINATION
*             G      =GRAVITY
*             TEMP   =TEMPERATURES OF THE BLOCKS
*             CONST  =STORAGE FOR CONSTANT PART OF BLOCK EQUATIONS
      COMMON /ALLCONV/ CONVRO(30,5),IDIFF,NOCONV,QCONV(30),
     1TAW(30),IHORQ
*             CONVRO- COL 1 =ICROBLK (CONVECTION AND RAD.OUT BLOCK NOS)
*                     COL 2 =CROAREA (CONVECTION AND RAD.OUT AREAS)
*                     COL 3 =PERCRNL (EITHER PERCENTS OR REY.NO LENGTHS
*                     COL 4 =IQORH (WHEN HEATING RATES ARE GIVEN-THE CODE
*                           IS  1 IF Q INPUTS AND 2 IF H AND TAW INPUTS)
*                       OR  =NRATAB(WHEN HEATING RATES ARE COMPUTED THE NO.
*                           OF THE RATIO TABLE IS GIVEN-ALL ONES IF ONLY
*                           ONE TABLE)
*                     COL 5 =HIPTS(NO OF POINTS IN HEAT INPUT TABLES)
*             IDIFF  =0 IF HEATING RATE TABLES ARE SAME FOR EACH CONV BLK
*                    =1 IF HEATING RATE TABLES ARE DIFF FOR EACH CONV BLK
*             NOCONV =NO OF CONVECTION BLOCKS
*             QCONV  =STORAGE FOR HEATING RATE IN PROGRAM
*             TAW    =STORAGE FOR ADIABATIC WALL TEMPERATURE
*             IHORQ  =TEST FOR OUTPUT HEADINGS(EITHER Q OR H-NOT INPUT)
      COMMON /TVDASTG/ ALT,VFS,PFS,TFS,MFS,RNFS,RHOSF3,VLOCL,MLOCL,
     1RNPFL,RNLOCL,TLOCL,RHOLOCL
*             ALT    =ALTITUDE FROM TABLE VALUES
*             VFS    =FREE-STREAM VELOCITY
*             PFS    =FREE-STREAM PRESSURE
*             TFS    =FREE-STREAM TEMPERATURE
*             MFS    =FREE-STREAM MACH NO
*             RNFS   =FREE-STREAM REYNOLDS NO
*             RHOSF3 =FREE-STREAM DENSITY (SLUGS/CUBIC FT.)
*             VLOCL  =LOCAL  VELOCITY
*             MLOCL  =LOCAL  MACH NO
*             RNPFL  =LOCAL  REYNOLDS NO /FT.
*             RNLOCL =LOCAL  REYNOLDS NO.
*             TLOCL  =LOCAL  TEMPERATURE
*             RHOLOCL=LOCAL  DENSITY
      COMMON /CVONLY/ HITIME(200),HTAB(200),TAWTAB(200),QTAB(200)
*             HITIME =TABLE OF TIMES FOR GIVEN HEAT INPUTS
*             HTAB   =TABLE FOR H VALUES WHEN H AND TAW ARE GIVEN INPUTS
*             TAWTAB =TABLE OF TAWVALUES WHEN H AND TAW ARE GIVEN INPUTS
*             QTAB   =TABLE OF Q VALUES WHEN Q IS GIVEN AS INPUT
      COMMON /ARRAYS/ ARRAY1(1800),ARRAY2(1800),ARRAY3(1800),ATAB(40),
     1VTAB(45)
*          TABLES FOR NORMAL SHOCK WAVE PARAMETERS IN IMPERFECT AIR
*             ARRAY1 =POP/P1 (PITOT TO FREE-STREAM PRESSURE RATIO)
*             ARRAY2 =ROP/R1 (NORMAL SHOCK STAGNATION DENSITY RATIO)
*             ARRAY3 =TOP/T1 (NORMAL SHOCK STAGNATION TEMPERATURE RATIO)
*             ATAB   =ALTITUDE TABLE (FOR ABOVE VALUES)
*             VTAB   =VELOCITY TABLE (FOR ABOVE VALUES)
```

```
      COMMON /VDONLY/ TRANRN,FPCONE
*             TRANRN =TRANSITION REYNOLDS NUMBER
*             FPCONE =DESIGNATION OF FLAT PLATE OR CONE
*                     FLAT PLATE = 1.
*                     CONE = SQ.RT(3) OR  1.7320508
      COMMON /STGONLY/ COHEN,RADNOSE
*             COHEN  =CODE IS 0 IF USE OF THESE EQ IS REQ. OVER 20000FT/S
*                     CODE IS 1 IF USE OF THESE EQ. IS FOR ALL RANGES(STAG)
*             RADNOSE=RADIUS OF THE NOSE (STAGNATION OPTION)
      COMMON /TRAJO/ NAVPTS,TIMEAV(100),ALTTAB(100),VELTAB(100),NRATPT,
     1MFSTAB(25),VERTAB(50),MACTAB(50),REYTAB(50),TEMTAB(50),RHOTAB(50)
*             NAVPTS =NO OF ALT AND VEL POINTS FOR GIVEN TRAJECTORY
*             TIMEAV =TIME TABLE FOR TRAJECTORY
*             ALTTAB =ALTITUDE TABLE FOR TRAJECTORY
*             VELTAB =VELOCITY TABLE FOR TRAJECTORY
*             NRATPT =NO OF POINTS FOR LOCAL/FREE-STREAM RATIO TABLES
*             MFSTAB =MACH NO TABLE FOR RATIO EVALUATIONS
*             VERTAB =VELOCITY RATIO TABLE
*             MACTAB =MACH NO  RATIO TABLE
*             REYTAB =REYNOLDS NO RATIO TABLE
*             TEMTAB =TEMPERATURE RATIO TABLE
*             RHOTAB =DENSITY RATIO TABLE
      COMMON /CDGEOM/ NTOUCH(5,100),ACOND(300)
*             NTOUCH =ROW 1 OF THIS ARRAY DESIGNATES THE NO OF CONDUCTION
*                     TERMS FOR EACH BLOCK(NO REPEATS) ROWS 2-5 ARE THE
*                     BLOCK NUMBERS OF THESE COND. TERMS(-INDICATES WD)
*             ACOND  =CONDUCTION AREAS CORRESPONDING TO NTOUCH VALUES
      COMMON  /MATPROP/ MAT(100),EMS(40),RHO(10),NCPPTS(10),TCPTAB(100),
     1CPTAB(100),NKMPTS(10),TTKTAB(100),KMTAB(100)
*             MAT    =MATERIAL NO OF EACH OF THE BLOCKS
*             EMS    =EMMISSIVITY OF MATERIALS
*             RHO    =DENSITIES OF MATERIALS
*             NCPPTS =NO OF POINTS IN EACH CP TABLE
*             TCPTAB =TEMPERATURE TABLES FOR ALL  CP TABLES
*             CPTAB  =SPECIFIC HEAT TABLES FOR MATERIALS
*             NKMPTS =NO OF POINTS IN EACH KM TABLE
*             TKMTAB =TEMPERATURE TABLES FOR ALL KM TABLES
*             KMTAB  =CONDUCTIVITY TABLES FOR MATERIALS
      COMMON /BLKDIM/  WD(100),VOL(100),LEN(100),NHOR
*             WD     =WIDTHS OF THE BLOCKS
*             VOL    =VOLUMES OF THE BLOCKS
*             LEN    =LENGTHS OF THE BLOCKS
*             NHOR   =NO OF HORIZONTAL BLOCKS
      COMMON  /RADONLY/NRADO,LINEAR
*             NRADO  =NO OF RADIATION-OUT BLOCKS
*             LINEAR =IF 0- T**4 IS USED AS CONST,IF 1-T**4 IS LINEARIZED
      COMMON /AGONLY/  NAIRGAP,ACRADI(4,30)
*             NAIRGAP=NO OF AIR GAPS
*             ACRADI =ROW 1 = NO OF AIR GAP BLOCK
*                     ROW 2 = NO OF BLK OPPOSITE AIR GAP (GT AIR GAP BLK)
*                     ROW 3 = DISTANCE ACROSS AIR GAP
*                     ROW 4 = AREA ON AIR GAP
      COMMON /JTONLY/  NJOINT,AJOINT(4,20)
*             NJOINT =NO OF JOINT TERMS
*             AJOINT =ROW 1 =NO OF JOINT BLOCK
*                     ROW 2 =BLK NO OF CONNECTING JOINT BLK
      COMMON /THIN/  TWDEPTH
*             TWDEPTH=THIN WALL DEPTH
      COMMON /INVRS/ INVERSE,INVBLK,TEMPINV,NOINV,INVBLKS(4),NOINVT,
```

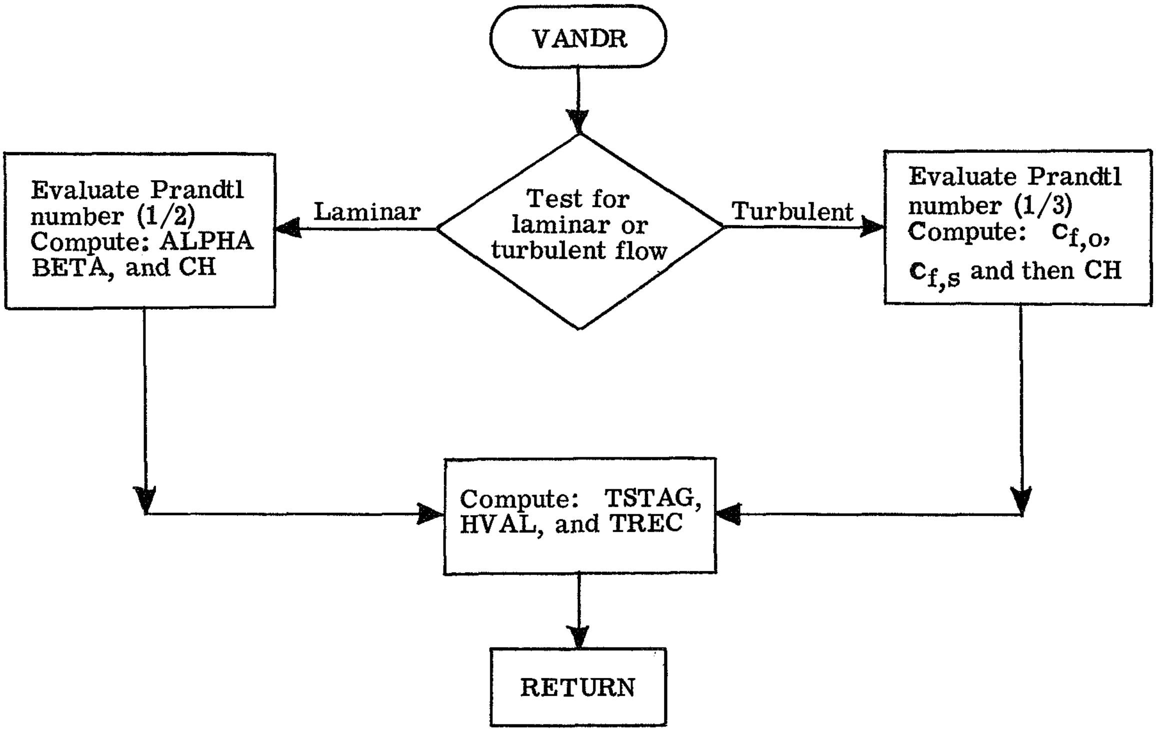
```
      1TINV(200),TEMINV(400)
*           INVERSE=IF 0,NO INVERSE SOL REQUIRED,IF 1 ,INVERSE REQUIRED
*           INVBLK =INVERSE BLOCK NUMBER (USED INTERNALLY)
*           TEMPINV=TEMPERATURE OF INVERSE BLOCK
*           NOINV  =NO OF INVERSE BLOCKS
*           INVBLKS=INVERSE BLOCK NUMBERS
*           NOINVT =NO OF TIME-TEMPERATURE POINTS AS INPUT
*           TINV   =TABLE OF TIMES FOR GIVEN TEMPERATURE INPUTS
*           TEMINV =TEMPERATURE TABLE FOR INVERSE BLOCKS
       COMMON /CVTRAJ/NRATIO
       DIMENSION COEF(100,20),TOTAL(30)
       DIMENSION ICROBLK(30),CROAREA(30),PERCRNL(30),IQORH(30),HIPTS(30),
      1KM1(10),KM2(10),KM3(10),KM4(10),KM5(10),KM6(10),KM7(10),KM8(10),
      2KM9(10),KM10(10),TKTAB1(10),TKTAB2(10),TKTAB3(10),TKTAB4(10),
      3TKTAB5(10),TKTAB6(10),TKTAB7(10),TKTAB8(10),TKTAB9(10),TKTAB10(10)
      4,CP1(10),CP2(10),CP3(10),CP4(10),CP5(10),CP6(10),CP7(10),CP8(10),
      5CP9(10),CP10(10),TCPTAB1(10),TCPTAB2(10),TCPTAB3(10),TCPTAB4(10),
      6TCPTAB5(10),TCPTAB6(10),TCPTAB7(10),TCPTAB8(10),TCPTAB9(10),
      7TCPTB10(10),EMS1(4),EMS2(4),EMS3(4),EMS4(4),EMS5(4),EMS6(4),
      8EMS7(4),EMS8(4),EMS9(4),EMS10(4),INDEX(30,2)
       DIMENSION NRATAB(30)
       DIMENSION TEMINV1(100),TEMINV2(100),TEMINV3(100),TEMINV4(100)
       REAL KMTAB,KM1,KM2,KM3,KM4,KM5,KM6,KM7,KM8,KM9,KM10
       REAL MFS,MLOCL,MFSTAB,MACTAB
       REAL  LEN
        INTEGER COHEN
        INTEGER PRFREQ
       REAL ICROBLK,IQORH,NRATAB
       EQUIVALENCE  (CONVRO(1,1),ICROBLK(1)),(CONVRO(1,2),CROAREA(1)),
      1(CONVRO(1,3),PERCRNL(1)),(CONVRO(1,4),IQORH(1)),(CONVRO(1,5),
      2HIPTS(1))
       EQUIVALENCE  (KMTAB(1),KM1(1)),(KMTAB(11),KM2(1)),(KMTAB(21),
      1KM3(1)),(KMTAB(31),KM4(1)),(KMTAB(41),KM5(1)),(KMTAB(51),KM6(1)),
      2(KMTAB(61),KM7(1)),(KMTAB(71),KM8(1)),(KMTAB(81),KM9(1)),
      3(KMTAB(91),KM10(1)),(TTKTAB(1),TKTAB1(1)),(TTKTAB(11),TKTAB2(1)),
      4(TTKTAB(21),TKTAB3(1)),(TTKTAB(31),TKTAB4(1)),(TTKTAB(41),TKTAB5
      5(1)),(TTKTAB(51),TKTAB6(1)),(TTKTAB(61),TKTAB7(1)),(TTKTAB(71),
      6TKTAB8(1)),(TTKTAB(81),TKTAB9(1)),(TTKTAB(91),TKTAB10(1))
       EQUIVALENCE (CPTAB(1),CP1(1)),(CPTAB(11),CP2(1)),(CPTAB(21),CP3
      1(1)),(CPTAB(31),CP4(1)),(CPTAB(41),CP5(1)),(CPTAB(51),CP6(1)),
      2(CPTAB(61),CP7(1)),(CPTAB(71),CP8(1)),(CPTAB(81),CP9(1)),(CPTAB
      3(91),CP10(1)),(TCPTAB(1),TCPTAB1(1)),(TCPTAB(11),TCPTAB2(1)),
      4(TCPTAB(21),TCPTAB3(1)),(TCPTAB(31),TCPTAB4(1)),(TCPTAB(41),
      5TCPTAB5(1)),(TCPTAB(51),TCPTAB6(1)),(TCPTAB(61),TCPTAB7(1)),
      6(TCPTAB(71),TCPTAB8(1)),(TCPTAB(81),TCPTAB9(1)),(TCPTAB(91),
      7TCPTB10(1))
       EQUIVALENCE (CONVRO(1,4),NRATAB(1))
       DIMENSION VERTAB1(25),VERTAB2(25),MACTAB1(25),MACTAB2(25),
      1REYTAB1(25),REYTAB2(25),TEMTAB1(25),TEMTAB2(25),RHOTAB1(25),
      2RHOTAB2(25)
       EQUIVALENCE(VERTAB1(1),VERTAB(1)),(VERTAB2(1),VERTAB(26)),
      1(MACTAB1(1),MACTAB(1)),(MACTAB2(1),MACTAB(26)),(REYTAB1(1),
      2REYTAB(1)),(REYTAB2(1),REYTAB(26)),(TEMTAB1(1),TEMTAB(1)),
      3(TEMTAB2(1),TEMTAB(26)),(RHOTAB1(1),RHOTAB(1)),(RHOTAB2(1),
      4RHOTAB(26))
       EQUIVALENCE (TEMINV1(1),TEMINV(1)),(TEMINV2(1),TEMINV(101)),
      1(TEMINV3(1),TEMINV(201)),(TEMINV4(1)   ,TEMINV(301))
       REAL  MACTAB1,MACTAB2
       EQUIVALENCE  (EMS(1),EMS1(1)),(EMS(5),EMS2(1)),(EMS(9),EMS3(1)),
      1(EMS(13),EMS4(1)),(EMS(17),EMS5(1)),(EMS(21),EMS6(1)),(EMS(25),
      2EMS7(1)),(EMS(29),EMS8(1)),(EMS(33),EMS9(1)),(EMS(37),EMS10(1))
       NAMELIST /CASEINP/ NO,NHOR,NVER,NOCONV,NRADO,NJOINT,NAIRGAP,
      1IGEOM,IDIM,ICONVQ,IDIFF,LINEAR,INVERSE,COHEN,TIME,TIMSTOP,DELTAT,
      2PRFREQ,TEMP,MAT,EMS1,EMS2,EMS3,EMS4,EMS5,EMS6,EMS7,EMS8,
      3EMS9,EMS10,RHO,NKMPTS,TKTAB1,KM1,TKTAB2,KM2,TKTAB3,KM3,TKTAB4,KM4,
      4TKTAB5,KM5,TKTAB6,KM6,TKTAB7,KM7,TKTAB8,KM8,TKTAB9,KM9,TKTAB10,
      5KM10,NCPPTS,TCPTAB1,CP1,TCPTAB2,CP2,TCPTAB3,CP3,TCPTAB4,CP4,
```

```
     6TCPTAB5,CP5,TCPTAB6,CP6,TCPTAB7,CP7,TCPTAB8,CP8,TCPTAB9,CP9,
     7TCPTB10,CP10,WD,RAD,LEN,THETA,VOL,TWDEPTH,ACOND,NTOUCH,ICROBLK,
     8CROAREA,PERCRNL,IQORH,HIPTS,TRANRN,FPCONE,RADNOSE,ACRADI,AJOINT,
     9HITIME,HTAB,TAWTAB,QTAB,NOINV,INVBLKS,NOINVT,TINV,TEMINV,
     XTEMINV1,TEMINV2,TEMINV3,TEMINV4,
     1NAVPTS,TIMEAV,ALTTAB,VELTAB,NRATPT,MFSTAB,VERTAB,MACTAB,REYTAB,
     2TEMTAB,RHOTAB,NRATAB,VERTAB1,VERTAB2,MACTAB1,MACTAB2,REYTAB1,
     3REYTAB2,TEMTAB1,TEMTAB2,RHOTAB1,RHOTAB2
*
    5 READ (5,1000)
 1000 FORMAT (80H1
     1
      IF (EOF,5)   6,7
    6 STOP
    7 READ (5,CASEINP)
      WRITE (6,1000)
      G= 32.1741
      IPFCT = 0
      IF (IDIM.NE.0) GO TO 15
      CALL THINWAL
      GO TO 5
*
*
*
   15 WRITE (6,CASEINP)
      WRITE(6,1000)
*
      IF (NAIRGAP.EQ.0.AND.NJOINT.EQ.0) 20,25
*
*  NO AIRGAPS OR JOINTS (SET ROUTE)
*
   20 IROUTE=1
      GO TO 30
*
*  EITHER AIRGAPS,JOINTS, OR BOTH  (SET ROUTE)
*
   25 IROUTE=2
*
   30 IF (INVERSE.EQ.0) GO TO 35
      INVRT=2
      MDUMMY=NO
      NDUMMY=NO
      GO TO  40
*
   35 INVRT=1
      MDUMMY= NO
      NDUMMY= NHOR+1
      IF (IDIM.EQ.1)  NDUMMY=2
*
*  UPDATE TIME BY DELTA TIME TO BEGIN COMPUTATION
*
   40 TIME = TIME + DELTAT
      GO TO (50,42) INVRT
*
   42 INVBLK= INVBLKS(1)
      IF (NOINV.EQ.1)  GO TO 45
      J=1
      TEMPINV=0.0
      DO 44   I=1,NOINV
      CALL FTLUP (TIME,SAVE,1,NOINVT,TINV(1),TEMINV(J))
      J=J+100
   44 TEMPINV= TEMPINV+SAVE
      GO TO 50
   45 CALL FTLUP(TIME,TEMPINV,1,NOINVT,TINV(1),TEMINV(1))
*
```

```
   50 DO 60 I=1,100
      DO 55 J=1,20
   55 COEF(I,J)=0.0
   60 CONST(I)=0.0
      CALL COND (COEF,MDUMMY,NDUMMY)
*
      CALL RADOUT (COEF,MDUMMY,NDUMMY)
*
      GO TO (65,90),INVRT
*
   65 CALL CONV (COEF,MDUMMY,NDUMMY)
*
      GO TO (80,70), IROUTE
*
   70 IF (NAIRGAP.EQ.0) GO TO 75
      CALL AIRGAP (COEF,MDUMMY,NDUMMY)
*
      IF (NJOINT.EQ.0) GO TO 80
   75 CALL JOINT (COEF,MDUMMY,NDUMMY)
*
*  SOLUTION OF THE COMPACT SYMMETRICAL MATRIX
*
   80 CALL COMPMTX(COEF,MDUMMY,NDUMMY)
      GO TO 100
*
   90 CALL INVSOL(COEF,MDUMMY,INDEX)
*
  100 IPFCT= IPFCT+1
      IF (IPFCT.NE.PRFREQ) GO TO 40
      IPFCT=0
      IF (INVERSE.NE.0)  GO TO 150
*
      WRITE (6,2000) TIME,DELTAT
 2000 FORMAT(1X///6X6HTIME =F10.4,6X13HDELTA TIME = F6.4//)
*
      IF (IHORQ.EQ.2) GO TO 120
*
      DO 110  I= 1,NOCONV
  110 TOTAL(I)=QCONV(I)*(TAW(I)-TEMP(I))
      J=NOCONV+1
      WRITE (6,3000)
 3000 FORMAT (1H ,6X9HBLOCK  NO11X11HTEMPERATURE8X12HCONVECTION Q11X
     18HH * AREA10X11HT(RECOVERY)//)
*
      DO 125  I=1,NOCONV
  125 WRITE (6,4000)  I,TEMP(I),TOTAL(I),QCONV(I),TAW(I)
 4000 FORMAT (10X,I3,6X,4E20.8)
*
      DO 126  I=J,NO
  126 WRITE (6,5000)  I,TEMP(I)
 5000 FORMAT (10X,I3,6X,E20.8)
*
      GO TO 150
*
  120 WRITE (6,6000)
 6000 FORMAT (1H ,6X9HBLOCK  NO11X11HTEMPERATURE8X12HCONVECTION Q//)
      DO 130  I=1,NOCONV
  130 WRITE (6,7000)  I,TEMP(I),QCONV(I)
 7000 FORMAT (10X,I3,6X,2E20.8)
      J=NOCONV+1
      DO 140  I=J,NO
  140 WRITE (6,5000)  I,TEMP(I)
*
  150 IF (TIME.LT.TIMSTOP) GO TO 40
*
      GO TO 5
*
      END
```

VANDR

Subroutine VANDR computes the heat-transfer coefficient using the Van Driest equations with options for either turbulent or laminar flow based on transition Reynolds number. The flow diagram for subroutine VANDR follows.



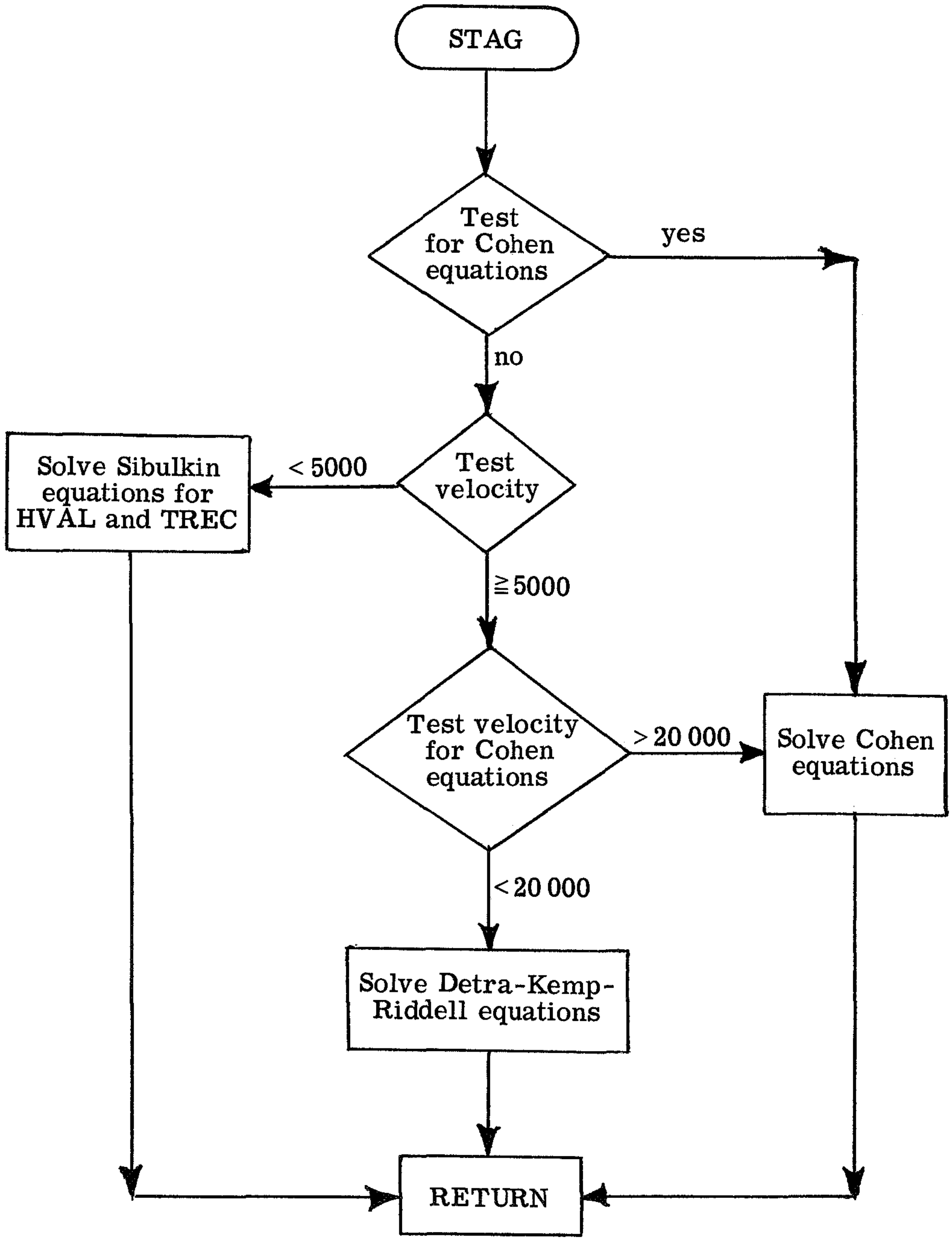
The program listing for VANDR is

```
      SUBROUTINE VANDR(I,HVAL,TREC)
*
*        A SUBROUTINE TO CALCULATE HEATING RATES USING VAN DRIEST EQUATIONS
*        THIS SUBROUTINE IS CALLED BY TWO SUBROUTINES - CONV AND THINWAL
*
      COMMON   TIME,DELTAT,TIMSTOP,PRFREQ,NO,IDIM,ICONVQ,G,
     1TEMP(100),CONST(100)
      COMMON /ALLCONV/ CONVRO(30,5),IDIFF,NOCONV,QCONV(30),
     1TAW(30),IHORQ
      COMMON /TVDASTG/ ALT,VFS,PFS,TFS,MFS,RNFS,RHOSF3,VLOCL,MLOCL,
     1RNPFL,RNLOCL,TLOCL,RHOLOCL
      COMMON /ARRAYS/ ARRAY1(1800),ARRAY2(1800),ARRAY3(1800),ATAB(40),
     1VTAB(45)
      COMMON /VDONLY/ TRANRN,FPCONE
      DIMENSION CFMTAB(18),CFTTAB(8),CFTAB(144),TPRTAB(10),PR2TAB(10),
     1PR3TAB(10)
      REAL  MFS,MLOCL
      DATA TPRTAB/400.,500.,600.,700.,800.,900.,1100.,1250.,2200.,3900.
     1/, PR3TAB/.900,.8936,.8880,.8840,.8809,.8790,.8790,.8805,.8932,
     2.9018/, PR2TAB/.8555,.8448,.8367,.8310,.827,.825,.825,.827,.8437,
     3.8567/
```

```
      DATA   CFMTAB/.5,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,12.0,
     113.0,14.0,15.0,16.0,18.0,20./, CFTTAB/.2,.4,.6,1.0,2.0,3.0,4.0,6./
      DATA   CFTAB/.0051,.004904,.00429,.003596,.002978,.002475,.002078,
     1.001764,.001516,.001317,.001156,.000913,.0008208,.0007425,
     2.0006755,.0006177,.0005236,.0004507,.004546,.004411,.00396,
     3.003416,.002901,.00246,.002097,.001804,.001565,.001371,.001211,
     4.0009676,.0008734,.0007931,.0007239,.000664,.0005657,.0004891,
     5.00417,.00407,.003709,.003258,.002812,.002417,.002084,.001809,
     6.001581,.001394,.001237,.000995,.0009016,.0008209,.007512,
     7.0006905,.0005905,.0005121,.003673,.003603,.003345,.003005,
     8.002652,.002325,.002036,.001788,.001582,.001407,.001258,.001025,
     9.0009322,.0008523,.0007828,.0007219,.0006208,.0005409,.002990,
     X.002948,.0028,.002593,.002357,.002126,.001907,.001711,.001539,
     1.001389,.001259,.001047,.0009598,.0008838,.0008168,.0007576,
     2.0006578,.0005777,.00261,.002578,.002475,.002328,.002153,.001974,
     3.001798,.001635,.001489,.001357,.001241,.001046,.0009652,.0008934,
     4.0008295,.0007724,.0006755,.0005967,.00235,.002328,.002251,
     5.002137,.002,.001854,.001708,.001568,.001441,.001323,.001218,
     6.001039,.000963,.000895,.0008341,.007793,.0006853,.0006082,
     7.002005,.002,.00195,.001875,.001779,.001675,.001566,.00146,
     8.001357,.001262,.001173,.001017,.00095,.0008887,.0008331,.0007825,
     9.0006944,.000621/
      RNLOCL=RNPFL * CONVRO(I,3)
      TRAT= TEMP(I)/TLOCL
      IF (RNLOCL.GE.TRANRN)  GO TO 150
*
*  LAMINAR FLOW
*
      CALL FTLUP(TEMP(I),PRNO,1,10,TPRTAB,PR2TAB)
      SAVE=SQRT(RNLOCL)
*
      ALPHA = .416594+MLOCL*(-.00246733+ MLOCL*(-8.17489E-4+ MLOCL*
     1 2.734033E-5))
*
      BETA= TRAT*(-.0134671+ MLOCL*(2.635807E-4 +MLOCL*(5.818944E-5-
     1 MLOCL*2.173257E-6)))
      CH = FPCONE*(ALPHA+ BETA)/ SAVE
*
      GO TO  200
*
*  TURBULENT FLOW
*
  150 CALL FTLUP(TEMP(I),PRNO,1,10,TPRTAB,PR3TAB)
*
      CALL DISCOT(MLOCL,TRAT,CFMTAB,CFTAB,CFTTAB,11,144,8,CFR)
*
      SAVE= -1826.3756+9737.6*TRAT
      A=(668.12+ SQRT(SAVE))/4868.8
      B=.0019-.0001*TRAT
      RNEXP=A+B*MLOCL
*
      SAVE1=1.0
      IF (FPCONE.NE.1.0) SAVE1=.5
*
      CFS= CFR/((SAVE1*RNLOCL/1.E6)**RNEXP)
      CH = .6*CFS
*
*  TURBURLENT  OR LAMINAR  FLOW
*
  200 CPIDEAL=.24
      TSTAG=TFS*(1.0+.2*MFS**2)
      HVAL= CH* RHOLOCL*CPIDEAL*VLOCL*G
      TREC= PRNO*(TSTAG -TLOCL)+ TLOCL
      IHORQ=1
*
  300 RETURN
      END
```

STAG

Subroutine STAG computes the stagnation heat-transfer coefficient by either the Sibulkin equations, the Detra-Kemp-Riddell equations, or the Cohen equations according to the value of the free-stream velocity. For locations not at the stagnation point, a ratio value designated in the input is used for the percentage of the stagnation value.



The program listing for subroutine STAG is

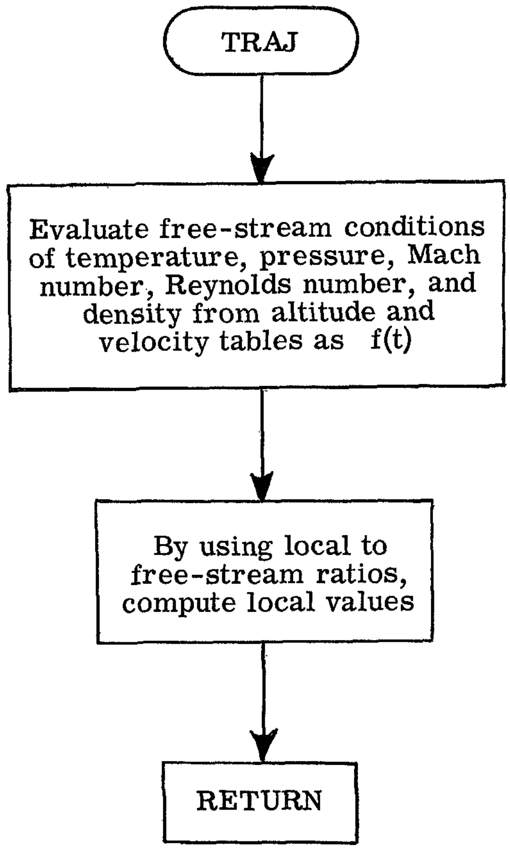
```
      SUBROUTINE STAG(I,HVAL,TREC,HSUBDIF)
*
*        A SUBROUTINE TO CALCULATE HEATING RATES AT STAGNATION -IT USES
*        SIBULKIN DETRA-KEMP-RIDDELL,AND COHEN EQUATIONS
*        THIS SUBROUTINE IS CALLED BY TWO SUBROUTINES - CONV AND THINWAL
*
      COMMON   TIME,DELTAT,TIMSTOP,PRFREQ,NO,IDIM,ICONVQ,G,
     1TEMP(100),CONST(100)
      COMMON /ALLCONV/ CONVRO(30,5),IDIFF,NOCONV,QCONV(30),
     1TAW(30),IHORQ
      COMMON /STGONLY/ COHEN,RADNOSE
      COMMON /TVDASTG/ ALT,VFS,PFS,TFS,MFS,RNFS,RHOSF3,VLOCL,MLOCL,
     1RNPFL,RNLOCL,TLOCL,RHOLOCL
      COMMON /ARRAYS/ ARRAY1(1800),ARRAY2(1800),ARRAY3(1800),ATAB(40),
     1VTAB(45)
      INTEGER  COHEN
      REAL  MFS,MLOCL
      DIMENSION TMPTAR(32),PRESTB(7),ZHRTTAB(224),ZPRTEM(12),ZTAB(84),
     1PRNOTB (84),UEURTAB(224)
      DATA  TMPTAB/0.0,800.,900.,1800.,2700.,3600.,4500.,5400.,6300.,
     17200.,8100.,9000.,9900.,10800.,11700.,12600.,13500.,14400.,15300.,
     216200.,17100.,18000.,18900.,19800.,20700.,21600.,22500.,23400.,
     324300.,25200.,26100.,27000./, PRESTB/-.6743202,.3256798,1.3256798,
     42.3256798,3.3256798,4.3256798,5.3256798/
      DATA  ZHRTTAB/0.0,3.5,3.52,3.65,3.80,
     14.41,8.02,8.19,8.03,9.82,16.8,23.46,23.58,22.54,21.93,22.29,24.26,
     228.65,35.75,43.74,49.15,50.96,50.64,49.48,48.07,46.64,45.26,43.97,
     342.76,41.64,40.58,39.60,0.0,3.5,3.52,3.65,3.8,4.07,6.16,8.02,7.77,
     48.09,10.55,16.68,21.58,21.97,21.24,20.69,20.72,21.65,23.85,27.66,
     533.0,38.79,43.28,45.57,46.09,45.64,44.74,43.69,42.61,41.55,40.53,
     639.57,0.0,3.5,3.52,3.65,3.8,3.97,4.81,7.13,7.62,7.53,8.14,10.48,
     715.14,19.3,20.35,20.01,19.54,19.34,19.6,20.49,22.17,24.78,28.28,
     832.31,36.13,39.01,40.66,41.26,41.17,40.69,40.01,39.24,0.0,3.5,
     93.52,3.65,3.8,3.93,4.27,5.55,7.08,7.28,7.33,7.96,9.73,12.93,16.46,
     X18.34,18.66,18.43,18.17,18.09,18.29,18.85,19.84,21.31,23.28,25.69,
     128.36,30.99,33.27,34.97,36.02,36.53,0.0,3.5,3.52,3.65,3.8,3.92,
     24.09,4.61,5.75,6.74,6.98,7.1,7.58,8.7,10.64,13.2,15.48,16.73,17.09
     3,17.04,16.91,16.84,16.9,17.13,17.57,18.24,19.16,20.32,21.72,23.29,
     424.98,26.66,0.,3.5,3.52,3.65,3.8,3.92,4.03,4.25,4.75,5.56,6.29,
     56.62,6.8,7.11,7.72,8.76,10.24,11.99,13.63,14.79,15.4,15.61,15.64,
     615.6,15.58,15.62,15.74,15.96,16.28,16.71,17.26,17.92,0.0,3.5,3.52,
     73.65,3.8,3.92,4.01,4.13,4.34,4.7,5.2,5.73,6.13,6.38,6.62,6.95,
     87.44,8.16,9.1,10.2,11.36,12.42,13.23,13.77,14.08,14.22,14.28,14.3,
     914.31,14.34,14.4,14.49/
      DATA  ZTAB/ 5*1.0,1.016,1.163,1.2,1.211,1.287,1.577,1.910,
     15 * 1.0,1.005,1.088,1.192,1.203,1.228,1.337,1.622,
     25 * 1.0,1.002,1.033,1.149,1.197,1.208,1.245,1.359,
     35 * 1.0,1.001,1.011,1.072,1.167,1.198,1.213,1.252,
     46 * 1.0,1.004,1.026,1.092,1.165,1.196,1.214,
```

```
     56 * 1.0,1.001,1.009,1.035,1.089,1.149,1.186,
     67 * 1.0,1.003,1.012,1.033,1.071,1.118/,
     7ZPRTEM/ 0.0,800.,900.,1800.,2700.,3600.,4500.,5400.,6300.,7200.,
     88100.,9000./,    PRNOTB  /.75,.75,.738,.756,.767,.614,.771,.714,
     9.606,.587,.764,.993,.75,.75,.738,.756,.767,.668,.654,.745,.658,
     X.58,.611,.799,.75,.75,.738,.756,.767,.724,.611,.74,.737,.619,.578,
     1.624,.75,.75,.738,.756,.767,.766,.645,.636,.744,.759,.610,.581,
     2.75,.75,.738,.755,.767,.773,.696,.627,.660,.762,.752,.611,
     3.75,.75,.738,.756,.767,.773,.751,.680,.631,.662,.743,.767,
     4.75,.75,.738,.756,.767,.773,.762,.740,.678,.640,.654,.702/
      DATA  UEURTAB/8*1.0,1.011,1.032,1.096,1.181,1.227,1.256,1.271,
     11.264,1.21,1.072,.826,.517,.261,.118,.055,.029,.018,.012,.009,
     2.008,.007,.007,.008,.008,
     38*1.0,1.01,1.024,1.055,1.128,1.209,1.257,1.286,1.303,1.307,1.280,
     41.207,1.068,.853,.595,.361,.2,.108,.063,.036,.024,.018,.015,.013,
     5.012,  8*1.0,1.01,1.022,1.038,1.074,1.146,1.228,1.276,1.317,1.337,
     61.347,1.343,1.314,1.251,1.143,.983,.782,.571,.387,.249,.158,.10,
     7.067,.042,.016,  8* 1.0,1.006,1.02,1.033,1.051,1.086,1.148,1.229,
     81.294,1.332,1.371,1.386,1.396,1.393,1.375,1.335,1.267,1.168,1.04,
     9.881,.711,.547,.408,.268,.212, 8* 1.0,1.003,1.016,1.029,1.043,
     X1.06,1.09,1.139,1.208,1.283,1.342,1.386,1.425,1.438,1.445,1.448,
     11.442,1.424,1.394,1.342,1.274,1.187,1.082,.94,.828,
     28* 1.0,1.001,1.008,1.022,1.036,1.052,1.067,1.09,1.124,1.175,1.238,
     31.307,1.368,1.418,1.468,1.496,1.501,1.511,1.52,1.516,1.508,1.492,
     41.468,1.415,1.387,   9 * 1.0,1.003,1.01,1.022,1.036,1.05,1.072,
     51.089,1.112,1.143,1.185,1.238,1.298,1.361,1.418,1.467,1.509,1.549,
     61.577,1.581,1.594,1.599,1.601,1.604/
      IF (COHEN.EQ.1) GO TO 50
      IF (VFS.GE.5000.) GO TO  50
*
*  SIBULKIN OPTION
*
      B = .80 +.541*MFS -.00574* MFS**2
*
      HVAL=B*G*RHOSF3*VFS*.24*CONVRO(I,3)/(SQRT(RNFS*RADNOSE))
      TREC= TFS*(1.0+.2*MFS**2)
      IHORQ=1
      RETURN
*
   50 IF  (VFS.GT.5000.) GO TO 60
      SAVE1 = (1.2*MFS**2)**3.5
      SAVE2= (6./(7.0*MFS**2-1.0))**2.5
      POPP1=  SAVE1   *SAVE2
      GO TO 70
   60 CALL DISCOT(VFS,ALT,VTAB,ARRAY1,ATAB,11,1800,40,POPP1)
   70 PESTAG=POPP1*PFS
      PELOG = ALOG10(PESTAG)
      CALL DISCOT(TEMP(I),PELOG,TMPTAB,ZHRTTAB,PRESTB,11,224,7,ZHRT)
      HSUBW= .0686*ZHRT*TEMP(I)
      HSUBR=(VFS**2)/(50000.) +.24*TFS
      IF (COHEN.EQ.1) GO TO 100
      IF (VFS.GT.20000.) GO TO 100
*
```

```
*  DETRA-KEMP-RIDDELL OPTION
*
      HVAL=(.447E-8 *SQRT(RHOSF3)* VFS**3.15)/(SQRT(RADNOSE)*(HSUBR-
     1130.))*CONVRO(I,3)
      GO TO 200
*
*  COHEN OPTION
*
  100 IF (VFS.GT.5000.)  GO TO 110
      TOPT1 = 1.0+.2*MFS**2
      TESTAG = TOPT1*TFS
      ROPR1=POPP1/TOPT1
      RHOSTAG=ROPR1*RHOSF3
      GO TO 120
  110 CALL DISCOT(VFS,ALT,VTAB,ARRAY2,ATAB,11,1800,40,ROPR1)
      RHOSTAG= ROPR1*RHOSF3
      CALL DISCOT(VFS,ALT,VTAB,ARRAY3,ATAB,11,1800,40,TOPT1)
      TESTAG= TOPT1*TFS
  120 CALL DISCOT(TEMP(I),PELOG,TMPTAB,UEURTAB,PRESTB,11,224,7,UWUREF)
      CALL DISCOT(TESTAG,PELOG,TMPTAB,UEURTAB,PRESTB,11,224,7,UEUREF)
*
      UREFS=(2.28E-8*TESTAG**1.5)/(TESTAG + 201.6)
      UREFW=(2.28E-8*TEMP(I)**1.5)/(TEMP(I)+201.6)
*
      UESTAG= UREFS* UEUREF
      UWSTAG= UREFW* UWUREF
*
      CALL DISCOT(TEMP(I),PELOG ,ZPRTEM,ZTAB,PRESTB,11,84,7,ZW)
      CALL DISCOT(TEMP(I),PELOG ,ZPRTEM,PRNOTB,PRESTB,11,84,7,PRWSTG)
*
      ROWSTAG = 5.825E-4* PESTAG/(ZW * TEMP(I))
*
      TSUBE=8465./(VFS**2/50000.)
*
      IF (TSUBE.LT..5) GO TO 150
      G1TE=1.0
      GO TO 160
  150 G1TE= 1.0+ .075*(1.0/(TSUBE)-2.0)**2
  160 SAVE1= PRWSTG**(-.6)
      SAVE2=(RHOSTAG*UESTAG)**.43
      SAVE3=(ROWSTAG*UWSTAG)**.07
      SAVE4=(2.0*(PESTAG-PFS)/RHOSTAG)**.25
*
      HVAL=.767*G*SAVE1*SAVE2*SAVE3*SAVE4*G1TE*CONVRO(I,3)/SQRT(RADNOSE)
*
  200 HSUBDIF = HSUBR-HSUBW
      IHORQ=2
*
      RETURN
      END
```

TRAJ

The subroutine TRAJ computes the free-stream and local conditions from a given trajectory of altitude and velocity as functions of time. The flow diagram for the subroutine TRAJ follows.

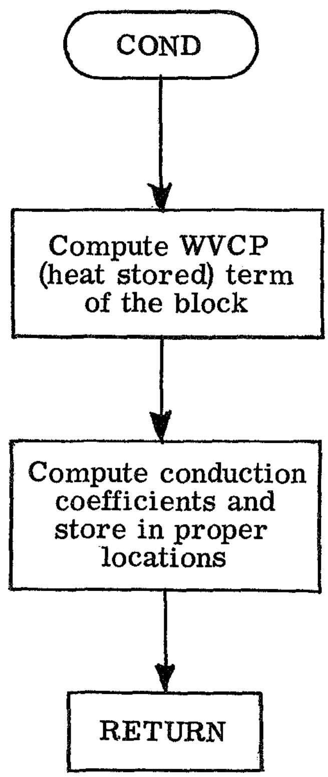


The program listing for subroutine TRAJ is

```
      SUBROUTINE TRAJ
*
*        A SUBROUTINE TO CALCULATE  FREE-STREAM AND LOCAL CONDITIONS FROM A
*        GIVEN TRAJ FOR USE IN VANDR AND STAG
*        THIS SUBROUTINE IS CALLED BY TWO SUBROUTINES - CONV AND THINWAL
*
      COMMON  TIME,DELTAT,TIMSTOP,PRFREQ,NO,IDIM,ICONVQ,G,
     1TEMP(100),CONST(100)
      DIMENSION ANS(4)
*
      COMMON /TRAJO/ NAVPTS,TIMEAV(100),ALTTAB(100),VELTAB(100),NRATPT,
     1MFSTAB(25),VERTAB(50),MACTAB(50),REYTAB(50),TEMTAB(50),RHOTAB(50)
      COMMON /TVDASTG/ALT,VFS,PFS,TFS,MFS,RNFS,RHOSF3,VLOCL,MLOCL,RNPFL,
     1RNLOCL,TLOCL,RHOLOCL
      COMMON /CVTRAJ/NRATIO
*
      REAL MFSTAB,MACTAB,MFS,MLOCL,MUFS
*
      CALL FTLUP(TIME,ALT,1,NAVPTS,TIMEAV,ALTTAB)
      CALL FTLUP(TIME,VFS,1,NAVPTS,TIMEAV,VELTAB)
      CALL AT62 (ALT,ANS)
      TFS=1.8 * ANS(3)
      RHOSF3=ANS(1)
      PFS =ANS(2)
      MFS = VFS /ANS(4)
      MUFS= 2.27E-8 * TFS**(3./2.)/(TFS+198.6)
      RNFS= RHOSF3 *ANS(4)* MFS/MUFS
*
      IF (ICONVQ.NE.3) GO TO 10
      J=1
*
*  LOCAL VALUES FOUND BY USING LOCAL TO FREE-STREAM RATIOS
*
      L=1
      IF (NRATPT.EQ.0)  J=0
      IF (IDIM.EQ.0)  GO TO 5
      IF (NRATIO.EQ.2)  L=26
    5 CALL FTLUP (MFS,TABVAL,J,NRATPT,MFSTAB,VERTAB(L))
      VLOCL = VFS * TABVAL
      CALL FTLUP (MFS,TABVAL,J,NRATPT,MFSTAB,MACTAB(L))
      MLOCL = MFS * TABVAL
      CALL FTLUP (MFS,TABVAL,J,NRATPT,MFSTAB,REYTAB(L))
      RNPFL = RNFS * TABVAL
      CALL FTLUP (MFS,TABVAL,J,NRATPT,MFSTAB,TEMTAB(L))
      TLOCL = TFS * TABVAL
      CALL FTLUP (MFS,TABVAL,J,NRATPT,MFSTAB,RHOTAB(L))
      RHOLOCL = RHOSF3 *TABVAL
*
   10 CONTINUE
      RETURN
      END
```

COND

Subroutine COND computes the conduction coefficient and stores it in the coefficient matrix for solution. The flow diagram for subroutine COND follows.



The program listing for subroutine COND is

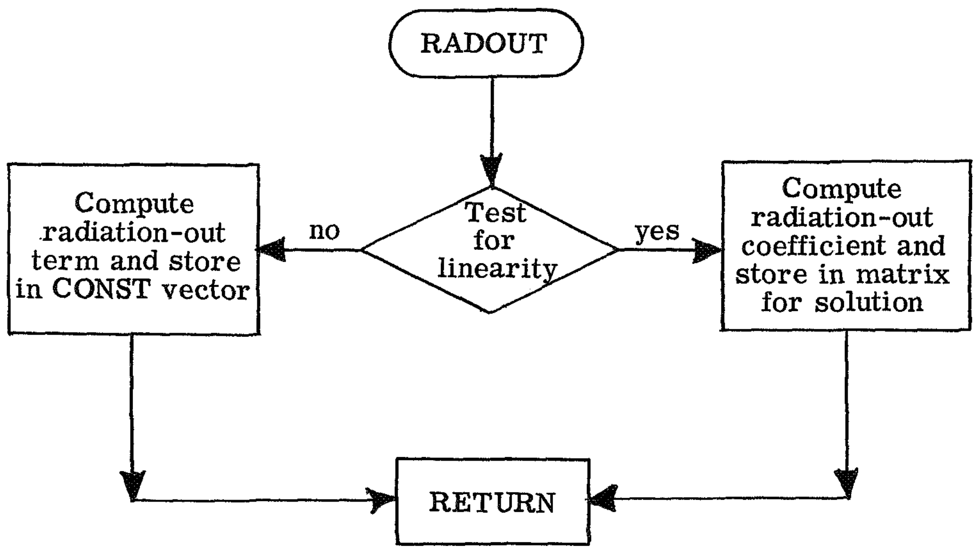
```
      SUBROUTINE   COND  (COEF,MDUMMY,NDUMMY)
*
*        A SUBROUTINE TO CALCULATE THE CONDUCTION TERMS
*        THIS SUBROUTINE IS CALLED BY THE MAIN PROGRAM
*
*
      DIMENSION  COEF(MDUMMY,NDUMMY)
      COMMON   TIME,DELTAT,TIMSTOP,PRFREQ,NO,IDIM,ICONVQ,G,
     1TEMP(100),CONST(100)
      COMMON /CDGEOM/ NTOUCH(5,100),ACOND(300)
      COMMON /ALLCONV/ CONVRQ(30,5),IDIFF,NOCONV,QCONV(30),
     1TAW(30),IHORQ
      COMMON   /MATPROP/ MAT(100),EMS(40),RHO(10),NCPPTS(10),TCPTAB(100),
     1CPTAB(100),NKMPTS(10),TTKTAB(100),KMTAB(100)
      COMMON /BLKDIM/   WD(100),VOL(100),LEN(100),NHOR,NVER
      COMMON /INVRS/ INVERSE,INVBLK,TEMPINV,NOINV,INVBLKS(4),NOINVT,
     1TINV(200),TEMINV(400)
      REAL LEN
      REAL KMTAB
*
    1 IF (INVERSE.EQ.1) INVRT=1
      NCDCT = 1
      DO 60  NBLOCK=1,NO
      IORDER=1
      IMAT =MAT(NBLOCK)
      I=IMAT *10-9
      NPRS=NCPPTS(IMAT)
      IF (NPRS.EQ.0) IORDER=0
      CALL FTLUP (TEMP(NBLOCK),VALCP,IORDER,NPRS,TCPTAB(I),CPTAB(I))
      WVCP = RHO(IMAT) * VOL(NBLOCK) * VALCP/ DELTAT
*
      NUMCD=NTOUCH(1,NBLOCK)
      IF (INVERSE.EQ.1)  GO TO 2
      COEF(NBLOCK,1)=COEF(NBLOCK,1) -WVCP
      CONST(NBLOCK)=CONST(NBLOCK) - WVCP*TEMP(NBLOCK)
      IF  (NUMCD.NE.0)  GO TO 3
      GO TO 60
    2 IF (NUMCD.NE.0) GO TO 3
      IF (INVBLK.EQ.NO)  GO TO 4
      COEF(NBLOCK,NBLOCK)=COEF(NBLOCK,NBLOCK)-WVCP
      CONST(NBLOCK)=CONST(NBLOCK)-WVCP*TEMP(NBLOCK)
      GO TO 60
    4 CONST(NBLOCK)=CONST(NBLOCK)+WVCP*(TEMPINV-TEMP(NBLOCK))
      GO TO 60
*
    3 IORDER=1
*
      NPRS=NKMPTS(IMAT)
      IF (NPRS.EQ.0) IORDER=0
      CALL FTLUP (TEMP(NBLOCK),VALK1,IORDER,NPRS,TTKTAB(I),KMTAB(I))
*
      DO  50   J=1,NUMCD
*
*  TEST TOUCHING BLOCK NOS AND FIND DISTANCES ACCORDING TO PROPER SIGN
*
      ITBLK=NTOUCH(J+1,NBLOCK)
      IF (ITBLK) 5, 51,10
*
```

```
*  SIGN WAS NEGATIVE USE WIDTHS OF BLOCKS
*
    5 ITBLK= IABS(ITBLK)
      D1=WD(NBLOCK)
      D2=WD(ITBLK)
      GO TO 15
*
*  SIGN WAS POSITIVE USE LENGTHS OF BLOCKS
*
   10 D1=LEN(NBLOCK)
      D2=LEN(ITBLK)
*
*  FIND NO OF MATERIAL AND KM VALUE FOR TOUCHING BLOCK
*
   15 IORDER=1
      IMAT =MAT(ITBLK)
      I=IMAT *10-9
      NPRS=NKMPTS(IMAT)
      IF (NPRS.EQ.0) IORDER=0
      CALL FTLUP (TEMP(ITBLK ),VALK2,IORDER,NPRS,TTKTAB(I),KMTAB(I))
*
*  PICK UP NEXT CONDUCTION AREA IN ARRAY AND COMPUTE COEF.OF TEMPERATURE
*    DIFFERENCE
*
      SAVE =2.0*ACOND(NCDCT)/((D1/VALK1)+D2/VALK2)
      NCDCT = NCDCT+1
*
      IF (INVERSE.EQ.1) GO TO 90
*
      COEF(NBLOCK,1)=COEF(NBLOCK,1) -SAVE
      ISAVE=ITBLK-NBLOCK+1
      COEF(NBLOCK,ISAVE)=SAVE
      COEF(ITBLK,1) =COEF(ITBLK,1) -SAVE
*
      GO TO 50
*
*  STORAGE OF CONDUCTION TERMS FOR INVERSE SOLUTION
   90 HOLD=    TEMPINV*SAVE
*
      GO TO  (100,200,300,400,500) INVRT
*
  100 COEF(INVBLK,2)=SAVE
      COEF(INVBLK,INVBLK)= CONVRO(1,2)
      IF (INVBLK.NE.1) GO TO 150
*
*  BLOCK NO 1 IS INVERSE BLOCK
*
      CONST(1)=CONST(1) + HOLD +WVCP*(TEMPINV-TEMP(1))
      CONST(2)=-HOLD
      INVRT=5
      IF (NOINV.EQ.1) GO TO 550
      DO 105  L=2,NOINV
      INO=INVBLKS(L)
      COEF(1,INO) =COEF(1,INO)+SAVE+WVCP
  105 COEF (2,INO)=COEF(2,INO)-SAVE
      GO TO 550
*
  150 COEF (INVBLK,1)=COEF(INVBLK,1)-SAVE -WVCP
      COEF (1,1)=SAVE
*
      IF (INVBLK.EQ.2) GO TO 160
      COEF (1,2)= COEF(1,2)-SAVE
      CONST(INVBLK)=CONST(INVBLK)-WVCP*TEMP(NBLOCK)
      INVRT=3
      GO TO 50
*
```

```
*  BLOCK NO 2 IS INVERSE BLOCK
*
  160 CONST(2) = CONST(2) -HOLD-WVCP* TEMP(NBLOCK)
      CONST(1) = HOLD
      INVRT=2
      IF (NOINV.EQ.1) GO TO 50
      DO 165  L=2,NOINV
      INO=INVBLKS(L)
      COEF(1,INO) =COEF(1,INO)+SAVE
  165 COEF (2,INO)=COEF(2,INO)-SAVE
      GO TO 50
*
  200 COEF(1,3)=COEF(1,3)+SAVE
      CONST(1)=CONST(1) + HOLD +WVCP*(TEMPINV-TEMP(2))
      CONST(3) =-HOLD
      INVRT=5
      IF (NOINV.EQ.1) GO TO 550
      DO 205  L=2,NOINV
      INO=INVBLKS(L)
      COEF(1,INO) =COEF(1,INO)+SAVE+WVCP
  205 COEF (3,INO)=COEF(3,INO)-SAVE
      GO TO 550
  300 IF (ITBLK.EQ.INVBLK) GO TO 350
*
      COEF(NBLOCK-1,ITBLK)=COEF(NBLOCK-1,ITBLK)+SAVE
      COEF(NBLOCK,ITBLK)= COEF(NBLOCK,ITBLK)-SAVE
      CONST(NBLOCK-1)=CONST(NBLOCK-1)-WVCP*TEMP(NBLOCK)
      GO TO 360
*
  350 INVRT=4
      CONST(NBLOCK-1)=CONST(NBLOCK-1)-HOLD-WVCP*TEMP(NBLOCK)
      CONST(NBLOCK)=CONST(NBLOCK)+HOLD
      IF (NOINV.EQ.1) GO TO 360
      DO 355  L=2,NOINV
      INO=INVBLKS(L)
      COEF(NBLOCK-1,INO) =COEF(NBLOCK-1,INO)-SAVE
  355 COEF(NBLOCK,INO) = COEF(NBLOCK,INO)+SAVE
  360 COEF(NBLOCK-1,NBLOCK)= COEF (NBLOCK-1,NBLOCK)-SAVE-WVCP
      COEF(NBLOCK,NBLOCK)=COEF(NBLOCK,NBLOCK)+SAVE
      GO TO 50
*
  400 COEF(NBLOCK-1,ITBLK)=COEF(NBLOCK-1,ITBLK)+SAVE
      CONST(NBLOCK-1)= CONST(NBLOCK-1)+HOLD+WVCP*(TEMPINV-TEMP(NBLOCK))
      CONST(ITBLK) =-HOLD
      INVRT=5
      IF (NOINV.EQ.1) GO TO 550
      DO 450  L=2,NOINV
      INO=INVBLKS(L)
      COEF(NBLOCK-1,INO) =COEF(NBLOCK-1,INO)+SAVE+WVCP
  450 COEF(ITBLK,INO)= COEF(ITBLK,INO)-SAVE
      GO TO 550
*
  500 COEF(NBLOCK,ITBLK)=COEF(NBLOCK,ITBLK)+SAVE
      COEF(NBLOCK,NBLOCK)=COEF(NBLOCK,NBLOCK)-SAVE-WVCP
      COEF(ITBLK,NBLOCK)=COEF(ITBLK,NBLOCK)+SAVE
      CONST(NBLOCK)=CONST(NBLOCK)-WVCP*TEMP(NBLOCK)
  550 COEF(ITBLK,ITBLK) = COEF(ITBLK,ITBLK)-SAVE
*
   50 CONTINUE
   60 CONTINUE
*
      RETURN
   51 WRITE (6,52) NBLOCK
   52 FORMAT (4X35HERROR - TOUCHING BLOCK=0  BLOCK NO=I4)
      STOP
      END
```

RADOUT

Subroutine RADOUT computes the radiation-out term of the heat-balance equation and stores it in the constant vector for solution or it computes a linearized form and stores values in the coefficient matrix and in the constant vector for solution. The flow diagram for subroutine RADOUT follows.

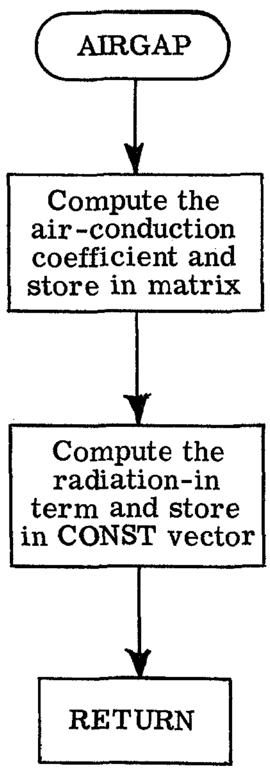


The program listing for subroutine RADOUT is

```
      SUBROUTINE RADOUT (COEF,MDUMMY,NDUMMY)
*
*        A SUBROUTINE TO CALCULATE THE RADIATION-OUT TERM EITHER LINEARIZED
*        OR NOT
*        THIS SUBROUTINE IS CALLED BY THE MAIN PROGRAM
*
      DIMENSION  COEF(MDUMMY,NDUMMY)
      COMMON  TIME,DELTAT,TIMSTOP,PRFREQ,NO,IDIM,ICONVQ,G,
     1TEMP(100),CONST(100)
      COMMON /ALLCONV/ CONVRO(30,5),IDIFF,NOCONV,QCONV(30),
     1TAW(30),IHORQ
      COMMON  /MATPROP/ MAT(100),EMS(40),RHO(10),NCPPTS(10),TCPTAB(100),
     1CPTAB(100),NKMPTS(10),TTKTAB(100),KMTAB(100)
      COMMON /INVRS/ INVERSE,INVBLK,TEMPINV,NOINV,INVBLKS(4),NOINVT,
     1TINV(200),TEMINV(400)
      COMMON  /RADONLY/NRADO,LINEAR
      REAL KMTAB
*
      IF(NRADO.EQ.0)     RETURN
      DO 20  I=1,NRADO
      IBLK= CONVRO(I,1)
      TCUBED= TEMP(IBLK)**3
      M= MAT(IBLK)
      J= M*4-3
*
      EMSS=EMS(J)+TEMP(IBLK)*(EMS(J+1)+TEMP(IBLK)*(EMS(J+2)+
     1TEMP(IBLK)*EMS(J+3)))
*
      HOLD = .4835E-12*EMSS* CONVRO(IBLK,2)* TCUBED
      HOLD1= HOLD*TEMP(IBLK)
      IF (LINEAR.EQ.0) GO TO 5
*
*   STORAGE FOR LINEAR EQ. (T**4)
*
      IF (INVERSE.EQ.0)  GO TO 4
      COEF(INVBLK,1)= COEF(INVBLK,1) -HOLD*2.0
      CONST(INVBLK)= CONST(INVBLK)-HOLD1
      GO TO 20
    4 COEF(IBLK,1) = COEF(IBLK,1) -HOLD*2.0
      CONST(IBLK) = CONST(IBLK)-HOLD1
      GO TO 20
*
*
    5 IF (INVERSE.EQ.0) GO TO 10
      IF (IBLK.NE.1)     GO TO 10
*
    6 CONST(INVBLK)=CONST(INVBLK)+HOLD1
      GO TO  20
*
   10 CONST(IBLK)=CONST(IBLK)+HOLD1
*
   20 CONTINUE
      RETURN
      END
```

AIRGAP

Subroutine AIRGAP computes the air-conduction coefficient and stores it in the coefficient matrix and also computes the radiation term across an air gap and stores this term in the constant vector for solution. The flow diagram for subroutine AIRGAP follows.

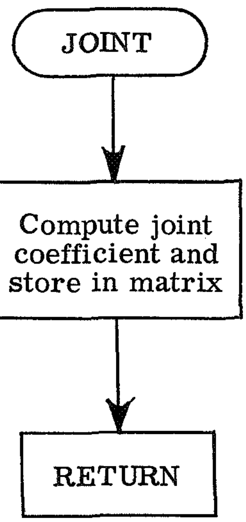


The program listing for subroutine AIRGAP is

```
      SUBROUTINE AIRGAP (COEF,MDUMMY,NDUMMY)
*
*        A SUBROUTINE TO CALCULATE THE AIR CONDUCTION AND THE RADIATION
*        ACROSS AN AIR GAP - CANNOT BE USED FOR INVERSE SOLUTION
*        THIS SUBROUTINE IS CALLED BY THE MAIN PROGRAM
*
      DIMENSION  COEF(MDUMMY,NDUMMY)
      COMMON   TIME,DELTAT,TIMSTOP,PRFREQ,NO,IDIM,ICONVQ,G,
     1TEMP(100),CONST(100)
      COMMON /AGONLY/   NAIRGAP,ACRADI(4,30)
      COMMON   /MATPROP/ MAT(100),EMS(40),RHO(10),NCPPTS(10),TCPTAB(100),
     1CPTAB(100),NKMPTS(10),TTKTAB(100),KMTAB(100)
      REAL KMTAB
      DIMENSION  ATTAB(9),AKMTAB(9)
      DATA  NAKMPT,ATTAB,AKMTAB /9,400.,600.,800.,1000.,1200.,1400.,
     11600.,1800.,2000.,3.18E-6,4.60E-6,5.87E-6,7.02E-6,8.07E-6,
     29.06E-6,9.94E-6,10.57E-6,10.87E-6 /
*
      DO  50   I=1,NAIRGAP
      NACBLK =ACRADI(1,I)
      CALL  FTLUP(TEMP(NACBLK),AK1,1,NAKMPT,ATTAB,AKMTAB)
      IOBLK= ACRADI(2,I)
      CALL  FTLUP  (TEMP(IOBLK),AK2,1,NAKMPT,ATTAB,AKMTAB)
      AIRK=(AK1+AK2)/2.0
*
      SAVE= AIRK * ACRADI(4,I)/ACRADI(3,I)
*
      COEF(NACBLK,1) =COEF(NACBLK,1)-SAVE
      INDEX=IOBLK-NACBLK+1
      IF (INDEX.GT.0)  GO TO 5
      INDEX= NACBLK-IOBLK+1
      COEF(IOBLK,INDEX)=COEF(IOBLK,INDEX)+SAVE
      GO TO 10
    5 COEF(NACBLK,INDEX)=SAVE +  COEF(NACBLK,INDEX)
   10 COEF(IOBLK,1) = COEF(IOBLK,1)- SAVE
*
*  COMPUTE RADIATION-IN ACROSS THE AIR GAP
*
      IMAT=MAT(NACBLK)
      J=IMAT*4-3
      EM1 =EMS(J)+TEMP(NACBLK)*(EMS(J+1)+TEMP(NACBLK)*(EMS(J+2)+
     1TEMP(NACBLK)* EMS(J+3)))
      IMAT=MAT(IOBLK)
      J=IMAT*4-3
      EM2 =EMS(J)+TEMP(IOBLK)*(EMS(J+1)+TEMP(IOBLK)*(EMS(J+2)+
     1TEMP(IOBLK)* EMS(J+3)))
*
*  STORAGE AS A CONSTANT PART OF THE EQUATION
*
      HOLD=.4835E-12*ACRADI(4,I)     /(1.0/ EM1 +1.0/EM2 -1.0)
      T1P4= TEMP(NACBLK)**4
      T2P4= TEMP(IOBLK) **4
      CONST(NACBLK)= CONST(NACBLK)+HOLD*(T1P4-T2P4)
      CONST(IOBLK) = CONST(IOBLK)+HOLD*(T2P4-T1P4)
   50 CONTINUE
      RETURN
      END
```

JOINT

Subroutine JOINT computes the joint conduction coefficient and stores it in the coefficient matrix. The flow diagram for subroutine JOINT follows.

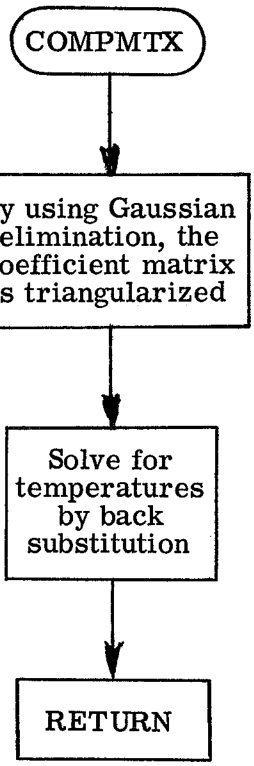


The program listing for subroutine JOINT is

```
      SUBROUTINE  JOINT (COEF,MDUMMY,NDUMMY)
*
*        A SUBROUTINE TO CALCULATE THE JOINT CONDUCTION TERM
*        THIS SUBROUTINE IS CALLED BY THE MAIN PROGRAM
*
      DIMENSION  COEF(MDUMMY,NDUMMY)
      COMMON  TIME,DELTAT,TIMSTOP,PRFREQ,NO,IDIM,ICONVQ,G,
     1TEMP(100),CONST(100)
      COMMON  /MATPROP/ MAT(100),EMS(40),RHO(10),NCPPTS(10),TCPTAB(100),
     1CPTAB(100),NKMPTS(10),TTKTAB(100),KMTAB(100)
      COMMON /BLKDIM/  WD(100),VOL(100),LEN(100),NHOR,NVER
      COMMON /JTONLY/  NJOINT,AJOINT(4,20)
      REAL KMTAB
      REAL LEN
*
* SET UP LOOP TO DO MORE THAN ONE JOINT TERM
*
      DO 50  J=1,NJOINT
*
      NBLOCK = AJOINT(1,J)
      JTBLK= AJOINT(2,J)
      IF (JTBLK) 5,20,10
*
* SIGN WAS NEGATIVE - USE WIDTHS
*
    5 JTBLK= IABS(JTBLK)
      D1=WD(NBLOCK)
      D2=WD(JTBLK)
      GO TO 15
*
* SIGN WAS POSITIVE - USE LENGTHS
*
   10 D1=LEN(NBLOCK)
      D2=LEN(JTBLK)
*
* FIND NO OF MATERIAL AND KM OF JOINT BLOCK
*
   15 IORDER=1
      IMAT = MAT(NBLOCK)
      I=IMAT*10-9
      NPRS=NKMPTS(IMAT)
      IF(NPRS.EQ.0)  IORDER=0
      CALLFTLUP(TEMP(NBLOCK),VALK1,IORDER,NPRS,TTKTAB(I),KMTAB(I))
*
      IORDER=1
      IMAT = MAT(JTBLK)
      I=IMAT*10-9
      NPRS=NKMPTS(IMAT)
      IF(NPRS.EQ.0)  IORDER=0
      CALL FTLUP(TEMP(JTBLK),VALK2,IORDER,NPRS,TTKTAB(I),KMTAB(I))
*
* PICK UP NEXT AREA AND HJ AND COMPUTE COEF.OF TEMPERATURE DIFFERENCE
*
      HOLD= 2.0*AJOINT(4,J)/(2.0/AJOINT(3,J)+D1/VALK1+D2/VALK2)
      COEF(NBLOCK,1)=COEF(NBLOCK,1)-HOLD
      COEF(JTBLK ,1)=COEF(JTBLK ,1)-HOLD
      INDEX = JTBLK-NBLOCK+1
      COEF(NBLOCK,INDEX) = HOLD
C
   50 CONTINUE
      RETURN
*
* ERROR
*
   20 WRITE (6, 100) NBLOCK
  100 FORMAT(1H1,42H JOINT BLOCK NO SHOULD NOT BE ZERO,NBLOCK=I6)
      CALL EXIT
*
      END
```

COMPMTX

Subroutine COMPMTX solves the symmetric coefficient matrix for temperatures. The technique used for solution is a Gaussian elimination with a back substitution. The flow diagram for subroutine COMPMTX follows.

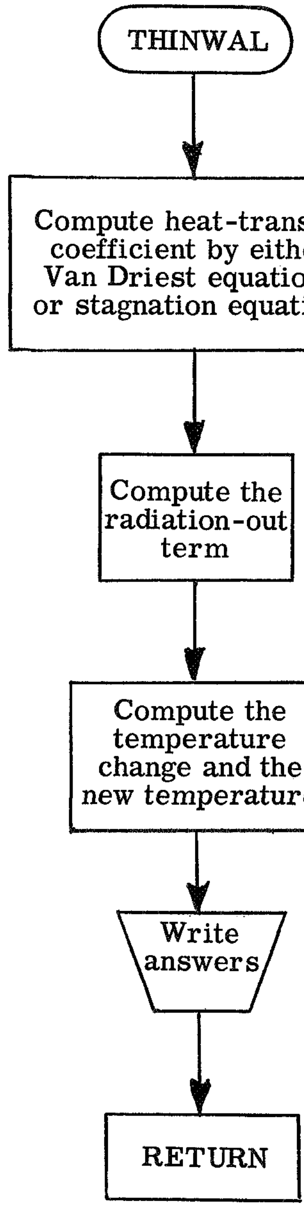


The program listing for subroutine COMPMTX is

```
      SUBROUTINE COMPMTX(COEF,MDUMMY,NDUMMY)
*
*        A SUBROUTINE WHICH SOLVES THE SYSTEM OF SIMULTANEOUS EQS.USING
*        A GAUSSIAN ELIMINATION ON THE SYMMETRICAL COEF. MATRIX - ONLY THE
*        UPPER PORTION OF THE MATRIX(ABOVE DIAGONAL) HAS BEEN STORED
*        THIS SUB. CANNOT BE USED FOR INVERSE AS IT IS NOT SYMMETRICAL
*        THIS SUBROUTINE IS CALLED BY THE MAIN PROGRAM
*
      DIMENSION  COEF(MDUMMY,NDUMMY)
      COMMON  TIME,DELTAT,TIMSTOP,PRFREQ,NO,IDIM,ICONVQ,G,
     1TEMP(100),CONST(100)
*
      ISTOP=NO-1
      IF (IDIM.EQ.1) GO TO 60
*
      DO 30  I=1,ISTOP
*
      DO 20  J=2,NDUMMY
*
      II= I+J-1
      IF (COEF(I,J).EQ.0.) GO TO 20
      M =1
*
      DO 10  K=J,NDUMMY
*
      IF (COEF(I,K).EQ.0.) GO TO 10
      COEF(II,M) =COEF(II,M) -COEF(I,K)*COEF(I,J)/COEF(I,1)
   10 M=M+1
*
      CONST(II)= CONST(II) - CONST(I)*COEF(I,J)/COEF(I,1)
   20 CONTINUE
   30 CONTINUE
*
*
      TEMP(NO)= CONST(NO)/COEF(NO,1)
      JSTOP= NDUMMY-1
*
      DO  50  I=1,ISTOP
      SUM =0.0
      M=NO-I
*
      DO 40 J=1,JSTOP
      IF ((M+J).GT.NO) GO TO 40
      SUM = SUM+ TEMP(M+J) * COEF(M,J+1)
   40 CONTINUE
*
   50 TEMP(M)= (CONST(M)-SUM)/COEF(M,1)
      GO TO  90
*
*
*  TRIANGULARIZATION AND SOLUTION OF ONE-DIMENSIONAL CONFIGURATION
*
   60 DO  70  I=2,NO
*
      II=I-1
      COEF(I,1)= COEF(I,1) - COEF(II,2)**2/COEF(II,1)
   70 CONST(I) = CONST(I)-CONST(II)* COEF(II,2) / COEF(II,1)
*
      TEMP(NO) = CONST(NO)/COEF(NO,1)
*
      DO 80  I=1,ISTOP
      M= NO-I
   80 TEMP(M)=(CONST(M)- COEF(M,2)*TEMP(M+1))/COEF(M,1)
*
   90 RETURN
      END
```

THINWAL

Subroutine THINWAL solves a simplified temperature equation by calling subroutine CONV for the convection term and subroutine RADOUT for the radiation-out term. The flow diagram for subroutine THINWAL follows.

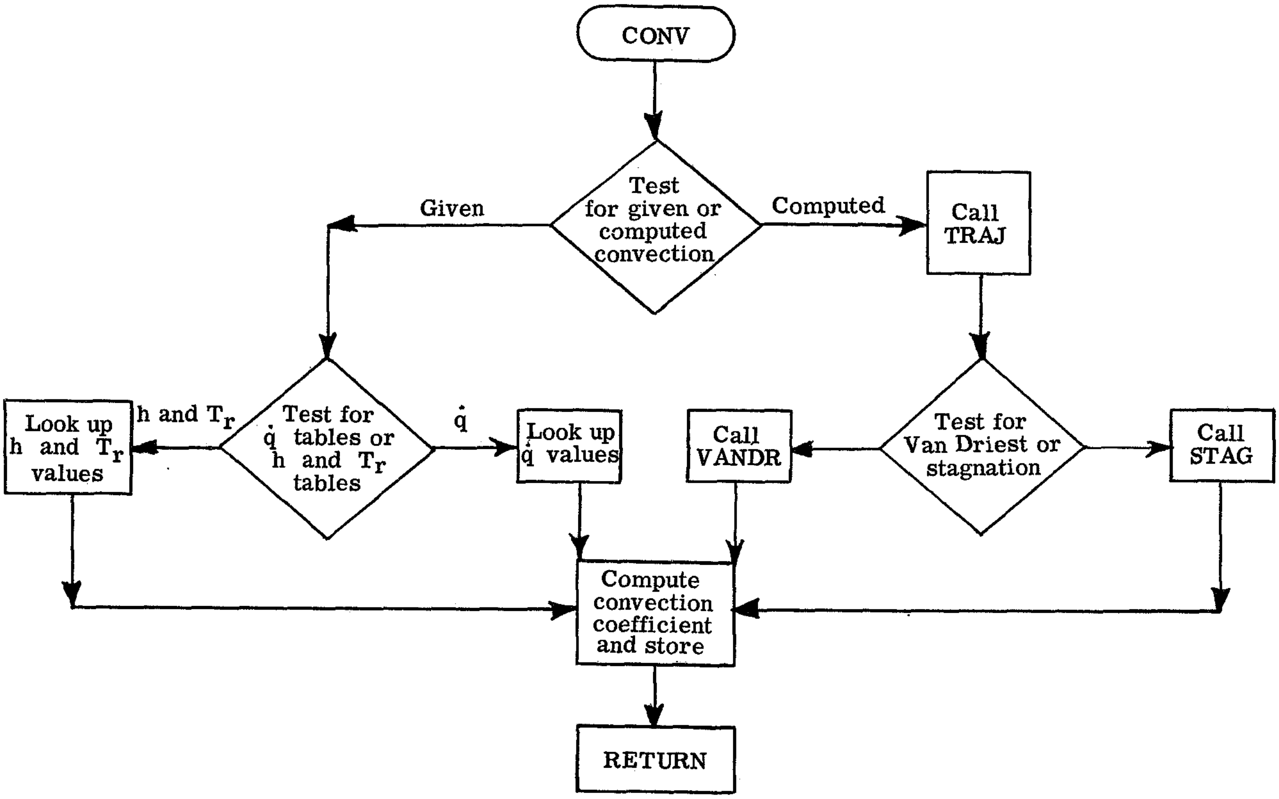


The program listing for subroutine THINWAL is

```
      SUBROUTINE THINWAL
*
*        A SUBROUTINE WHICH CALCULATES THIN WALL HEATING RATES AND TEMPS.
*        THIS SUBROUTINE IS CALLED BY THE MAIN PROGRAM
*
*
      COMMON  TIME,DELTAT,TIMSTOP,PRFREQ,NO,IDIM,ICONVQ,G,
     1TEMP(100),CONST(100)
      COMMON /THIN/  TWDEPTH
      COMMON  /MATPROP/ MAT(100),EMS(40),RHO(10),NCPPTS(10),TCPTAB(100),
     1CPTAB(100),NKMPTS(10),TTKTAB(100),KMTAB(100)
      COMMON /ALLCONV/ CONVRO(30,5),IDIFF,NOCONV,QCONV(30),
     1TAW(30),IHORQ
      REAL KMTAB
      INTEGER PRFREQ
*
      I=1
      IPFCT=0
    5 TIME= TIME+DELTAT
      CALL TRAJ
      IF (ICONVQ.EQ.3) GO TO 15
*
      CALL STAG(I,HVAL,TREC,HSUBDIF)
      GO TO (20,10),IHORQ
   10 CONVQ= HVAL *HSUBDIF
      GO TO 25
   15 CALL VANDR(I,HVAL,TREC)
   20 CONVQ= HVAL * (TREC-TEMP(1))
*
   25 IORDER=1
      IF (NCPPTS(1).EQ.0) IORDER=0
*
      CALL FTLUP (TEMP(1),CPVAL,IORDER,NCPPTS(1),TCPTAB(1),CPTAB(1))
      SAVE = DELTAT/(RHO(1)* CPVAL*TWDEPTH)
*
      IF (EMS(2).EQ.0..AND.EMS(3).EQ.0..AND.EMS(4).EQ.0.)  GO TO  30
*
      EMSVAL=EMS(1)+TEMP(1)*(EMS(2)+TEMP(1)*(EMS(3)+TEMP(1)*EMS(4)))
      GO TO 40
*
   30 EMSVAL= EMS(1)
*
   40 RADIAT= .4835E-12 *EMSVAL * TEMP**4
*
      DELTEMP= SAVE *(CONVQ -RADIAT)
*
      TEMP(1)= TEMP(1)+ DELTEMP
*
      IPFCT=IPFCT +1
      IF (IPFCT.NE.PRFREQ) GO TO 50
      IPFCT = 0
*
*  PRINT  ANSWERS
      WRITE (6,100) TIME,CONVQ,RADIAT,DELTEMP,TEMP(1)
  100 FORMAT (2X5HTIME=F8.4  ,3X12HCONVECTION= E15.8,3X11HRADIATION= E15
     1.8,3X12HDELTA TEMP= E15.8,3X13HTEMPERATURE= F15.8)
   50 IF(TIME.LT.TIMSTOP) GO TO 5
      RETURN
      END
```

CONV

Subroutine CONV computes the heating-rate coefficient from either tabular input values or from computed values obtained in either the VANDR subroutine or the STAG subroutine. The flow diagram for subroutine CONV follows.



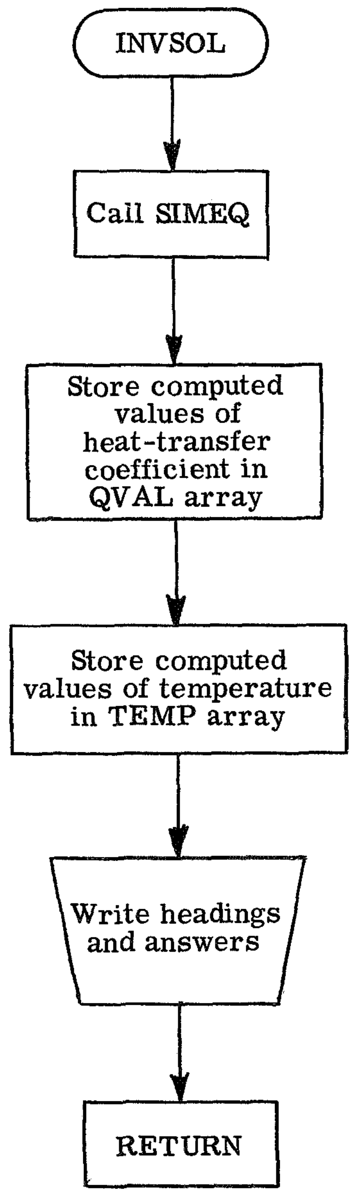
The program listing for subroutine CONV is

```
      SUBROUTINE   CONV (COEF,MDUMMY,NDUMMY)
*
*        A SUBROUTINE TO CALCULATE THE CONVECTION TERMS USING GIVEN Q HEAT
*        RATES OR H AND TAW VALUE OR IT USES HEATING RATES CALCULATED
*        BY EITHER STAG OR VANDR SUBROUTINES
*        THIS SUBROUTINE IS CALLED BY TWO SUBROUTINES - CONV AND THINWAL
*
      DIMENSION  COEF(MDUMMY,NDUMMY)
      COMMON   TIME,DELTAT,TIMSTOP,PRFREQ,NO,IDIM,ICONVQ,G,
     1TEMP(100),CONST(100)
      COMMON /ALLCONV/ CONVRO(30,5),IDIFF,NOCONV,QCONV(30),
     1TAW(30),IHORQ
      COMMON /CVONLY/ HITIME(200),HTAB(200),TAWTAB(200),QTAB(200)
      COMMON /CVTRAJ/NRATIO
*
      GO TO (5,10,10) ICONVQ
*
    5 LOCQ=1
      LOCH=1
      IORD=1
*
```

```
*   SET UP LOOP FOR MORE THAN ONE CONV. TERMS
*
    10 DO 100   I=1,NOCONV
       ICBLK = CONVRO(I,1)
       I=I
*
       GO TO (20,50,50) ICONVQ
*
*   GIVEN HEATING INPUT
*
    20 IQH =CONVRO(I,4)
       IPTS=CONVRO(I,5)
       IF (IPTS.GT.1) GO TO 25
       IORD=0
       IPTS=0
*
    25 IF (IQH.EQ.1) GO TO 35
*
*   H AND TAW TABLES
*
       CALL FTLUP(TIME,HVAL    ,IORD,IPTS,HITIME,HTAB(LOCH))
       CALL FTLUP(TIME,TREC      ,IORD,IPTS,HITIME,TAWTAB(LOCH))
*
       IF (IDIFF.EQ.0) GO TO 30
       IF(IPTS.EQ.0) IPTS=1
       LOCH=LOCH+IPTS
    30 IORD=1
       IHORQ = 1
       GO TO 80
*
*   Q TABLES
*
    35 CALL FTLUP(TIME,QCONV(I),IORD,IPTS,HITIME,QTAB(LOCQ))
       IHORQ = 2
*
       IF (IDIFF.EQ.0) GO TO 40
       IF (IPTS.EQ.0) IPTS=1
       LOCQ=LOCQ+IPTS
    40 IORD=1
       QCONV(I)=QCONV(I)*CONVRO(I,2)
       CONST(ICBLK)=CONST(ICBLK)-QCONV(I)
       GO TO 100
*
*   STAG. AND VAN DRIEST HEATING
*
    50 NRATIO=CONVRO(I,4)
       CALL  TRAJ
       GO TO (105,60,70) ICONVQ
    60 CALL STAG(I,HVAL,TREC,HSUBDIF)
*
       GO TO (80,90) IHORQ
*
    70 CALL VANDR(I,HVAL,TREC)
*
    80 QCONV(I)= HVAL*CONVRO(I,2)
       TAW(I)= TREC
       COEF(ICBLK,1)=-QCONV(I)+COEF(ICBLK,1)
       CONST(ICBLK)=CONST(ICBLK)-QCONV(I)*TREC
*
       GO TO 100
*
    90 QCONV(I)=HVAL*CONVRO(I,2)*HSUBDIF
       CONST(ICBLK)=CONST(ICBLK)-QCONV(I)
   100 CONTINUE
       RETURN
*
   105 WRITE (6,110)
   110 FORMAT (1H1,46H ERROR IN ICONVQ-SHOULD NOT BE 1 AT THIS POINT)
       STOP
       END
```

INVSOL

Subroutine INVSOL solves the coefficient matrix by subroutine SIMEQ and then stores and writes out the computed values of the temperatures and the heating rates. The flow diagram for subroutine INVSOL follows.



The program listing for subroutine INVSOL is

```
      SUBROUTINE INVSOL (COEF,MDUMMY)
*
*        A SUBROUTINE WHICH HANDLES SOLUTION AND STORAGE OF ANSWERS FOR THE
*        INVERSE SOLUTIONS - IT USES SIMEQ(A LIBRARY ROUTINE) FOR THE
*        SOLUTION OF THE SIMULTANEOUS EQUATIONS
*        THIS SUBROUTINE IS CALLED BY THE MAIN PROGRAM
*
      DIMENSION COEF(MDUMMY,MDUMMY),IPIVOT(30)
      COMMON  TIME,DELTAT,TIMSTOP,PRFREQ,NO,IDIM,ICONVQ,G,
     1TEMP(100),CONST(100)
      COMMON /INVRS/ INVERSE,INVBLK,TEMPINV,NOINV,INVBLKS(4),NOINVT,
     1TINV(200),TEMINV(400)
*
      CALL SIMEQ(COEF,MDUMMY,CONST,1,DUM,IPIVOT,MDUMMY,ISC)
*
      QVAL= CONST(INVBLK)
*
      DO 10  I=1,MDUMMY
*
   10 TEMP(I)=CONST(I)
      IF (NOINV.GT.1) GO TO 20
      TEMP(INVBLK)= TEMPINV
      GO TO 40
*
   20 SUM=0.0
      DO 30  J=2,NOINV
      L= INVBLKS(J)
   30 SUM= SUM + CONST(L)
      TEMP(INVBLK) = TEMPINV -SUM
*
   40 WRITE (6,50) TIME,DELTAT,QVAL
   50 FORMAT (1H0,6X7HTIME = F8.4,6X13HDELTA TIME = F8.5,6X4HQ = E16.8//
     19X8HBLOCK NO,13X11HTEMPERATURE/)
*
      WRITE  (6,60) (I,TEMP(I),I=1,NO)
   60 FORMAT      (12XI3,12XE16.8)
      RETURN
      END
```

BLOCK DATA

Subroutine BLOCK DATA contains the tables for normal-shock-wave parameters in imperfect air. The program listing for BLOCK DATA follows:

```
      BLOCK DATA
*
*   DATA STATEMENT INFORMATION
*        TABLES FOR NORMAL SHOCK WAVE PARAMETERS IN IMPERFECT AIR
*
*   ATAB    =   ALTITUDE TABLE   (10000FT-400000FT)
*   VTAB    =   VELOCITY TABLE   (6000-50000 FT/SEC)
*   ARRAY1 =   POP/P1 , PITOT TO FREE-STREAM PRESSURE RATIOS
*   ARRAY2 =   ROP/R1 , NORMAL SHOCK STAGNATION DENSITY RATIOS
*   ARRAY3 =   TOP/T1 , NORMAL SHOCK STAGNATION TEMPERATURE RATIOS
*
      COMMON /ARRAYS/ ARRAY1(1800),ARRAY2(1800),ARRAY3(1800)
     1,ATAB(40),VTAB(45)
      DATA ((ARRAY1(I), I= 1,90)=
     X40.829,  55.548,  72.670,  92.148,  114.038, 138.280, 164.896,
     X193.791, 224.864, 258.265, 293.908, 332.066, 372.598, 415.635,
     X461.145, 508.870, 558.998, 611.397, 666.274, 723.309, 782.746,
     X844.403, 908.371, 974.556, 1043.079,1113.746,1186.627,1261.598,
     X1338.723,0.0,    0.0,     0.0,     0.0,     0.0,     0.0,
     X0.0,     0.0,    0.0,     0.0,     0.0,     0.0,     0.0,
     X0.0,     0.0,    0.0,
     X44.034,  59.925,  78.415,  99.452,  123.093, 149.303, 178.025,
     X209.256, 242.816, 278.919, 317.404, 358.552, 402.476, 448.928,
     X498.001, 549.593, 603.781, 660.481, 719.679, 781.359, 845.530,
     X912.226, 981.238, 1052.855,1126.818,1203.137,1281.751,1362.677,
     X1445.974,0.0,    0.0,     0.0,     0.0,     0.0,     0.0,
     X0.0,     0.0,    0.0,     0.0,     0.0,     0.0,     0.0,
     X0.0,     0.0,    0.0)
      DATA ((ARRAY1(I), I= 91,180)=
     X47.787,  65.054,  85.145,  108.006, 133.702, 162.244, 193.459,
     X227.358, 263.933, 303.082, 344.944, 389.806, 437.498, 487.986,
     X541.330, 597.555, 656.392, 718.058, 782.389, 849.449, 919.198,
     X991.700, 1066.742,1144.459,1224.906,1307.860,1393.343,1481.233,
     X1571.628,0.0,    0.0,     0.0,     0.0,     0.0,     0.0,
     X0.0,     0.0,    0.0,     0.0,     0.0,     0.0,     0.0,
     X0.0,     0.0,    0.0,
     X50.435,  68.672,  89.896,  114.047, 141.228, 171.420, 204.449,
     X240.248, 278.810, 320.187, 364.587, 411.791, 462.281, 515.684,
     X572.104, 631.419, 693.661, 758.842, 826.859, 897.682, 971.509,
     X1048.037,1127.402,1209.658,1294.578,1382.184,1472.453,1565.254,
     X1660.735,1759.011,0.0,    0.0,     0.0,     0.0,     0.0,
     X0.0,     0.0,    0.0,     0.0,     0.0,     0.0,     0.0,
     X0.0,     0.0,    0.0)
      DATA ((ARRAY1(I), I= 181,270)=
     X50.437,  68.677,  89.911,  114.081, 141.332, 171.528, 204.551,
     X240.449, 279.033, 320.398, 364.700, 412.208, 462.634, 516.143,
     X572.582, 631.966, 694.372, 759.482, 827.592, 898.509, 972.303,
     X1049.014,1128.436,1210.732,1295.692,1383.363,1473.529,1566.326,
     X1661.780,1760.113,0.0,    0.0,     0.0,     0.0,     0.0,
     X0.0,     0.0,    0.0,     0.0,     0.0,     0.0,     0.0,
     X0.0,     0.0,    0.0,
     X50.440,  68.682,  89.924,  114.124, 141.434, 171.635, 204.726,
     X240.645, 279.260, 320.637, 364.911, 412.412, 462.976, 516.619,
     X573.057, 632.499, 694.922, 760.266, 828.301, 899.295, 973.160,
     X1049.867,1129.443,1211.713,1296.792,1384.372,1474.581,1567.308,
     X1662.737,1761.119,1862.512,0.0,   0.0,     0.0,     0.0,
     X0.0,     0.0,    0.0,     0.0,     0.0,     0.0,     0.0,
     X0.0,     0.0,    0.0)
```

```
 DATA ((ARRAY1(I), I= 271,360)=
X50.152,  68.290,  89.420,  113.516, 140.698, 170.743, 203.720,
X239.414, 277.779, 318.913, 363.117, 410.313, 460.617, 513.994,
X570.257, 629.449, 691.475, 756.406, 824.290, 894.841, 968.341,
X1044.678,1123.803,1205.705,1290.241,1377.356,1466.981,1559.135,
X1654.045,1751.944,1852.888,0.0,     0.0,     0.0,     0.0,
X0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
X0.0,     0.0,     0.0,
X49.474,  67.366,  88.220,  112.028, 138.843, 168.567, 201.085,
X236.285, 274.113, 314.690, 358.361, 405.105, 454.747, 507.316,
X562.929, 621.313, 682.594, 746.681, 813.608, 883.387, 955.876,
X1031.237,1109.353,1190.166,1273.543,1359.447,1447.754,1538.632,
X1632.322,1728.973,1828.673,0.0,     0.0,     0.0,     0.0,
X0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
X0.0,     0.0,     0.0)
 DATA ((ARRAY1(I), I= 361,450)=
X48.815,  66.469,  87.056,  110.596, 137.070, 166.455, 198.560,
X233.326, 270.589, 310.666, 353.730, 399.816, 448.903, 500.871,
X555.749, 613.479, 673.899, 737.182, 803.274, 872.124, 943.730,
X1018.138,1095.255,1174.999,1257.270,1341.928,1429.008,1518.669,
X1611.153,1706.636,1805.129,1906.603,0.0,     0.0,     0.0,
X0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
X0.0,     0.0,     0.0,
X48.175,  65.596,  85.927,  109.200, 135.368, 164.349, 196.068,
X230.331, 267.159, 306.673, 349.215, 394.784, 443.275, 494.668,
X548.810, 605.734, 665.503, 727.947, 793.185, 861.172, 931.874,
X1005.355,1081.502,1160.201,1241.363,1324.846,1410.718,1499.204,
X1590.384,1684.908,1782.249,1882.493,1985.600,0.0,     0.0,
X0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
X0.0,     0.0,     0.0)
 DATA ((ARRAY1(I), I= 451,540)=
X47.061,  64.077,  83.953,  106.720, 132.313, 160.635, 191.616,
X225.124, 261.051, 299.630, 341.310, 385.868, 433.261, 483.390,
X536.348, 591.967, 650.382, 711.437, 775.199, 841.615, 910.721,
X982.527, 1056.900,1133.797,1213.049,1294.512,1378.317,1464.777,
X1554.108,1646.391,1741.585,1839.612,1940.455,2044.039,0.0,
X0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
X0.0,     0.0,     0.0,
X45.434,  61.859,  81.064,  103.071, 127.807, 155.176, 185.048,
X217.418, 252.052, 289.329, 329.561, 372.613, 418.347, 466.819,
X517.959, 571.721, 628.050, 687.037, 748.588, 812.760, 879.466,
X948.807, 1020.637,1094.816,1171.253,1249.807,1330.637,1414.137,
X1500.484,1589.654,1681.638,1776.385,1873.788,1973.874,2076.634,
X0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
X0.0,     0.0,     0.0)
 DATA ((ARRAY1(I), I= 541,630)=
X43.920,  59.794,  78.377,  99.670,  123.601, 150.053, 178.975,
X210.198, 243.667, 279.731, 318.603, 360.252, 404.547, 451.417,
X500.808, 552.825, 607.306, 664.258, 723.805, 785.817, 850.342,
X917.355, 986.777, 1058.467,1132.265,1208.087,1286.188,1366.979,
X1450.487,1536.786,1625.789,1717.440,1811.678,1908.474,2007.863,
X2109.758,0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
X0.0,     0.0,     0.0,
X42.505,  57.868,  75.869,  96.498,  119.664, 145.260, 173.280,
X203.474, 235.812, 270.744, 308.393, 348.723, 391.613, 436.931,
X484.826, 535.093, 587.851, 642.989, 700.602, 760.668, 823.080,
X887.967, 955.124, 1024.473,1095.803,1169.103,1244.670,1322.879,
X1403.816,1487.403,1573.615,1662.382,1753.611,1847.367,1943.577,
X2042.239,2143.298,0.0,     0.0,     0.0,     0.0,     0.0,
X0.0,     0.0,     0.0)
```

```
 DATA ((ARRAY1(I), I= 631,720)=
X41.182,  56.067,  73.526,  93.529,  115.974, 140.798, 167.908,
X197.175, 228.431, 262.265, 298.849, 337.947, 379.482, 423.452,
X469.795, 518.505, 569.599, 623.024, 678.877, 737.066, 797.591,
X860.420, 925.464, 992.608, 1061.628,1132.570,1205.790,1281.624,
X1360.101,1441.164,1524.752,1610.769,1699.224,1790.093,1883.347,
X1978.953,2076.910,2177.210,0.0,     0.0,     0.0,     0.0,
X0.0,     0.0,     0.0,
X40.508,  55.155,  72.349,  92.037,  114.136, 138.567, 165.234,
X193.990, 224.692, 258.017, 294.024, 332.515, 373.388, 416.677,
X462.243, 510.228, 560.444, 613.059, 668.020, 725.218, 784.775,
X846.600, 910.570, 976.585, 1044.412,1114.127,1186.181,1260.838,
X1338.123,1417.921,1500.205,1584.854,1671.910,1761.329,1853.103,
X1947.208,2043.580,2142.295,2243.314,0.0,     0.0,     0.0,
X0.0,     0.0,     0.0)
 DATA ((ARRAY1(I), I= 721,810)=
X40.509,  55.164,  72.378,  92.089,  114.216, 138.657, 165.342,
X194.051, 224.765, 258.158, 294.181, 332.680, 373.616, 416.886,
X462.506, 510.453, 560.766, 613.401, 668.435, 725.618, 785.168,
X847.025, 910.996, 977.000, 1044.771,1114.468,1186.565,1261.317,
X1338.687,1418.584,1500.907,1585.604,1672.730,1762.263,1854.105,
X1948.258,2044.732,2143.525,2244.651,2348.002,2453.712,0.0,
X0.0,     0.0,     0.0,
X41.227,  56.155,  73.698,  93.788,  116.330, 141.218, 168.393,
X197.594, 228.908, 262.914, 299.624, 338.837, 380.509, 424.560,
X471.075, 519.893, 571.133, 624.733, 680.702, 738.983, 799.685,
X862.667, 927.844, 994.985, 1063.924,1134.847,1208.308,1284.498,
X1363.358,1444.776,1528.654,1614.961,1703.776,1794.990,1888.535,
X1984.440,2082.754,2183.359,2286.311,2391.627,2499.293,2609.296,
X0.0,     0.0,     0.0)
 DATA ((ARRAY1(I), I= 811,900)=
X42.163,  57.445,  75.415,  95.995,  119.060, 144.535, 172.322,
X202.178, 234.256, 269.074, 306.673, 346.780, 389.431, 434.567,
X482.147, 532.133, 584.530, 639.412, 696.622, 756.354, 818.482,
X882.945, 949.622, 1018.289,1088.766,1161.303,1236.541,1314.601,
X1395.376,1478.744,1564.636,1653.070,1743.986,1837.308,1933.078,
X2031.255,2131.840,2234.834,2340.210,2447.994,2558.148,2670.728,
X2785.623,0.0,     0.0,
X43.142,  58.797,  77.212,  98.304,  121.911, 147.990, 176.452,
X206.986, 239.822, 275.517, 313.996, 355.094, 398.793, 445.027,
X493.709, 544.913, 598.585, 654.716, 713.418, 774.584, 838.184,
X904.161, 972.356, 1042.641,1114.735,1188.981,1266.097,1346.105,
X1428.850,1514.249,1602.278,1692.847,1785.893,1881.504,1979.578,
X2080.132,2183.172,2288.672,2396.623,2507.046,2619.874,2735.070,
X2852.915,2973.257,3095.966)
 DATA ((ARRAY1(I), I= 901,990)=
X45.015,  61.375,  80.622,  102.659, 127.320, 154.569, 184.294,
X216.149, 250.386, 287.748, 327.910, 370.915, 416.560, 464.803,
X515.679, 569.163, 625.248, 683.866, 745.192, 809.097, 875.452,
X944.356, 1015.574,1088.963,1164.153,1241.701,1322.299,1405.916,
X1492.355,1581.613,1673.567,1768.183,1865.387,1965.235,2067.749,
X2172.843,2280.488,2390.730,2503.628,2618.919,2736.765,2857.223,
X2980.348,3106.147,3234.352,
X47.308,  64.533,  84.793,  107.978, 133.923, 162.610, 193.878,
X227.384, 263.357, 302.707, 345.005, 390.259, 438.209, 488.986,
X542.520, 598.818, 657.809, 719.506, 783.987, 851.204, 921.027,
X993.556, 1068.484,1145.599,1224.570,1306.165,1391.041,1479.049,
X1570.042,1664.018,1760.830,1860.376,1962.672,2067.792,2175.679,
X2286.254,2399.531,2515.546,2634.238,2755.570,2879.559,3006.329,
X3135.901,3268.127,3403.187)
```

```
 DATA ((ARRAY1(I), I= 991,1080)=
X49.848,  68.031,  89.413,  113.869, 141.248, 171.510, 204.491,
X239.794, 277.756, 319.289, 363.926, 411.631, 462.236, 515.831,
X572.271, 631.673, 693.856, 758.960, 826.966, 897.815, 971.564,
X1048.048,1127.083,1208.317,1291.494,1377.615,1467.228,1560.136,
X1656.203,1755.374,1857.527,1962.585,2070.587,2181.486,2295.266,
X2411.900,2531.420,2653.712,2778.956,2906.912,3037.713,3171.562,
X3308.126,3447.770,3589.705,
X52.679,  71.925,  94.565,  120.444, 149.424, 181.431, 216.318,
X253.602, 293.820, 337.726, 385.001, 435.417, 489.022, 545.698,
X605.442, 668.230, 734.043, 802.897, 874.879, 949.835, 1027.860,
X1108.776,1192.363,1278.169,1366.080,1457.232,1552.135,1650.507,
X1752.210,1857.180,1965.294,2076.449,2190.728,2308.101,2428.493,
X2551.881,2678.324,2807.719,2940.156,3075.575,3214.069,3355.639,
X3500.234,3647.849,3798.275)
 DATA ((ARRAY1(I), I= 1081,1170)=
X55.852,  76.292,  100.342, 127.814, 158.586, 192.542, 229.555,
X269.039, 311.799, 358.406, 408.631, 462.183, 519.052, 579.216,
X642.616, 709.235, 779.080, 852.196, 928.578, 1008.186,1090.988,
X1176.855,1265.520,1356.474,1449.667,1546.508,1647.336,1751.821,
X1859.858,1971.301,2086.094,2204.115,2325.433,2450.014,2577.821,
X2708.773,2843.012,2980.386,3120.949,3264.751,3411.804,3562.189,
X3715.611,3872.297,4032.001,
X59.435,  81.221,  106.869, 136.137, 168.931, 205.105, 244.513,
X286.483, 332.072, 381.839, 435.317, 492.372, 552.912, 617.080,
X684.577, 755.563, 829.941, 907.817, 989.220, 1074.028,1162.273,
X1253.693,1348.094,1444.861,1544.026,1647.299,1754.801,1866.219,
X1981.360,2100.152,2222.481,2348.253,2477.498,2610.232,2746.386,
X2885.942,3028.975,3175.342,3325.141,3478.282,3634.986,3795.271,
X3958.822,4125.702,4296.070)
 DATA ((ARRAY1(I), I= 1171,1260)=
X60.385,  82.559,  108.664, 138.436, 171.767, 208.536, 248.601,
X291.161, 337.580, 388.209, 442.599, 500.621, 562.224, 627.327,
X696.029, 768.177, 843.864, 922.956, 1005.678,1091.955,1181.662,
X1274.610,1370.481,1468.654,1569.406,1674.527,1783.949,1897.317,
X2014.385,2135.170,2259.576,2387.557,2518.980,2653.919,2792.391,
X2934.291,3079.663,3228.554,3380.953,3536.714,3695.931,3858.611,
X4024.869,4194.950,4368.029,
X60.389,  82.602,  108.752, 138.555, 171.905, 208.705, 248.753,
X291.245, 337.779, 388.391, 442.851, 500.968, 562.594, 627.747,
X696.401, 768.622, 844.387, 923.616, 1006.250,1092.539,1182.295,
X1275.333,1371.142,1469.144,1569.938,1675.250,1784.810,1898.324,
X2015.547,2136.355,2260.785,2388.898,2520.497,2655.566,2794.128,
X2936.155,3081.660,3230.591,3383.048,3539.021,3698.597,3861.424,
X4027.656,4197.444,4370.944)
 DATA ((ARRAY1(I), I= 1261,1350)=
X60.394,  82.648,  108.823, 138.672, 172.023, 208.851, 248.876,
X291.360, 337.951, 388.663, 443.177, 501.267, 562.927, 628.109,
X696.814, 769.000, 844.802, 924.192, 1006.896,1093.139,1182.894,
X1275.901,1371.771,1469.602,1570.469,1675.993,1785.690,1899.305,
X2016.677,2137.635,2262.087,2390.153,2521.917,2657.057,2795.735,
X2937.900,3083.657,3232.671,3385.135,3541.175,3700.914,3864.102,
X4030.684,4200.578,4373.841,
X59.003,  80.789,  106.387, 135.575, 168.142, 204.131, 243.185,
X284.646, 330.221, 379.825, 433.041, 489.848, 550.116, 613.806,
X680.912, 751.477, 825.425, 902.950, 983.926, 1068.228,1155.892,
X1246.673,1340.330,1435.641,1534.287,1637.561,1744.875,1855.945,
X1970.661,2088.925,2210.640,2335.731,2464.398,2596.428,2731.933,
X2870.793,3013.211,3159.073,3308.197,3460.457,3616.430,3775.875,
X3938.951,4105.511,4274.803)
```

```
      DATA ((ARRAY1(I), I= 1351,1440)=
     X56.262,  77.076,  101.510, 129.330, 160.403, 194.677, 231.856,
     X271.342, 314.861, 362.186, 412.949, 467.120, 524.535, 585.303,
     X649.251, 716.476, 787.020, 860.802, 937.979, 1018.439,1102.061,
     X1188.624,1277.729,1368.368,1462.533,1561.102,1663.515,1769.519,
     X1878.897,1991.654,2107.748,2227.156,2349.773,2475.625,2604.771,
     X2737.106,2872.732,3011.651,3153.968,3299.400,3447.981,3599.819,
     X3755.096,3913.777,4075.768,
     X53.772,  73.704,  97.078,  123.646, 153.311, 186.093, 221.531,
     X259.241, 300.868, 346.155, 394.689, 446.366, 501.297, 559.297,
     X620.472, 684.672, 752.032, 822.592, 896.233, 973.114, 1053.019,
     X1135.743,1220.766,1307.168,1397.303,1491.578,1589.466,1690.764,
     X1795.365,1903.132,2014.043,2128.139,2245.337,2365.638,2489.027,
     X2615.503,2745.075,2877.672,3013.506,3152.509,3294.637,3439.885,
     X3588.090,3739.456,3894.296)
      DATA ((ARRAY1(I), I= 1441,1530)=
     X51.223,  70.243,  92.513,  117.803, 146.094, 177.230, 210.927,
     X246.818, 286.545, 329.670, 375.864, 425.111, 477.329, 532.603,
     X590.864, 652.025, 716.111, 783.298, 853.463, 926.598, 1002.697,
     X1081.398,1162.271,1244.406,1330.355,1420.219,1513.454,1609.884,
     X1709.510,1812.181,1917.810,2026.415,2138.023,2252.622,2370.114,
     X2490.587,2613.956,2740.268,2869.518,3001.701,3137.069,3275.431,
     X3416.751,3560.978,3708.129,
     X47.870,  65.667,  86.475,  110.078, 136.464, 165.618, 196.952,
     X230.474, 267.664, 307.904, 351.077, 397.068, 445.831, 497.383,
     X551.750, 608.929, 668.765, 731.448, 796.985, 865.275, 936.327,
     X1009.763,1085.156,1161.770,1242.134,1326.127,1413.244,1503.291,
     X1596.302,1692.191,1790.864,1892.261,1996.481,2103.480,2213.191,
     X2325.659,2440.907,2558.979,2679.698,2803.154,2929.320,3058.489,
     X3190.531,3325.316,3462.808)
      DATA ((ARRAY1(I), I= 1531,1620)=
     X44.942,  61.663,  81.192,  103.329, 128.057, 155.370, 184.722,
     X216.200, 251.122, 288.874, 329.362, 372.484, 418.257, 466.591,
     X517.575, 571.239, 627.340, 686.101, 747.573, 811.622, 878.275,
     X947.126, 1017.712,1089.515,1164.999,1243.843,1325.608,1410.088,
     X1497.266,1587.243,1679.790,1774.955,1872.691,1973.017,2075.948,
     X2181.401,2289.512,2400.236,2513.529,2629.373,2747.739,2868.734,
     X2992.617,3119.063,3248.136,
     X42.363,  58.135,  76.526,  97.396,  120.705, 146.350, 173.930,
     X203.620, 236.508, 272.069, 310.219, 350.839, 393.887, 439.431,
     X487.415, 537.917, 590.793, 646.123, 703.982, 764.311, 827.034,
     X891.869, 958.214, 1025.802,1096.968,1171.253,1248.272,1327.859,
     X1409.949,1494.673,1581.848,1671.449,1763.445,1857.961,1954.873,
     X2054.158,2155.919,2260.179,2366.861,2475.939,2587.411,2701.360,
     X2817.990,2937.139,3058.678)
      DATA ((ARRAY1(I), I= 1621,1710)=
     X38.178,  52.389,  68.932,  87.686,  108.623, 131.749, 156.429,
     X183.192, 212.777, 244.795, 279.076, 315.623, 354.338, 395.261,
     X438.415, 483.831, 531.383, 581.104, 633.185, 687.405, 743.779,
     X802.093, 861.629, 922.410, 986.496, 1053.336,1122.628,1194.207,
     X1268.055,1344.197,1422.625,1503.210,1585.974,1670.928,1758.089,
     X1847.382,1939.852,2032.581,2128.539,2226.634,2326.928,2429.400,
     X2534.259,2641.410,2750.733,
     X.000,    .000,    .000,    .000,    98.341,  119.298, 141.545,
     X165.814, 192.609, 221.587, 252.582, 285.627, 320.680, 357.707,
     X396.732, 437.814, 480.784, 525.772, 572.856, 621.950, 672.968,
     X725.685, 779.457, 834.472, 892.517, 953.019, 1015.726,1080.514,
     X1147.353,1216.241,1287.173,1360.081,1434.930,1511.788,1590.604,
     X1671.369,1754.139,1838.956,1925.746,2014.483,2105.200,2197.979,
     X2292.931,2389.870,2488.785)
```

```
 DATA ((ARRAY1(I), I= 1711,1800)=
X.000,     0.0,      .000,     .000,     .000,     108.989, 129.273,
X151.476, 175.976, 202.424, 230.727, 260.910, 292.891, 326.704,
X362.342, 399.801, 439.044, 480.164, 523.171, 567.977, 614.565,
X662.658, 711.658, 761.929, 814.985, 870.265, 927.545, 986.709,
X1047.757,1110.697,1175.446,1241.962,1310.312,1380.457,1452.399,
X1526.171,1601.811,1679.252,1758.451,1839.460,1922.361,2007.264,
X2094.014,2182.461,2272.705,
X0.0,      0.0,      0.0,      .000,     .000,     .000,     113.227,
X132.383,153.849,177.10,202.01,228.541,256.546,286.134,
X317.244, 350.088, 384.425, 420.452, 458.113, 497.384, 538.077,
X580.172, 622.981, 667.059, 713.544, 761.968, 812.106, 863.919,
X917.374, 972.465, 1029.105,1087.340,1147.173,1208.581,1271.583,
X1336.203,1402.472,1470.266,1539.601,1610.492,1683.113,1757.460,
X1833.418,1910.861,1989.757)
 DATA ((ARRAY2(I), I= 1,90)=
X6.259,   6.737,   7.199,   7.678,   8.176,   8.662,   9.091,
X9.440,   9.666,   9.822,   9.975,   10.181,  10.431,  10.710,
X10.998,  11.276,  11.541,  11.785,  12.009,  12.204,  12.373,
X12.511,  12.617,  12.690,  12.729,  12.730,  12.697,  12.631,
X12.543,  0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
X0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
X0.0,     0.0,     0.0,
X6.318,   6.788,   7.247,   7.733,   8.247,   8.751,   9.194,
X9.545,   9.776,   9.929,   10.083,  10.296,  10.564,  10.855,
X11.153,  11.443,  11.721,  11.976,  12.207,  12.410,  12.583,
X12.726,  12.833,  12.906,  12.940,  12.935,  12.889,  12.807,
X12.703,  0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
X0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
X0.0,     0.0,     0.0)
 DATA ((ARRAY2(I), I= 91,180)=
X6.376,   6.837,   7.294,   7.789,   8.320,   8.846,   9.303,
X9.652,   9.894,   10.038,  10.195,  10.423,  10.703,  11.008,
X11.321,  11.627,  11.913,  12.179,  12.419,  12.629,  12.808,
X12.955,  13.064,  13.135,  13.165,  13.151,  13.091,  12.992,
X12.869,  0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
X0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
X0.0,     0.0,     0.0,
X6.414,   6.869,   7.329,   7.839,   8.396,   8.947,   9.425,
X9.782,   10.016,  10.153,  10.320,  10.558,  10.859,  11.184,
X11.513,  11.833,  12.134,  12.413,  12.663,  12.882,  13.069,
X13.219,  13.331,  13.401,  13.425,  13.398,  13.320,  13.199,
X13.056,  12.918,  0.0,     0.0,     0.0,     0.0,     0.0,
X0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
X0.0,     0.0,     0.0)
 DATA ((ARRAY2(I), I= 181,270)=
X6.418,   6.876,   7.346,   7.881,   8.473,   9.052,   9.549,
X9.918,   10.146,  10.272,  10.445,  10.712,  11.033,  11.381,
X11.729,  12.067,  12.387,  12.677,  12.940,  13.169,  13.364,
X13.520,  13.634,  13.701,  13.716,  13.673,  13.572,  13.424,
X13.259,  13.111,  0.0,     0.0,     0.0,     0.0,     0.0,
X0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
X0.0,     0.0,     0.0,
X6.421,   6.881,   7.362,   7.927,   8.554,   9.161,   9.679,
X10.055,  10.272,  10.389,  10.573,  10.861,  11.209,  11.581,
X11.948,  12.304,  12.638,  12.946,  13.219,  13.459,  13.661,
X13.822,  13.937,  14.000,  14.005,  13.943,  13.815,  13.640,
X13.457,  13.304,  13.195,  0.0,     0.0,     0.0,     0.0,
X0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
X0.0,     0.0,     0.0)
```

```
 DATA ((ARRAY2(I), I= 271,360)=
X6.419,   6.882,   7.379,   7.975,   8.638,   9.273,   9.814,
X10.193,  10.393,  10.500,  10.705,  11.017,  11.389,  11.783,
X12.173,  12.547,  12.897,  13.217,  13.506,  13.754,  13.964,
X14.130,  14.246,  14.303,  14.295,  14.210,  14.052,  13.849,
X13.653,  13.501,  13.401,  0.0,     0.0,     0.0,     0.0,
X0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
X0.0,     0.0,     0.0,
X6.413,   6.880,   7.395,   8.027,   8.724,   9.391,   9.950,
X10.330,  10.500,  10.609,  10.836,  11.180,  11.577,  11.991,
X12.402,  12.794,  13.160,  13.495,  13.795,  14.056,  14.273,
X14.443,  14.558,  14.609,  14.584,  14.472,  14.280,  14.051,
X13.849,  13.704,  13.616,  0.0,     0.0,     0.0,     0.0,
X0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
X0.0,     0.0,     0.0)
 DATA ((ARRAY2(I), I= 361,450)=
X6.405,   6.878,   7.414,   8.085,   8.816,   9.513,   10.090,
X10.469,  10.609,  10.719,  10.968,  11.338,  11.762,  12.200,
X12.632,  13.043,  13.424,  13.773,  14.086,  14.356,  14.580,
X14.754,  14.869,14.911,    14.867,  14.723,  14.496,  14.247,
X14.045,  13.911,  13.837,  13.804,  0.0,     0.0,     0.0,
X0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
X0.0,     0.0,     0.0,
X6.398,   6.876,   7.437,   8.146,   8.913,   9.636,   10.230,
X10.602,  10.717,  10.825,  11.102,  11.502,  11.951,  12.413,
X12.863,  13.290,  13.690,  14.052,  14.376,  14.657,  14.888,
X15.066,  15.178,  15.210,  15.143,  14.964,  14.700,  14.437,
X14.242,  14.123,  14.064,  14.045,  14.051,  0.0,     0.0,
X0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
X0.0,     0.0,     0.0)
 DATA ((ARRAY2(I), I= 451,540)=
X6.385,   6.871,   7.462,   8.211,   9.014,   9.763,   10.372,
X10.736,  10.818,  10.931,  11.242,  11.671,  12.146,  12.627,
X13.097,  13.543,  13.959,  14.336,  14.673,  14.963,  15.202,
X15.383,  15.491,  15.511,  15.417,  15.196,  14.897,  14.628,
X14.446,  14.345,  14.301,  14.296,  14.315,  14.349,  0.0,
X0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
X0.0,     0.0,     0.0,
X6.363,   6.862,   7.487,   8.278,   9.117,   9.892,   10.512,
X10.865,  10.910,  11.039,  11.382,  11.841,  12.339,  12.844,
X13.334,  13.799,  14.230,  14.622,  14.971,  15.271,  15.517,
X15.700,  15.805,  15.809,  15.683,  15.417,  15.087,  14.819,
X14.654,  14.572,  14.546,  14.556,  14.587,  14.631,  14.682,
X0.0,     0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
X0.0,     0.0,     0.0)
 DATA ((ARRAY2(I), I= 541,630)=
X6.342,   6.854,   7.516,   8.347,   9.219,   10.019,  10.652,
X10.984,  10.995,  11.145,  11.520,  12.007,  12.532,  13.058,
X13.566,  14.049,  14.496,  14.901,  15.263,  15.573,  15.825,
X16.010,  16.109,  16.096,  15.934,  15.620,  15.266,  15.088,
X14.862,  14.800,  14.790,  14.814,  14.857,  14.912,  14.972,
X15.034,  0.0,     0.0,     0.0,     0.0,     0.0,     0.0,
X0.0,     0.0,     0.0,
X6.322,   6.848,   7.548,   8.419,   9.322,   10.143,  10.788,
X11.095,  11.073,  11.250,  11.658,  12.172,  12.721,  13.266,
X13.795,  14.293,  14.755,  15.174,  15.547,  15.867,  16.126,
X16.312,  16.405,  16.372,  16.170,  15.807,  15.439,  15.195,
X15.070,  15.027,  15.034,  15.071,  15.126,  15.191,  15.261,
X15.330,  15.397,  0.0,     0.0,     0.0,     0.0,     0.0,
X0.0,     0.0,     0.0)
```

```
      DATA ((ARRAY2(I), I= 631,720)=
     X6.301,  6.845,  7.584,  8.492,  9.424,  10.268, 10.918,
     X11.199, 11.145, 11.351, 11.795, 12.336, 12.905, 13.472,
     X14.017, 14.531, 15.008, 15.440, 15.826, 16.155, 16.420,
     X16.607, 16.693, 16.637, 16.390, 15.981, 15.607, 15.381,
     X15.278, 15.253, 15.276, 15.326, 15.393, 15.468, 15.546,
     X15.624, 15.698, 15.768, 0.0,    0.0,    0.0,    0.0,
     X0.0,    0.0,    0.0,
     X6.291,  6.850,  7.626,  8.567,  9.525,  10.390, 11.046,
     X11.295, 11.215, 11.448, 11.927, 12.492, 13.082, 13.668,
     X14.229, 14.760, 15.249, 15.695, 16.091, 16.429, 16.699,
     X16.888, 16.964, 16.886, 16.592, 16.141, 15.768, 15.562,
     X15.480, 15.472, 15.509, 15.571, 15.648, 15.732, 15.819,
     X15.905, 15.985, 16.061, 16.130, 0.0,    0.0,    0.0,
     X0.0,    0.0,    0.0)
      DATA ((ARRAY2(I), I= 721,810)=
     X6.292,  6.864,  7.673,  8.645,  9.629,  10.511, 11.173,
     X11.385, 11.288, 11.550, 12.059, 12.646, 13.255, 13.857,
     X14.435, 14.979, 15.483, 15.942, 16.349, 16.694, 16.969,
     X17.158, 17.227, 17.122, 16.779, 16.292, 15.926, 15.741,
     X15.678, 15.686, 15.735, 15.809, 15.896, 15.990, 16.085,
     X16.178, 16.266, 16.348, 16.422, 16.487, 16.544, 0.0,
     X0.0,    0.0,    0.0,
     X6.304,  6.887,  7.726,  8.726,  9.733,  10.633, 11.300,
     X11.475, 11.366, 11.655, 12.190, 12.797, 13.424, 14.041,
     X14.636, 15.194, 15.712, 16.181, 16.597, 16.950, 17.232,
     X17.421, 17.481, 17.349, 16.954, 16.437, 16.083, 15.919,
     X15.873, 15.896, 15.959, 16.044, 16.141, 16.243, 16.346,
     X16.445, 16.540, 16.628, 16.707, 16.781, 16.838, 16.889,
     X0.0,    0.0,    0.0)
      DATA ((ARRAY2(I), I= 811,900)=
     X6.320,  6.914,  7.783,  8.812,  9.840,  10.759, 11.427,
     X11.561, 11.445, 11.765, 12.326, 12.951, 13.597, 14.232,
     X14.841, 15.414, 15.943, 16.425, 16.849, 17.212, 17.500,
     X17.689, 17.739, 17.575, 17.125, 16.583, 16.246, 16.103,
     X16.076, 16.114, 16.189, 16.285, 16.392, 16.502, 16.613,
     X16.719, 16.820, 16.914, 16.999, 17.075, 17.143, 17.194,
     X17.236, 0.0,    0.0,
     X6.336,  6.943,  7.844,  8.901,  9.951,  10.887, 11.559,
     X11.646, 11.526, 11.879, 12.464, 13.111, 13.775, 14.427,
     X15.051, 15.639, 16.182, 16.675, 17.112, 17.485, 17.776,
     X17.964, 18.002, 17.804, 17.293, 16.733, 16.417, 16.296,
     X16.287, 16.339, 16.427, 16.535, 16.651, 16.770, 16.889,
     X17.003, 17.111, 17.211, 17.302, 17.383, 17.453, 17.510,
     X17.558, 17.591, 17.610)
      DATA ((ARRAY2(I), I= 901,990)=
     X6.364,  6.982,  7.910,  8.994,  10.066, 11.020, 11.694,
     X11.732, 11.610, 11.996, 12.599, 13.272, 13.954, 14.622,
     X15.261, 15.864, 16.421, 16.925, 17.374, 17.754, 18.050,
     X18.237, 18.264, 18.028, 17.454, 16.884, 16.590, 16.490,
     X16.498, 16.565, 16.665, 16.783, 16.909, 17.036, 17.163,
     X17.285, 17.400, 17.507, 17.604, 17.690, 17.765, 17.827,
     X17.878, 17.921, 17.938,
     X6.396,  7.024,  7.982,  9.092,  10.187, 11.160, 11.835,
     X11.820, 11.702, 12.119, 12.744, 13.442, 14.140, 14.825,
     X15.480, 16.098, 16.669, 17.187, 17.646, 18.034, 18.335,
     X18.523, 18.536, 18.255, 17.615, 17.044, 16.775, 16.696,
     X16.721, 16.803, 16.915, 17.044, 17.180, 17.317, 17.452,
     X17.582, 17.703, 17.817, 17.920, 18.011, 18.091, 18.158,
     X18.213, 18.253, 18.280)
```

```
 DATA ((ARRAY2(I), I= 991,1080)=
X6.428,   7.070,   8.058,   9.196,   10.315,  11.307,  11.982,
X11.906,  11.800,  12.252,  12.901,  13.620,  14.337,  15.041,
X15.713,  16.346,  16.931,  17.462,  17.933,  18.330,  18.639,
X18.825,  18.822,  18.489,  17.780,  17.216,  16.973,  16.917,
X16.960,  17.057,  17.183,  17.323,  17.469,  17.616,  17.760,
X17.897,  18.026,  18.146,  18.255,  18.353,  18.438,  18.511,
X18.569,  18.614,  18.641,
X6.460,   7.118,   8.140,   9.307,   10.453,  11.464,  12.136,
X11.990,  11.907,  12.391,  13.068,  13.804,  14.547,  15.269,
X15.960,  16.609,  17.209,  17.754,  18.238,  18.646,  18.961,
X19.146,  19.123,  18.728,  17.949,  17.401,  17.188,  17.156,
X17.217,  17.330,  17.469,  17.620,  17.778,  17.935,  18.087,
X18.233,  18.370,  18.497,  18.613,  18.717,  18.809,  18.887,
X18.951,  18.999,  19.030)
 DATA ((ARRAY2(I), I= 1081,1170)=
X6.494,   7.171,   8.229,   9.426,   10.599,  11.630,  12.297,
X12.072,  12.020,  12.541,  13.247,  14.004,  14.772,  15.513,
X16.222,  16.889,  17.505,  18.066,  18.562,  18.982,  19.304,
X19.487,  19.442,  18.971,  18.125,  17.602,  17.420,  17.412,
X17.494,  17.622,  17.775,  17.939,  18.108,  18.276,  18.438,
X18.593,  18.739,  18.874,  18.997,  19.108,  19.206,  19.290,
X19.359,  19.412,  19.447,
X6.528,   7.227,   8.325,   9.554,   10.757,  11.809,  12.465,
X12.155,  12.143,  12.705,  13.439,  14.220,  15.005,  15.775,
X16.503,  17.188,  17.822,  18.398,  18.910,  19.341,  19.670,
X19.851,  19.779,  19.219,  18.312,  17.822,  17.674,  17.691,
X17.792,  17.937,  18.104,  18.282,  18.463,  18.641,  18.814,
X18.979,  19.134,  19.278,  19.409,  19.527,  19.632,  19.723,
X19.798,  19.855,  19.895)
 DATA ((ARRAY2(I), I= 1171,1260)=
X6.540,   7.281,   8.428,   9.696,   10.931,  12.005,  12.641,
X12.230,  12.280,  12.889,  13.657,  14.467,  15.278,  16.069,
X16.824,  17.531,  18.187,  18.781,  19.308,  19.753,  20.090,
X20.267,  20.159,  19.479,  18.522,  18.079,  17.969,  18.015,
X18.138,  18.302,  18.486,  18.680,  18.875,  19.066,  19.251,
X19.427,  19.593,  19.746,  19.888,  20.015,  20.127,  20.224,
X20.305,  20.370,  20.414,
X6.545,   7.337,   8.537,   9.844,   11.111,  12.207,  12.811,
X12.301,  12.425,  13.079,  13.882,  14.724,  15.561,  16.374,
X17.154,  17.886,  18.564,  19.179,  19.721,  20.179,  20.524,
X20.698,  20.545,  19.725,  18.743,  18.352,  18.280,  18.354,
X18.500,  18.683,  18.884,  19.093,  19.303,  19.508,  19.706,
X19.894,  20.071,  20.234,  20.385,  20.522,  20.644,  20.749,
X20.836,  20.906,  20.956)
 DATA ((ARRAY2(I), I= 1261,1350)=
X6.552,   7.398,   8.647,   9.994,   11.292,  12.411,  12.971,
X12.376,  12.575,  13.275,  14.112,  14.982,  15.846,  16.684,
X17.484,  18.241,  18.941,  19.578,  20.138,  20.609,  20.961,
X21.127,  20.926,  19.953,  18.973,  18.635,  18.599,  18.700,
X18.868,  19.070,  19.288,  19.512,  19.736,  19.954,  20.165,
X20.365,  20.554,  20.728,  20.888,  21.034,  21.165,  21.279,
X21.375,  21.450,  21.505,
X6.551,   7.462,   8.762,   10.149,  11.479,  12.618,  13.116,
X12.450,  12.730,  13.478,  14.347,  15.248,  16.141,  17.005,
X17.830,  18.607,  19.330,  19.989,  20.570,  21.057,  21.417,
X21.573,  21.312,  20.168,  19.218,  18.936,  18.937,  19.065,
X19.255,  19.477,  19.712,  19.951,  20.189,  20.421,  20.645,
X20.857,  21.057,  21.244,  21.416,  21.570,  21.709,  21.831,
X21.936,  22.020,  22.080)
```

```
      DATA ((ARRAY2(I), I= 1351,1440)=
     X6.542,    7.527,    8.878,    10.303,   11.666,   12.821,   13.234,
     X12.524,   12.888,   13.682,   14.586,   15.518,   16.437,   17.329,
     X18.178,   18.977,   19.721,   20.399,   21.001,   21.504,   21.873,
     X22.021,   21.684,   20.368,   19.472,   19.243,   19.280,   19.435,
     X19.647,   19.887,   20.140,   20.396,   20.649,   20.894,   21.130,
     X21.354,   21.565,   21.762,   21.945,   22.111,   22.258,   22.388,
     X22.499,   22.590,   22.658,
     X6.537,    7.593,    8.992,    10.452,   11.843,   13.015,   13.326,
     X12.600,   13.042,   13.880,   14.817,   15.774,   16.721,   17.637,
     X18.512,   19.333,   20.097,   20.794,   21.412,   21.932,   22.308,
     X22.447,   22.028,   20.553,   19.724,   19.544,   19.613,   19.792,
     X20.025,   20.283,   20.552,   20.822,   21.089,   21.348,   21.597,
     X21.833,   22.055,   22.262,   22.453,   22.629,   22.786,   22.925,
     X23.042,   23.138,   23.213)
      DATA ((ARRAY2(I), I= 1441,1530)=
     X6.535,    7.661,    9.103,    10.596,   12.019,   13.197,   13.390,
     X12.679,   13.195,   14.072,   15.038,   16.023,   16.994,   17.935,
     X18.834,   19.678,   20.461,   21.178,   21.813,   22.345,   22.731,
     X22.857,   22.344,   20.732,   19.975,   19.841,   19.940,   20.140,
     X20.392,   20.668,   20.952,   21.237,   21.517,   21.789,   22.050,
     X22.298,   22.531,   22.748,   22.949,   23.132,   23.298,   23.445,
     X23.571,   23.675,   23.753,
     X6.529,    7.727,    9.212,    10.736,   12.183,   13.376,   13.419,
     X12.758,   13.347,   14.260,   15.257,   16.269,   17.264,   18.228,
     X19.148,   20.017,   20.820,   21.554,   22.206,   22.751,   23.145,
     X23.258,   22.634,   20.909,   20.227,   20.137,   20.263,   20.486,
     X20.756,   21.048,   21.348,   21.646,   21.940,   22.224,   22.497,
     X22.756,   23.000,   23.228,   23.439,   23.632,   23.805,   23.959,
     X24.093,   24.203,   24.287)
      DATA ((ARRAY2(I), I= 1531,1620)=
     X6.530,    7.792,    9.315,    10.868,   12.338,   13.533,   13.428,
     X12.838,   13.490,   14.437,   15.462,   16.498,   17.517,   18.502,
     X19.444,   20.333,   21.155,   21.907,   22.575,   23.132,   23.531,
     X23.632,   22.888,   21.081,   20.469,   20.418,   20.571,   20.813,
     X21.099,   21.406,   21.720,   22.032,   22.338,   22.634,   22.917,
     X23.187,   23.441,   23.679,   23.899,   24.100,   24.282,   24.443,
     X24.584,   24.700,   24.790,
     X6.535,    7.856,    9.412,    10.995,   12.490,   13.681,   13.422,
     X12.918,   13.626,   14.604,   15.656,   16.715,   17.754,   18.760,
     X19.721,   20.629,   21.470,   22.239,   22.921,   23.491,   23.895,
     X23.981,   23.106,   21.248,   20.703,   20.686,   20.861,   21.122,
     X21.424,   21.745,   22.072,   22.397,   22.713,   23.021,   23.315,
     X23.593,   23.857,   24.104,   24.333,   24.542,   24.731,   24.899,
     X25.046,   25.169,   25.264)
      DATA ((ARRAY2(I), I= 1621,1710)=
     X6.529,    7.916,    9.507,    11.115,   12.629,   13.819,   13.370,
     X12.994,   13.760,   14.774,   15.851,   16.934,   17.995,   19.022,
     X20.003,   20.931,   21.791,   22.575,   23.275,   23.856,   24.264,
     X24.331,   23.294,   21.419,   20.943,   20.961,   21.161,   21.441,
     X21.758,   22.094,   22.435,   22.772,   23.101,   23.419,   23.724,
     X24.013,   24.286,   24.542,   24.779,   24.997,   25.195,   25.370,
     X25.524,   25.653,   25.753,
     X.000,     .000,     .000,     .000,     12.752,   13.932,   13.310,
     X13.065,   13.885,   14.928,   16.027,   17.132,   18.215,   19.260,
     X20.259,   21.205,   22.080,   22.880,   23.592,   24.187,   24.598,
     X24.644,   23.438,   21.578,   21.164,   21.214,   21.435,   21.732,
     X22.064,   22.413,   22.766,   23.114,   23.454,   23.782,   24.096,
     X24.394,   24.676,   24.941,   25.187,   25.412,   25.617,   25.800,
     X25.960,   26.095,   26.199)
```

```
 DATA ((ARRAY2(I), I= 1711,1800)=
X.000,    0.0,    .000,    .000,    .000,    14.023,  13.252,
X13.132,  13.999, 15.068,  16.188,  17.313,  18.413,  19.476,
X20.492,  21.451, 22.342,  23.158,  23.884,  24.487,  24.903,
X24.925,  23.547, 21.728,  21.368,  21.445,  21.685,  21.997,
X22.343,  22.704, 23.068,  23.426,  23.775,  24.112,  24.435,
X24.742,  25.033, 25.306,  25.558,  25.791,  26.002,  26.194,
X26.361,  26.499, 26.607,
X0.0,     0.0,    0.0,     .000,    .000,    .000,    13.150,
X13.18,14.38,15.18,16.32,17.47,18.612,19.693,
X20.71,   21.699, 22.604,  23.437,  24.177,  24.790,  25.205,
X25.199,  23.614, 21.876,  21.574,  21.681,  21.940,  22.268,
X22.629,  23.002, 23.376,  23.744,  24.104,  24.451,  24.783,
X25.100,  25.401, 25.681,  25.941,  26.180,  26.400,  26.597,
X26.770,  26.914, 27.025)
 DATA ((ARRAY3(I), I= 1,90)=
X6.475,   8.179,  10.001,  11.835,  13.605,  15.299,  16.981,
X18.722,  20.647, 22.744,  24.879,  26.885,  28.708,  30.365,
X31.899,  33.343, 34.725,  36.068,  37.395,  38.719,  40.063,
X41.443,  42.883, 44.409,  46.052,  47.850,  49.839,  52.056,
X54.503,  0.0,    0.0,     0.0,     0.0,     0.0,     0.0,
X0.0,     0.0,    0.0,     0.0,     0.0,     0.0,     0.0,
X0.0,     0.0,    0.0,
X6.930,   8.774,  10.739,  12.700,  14.569,  16.343,  18.102,
X19.943,  21.979, 24.223,  26.500,  28.618,  30.525,  32.252,
X33.846,  35.346, 36.780,  38.177,  39.555,  40.937,  42.340,
X43.785,  45.298, 46.912,  48.660,  50.587,  52.745,  55.169,
X57.854,  0.0,    0.0,     0.0,     0.0,     0.0,     0.0,
X0.0,     0.0,    0.0,     0.0,     0.0,     0.0,     0.0,
X0.0,     0.0,    0.0)
 DATA ((ARRAY3(I), I= 91,180)=
X7.462,   9.470,  11.602,  13.706,  15.684,  17.548,  19.395,
X21.352,  23.517, 25.938,  28.380,  30.622,  32.624,  34.429,
X36.090,  37.651, 39.147,  40.604,  42.042,  43.487,  44.957,
X46.477,  48.076, 49.791,  51.664,  53.750,  56.111,  58.784,
X61.755,  0.0,    0.0,     0.0,     0.0,     0.0,     0.0,
X0.0,     0.0,    0.0,     0.0,     0.0,     0.0,     0.0,
X0.0,     0.0,    0.0,
X7.838,   9.961,  12.203,  14.384,  16.401,  18.286,  20.158,
X22.160,  24.412, 26.954,  29.491,  31.783,  33.808,  35.625,
X37.294,  38.864, 40.368,  41.832,  43.284,  44.741,  46.232,
X47.778,  49.415, 51.183,  53.135,  55.338,  57.858,  60.741,
X63.943,  67.322, 0.0,     0.0,     0.0,     0.0,     0.0,
X0.0,     0.0,    0.0,     0.0,     0.0,     0.0,     0.0,
X0.0,     0.0,    0.0)
 DATA ((ARRAY3(I), I= 181,270)=
X7.839,   9.961,  12.184,  14.305,  16.229,  18.016,  19.800,
X21.737,  23.963, 26.496,  28.979,  31.177,  33.097,  34.814,
X36.391,  37.872, 39.292,  40.680,  42.054,  43.443,  44.864,
X46.351,  47.933, 49.659,  51.587,  53.800,  56.370,  59.330,
X62.594,  65.974, 0.0,     0.0,     0.0,     0.0,     0.0,
X0.0,     0.0,    0.0,     0.0,     0.0,     0.0,     0.0,
X0.0,     0.0,    0.0,
X7.840,   9.959,  12.159,  14.213,  16.044,  17.737,  19.440,
X21.318,  23.531, 26.061,  28.485,  30.586,  32.406,  34.028,
X35.516,  36.915, 38.259,  39.571,  40.878,  42.198,  43.560,
X44.987,  46.522, 48.212,  50.129,  52.361,  54.993,  58.035,
X61.341,  64.684, 67.911,  0.0,     0.0,     0.0,     0.0,
X0.0,     0.0,    0.0,     0.0,     0.0,     0.0,     0.0,
X0.0,     0.0,    0.0)
```

```
 DATA ((ARRAY3(I), I= 271,360)=
X7.799,  9.903,  12.062, 14.029, 15.755, 17.349, 18.966,
X20.784, 22.983, 25.499, 27.842, 29.833, 31.544, 33.065,
X34.461, 35.776, 37.038, 38.275, 39.508, 40.760, 42.054,
X43.422, 44.903, 46.556, 48.456, 50.709, 53.404, 56.507,
X59.807, 63.060, 66.137, 0.0,    0.0,    0.0,    0.0,
X0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
X0.0,    0.0,    0.0,
X7.704,  9.773,  11.868, 13.729, 15.337, 16.827, 18.352,
X20.107, 22.301, 24.768, 27.004, 28.869, 30.462, 31.877,
X33.177, 34.400, 35.580, 36.737, 37.894, 39.071, 40.295,
X41.596, 43.018, 44.624, 46.505, 48.776, 51.522, 54.649,
X57.881, 60.984, 63.874, 0.0,    0.0,    0.0,    0.0,
X0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
X0.0,    0.0,    0.0)
 DATA ((ARRAY3(I), I= 361,450)=
X7.610,  9.646,  11.673, 13.427, 14.927, 16.320, 17.764,
X19.466, 21.661, 24.083, 26.207, 27.950, 29.435, 30.754,
X31.965, 33.108, 34.210, 35.295, 36.381, 37.492, 38.651,
X39.892, 41.262, 42.830, 44.701, 47.003, 49.803, 52.932,
X56.063, 58.999, 61.699, 64.190, 0.0,    0.0,    0.0,
X0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
X0.0,    0.0,    0.0,
X7.520,  9.521,  11.477, 13.127, 14.525, 15.828, 17.200,
X18.860, 21.060, 23.436, 25.447, 27.077, 28.462, 29.692,
X30.822, 31.891, 32.923, 33.941, 34.964, 36.012, 37.113,
X38.301, 39.624, 41.162, 43.030, 45.376, 48.230, 51.331,
X54.327, 57.093, 59.607, 61.915, 64.067, 0.0,    0.0,
X0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
X0.0,    0.0,    0.0)
 DATA ((ARRAY3(I), I= 451,540)=
X7.362,  9.307,  11.165, 12.694, 13.982, 15.192, 16.484,
X18.096, 20.289, 22.581, 24.457, 25.961, 27.238, 28.373,
X29.419, 30.408, 31.366, 32.311, 33.266, 34.246, 35.281,
X36.406, 37.676, 39.172, 41.032, 43.409, 46.279, 49.287,
X52.110, 54.661, 56.971, 59.087, 61.055, 62.906, 0.0,
X0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
X0.0,    0.0,    0.0,
X7.131,  8.995,  10.731, 12.126, 13.298, 14.409, 15.616,
X17.173, 19.339, 21.511, 23.231, 24.602, 25.766, 26.803,
X27.759, 28.665, 29.543, 30.413, 31.292, 32.201, 33.164,
X34.220, 35.423, 36.871, 38.711, 41.098, 43.931, 46.783,
X49.384, 51.707, 53.799, 55.714, 57.493, 59.168, 60.762,
X0.0,    0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
X0.0,    0.0,    0.0)
 DATA ((ARRAY3(I), I= 541,630)=
X6.916,  8.703,  10.321, 11.595, 12.665, 13.690, 14.825,
X16.339, 18.481, 20.528, 22.107, 23.361, 24.425, 25.374,
X26.252, 27.085, 27.894, 28.696, 29.511, 30.355, 31.256,
X32.249, 33.398, 34.803, 36.631, 39.034, 41.809, 44.490,
X46.880, 48.994, 50.896, 52.634, 54.249, 55.770, 57.218,
X58.610, 0.0,    0.0,    0.0,    0.0,    0.0,    0.0,
X0.0,    0.0,    0.0,
X6.715,  8.427,  9.932,  11.098, 12.079, 13.030, 14.101,
X15.584, 17.701, 19.622, 21.072, 22.222, 23.199, 24.072,
X24.881, 25.649, 26.397, 27.140, 27.896, 28.683, 29.527,
X30.468, 31.566, 32.936, 34.765, 37.181, 39.875, 42.379,
X44.573, 46.503, 48.234, 49.817, 51.288, 52.674, 53.993,
X55.265, 56.496, 0.0,    0.0,    0.0,    0.0,    0.0,
X0.0,    0.0,    0.0)
```

```
      DATA ((ARRAY3(I), I= 631,720)=
     X6.527,   8.166,   9.564,   10.633,  11.536,  12.421,  13.437,
     X14.900,  16.987,  18.784,  20.118,  21.175,  22.075,  22.880,
     X23.627,  24.339,  25.033,  25.724,  26.428,  27.164,  27.958,
     X28.849,  29.904,  31.247,  33.083,  35.507,  38.099,  40.428,
     X42.441,  44.206,  45.787,  47.234,  48.579,  49.846,  51.055,
     X52.218,  53.347,  54.449,  0.0,     0.0,     0.0,     0.0,
     X0.0,     0.0,     0.0,
     X6.431,   8.025,   9.346,   10.344,  11.194,  12.033,  13.016,
     X14.484,  16.565,  18.275,  19.519,  20.510,  21.356,  22.113,
     X22.817,  23.488,  24.144,  24.799,  25.467,  26.167,  26.928,
     X27.787,  28.818,  30.157,  32.030,  34.485,  36.998,  39.193,
     X41.073,  42.717,  44.190,  45.538,  46.791,  47.974,  49.102,
     X50.188,  51.242,  52.273,  53.288,  0.0,     0.0,     0.0,
     X0.0,     0.0,     0.0)
      DATA ((ARRAY3(I), I= 721,810)=
     X6.430,   8.006,   9.279,   10.233,  11.048,  11.860,  12.828,
     X14.329,  16.430,  18.084,  19.267,  20.215,  21.026,  21.752,
     X22.428,  23.074,  23.706,  24.337,  24.985,  25.665,  26.405,
     X27.250,  28.276,  29.637,  31.581,  34.092,  36.553,  38.656,
     X40.444,  42.006,  43.405,  44.686,  45.878,  47.004,  48.077,
     X49.112,  50.117,  51.100,  52.067,  53.026,  53.982,  0.0,
     X0.0,     0.0,     0.0,
     X6.531,   8.120,   9.371,   10.303,  11.102,  11.903,  12.877,
     X14.440,  16.591,  18.215,  19.368,  20.294,  21.087,  21.800,
     X22.462,  23.096,  23.716,  24.339,  24.977,  25.650,  26.386,
     X27.232,  28.273,  29.683,  31.738,  34.334,  36.777,  38.826,
     X40.561,  42.075,  43.432,  44.675,  45.833,  46.926,  47.970,
     X48.975,  49.953,  50.909,  51.851,  52.782,  53.717,  54.651,
     X0.0,     0.0,     0.0)
      DATA ((ARRAY3(I), I= 811,900)=
     X6.662,   8.271,   9.506,   10.421,  11.209,  12.004,  12.989,
     X14.630,  16.833,  18.433,  19.564,  20.473,  21.252,  21.954,
     X22.607,  23.233,  23.846,  24.461,  25.094,  25.764,  26.500,
     X27.351,  28.413,  29.886,  32.071,  34.753,  37.177,  39.180,
     X40.871,  42.346,  43.669,  44.881,  46.011,  47.078,  48.097,
     X49.080,  50.036,  50.971,  51.893,  52.807,  53.716,  54.636,
     X55.560,  0.0,     0.0,
     X6.800,   8.426,   9.645,   10.542,  11.319,  12.110,  13.109,
     X14.837,  17.085,  18.658,  19.766,  20.659,  21.426,  22.117,
     X22.762,  23.378,  23.984,  24.592,  25.220,  25.886,  26.622,
     X27.480,  28.567,  30.109,  32.438,  35.192,  37.586,  39.541,
     X41.188,  42.624,  43.914,  45.096,  46.198,  47.240,  48.236,
     X49.197,  50.131,  51.046,  51.948,  52.844,  53.738,  54.636,
     X55.543,  56.467,  57.412)
      DATA ((ARRAY3(I), I= 901,990)=
     X7.064,   8.744,   9.974,   10.876,  11.659,  12.462,  13.496,
     X15.352,  17.685,  19.264,  20.378,  21.270,  22.041,  22.736,
     X23.386,  24.008,  24.619,  25.234,  25.869,  26.546,  27.297,
     X28.181,  29.314,  30.964,  33.491,  36.356,  38.762,  40.710,
     X42.348,  43.778,  45.062,  46.240,  47.339,  48.379,  49.373,
     X50.332,  51.266,  52.180,  53.082,  53.978,  54.873,  55.772,
     X56.682,  57.603,  58.559,
     X7.387,   9.134,   10.384,  11.295,  12.090,  12.912,  13.991,
     X16.002,  18.431,  20.025,  21.153,  22.049,  22.829,  23.533,
     X24.192,  24.824,  25.444,  26.070,  26.717,  27.409,  28.181,
     X29.097,  30.290,  32.076,  34.836,  37.817,  40.242,  42.193,
     X43.833,  45.264,  46.551,  47.733,  48.836,  49.880,  50.879,
     X51.843,  52.781,  53.701,  54.609,  55.510,  56.412,  57.318,
     X58.236,  59.172,  60.132)
```

```
      DATA ((ARRAY3(I), I= 991,1080)=
     X7.744,   9.563,    10.831,   11.754,   12.563,   13.405,   14.535,
     X16.727,  19.251,   20.858,   21.997,   22.902,   23.692,   24.407,
     X25.074,  25.717,   26.349,   26.986,   27.647,   28.355,   29.150,
     X30.101,  31.365,   33.311,   36.330,   39.414,   41.855,   43.809,
     X45.450,  46.886,   48.176,   49.362,   50.471,   51.520,   52.524,
     X53.494,  54.438,   55.364,   56.279,   57.188,   58.096,   59.011,
     X59.939,  60.884,   61.856,
     X8.143,   10.037,   11.324,   12.259,   13.083,   13.948,   15.139,
     X17.540,  20.154,   21.775,   22.925,   23.846,   24.643,   25.368,
     X26.048,  26.701,   27.346,   27.996,   28.671,   29.398,   30.217,
     X31.209,  32.552,   34.690,   37.994,   41.169,   43.622,   45.578,
     X47.224,  48.663,   49.959,   51.151,   52.265,   53.321,   54.332,
     X55.308,  56.260,   57.193,   58.115,   59.032,   59.950,   60.875,
     X61.812,  62.770,   63.756)
      DATA ((ARRAY3(I), I= 1081,1170)=
     X8.588,   10.562,   11.869,   12.818,   13.658,   14.549,   15.812,
     X18.456,  21.155,   22.791,   23.953,   24.890,   25.695,   26.434,
     X27.125,  27.791,   28.448,   29.114,   29.806,   30.553,   31.400,
     X32.437,  33.871,   36.244,   39.855,   43.105,   45.567,   47.527,
     X49.177,  50.623,   51.925,   53.125,   54.247,   55.310,   56.329,
     X57.314,  58.274,   59.216,   60.147,   61.074,   62.002,   62.938,
     X63.887,  64.859,   65.860,
     X9.091,   11.148,   12.475,   13.439,   14.298,   15.220,   16.568,
     X19.494,  22.271,   23.923,   25.099,   26.049,   26.873,   27.620,
     X28.326,  29.006,   29.678,   30.360,   31.071,   31.840,   32.719,
     X33.806,  35.349,   38.010,   41.942,   45.252,   47.721,   49.687,
     X51.344,  52.798,   54.109,   55.317,   56.449,   57.522,   58.550,
     X59.545,  60.515,   61.467,   62.409,   63.347,   64.287,   65.236,
     X66.199,  67.186,   68.205)
      DATA ((ARRAY3(I), I= 1171,1260)=
     X9.219,   11.236,   12.501,   13.426,   14.258,   15.166,   16.555,
     X19.665,  22.347,   23.919,   25.045,   25.959,   26.755,   27.476,
     X28.157,  28.816,   29.468,   30.130,   30.824,   31.579,   32.446,
     X33.535,  35.128,   38.015,   42.064,   45.234,   47.574,   49.439,
     X51.014,  52.398,   53.648,   54.802,   55.883,   56.909,   57.892,
     X58.844,  59.773,   60.686,   61.589,   62.489,   63.392,   64.304,
     X65.231,  66.183,   67.156,
     X9.211,   11.142,   12.323,   13.196,   13.990,   14.872,   16.293,
     X19.533,  22.058,   23.528,   24.587,   25.451,   26.206,   26.894,
     X27.539,  28.166,   28.789,   29.422,   30.087,   30.815,   31.658,
     X32.732,  34.359,   37.460,   41.500,   44.467,   46.645,   48.384,
     X49.856,  51.152,   52.325,   53.408,   54.424,   55.389,   56.315,
     X57.212,  58.087,   58.947,   59.799,   60.649,   61.502,   62.364,
     X63.242,  64.144,   65.079)
      DATA ((ARRAY3(I), I= 1261,1350)=
     X9.201,   11.041,   12.145,   12.970,   13.729,   14.587,   16.053,
     X19.401,  21.768,   23.144,   24.142,   24.959,   25.676,   26.331,
     X26.948,  27.543,   28.137,   28.744,   29.382,   30.084,   30.904,
     X31.966,  33.635,   36.966,   40.933,   43.703,   45.732,   47.356,
     X48.735,  49.951,   51.052,   52.071,   53.027,   53.936,   54.809,
     X55.654,  56.480,   57.292,   58.097,   58.900,   59.707,   60.524,
     X61.356,  62.213,   63.102,
     X8.992,   10.687,   11.691,   12.451,   13.160,   13.977,   15.475,
     X18.828,  20.978,   22.233,   23.150,   23.906,   24.569,   25.177,
     X25.751,  26.308,   26.859,   27.425,   28.025,   28.685,   29.464,
     X30.492,  32.177,   35.673,   39.419,   41.933,   43.777,   45.257,
     X46.517,  47.631,   48.642,   49.577,   50.455,   51.291,   52.094,
     X52.872,  53.633,   54.382,   55.124,   55.865,   56.610,   57.365,
     X58.136,  58.931,   59.757)
```

```
 DATA ((ARRAY3(I), I= 1351,1440)=
X8.591,   10.099,  10.988,  11.672,  12.319,  13.082,  14.598,
X17.838,  19.735,  20.852,  21.676,  22.358,  22.958,  23.509,
X24.031,  24.537,  25.042,  25.557,  26.104,  26.713,  27.436,
X28.408,  30.080,  33.641,  37.044,  39.270,  40.906,  42.226,
X43.352,  44.349,  45.255,  46.095,  46.884,  47.635,  48.357,
X49.057,  49.742,  50.416,  51.085,  51.754,  52.426,  53.109,
X53.806,  54.526,  55.276,
X8.222,   9.564,   10.356,  10.976,  11.570,  12.289,  13.836,
X16.935,  18.617,  19.618,  20.363,  20.982,  21.528,  22.030,
X22.506,  22.970,  23.432,  23.906,  24.409,  24.970,  25.645,
X26.570,  28.239,  31.828,  34.905,  36.887,  38.350,  39.534,
X40.547,  41.446,  42.263,  43.022,  43.735,  44.415,  45.068,
X45.702,  46.323,  46.934,  47.541,  48.147,  48.758,  49.378,
X50.013,  50.669,  51.354)
 DATA ((ARRAY3(I), I= 1441,1530)=
X7.842,   9.029,   9.737,   10.299,  10.844,  11.520,  13.107,
X16.027,  17.517,  18.414,  19.087,  19.649,  20.146,  20.604,
X21.039,  21.462,  21.885,  22.320,  22.784,  23.301,  23.929,
X24.806,  26.476,  30.040,  32.799,  34.564,  35.874,  36.937,
X37.849,  38.659,  39.398,  40.083,  40.728,  41.343,  41.934,
X42.508,  43.070,  43.624,  44.175,  44.725,  45.280,  45.843,
X46.421,  47.019,  47.644,
X7.355,   8.380,   9.000,   9.501,   9.994,   10.624,  12.237,
X14.906,  16.201,  16.992,  17.591,  18.093,  18.538,  18.948,
X19.339,  19.719,  20.100,  20.493,  20.913,  21.384,  21.958,
X22.779,  24.431,  27.867,  30.289,  31.837,  32.991,  33.931,
X34.740,  35.459,  36.116,  36.725,  37.300,  37.847,  38.374,
X38.886,  39.388,  39.882,  40.374,  40.865,  41.361,  41.865,
X42.383,  42.919,  43.481)
 DATA ((ARRAY3(I), I= 1531,1620)=
X6.923,   7.814,   8.364,   8.815,   9.265,   9.855,   11.498,
X13.926,  15.063,  15.768,  16.305,  16.757,  17.159,  17.530,
X17.883,  18.229,  18.574,  18.931,  19.314,  19.746,  20.277,
X21.047,  22.692,  25.979,  28.119,  29.490,  30.517,  31.356,
X32.080,  32.724,  33.313,  33.860,  34.376,  34.868,  35.342,
X35.803,  36.254,  36.699,  37.141,  37.584,  38.031,  38.486,
X38.953,  39.438,  39.948,
X6.537,   7.318,   7.810,   8.219,   8.632,   9.189,   10.860,
X13.064,  14.071,  14.704,  15.189,  15.599,  15.965,  16.303,
X16.625,  16.940,  17.256,  17.582,  17.933,  18.331,  18.824,
X19.552,  21.201,  24.323,  26.227,  27.451,  28.372,  29.128,
X29.780,  30.363,  30.895,  31.390,  31.857,  32.302,  32.732,
X33.149,  33.558,  33.961,  34.363,  34.765,  35.171,  35.584,
X36.009,  36.451,  36.916)
 DATA ((ARRAY3(I), I= 1621,1710)=
X5.919,   6.561,   6.979,   7.333,   7.696,   8.206,   9.850,
X11.728,  12.574,  13.116,  13.534,  13.890,  14.207,  14.501,
X14.782,  15.056,  15.332,  15.618,  15.926,  16.277,  16.716,
X17.378,  18.977,  21.782,  23.394,  24.437,  25.227,  25.877,
X26.439,  26.942,  27.402,  27.830,  28.234,  28.619,  28.991,
X29.353,  29.707,  30.057,  30.405,  30.754,  31.106,  31.466,
X31.835,  32.220,  32.625,
X.000,    .000,    .000,    .000,    6.915,   7.389,   8.995,
X10.599,  11.320,  11.788,  12.153,  12.464,  12.742,  13.001,
X13.247,  13.490,  13.733,  13.985,  14.258,  14.570,  14.964,
X15.572,  17.130,  19.640,  21.021,  21.922,  22.607,  23.172,
X23.663,  24.101,  24.503,  24.877,  25.230,  25.568,  25.893,
X26.209,  26.520,  26.826,  27.131,  27.437,  27.747,  28.062,
X28.387,  28.726,  29.083)
```

```
 DATA ((ARRAY3(I), I= 1711,1800)=                                              2137
X.000,    0.0,     .000,    .000,    .000,    6.724,   8.292,                 2138
X9.671,   10.295,  10.706,  11.028,  11.304,  11.551,  11.781,                2139
X12.001,  12.217,  12.435,  12.660,  12.905,  13.186,  13.543,                2140
X14.106,  15.634,  17.885,  19.086,  19.874,  20.477,  20.976,                2141
X21.409,  21.797,  22.153,  22.484,  22.797,  23.096,  23.385,                2142
X23.666,  23.941,  24.213,  24.484,  24.756,  25.031,  25.312,                2143
X25.602,  25.903,  26.222,                                                    2144
X0.0,     0.0,     0.0,     .000,    .000,    .000,    7.349,                 2145
X8.44,8.97,9.31,9.59,9.82,10.032,10.227,                                      2146
X10.42,   10.599,  10.785,  10.979,  11.189,  11.432,  11.743,                2147
X12.246,  13.695,  15.604,  16.600,  17.260,  17.767,  18.188,                2148
X18.554,  18.883,  19.184,  19.465,  19.730,  19.984,  20.229,                2149
X20.468,  20.702,  20.934,  21.164,  21.395,  21.630,  21.869,                2150
X22.116,  22.373,  22.645)                                                    2151
180000.,90000.,100000.,110000.,120000.,130000.,140000.,150000.,               2153
 DATA (ATAB=10000.,20000.,30000.,40000.,50000.,60000.,70000.,                  2152
2160000.,170000.,180000.,190000.,200000.,210000.,220000.,230000.,             2154
3240000.,250000.,260000.,270000.,280000.,290000.,300000.,310000.,             2155
4320000.,330000.,340000.,350000.,360000.,370000.,380000.,390000.,             2156
5400000.),(VTAB=6000.,7000.,8000.,9000.,10000.,11000.,12000.,                 2157
613000.,14000.,15000.,16000.,17000.,18000.,19000.,20000.,21000.,              2158
722000.,23000.,24000.,25000.,26000.,27000.,28000.,29000.,30000.,              2159
831000.,32000.,33000.,34000.,35000.,36000.,37000.,38000.,39000.,              2160
940000.,41000.,42000.,43000.,44000.,45000.,46000.,47000.,48000.,              2161
X49000.,50000.)                                                               2162
 END                                                                          2163
```

PROGRAM USAGE AND RESTRICTIONS

Program D1244 was written in FORTRAN IV language for the Control Data 6000 series computer using the scope 3.0 operating system. The program requires a field length of 130 000 octal locations for compilation and 70 000 octal locations for production running. (The additional 40 000 locations are required for compilation because the BLOCK DATA subprogram requires the entire 130 000 locations for compilation.) The computing time depends on the elapsed real time of the problem, the computing time interval, and the physical size of the system (number of blocks). For parametric studies it is suggested that the time interval and the number of blocks be varied to determine the most expeditious configuration.

A list of restrictions and limitations of the programed equations follows:

(1) For any trajectory input, the free-stream thermodynamic properties are based on the 1962 U.S. Standard Atmosphere tables (ref. 2) and the stagnation conditions on the body are for air in chemical equilibrium (ref. 15).

(2) The aerodynamic heating computations are restricted to air or gases with the same thermodynamic and transport properties. Further restrictions in the aerodynamic heating options are: (a) the Van Driest option (refs. 3 and 4) is restricted to $\gamma = 1.4$, $c_p = 0.24$ Btu/lbm-$^{\circ}$R, $\mu \propto \theta = \frac{198\ ^{\circ}R}{T_e} = 0.505$, $0 \leqq M_e \leqq 20$, $0.2 \leqq T_w/T_e \leqq 6.0$ and for turbulent flow the additional restriction of $10^5 \leqq N_{Re,e} \leqq 10^8$; (b) the Sibulkin option (ref. 5) is limited to a perfect gas ($\gamma = 1.4$) and thus $V_\infty \leqq 5000$ ft/sec; (c) the Detra, Kemp, and Riddell option (ref. 6) is used in the velocity range $5000 \leqq V_\infty \leqq 20\,000$ ft/sec, and (d) the Cohen option (ref. 7) is employed for velocities up to 41 000 ft/sec.

For local or free-stream conditions outside of the theoretical or correlation equation ranges or for gas mixtures other than air (such as in the entry into planetary atmospheres or wind-tunnel environments), it is recommended that the heating rates or the heat-transfer coefficients and recovery temperatures be generated by separate computations and then these results be put into the program to determine the thermal response of the skin.

Additional restrictions in the input data are:

(1) The number of blocks are 100 or less except for the inverse solution which are 30 or less.

(2) The product of the number of horizontal blocks times the total number of blocks cannot exceed 2000.

(3) All convective blocks must be numbered first (that is, ascending order beginning with the digit one) before nonconvective blocks are numbered.

(4) The number of materials is limited to 10 and these materials can have variable thermal properties.

Figure 4 shows the deck setup for D1244. For non-Langley users the Langley library subroutines described in appendix C should be included with program subroutines in the deck.

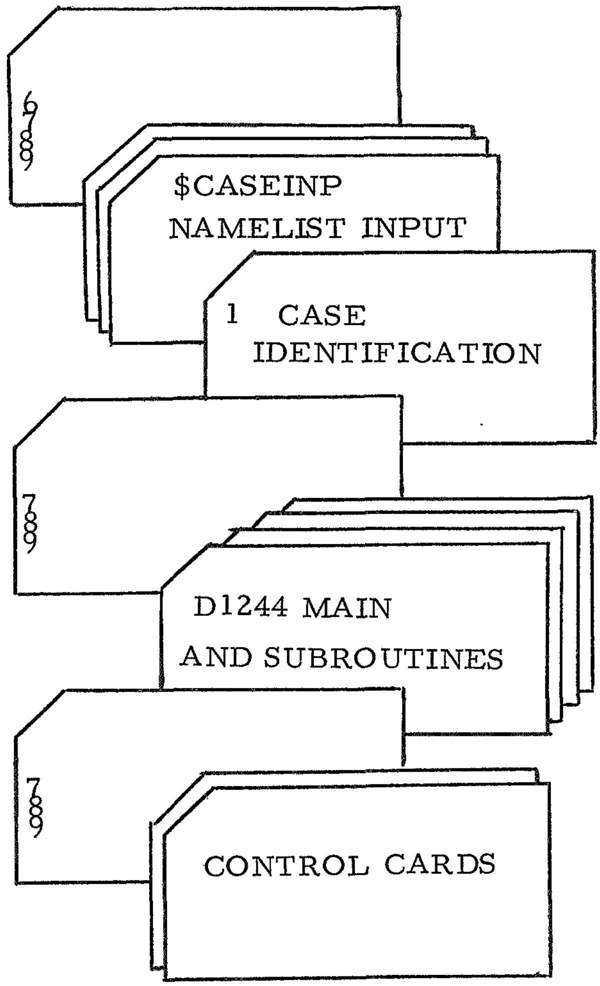


Figure 4.- D1244 deck configuration.

PROGRAM INPUT FORMAT

The first card of each case is an identification card and the remaining cards are loaded by the FORTRAN IV NAMELIST. The following is a description of how the cards should be set up for cases:

Card 1: Column 1 should have a 1 punch. The remainder of the card is used for identification of the case.

Card 2: Columns 2 to 9 – $CASEINP

Card 3, . . . are data cards. Data begins in column 2 and each card must end with a comma.

Last card: Column 2 – $

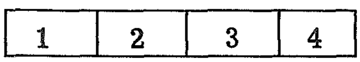
Input Data Cards

The following is a list of input parameters and their identifications. These cards must begin in column 2 and each card must end with a comma.

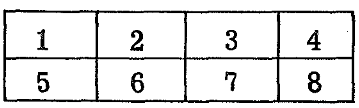
NO number of blocks

NHOR the greatest difference between the numbers of any two blocks which have a mutual heating term, either conduction, joint or air-conduction. Note that all convective blocks must be numbered before nonconvective blocks are numbered. NHOR*NO cannot exceed 2000.

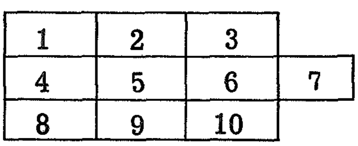
Examples:



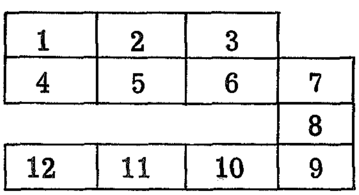
NHOR = 1



NHOR = 4



NHOR = 4



NHOR = 8

NAIRGAP = 3

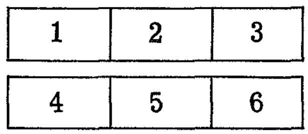
NOCONV number of convection blocks

NRADO number of radiation-out blocks

NJOINT number of joints

NAIRGAP number of air gaps

Example:



NAIRGAP = 3

IDIM
0 – thin wall
1 – one dimensional
2 – two dimensional

ICONVQ
0 – inverse solution
1 – heating rates are given
2 – stagnation option
3 – Van Driest option

IDIFF
0 – heating rate tables are same for each convection block
1 – heating rate tables are different for each convection block

LINEAR
0 – use T^4 for radiation-out term
1 – linearize T^4 term

INVERSE
0 – no inverse solution
1 – inverse solution required

COHEN
0 – use Cohen above 20 000 ft/sec
1 – use Cohen for all ranges

These controls have been specified to describe the kind of configuration and are used for the most part as routing controls for various options in the program. From this point on, only the necessary data pertinent to each particular configuration needs to be included, that is, if the configuration has no joint term, then any input relating to joints may be ignored. This statement also applies to any other option in the program.

TIME initial starting time, sec

TIMSTOP stopping time, sec

DELTAT	computing interval (delta time), sec
PRFREQ	print frequency
TEMP	initial temperatures of blocks, ^{o}R
MAT	material numbers of blocks (each material is given a number) (one per block)
EMS1 EMS2 . . . EMS10	emissivities of materials (four locations are saved for each material and the coefficients of a cubic polynomial are stored in these locations beginning with the constant value)
RHO	densities of materials, lb/ft^3
NKMPTS	number of table values for k_m of each material
TKTAB1 TKTAB2 . . . TKTAB10	independent temperature tables for each k_m table, ^{o}R
KM1 KM2 . . . KM10	k_m (conductivity of material) tables, Btu/ft-sec-^{o}R
NCPPTS	number of table values for c_p (specific heat) of each material
TCPTAB1 TCPTAB2 . . . TCPTB10	independent temperature tables for each c_p table, ^{o}R

CP1, CP2, ..., CP10	c_p (specific heat of material) tables, Btu/lbm-^{o}R
WD	widths of blocks, ft
LEN	lengths of blocks, ft
VOL	volumes of blocks, ft^3
TWDEPTH	thin-wall depth, ft
ACOND	conduction areas (the areas are listed beginning with the first block – for example, if block 1 touches 2 and 5, the areas between 1 and 2 and between 1 and 5 are given in that order; then if block 2 touches 1, 3, and 6 only the areas of the blocks ahead of the operating block number are given, for example, only the areas between 2 and 3 and 2 and 6 are given. No area is repeated. There are no blanks left and the areas are listed consecutively.
NTOUCH	conduction block information (see NTOUCH (m,n) explanation under section "Arrays")
ICROBLK	block numbers of convection and radiation-out terms
CROAREA	convection and radiation-out areas, ft^2
PERCRNL	Reynolds number lengths (if Van Driest), ft percents (if stagnation)
IQORH	$\dot{q}$, Btu/ft^2-sec, or h, Btu/ft^2-sec-^{o}R, and T_r, ^{o}R, inputs 1 – if $\dot{q}$ inputs 2 – if h and T_r input
HIPTS	number of values in each input table
TRANRN	transition Reynolds number
FPCONE	1.0 – flat plate; $\sqrt{3}$ – cone: Van Driest option

RADNOSE	nose radius, ft
ACRADI	air gap information (see section "Arrays")
AJOINT	joint information (see section "Arrays")

If heating input is given, the following tables are required:

HITIME	table of times for heat input, sec
HTAB	h tables (heat-transfer coefficient), Btu/ft^2-sec-$^{\circ}$R
TAWTAB	T_r tables (recovery temperature), $^{\circ}$R
QTAB	$\dot{q}$ tables, Btu/ft^2-sec

If an inverse solution is required, use the following information:

NOINV	number of inverse blocks (number of blocks with known temperature time histories)
INVBLKS	inverse block numbers
NOINVT	number of values in inverse temperature table
TINV	time table for inverse temperatures, sec
TEMINV	temperature table for inverse blocks, $^{\circ}$R

If there is more than one inverse block, the tables are as follows:

TEMINV1	table 1
TEMINV2	table 2
TEMINV3	table 3
TEMINV4	table 4

The following information is given for a trajectory when the Van Driest or stagnation options are used:

NAVPTS	number of values in altitude and velocity tables
TIMEAV	time table, sec
ALTTAB	altitude table, ft
VELTAB	velocity table, ft/sec

The following information is required for the Van Driest option and is used to convert free-stream conditions to local conditions:

NRATPT	number of ratio table points
MFSTAB	independent free-stream Mach number values
VERTAB	local to free-stream velocity ratios
MACTAB	local to free-stream Mach number ratios
REYTAB	local to free-stream Reynolds number ratios
TEMTAB	local to free-stream temperature ratios
RHOTAB	local to free-stream density ratios

If the model is at an angle of attack and two blocks with convective heating require different tables, use the following notation:

VERTAB1
VERTAB2
MACTAB1
MACTAB2
REYTAB1
REYTAB2
TEMTAB1
TEMTAB2
RHOTAB1
RHOTAB2

NRATAB(1)	a ratio table number for each convection block (If only one set of ratio tables is given, each convection block will be represented by a 1.)

ARRAYS

All (m,n) arrays when read into the machine are stored in consecutive locations, the m varying most rapidly.

NTOUCH(m,n).- NTOUCH(5,100) is a two-dimensional array where the n varies as the block number and m goes from 1 to 5 with:

1	total number of conduction blocks
2-5	numbers of the conduction blocks

If the conduction blocks require the width for the computation, the sign of the conduction block must be negative.

Example:

1	2	3
4	5	6

The NTOUCH array would be entered:

NTOUCH(1,1) = 2,2, -4,

NTOUCH(1,2) = 2,3, -5,

NTOUCH(1,3) = 1, -6,

NTOUCH(1,4) = 1,5,

NTOUCH(1,5) = 1,6,

NTOUCH(1,6) = 0,

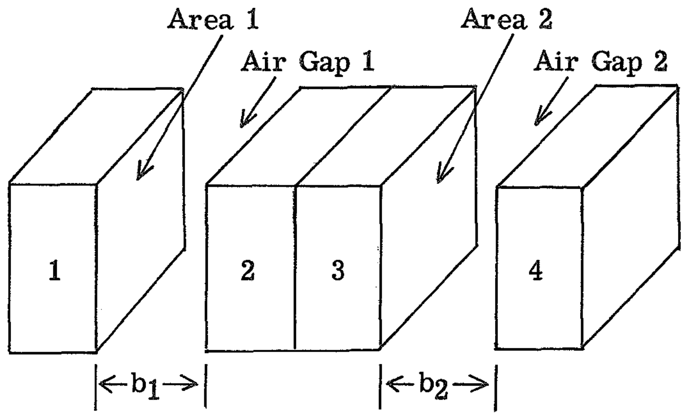
Another more simplified way of entering this same data would be:

NTOUCH = 2, 2, -4, 2*0, 2, 3, -5, 2*0, 1, -6, 3*0, 1, 5, 3*0, 1, 6, 4*0,

ACRADI(m,n).- ACRADI(4,20) is a two-dimensional array where the n array varies as the number of the joint and the m goes from 1 to 4 with

1	number of air gap block
2	block on opposite side of air gap
3	distance across air gap, ft
4	area on air gap, ft^2

Example:



Sketch (c)

$$ACRADI(1,1) = 1, 2, b_1, A_1,$$

$$ACRADI(1,2) = 3, 4, b_2, A_2,$$

Another more simplified way of entering this same data would be:

$$ACRADI = 1, 2, b_1, A_1, 3, 4, b_2, A_2,$$

AJOINT(m,n).- AJOINT(4,20) is a two-dimensional array where the n array varies as the number of the joint and the m goes from 1 to 4 with

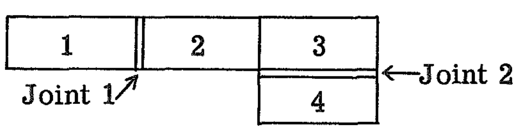
1 number of joint block

2 connecting block (other side of joint); block number is negative if the direction of heating across joint is in a vertical direction

3 h_j of joint, Btu/ft^2-sec-^{o}R

4 area of joint, ft^2

Example:



Sketch (d)

$$AJOINT(1,1) = 1, 2, h_{j,1}, A_1,$$

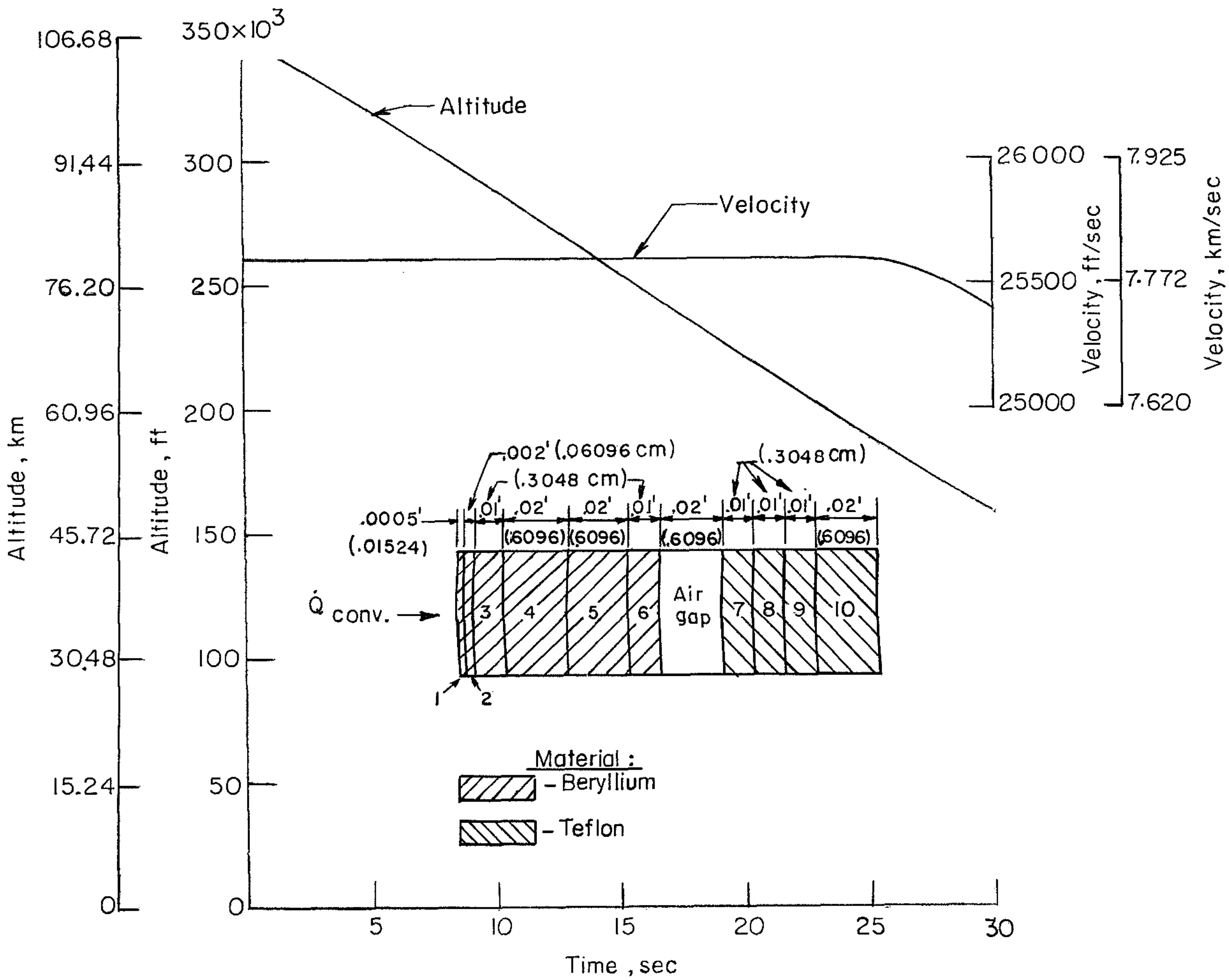
$$AJOINT(1,2) = 3, -4, h_{j,2}, A_2,$$

Another more simplified way of entering this same data would be:

$$AJOINT = 1, 2, h_{j,1}, A_1, 3, -4, h_{j,2}, A_2,$$

Two sample input listings and results are given in this section.

Case I is a one-dimensional stagnation heating analysis of a 1-foot-diameter sphere, composed of an exterior shell of beryllium and an interior shell of teflon, separated by an air gap. Note that the input volumes and conduction areas for the blocks were calculated for a flat-face slab rather than for a sphere simply for ease of computation.



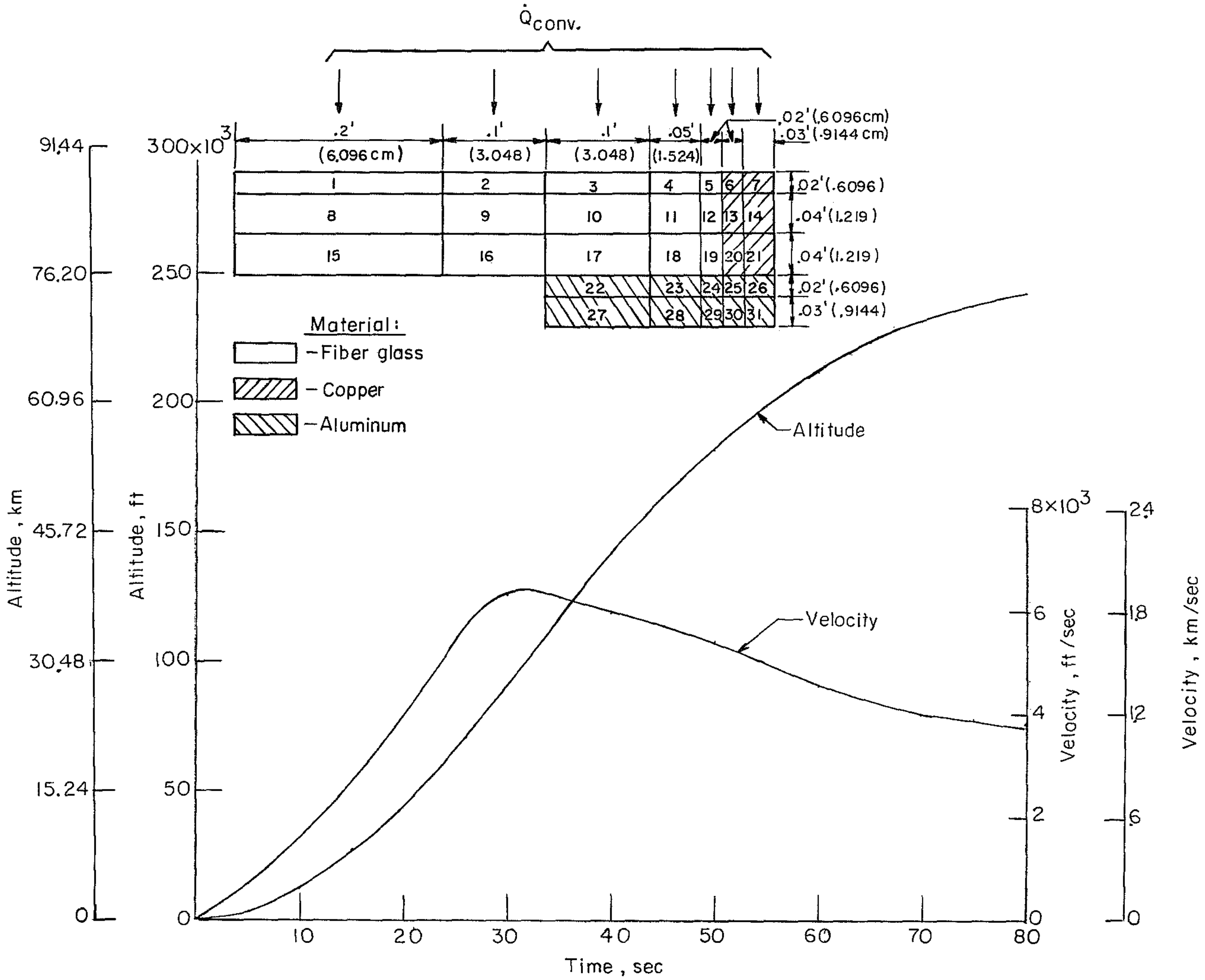
(a) Case I.

Figure 5.- Trajectory and configuration samples.

Case II is a two-dimensional analysis of a section of an 11° sharp cone composed of fiber glass, copper, and aluminum.

The geometry and trajectory are shown in figures 5(a) and 5(b) for cases I and II, respectively.

Listing of the input deck and the program results for the two sample cases follow.



(b) Case II.

Figure 5.- Concluded.

Case I

```
BERYLLIUM TEFLON HEAT SHIELD 1-D STAG. POINT THERMAL ANALYSIS

$CASEINP

NO      =  10,

NHOR    =  1,

NVER    =  0,

NOCONV  =  1,

NRADO   =  1,

NJOINT  =  0,

NAIRGAP =  1,

IGEOM   =  0,

IDIM    =  1,

ICONVO  =  2,

IDIFF   =  0,

LINEAR  =  1,

INVERSE =  0,

COHEN   =  1,

TIME    =  0.0,

TIMSTOP =  0.35E+02,

DELTAT  =  0.2E+00,

PRFREQ  =  10,

TEMP    =  0.5E+03,  0.5E+03,  0.5E+03,  0.5E+03,  0.5E+03,  0.5E+03,  0.5E+03,
           0.5E+03,  0.5E+03,  0.5E+03,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

MAT     =  1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 0, 0, 0, 0, 0, 0, 0,
           0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
           0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
           0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
           0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
           0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

EMS1    =  0.2E+00,  0.0,  0.0,  0.0,

EMS2    =  0.8E+00,  0.0,  0.0,  0.0,

EMS3    =  0.0,  0.0,  0.0,  0.0,

EMS4    =  0.0,  0.0,  0.0,  0.0,
```

```
EMS5    =  0.0,  0.0,  0.0,  0.0,

EMS6    =  0.0,  0.0,  0.0,  0.0,

EMS7    =  0.0,  0.0,  0.0,  0.0,

EMS8    =  0.0,  0.0,  0.0,  0.0,

EMS9    =  0.0,  0.0,  0.0,  0.0,

EMS10   =  0.0,  0.0,  0.0,  0.0,

RHO     =  0.116E+03,  0.137E+03,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,

NKMPTS  =  6,  2,  0,  0,  0,  0,  0,  0,  0,  0,

TKTAB1  =  0.5E+03,  0.1E+04,  0.15E+04,  0.2E+04,  0.25E+04,  0.28E+04,  0.0,
           0.0,  0.0,  0.0,

KM1     =  0.284E-01,  0.227E-01,  0.173E-01,  0.125E-01,  0.94E-02,  0.74E-02,
           0.0,  0.0,  0.0,  0.0,

TKTAB2  =  0.5E+03,  0.15E+04,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

KM2     =  0.4E-04,  0.8E-04,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

TKTAB3  =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

KM3     =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

TKTAB4  =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

KM4     =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

TKTAB5  =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

KM5     =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

TKTAB6  =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

KM6     =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

TKTAB7  =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

KM7     =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

TKTAB8  =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

KM8     =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

TKTAB9  =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

KM9     =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

TKTAB10 =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

KM10    =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

NCPPTS  =  4,  0,  0,  0,  0,  0,  0,  0,  0,  0,

TCPTAB1 =  0.5E+03,  0.8E+03,  0.11E+04,  0.28E+04,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,

CP1     =  0.425E+00,  0.55E+00,  0.617E+00,  0.86E+00,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,
```

```
TCPTAB2 =  0.5E+03,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

CP2     =  0.295E+00,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

TCPTAB3 =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

CP3     =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

TCPTAB4 =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

CP4     =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

TCPTAB5 =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

CP5     =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

TCPTAB6 =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

CP6     =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

TCPTAB7 =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

CP7     =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

TCPTAB8 =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

CP8     =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

TCPTAB9 =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

CP9     =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

TCPTB10 =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

CP10    =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

WD      =  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,
           0.1E+01,  0.1E+01,  0.1E+01,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

RAD     =  0.0,

LEN     =  0.5E-03,  0.2E-02,  0.1E-01,  0.2E-01,  0.2E-01,  0.1E-01,  0.1E-01,
           0.1E-01,  0.1E-01,  0.2E-01,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0.

THETA   =  0.0,

VOL     =  0.5E-03,  0.2E-02,  0.1E-01,  0.2E-01,  0.2E-01,  0.1E-01,  0.1E-01,
           0.1E-01,  0.1E-01,  0.2E-01,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
```

```
TWDEPTH  =  0.0,

ACOND    =  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,
            0.1E+01,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

NTOUCH   =  1,  2,  0,  0,  0,  1,  3,  0,  0,  0,  1,  4,  0,  0,  0,  1,  5,
            0,  0,  0,  1,  6,  0,  0,  0,  0,  0,  0,  0,  0,  1,  8,  0,  0,
            0,  1,  9,  0,  0,  0,  1,  10,  0,  0,  0,  0,  0,  0,  0,  0,  0,
            0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
            0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
            0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
            0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
            0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
            0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
            0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
            0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
            0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
            0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
            0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
            0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
            0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
            0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
            0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
            0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
            0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
            0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
            0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
            0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
            0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
            0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
            0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
            0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
            0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
            0,  0,  0,  0,  0,  0,  0,
```

```
ICROBLK = 0.1E+01, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,

CROAREA = 0.1E+01, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,

PERCRNL = 0.1E+01, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,

IOORH   = 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,

HIPTS   = 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,

TRANRN  = 0.0,

FPCCNE  = 0.0,

RADNOSE = 0.5E+00,

ACRADI  = 0.6E+01, 0.7E+01, 0.2E-01, 0.1E+01, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0,

AJOINT  = 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0,

HITIME  = 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0,
```

```
HTAB     =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,

TAWTAB   =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,

QTAB     =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,
```

```
NOINV   =  0,

INVBLKS =  0,  0,  0,  0,

NOINVT  =  0,

TINV    =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,

TEMINV  =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,

TEMINV1 =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,
```

```
TEMINV2 =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,

TEMINV3 =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,

TEMINV4 =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,

NAVPTS  =  7,

TIMEAV  =  0.0,  0.5E+01,  0.1E+02,  0.15E+02,  0.2E+02,  0.25E+02,  0.3E+02,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,

ALTTAB  =  0.35E+06,  0.32E+06,  0.288E+06,  0.255E+06,  0.22E+06,  0.188E+06,
           0.158E+06,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

VELTAB  =  0.256E+05,  0.256E+05,  0.256E+05,  0.256E+05,  0.256E+05,
           0.256E+05,  0.254E+05,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

NRATPT  =  0,

MFSTAB  =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,
```

```
VERTAB  =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

MACTAB  =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

REYTAB  =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

TEMTAB  =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

RHOTAB  =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

NRATAB  =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

VERTAB1 =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,

VERTAB2 =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,

MACTAB1 =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,

MACTAB2 =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,

REYTAB1 =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,

REYTAB2 =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,

TEMTAB1 =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,

TEMTAB2 =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,

RHOTAB1 =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,

RHOTAB2 =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,

$END
```

BERYLLIUM TEFLON HEAT SHIELD 1-D STAG. POINT THERMAL ANALYSIS

TIME = 2.0000 DELTA TIME = .2000

BLOCK NO	TEMPERATURE	CONVECTION Q
1	5.11780199E+02	9.40034716E+00
2	5.11369762E+02	
3	5.09491350E+02	
4	5.05847229E+02	
5	5.03276949E+02	
6	5.02670726E+02	
7	5.00001193E+02	
8	5.00000007E+02	
9	5.00000000E+02	
10	5.00000000E+02	

TIME = 4.0000 DELTA TIME = .2000

BLOCK NO	TEMPERATURE	CONVECTION Q
1	5.21673386E+02	1.29076652E+01
2	5.21107218E+02	
3	5.18510074E+02	
4	5.13385437E+02	
5	5.09581996E+02	
6	5.08658671E+02	
7	5.00007892E+02	
8	5.00000091E+02	
9	5.00000001E+02	
10	5.00000000E+02	

TIME = 6.0000 DELTA TIME = .2000

BLOCK NO	TEMPERATURE	CONVECTION Q
1	5.35321568E+02	1.81731806E+01
2	5.34520391E+02	
3	5.30851149E+02	
4	5.23652568E+02	
5	5.18336977E+02	
6	5.17048039E+02	
7	5.00023170E+02	
8	5.00000388E+02	
9	5.00000005E+02	
10	5.00000000E+02	

TIME = 8.0000 DELTA TIME = .2000

BLOCK NO	TEMPERATURE	CONVECTION Q
1	5.55223117E+02	2.63706500E+01
2	5.54051797E+02	
3	5.48696798E+02	
4	5.38256656E+02	
5	5.30611226E+02	
6	5.28763464E+02	
7	5.00050513E+02	
8	5.00001092E+02	
9	5.00000019E+02	
10	5.00000000E+02	

TIME = 10.0000 DELTA TIME = .2000

BLOCK NO	TEMPERATURE	CONVECTION Q
1	5.83945815E+02	3.82565228E+01
2	5.82227429E+02	
3	5.74382088E+02	
4	5.59146202E+02	
5	5.48044705E+02	
6	5.45367297E+02	
7	5.00095304E+02	
8	5.00002475E+02	
9	5.00000053E+02	
10	5.00000000E+02	

TIME = 12.0000 DELTA TIME = .2000

BLOCK NO	TEMPERATURE	CONVECTION Q
1	6.25152117E+02	5.55400452E+01
2	6.22616990E+02	
3	6.11072052E+02	
4	5.88800374E+02	
5	5.72675364E+02	
6	5.68796108E+02	
7	5.00165768E+02	
8	5.00004933E+02	
9	5.00000124E+02	
10	5.00000001E+02	

TIME = 14.0000 DELTA TIME = .2000

BLOCK NO	TEMPERATURE	CONVECTION Q
1	6.83559793E+02	7.99628188E+01
2	6.79823567E+02	
3	6.62863480E+02	
4	6.30391809E+02	
5	6.07051232E+02	
6	6.01453122E+02	
7	5.00274345E+02	
8	5.00009047E+02	
9	5.00000257E+02	
10	5.00000002E+02	

TIME = 16.0000 DELTA TIME = .2000

BLOCK NO	TEMPERATURE	CONVECTION Q
1	7.62434318E+02	1.10941673E+02
2	7.57076597E+02	
3	7.32834587E+02	
4	6.86688033E+02	
5	6.53678206E+02	
6	6.45778923E+02	
7	5.00439433E+02	
8	5.00015680E+02	
9	5.00000491E+02	
10	5.00000005E+02	

TIME = 18.0000 DELTA TIME = .2000

BLOCK NO	TEMPERATURE	CONVECTION Q
1	8.66096689E+02	1.51137020E+02
2	8.58463319E+02	
3	8.24120091E+02	
4	7.59640101E+02	
5	7.14121347E+02	
6	7.03285488E+02	
7	5.00687029E+02	
8	5.00026098E+02	
9	5.00000885E+02	
10	5.00000009E+02	

TIME = 20.0000 DELTA TIME = .2000

BLOCK NO	TEMPERATURE	CONVECTION Q
1	1.00148466E+03	2.01099182E+02
2	9.90677076E+02	
3	9.42403625E+02	
4	8.52740917E+02	
5	7.90299554E+02	
6	7.75560394E+02	
7	5.01053912E+02	
8	5.00042130E+02	
9	5.00001525E+02	
10	5.00000017E+02	

TIME = 22.0000 DELTA TIME = .2000

BLOCK NO	TEMPERATURE	CONVECTION Q
1	1.16823694E+03	2.55142325E+02
2	1.15340179E+03	
3	1.08769559E+03	
4	9.67286853E+02	
5	8.83975806E+02	
6	8.64354199E+02	
7	5.01593384E+02	
8	5.00066422E+02	
9	5.00002540E+02	
10	5.00000030E+02	

TIME = 24.0000 DELTA TIME = .2000

BLOCK NO	TEMPERATURE	CONVECTION Q
1	1.37575499E+03	3.20239032E+02
2	1.35506640E+03	
3	1.26496599E+03	
4	1.10423689E+03	
5	9.95706342E+02	
6	9.70388923E+02	
7	5.02382643E+02	
8	5.00102799E+02	
9	5.00004117E+02	
10	5.00000052E+02	

TIME = 26.0000 DELTA TIME = .2000

BLOCK NO	TEMPERATURE	CONVECTION Q
1	1.63875754E+03	3.96175542E+02
2	1.60916522E+03	
3	1.48326229E+03	
4	1.26743914E+03	
5	1.12603677E+03	
6	1.09353344E+03	
7	5.03533195E+02	
8	5.00156801E+02	
9	5.00006530E+02	
10	5.00000087E+02	

TIME = 28.0000 DELTA TIME = .2000

BLOCK NO	TEMPERATURE	CONVECTION Q
1	1.96732871E+03	4.81465142E+02
2	1.92355248E+03	
3	1.74541101E+03	
4	1.45663610E+03	
5	1.27513959E+03	
6	1.23407744E+03	
7	5.05206486E+02	
8	5.00236444E+02	
9	5.00010175E+02	
10	5.00000142E+02	

TIME = 30.0000 DELTA TIME = .2000

BLOCK NO	TEMPERATURE	CONVECTION Q
1	2.40131030E+03	5.91120858E+02
2	2.33353971E+03	
3	2.06653335E+03	
4	1.67238067E+03	
5	1.44119507E+03	
6	1.39009884E+03	
7	5.07636314E+02	
8	5.00353387E+02	
9	5.00015630E+02	
10	5.00000226E+02	

TIME = 32.0000 DELTA TIME = .2000

BLOCK NO	TEMPERATURE	CONVECTION Q
1	2.99948832E+03	7.31050211E+02
2	2.87620191E+03	
3	2.46416105E+03	
4	1.91895938E+03	
5	1.62251715E+03	
6	1.55919665E+03	
7	5.11142316E+02	
8	5.00524391E+02	
9	5.00023734E+02	
10	5.00000355E+02	

TIME = 34.0000 DELTA TIME = .2000

BLOCK NO	TEMPERATURE	CONVECTION Q
1	4.16635882E+03	9.40792075E+02
2	3.79946068E+03	
3	2.92821333E+03	
4	2.18116858E+03	
5	1.81381590E+03	
6	1.73733320E+03	
7	5.16142873E+02	
8	5.00773097E+02	
9	5.00035694E+02	
10	5.00000550E+02	

TIME = 36.0000 DELTA TIME = .2000

BLOCK NO	TEMPERATURE	CONVECTION Q
1	5.88421157E+03	1.21131504E+03
2	5.99047039E+03	
3	3.77938833E+03	
4	2.37514000E+03	
5	1.99587906E+03	
6	1.91165200E+03	
7	5.23091166E+02	
8	5.01131404E+02	
9	5.00053229E+02	
10	5.00000841E+02	

Case II

```
FIBERGLASS HEAT SHIELD 11 DEGREE CONE

$CASEINP

NO       =  31,

NHOR     =  7,

NVER     =  0,

NOCONV   =  7,

NRADO    =  7,

NJOINT   =  0,

NAIRGAP  =  0,

IGECM    =  0,

IDIM     =  2,

ICONVQ   =  3,

IDIFF    =  0,

LINEAR   =  0,

INVERSE  =  0,

COHEN    =  0,

TIME     =  0.0,

TIMSTOP  =  0.8E+02,

DELTAT   =  0.2E+00,

PRFREQ   =  25,

TEMP     =  0.52E+03,  0.52E+03,  0.52E+03,  0.52E+03,  0.52E+03,  0.52E+03,
            0.52E+03,  0.52E+03,  0.52E+03,  0.52E+03,  0.52E+03,  0.52E+03,
            0.52E+03,  0.52E+03,  0.52E+03,  0.52E+03,  0.52E+03,  0.52E+03,
            0.52E+03,  0.52E+03,  0.52E+03,  0.52E+03,  0.52E+03,  0.52E+03,
            0.52E+03,  0.52E+03,  0.52E+03,  0.52E+03,  0.52E+03,  0.52E+03,
            0.52E+03,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,

MAT      =  1,  1,  1,  1,  1,  2,  2,  1,  1,  1,  1,  1,  2,  2,  1,  1,  1,
            1,  1,  2,  2,  3,  3,  3,  3,  3,  3,  3,  3,  3,  3,  0,  0,  0,
            0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
            0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
            0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
            0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,

EMS1     =  0.88E+00,  0.0,  0.0,  0.0,

EMS2     =  0.25E+00,  0.0,  0.0,  0.0,

EMS3     =  0.0,  0.0,  0.0,  0.0,

EMS4     =  0.0,  0.0,  0.0,  0.0,

EMS5     =  0.0,  0.0,  0.0,  0.0,

EMS6     =  0.0,  0.0,  0.0,  0.0,
```

```
EMS7     =  0.0,  0.0,  0.0,  0.0,

EMS8     =  0.0,  0.0,  0.0,  0.0,

EMS9     =  0.0,  0.0,  0.0,  0.0,

EMS10    =  0.0,  0.0,  0.0,  0.0,

RHO      =  0.119E+03,  0.543E+03,  0.169E+03,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,

NKMPTS   =  7,  4,  5,  0,  0,  0,  0,  0,  0,  0,

TKTAB1   =  0.565E+03,  0.8E+03,  0.9E+03,  0.1E+04,  0.11E+04,  0.12E+04,
            0.2E+04,  0.0,  0.0,  0.0,

KM1      =  0.1722E-04,  0.25E-04,  0.2639E-04,  0.25E-04,  0.1944E-04,
            0.1111E-04,  0.1111E-04,  0.0,  0.0,  0.0,

TKTAB2   =  0.5E+03,  0.1E+04,  0.15E+04,  0.2E+04,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,

KM2      =  0.6167E-01,  0.575E-01,  0.5444E-01,  0.5222E-01,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,

TKTAB3   =  0.52E+03,  0.8E+03,  0.1E+04,  0.12E+04,  0.13E+04,  0.0,  0.0,
            0.0,  0.0,  0.0,

KM3      =  0.286E-01,  0.308E-01,  0.303E-01,  0.286E-01,  0.275E-01,  0.0,
            0.0,  0.0,  0.0,  0.0,

TKTAB4   =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

KM4      =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

TKTAB5   =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

KM5      =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

TKTAB6   =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

KM6      =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

TKTAB7   =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

KM7      =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

TKTAB8   =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

KM8      =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

TKTAB9   =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

KM9      =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

TKTAB10 =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

KM10     =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

NCPPTS   =  6,  4,  6,  0,  0,  0,  0,  0,  0,  0,

TCPTAB1 =  0.55E+03,  0.7E+03,  0.9E+03,  0.11E+04,  0.12E+04,  0.13E+04,  0.0,
            0.0,  0.0,  0.0,

CP1      =  0.218E+00,  0.255E+00,  0.296E+00,  0.325E+00,  0.337E+00,
            0.337E+00,  0.0,  0.0,  0.0,  0.0,

TCPTAB2 =  0.5E+03,  0.1E+04,  0.15E+04,  0.2E+04,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,

CP2      =  0.92E-01,  0.99E-01,  0.107E+00,  0.115E+00,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,
```

```
TCPTAB3 =  0.52E+03,  0.7E+03,  0.9E+03,  0.11E+04,  0.12E+04,  0.13E+04,  0.0,
           0.0,  0.0,  0.0,

CP3     =  0.202E+00,  0.22E+00,  0.233E+00,  0.24E+00,  0.265E+00,  0.276E+00,
           0.0,  0.0,  0.0,  0.0,

TCPTAB4 =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

CP4     =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

TCPTAB5 =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

CP5     =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

TCPTAB6 =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

CP6     =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

TCPTAB7 =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

CP7     =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

TCPTAB8 =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

CP8     =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

TCPTAB9 =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

CP9     =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

TCPTB10 =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

CP10    =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

WD      =  0.2E-01,  0.2E-01,  0.2E-01,  0.2E-01,  0.2E-01,  0.2E-01,  0.2E-01,
           0.4E-01,  0.4E-01,  0.4E-01,  0.4E-01,  0.4E-01,  0.4E-01,  0.4E-01,
           0.4E-01,  0.4E-01,  0.4E-01,  0.4E-01,  0.4E-01,  0.4E-01,  0.4E-01,
           0.2E-01,  0.2E-01,  0.2E-01,  0.2E-01,  0.2E-01,  0.3E-01,  0.3E-01,
           0.3E-01,  0.3E-01,  0.3E-01,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

RAD     =  0.0,

LEN     =  0.2E+00,  0.1E+00,  0.1E+00,  0.5E-01,  0.2E-01,  0.2E-01,  0.3E-01,
           0.2E+00,  0.1E+00,  0.1E+00,  0.5E-01,  0.2E-01,  0.2E-01,  0.3E-01,
           0.2E+00,  0.1E+00,  0.1E+00,  0.5E-01,  0.2E-01,  0.2E-01,  0.3E-01,
           0.1E+00,  0.5E-01,  0.2E-01,  0.2E-01,  0.3E-01,  0.1E+00,  0.5E-01,
           0.2E-01,  0.2E-01,  0.3E-01,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

THETA   =  0.0,

VOL     =  0.4E-02,  0.2E-02,  0.2E-02,  0.1E-02,  0.4E-03,  0.4E-03,  0.6E-03,
           0.8E-02,  0.4E-02,  0.4E-02,  0.2E-02,  0.8E-03,  0.8E-03,
           0.12E-02,  0.8E-02,  0.4E-02,  0.4E-02,  0.2E-02,  0.8E-03,
           0.8E-03,  0.12E-02,  0.2E-02,  0.1E-02,  0.4E-03,  0.4E-03,
           0.6E-03,  0.3E-02,  0.15E-02,  0.6E-03,  0.6E-03,  0.9E-03,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,
```

```
TWDEPTH =  0.0,

ACOND   =  0.2E-01,  0.2E+00,  0.2E-01,  0.1E+00,  0.2E-01,  0.1E+00,  0.2E-01,
           0.5E-01,  0.2E-01,  0.2E-01,  0.2E-01,  0.2E-01,  0.3E-01,  0.4E-01,
           0.2E+00,  0.4E-01,  0.1E+00,  0.4E-01,  0.1E+00,  0.4E-01,  0.5E-01,
           0.4E-01,  0.2E-01,  0.4E-01,  0.2E-01,  0.3E-01,  0.4E-01,  0.4E-01,
           0.4E-01,  0.1E+00,  0.4E-01,  0.5E-01,  0.4E-01,  0.2E-01,  0.4E-01,
           0.2E-01,  0.3E-01,  0.2E-01,  0.1E+00,  0.2E-01,  0.5E-01,  0.2E-01,
           0.2E-01,  0.2E-01,  0.2E-01,  0.3E-01,  0.3E-01,  0.3E-01,  0.3E-01,
           0.3E-01,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

NTOUCH  =  2,  2, -8,  0,  0,  2,  3, -9,  0,  0,  2,  4, -10,  0,  0,  2,  5,
         -11,  0,  0,  2,  6, -12,  0,  0,  2,  7, -13,  0,  0,  1, -14,  0,
           0,  0,  2,  9, -15,  0,  0,  2,  10, -16,  0,  0,  2,  11, -17,  0,
           0,  2,  12, -18,  0,  0,  2,  13, -19,  0,  0,  2,  14, -20,  0,  0,
           1, -21,  0,  0,  0,  1,  16,  0,  0,  0,  1,  17,  0,  0,  0,  2,
           18, -22,  0,  0,  2,  19, -23,  0,  0,  2,  20, -24,  0,  0,  2,
           21, -25,  0,  0,  1, -26,  0,  0,  0,  2,  23, -27,  0,  0,  2,  24,
         -28,  0,  0,  2,  25, -29,  0,  0,  2,  26, -30,  0,  0,  1, -31,  0,
           0,  0,  1,  28,  0,  0,  0,  1,  29,  0,  0,  0,  1,  30,  0,  0,
           0,  1,  31,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
           0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
           0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
           0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
           0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
           0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
           0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
           0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
           0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
           0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
           0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
           0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
           0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
           0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
           0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
           0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
           0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
           0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
           0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
           0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
```

```
ICROBLK =  0.1E+01,  0.2E+01,  0.3E+01,  0.4E+01,  0.5E+01,  0.6E+01,  0.7E+01,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,

CROAREA =  0.2E+00,  0.1E+00,  0.1E+00,  0.5E-01,  0.2E-01,  0.2E-01,  0.3E-01,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,

PERCRNL =  0.3E+01,  0.315E+01,  0.325E+01,  0.3325E+01,  0.336E+01,
           0.338E+01,  0.3405E+01,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,

IQORH   =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

HIPTS   =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

TRANRN  =  0.1E+07,

FPCONE  =  0.1732E+01,

RADNOSE =  0.5E+00,

ACRADI  =  0.6E+01,  0.7E+01,  0.2E-01,  0.1E+01,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,

AJOINT  =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,

HITIME  =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,
```

```
HTAB    =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,

TAWTAB  =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,

QTAB    =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,

NOINV   =  0,

INVBLKS =  0,  0,  0,  0,

NOINVT  =  0,
```

```
TINV     = 0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,

TEMINV   = 0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,

TEMINV1  = 0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,
```

```
TEMINV2 = 0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,

TEMINV3 = 0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,

TEMINV4 = 0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,

NAVPTS  = 18,

TIMEAV  = 0.0,  0.5E+01,  0.1E+02,  0.15E+02,  0.2E+02,  0.25E+02,  0.3E+02,
          0.32E+02,  0.35E+02,  0.4E+02,  0.45E+02,  0.5E+02,  0.55E+02,
          0.6E+02,  0.65E+02,  0.7E+02,  0.75E+02,  0.8E+02,  0.0,  0.0,  0.0,
          0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,  0.0,

ALTTAB  = 0.0,  0.3E+04,  0.12E+05,  0.27E+05,  0.43E+05,  0.67E+05,
          0.91E+05,  0.101E+06,  0.117E+06,  0.142E+06,  0.164E+06,
          0.182E+06,  0.199E+06,  0.212E+06,  0.224E+06,  0.232E+06,
          0.238E+06,  0.244E+06,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

VELTAB  = 0.0,  0.71E+03,  0.16E+04,  0.275E+04,  0.4E+04,  0.54E+04,
          0.63E+04,  0.64E+04,  0.625E+04,  0.595E+04,  0.57E+04,  0.54E+04,
          0.5E+04,  0.455E+04,  0.425E+04,  0.4E+04,  0.39E+04,  0.38E+04,
          0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
          0.0,  0.0,  0.0,  0.0,  0.0,
```

```
NRATPT  =  10,

MFSTAB  =  0.0,  0.1E+01,  0.105E+01,  0.12E+01,  0.2E+01,  0.3E+01,  0.4E+01,
           0.5E+01,  0.6E+01,  0.7E+01,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

VERTAB  =  0.1E+01,  0.1E+01,  0.9652E+00,  0.9727E+00,  0.9815E+00,
           0.988E+00,  0.9903E+00,  0.9922E+00,  0.9933E+00,  0.9932E+00,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

MACTAB  =  0.1E+01,  0.1E+01,  0.958E+00,  0.965E+00,  0.967E+00,  0.963E+00,
           0.961E+00,  0.954E+00,  0.946E+00,  0.937E+00,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,

REYTAB  =  0.1E+01,  0.1E+01,  0.9906E+00,  0.995E+00,  0.1037E+01,  0.109E+01,
           0.1181E+01,  0.1274E+01,  0.1386E+01,  0.1502E+01,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,

TEMTAB  =  0.1E+01,  0.1E+01,  0.1015E+01,  0.1016E+01,  0.103E+01,
           0.1053E+01,  0.1062E+01,  0.1081E+01,  0.1102E+01,  0.1124E+01,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

RHOTAB  =  0.1E+01,  0.1E+01,  0.1038E+01,  0.104E+01,  0.108E+01,  0.1147E+01,
           0.1248E+01,  0.1363E+01,  0.1502E+01,  0.1653E+01,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,

NRATAB  =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

VERTAB1 =  0.1E+01,  0.1E+01,  0.9652E+00,  0.9727E+00,  0.9815E+00,
           0.988E+00,  0.9903E+00,  0.9922E+00,  0.9933E+00,  0.9932E+00,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,

VERTAB2 =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,

MACTAB1 =  0.1E+01,  0.1E+01,  0.958E+00,  0.965E+00,  0.967E+00,  0.963E+00,
           0.961E+00,  0.954E+00,  0.946E+00,  0.937E+00,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,

MACTAB2 =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,

REYTAB1 =  0.1E+01,  0.1E+01,  0.9906E+00,  0.995E+00,  0.1037E+01,  0.109E+01,
           0.1181E+01,  0.1274E+01,  0.1386E+01,  0.1502E+01,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,

REYTAB2 =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,
```

```
TEMTAB1 =  0.1E+01,  0.1E+01,  0.1015E+01,  0.1016E+01,  0.103E+01,
           0.1053E+01,  0.1062E+01,  0.1081E+01,  0.1102E+01,  0.1124E+01,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,

TEMTAB2 =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,

RHOTAB1 =  0.1E+C1,  0.1E+01,  0.1C38E+01,  0.104E+01,  0.108E+01,  0.1147E+01,
           0.1248E+01,  0.1363E+01,  C.1502E+01,  0.1653E+C1,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,

RHOTAB2 =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,

$END
```

FIBERGLASS HEAT SHIELD 11 DEGREE CONE

TIME = 5.0000 DELTA TIME = .2000

BLOCK NO	TEMPERATURE	CONVECTION Q	H * AREA	T(RECOVERY)
1	5.20816634E+02	9.58868875E-02	3.89834461E-03	5.45413456E+02
2	5.20808338E+02	4.75939792E-02	1.93431095E-03	5.45413472E+02
3	5.20803056E+02	4.73714066E-02	1.92485109E-03	5.45413482E+02
4	5.20799215E+02	2.36047757E-02	9.58987258E-04	5.45413490E+02
5	5.20797421E+02	9.42710033E-03	3.82965269E-04	5.45413493E+02
6	5.20236045E+02	9.63516064E-03	3.82674981E-04	5.45414489E+02
7	5.20235851E+02	1.44361504E-02	5.73349128E-04	5.45414490E+02
8	5.20000107E+02			
9	5.20000104E+02			
10	5.20000101E+02			
11	5.20000099E+02			
12	5.20000316E+02			
13	5.20097500E+02			
14	5.20097521E+02			
15	5.20000000E+02			
16	5.20000000E+02			
17	5.20000000E+02			
18	5.19999999E+02			
19	5.20000050E+02			
20	5.20031885E+02			
21	5.20032146E+02			
22	5.19999948E+02			
23	5.20000454E+02			
24	5.20004562E+02			
25	5.20012324E+02			
26	5.20014705E+02			
27	5.19999942E+02			
28	5.20000214E+02			
29	5.20002149E+02			
30	5.20004084E+02			
31	5.20005315E+02			

TIME = 10.0000 DELTA TIME = .2000

BLOCK NO	TEMPERATURE	CONVECTION Q	H * AREA	T(RECOVERY)
1	5.39082606E+02	7.46835483E-01	5.84173630E-03	6.66927385E+02
2	5.38948365E+02	3.70940335E-01	2.89841432E-03	6.66928803E+02
3	5.38862792E+02	3.69360636E-01	2.88412222E-03	6.66929707E+02
4	5.38800484E+02	1.84105558E-01	1.43686669E-03	6.66930366E+02
5	5.38686105E+02	7.35890458E-02	5.73814126E-04	6.66931546E+02
6	5.23412262E+02	8.28549569E-02	5.76665038E-04	6.67091784E+02
7	5.23408111E+02	1.24140791E-01	8.63986497E-04	6.67091828E+02
8	5.20020097E+02			
9	5.20019944E+02			
10	5.20019846E+02			
11	5.20019783E+02			
12	5.20032033E+02			
13	5.21963883E+02			
14	5.21965115E+02			
15	5.20000013E+02			
16	5.20000012E+02			
17	5.20000027E+02			
18	5.20000143E+02			
19	5.20006277E+02			
20	5.21007115E+02			
21	5.21018889E+02			
22	5.20015106E+02			
23	5.20108842E+02			
24	5.20309272E+02			
25	5.20554383E+02			
26	5.20647155E+02			
27	5.20014812E+02			
28	5.20099885E+02			
29	5.20235918E+02			
30	5.20332077E+02			
31	5.20401912E+02			

TIME = 15.0000 DELTA TIME = .2000

BLOCK NO	TEMPERATURE	CONVECTION Q	H * AREA	T(RECOVERY)
1	6.05263040E+02	2.05336967E+00	5.42783841E-03	9.83566431E+02
2	6.04725993E+02	1.02020551E+00	2.69287949E-03	9.83579005E+02
3	6.04383309E+02	1.01607194E+00	2.67948832E-03	9.83587028E+02
4	6.04133285E+02	5.06532122E-01	1.33487762E-03	9.83592881E+02
5	6.03402164E+02	2.02722464E-01	5.33186052E-04	9.83611779E+02
6	5.33801208E+02	2.47405236E-01	5.47257217E-04	9.85883390E+02
7	5.33781291E+02	3.70683812E-01	8.19910551E-04	9.85884037E+02
8	5.20151604E+02			
9	5.20150565E+02			
10	5.20149902E+02			
11	5.20149505E+02			
12	5.20238728E+02			
13	5.29088718E+02			
14	5.29095970E+02			
15	5.20000155E+02			
16	5.20000154E+02			
17	5.20000504E+02			
18	5.20002073E+02			
19	5.20055839E+02			
20	5.25481168E+02			
21	5.25547589E+02			
22	5.20208953E+02			
23	5.20967924E+02			
24	5.22148585E+02			
25	5.23410361E+02			
26	5.23916931E+02			
27	5.20207404E+02			
28	5.20922215E+02			
29	5.21790719E+02			
30	5.22356715E+02			
31	5.22776174E+02			

TIME = 20.0000 DELTA TIME = .2000

BLOCK NO	TEMPERATURE	CONVECTION Q	H * AREA	T(RECOVERY)
1	7.19901152E+02	3.21041488E+00	3.77920930E-03	1.56939495E+03
2	7.18730652E+02	1.59491496E+00	1.87480330E-03	1.56944121E+03
3	7.17983316E+02	1.58835080E+00	1.86538374E-03	1.56947074E+03
4	7.17436333E+02	7.91789701E-01	9.29269537E-04	1.56949235E+03
5	7.15010487E+02	3.17343980E-01	3.71346627E-04	1.56958666E+03
6	5.50358105E+02	4.02957925E-01	3.91775469E-04	1.57890113E+03
7	5.50303022E+02	6.03744129E-01	5.86956012E-04	1.57890502E+03
8	5.20552478E+02			
9	5.20548877E+02			
10	5.20546582E+02			
11	5.20545344E+02			
12	5.20863240E+02			
13	5.42127392E+02			
14	5.42146416E+02			
15	5.20000804E+02			
16	5.20000800E+02			
17	5.20003418E+02			
18	5.20011568E+02			
19	5.20228489E+02			
20	5.35022358E+02			
21	5.35201158E+02			
22	5.21093061E+02			
23	5.23745059E+02			
24	5.27047987E+02			
25	5.30244960E+02			
26	5.31581050E+02			
27	5.21089332E+02			
28	5.23632895E+02			
29	5.26192056E+02			
30	5.27769267E+02			
31	5.28949074E+02			

TIME = 25.0000 DELTA TIME = .2000

BLOCK NO	TEMPERATURE	CONVECTION Q	H * AREA	T(RECOVERY)
1	8.39737644E+02	2.97686418E+00	1.76006039E-03	2.53107971E+03
2	8.37907291E+02	1.47819599E+00	8.72991046E-04	2.53116176E+03
3	8.36738498E+02	1.47167297E+00	8.68512313E-04	2.53121415E+03
4	8.35877811E+02	7.33469519E-01	4.32629689E-04	2.53125272E+03
5	8.30079765E+02	2.94288380E-01	1.72965450E-04	2.53150847E+03
6	5.63070798E+02	3.65777251E-01	1.83661656E-04	2.55465281E+03
7	5.62967860E+02	5.48031407E-01	2.75157815E-04	2.55466636E+03
8	5.21330847E+02			
9	5.21322456E+02			
10	5.21317117E+02			
11	5.21314599E+02			
12	5.22049830E+02			
13	5.54792559E+02			
14	5.54821944E+02			
15	5.20002619E+02			
16	5.20002604E+02			
17	5.20013535E+02			
18	5.20038452E+02			
19	5.20594461E+02			
20	5.46414434E+02			
21	5.46710834E+02			
22	5.23349354E+02			
23	5.28907488E+02			
24	5.34607038E+02			
25	5.39625288E+02			
26	5.41801133E+02			
27	5.23344250E+02			
28	5.28740280E+02			
29	5.33355589E+02			
30	5.36060849E+02			
31	5.38095677E+02			

TIME = 30.0000 DELTA TIME = .2000

BLOCK NO	TEMPERATURE	CONVECTION Q	H * AREA	T(RECOVERY)
1	9.19968463E+02	1.80797772E+00	7.55667458E-04	3.31252580E+03
2	9.17683078E+02	8.97441510E-01	3.74739226E-04	3.31252580E+03
3	9.16224612E+02	8.93270975E-01	3.72770743E-04	3.31252580E+03
4	9.15138280E+02	4.45124950E-01	1.85670838E-04	3.31252580E+03
5	9.04452283E+02	1.78779887E-01	7.42418724E-05	3.31252580E+03
6	5.66192595E+02	2.17860318E-01	7.83154690E-05	3.34802256E+03
7	5.66055597E+02	3.26410074E-01	1.17329632E-04	3.34804739E+03
8	5.22423563E+02			
9	5.22408833E+02			
10	5.22399472E+02			
11	5.22395820E+02			
12	5.23655717E+02			
13	5.60609882E+02			
14	5.60640179E+02			
15	5.20006304E+02			
16	5.20006272E+02			
17	5.20037480E+02			
18	5.20090594E+02			
19	5.21133501E+02			
20	5.53881156E+02			
21	5.54218007E+02			
22	5.27128936E+02			
23	5.35023185E+02			
24	5.41784438E+02			
25	5.47229090E+02			
26	5.49665635E+02			
27	5.27125132E+02			
28	5.34856006E+02			
29	5.40541333E+02			
30	5.43723256E+02			
31	5.46120278E+02			

TIME = 35.0000 DELTA TIME = .2000

BLOCK NO	TEMPERATURE	CONVECTION Q	H * AREA	T(RECOVERY)
1	9.55786095E+02	6.45478187E-01	2.76288996E-04	3.29202893E+03
2	9.53287414E+02	3.20441439E-01	1.37014474E-04	3.29202893E+03
3	9.51692595E+02	3.18977663E-01	1.36295651E-04	3.29202893E+03
4	9.50481632E+02	1.58960799E-01	6.78870759E-05	3.29202893E+03
5	9.33953819E+02	6.40682195E-02	2.71697111E-05	3.29202893E+03
6	5.62996163E+02	7.91046139E-02	2.86144944E-05	3.32749072E+03
7	5.62846067E+02	1.18520273E-01	4.28695674E-05	3.32751760E+03
8	5.23679154E+02			
9	5.23657608E+02			
10	5.23643990E+02			
11	5.23639943E+02			
12	5.25418856E+02			
13	5.60328907E+02			
14	5.60351612E+02			
15	5.20012324E+02			
16	5.20012272E+02			
17	5.20080339E+02			
18	5.20168048E+02			
19	5.21758317E+02			
20	5.56177741E+02			
21	5.56476228E+02			
22	5.31708032E+02			
23	5.40155656E+02			
24	5.46391563E+02			
25	5.51038254E+02			
26	5.53171325E+02			
27	5.31707429E+02			
28	5.40027209E+02			
29	5.45427725E+02			
30	5.48334685E+02			
31	5.50522014E+02			

TIME = 40.0000 DELTA TIME = .2000

BLOCK NO	TEMPERATURE	CONVECTION Q	H * AREA	T(RECOVERY)
1	9.59446992E+02	1.35320671E-01	6.95421581E-05	2.90532665E+03
2	9.57265773E+02	6.61120525E-02	3.39373648E-05	2.90532665E+03
3	9.56007140E+02	6.51337731E-02	3.34135953E-05	2.90532665E+03
4	9.54715000E+02	3.22212355E-02	1.65185292E-05	2.90532665E+03
5	9.31937705E+02	1.29877646E-02	6.58145197E-06	2.90532665E+03
6	5.57980196E+02	1.60288580E-02	6.70340521E-06	2.94913178E+03
7	5.57824752E+02	2.39569927E-02	1.00182146E-05	2.94916829E+03
8	5.24976180E+02			
9	5.24948547E+02			
10	5.24931517E+02			
11	5.24927761E+02			
12	5.27141825E+02			
13	5.56968141E+02			
14	5.56978366E+02			
15	5.20020869E+02			
16	5.20020799E+02			
17	5.20143264E+02			
18	5.20264959E+02			
19	5.22386400E+02			
20	5.54876421E+02			
21	5.55096523E+02			
22	5.36093595E+02			
23	5.43463024E+02			
24	5.48245515E+02			
25	5.51562403E+02			
26	5.53118135E+02			
27	5.36096627E+02			
28	5.43384132E+02			
29	5.47631082E+02			
30	5.49837804E+02			
31	5.51493284E+02			

TIME = 45.0000 DELTA TIME = .2000

BLOCK NO	TEMPERATURE	CONVECTION Q	H * AREA	T(RECOVERY)
1	9.58685312E+02	7.75914708E-02	4.39214468E-05	2.72528174E+03
2	9.56454328E+02	3.79133376E-02	2.14341645E-05	2.72528174E+03
3	9.55162019E+02	3.73554865E-02	2.11033673E-05	2.72528174E+03
4	9.53800674E+02	1.84815068E-02	1.04327995E-05	2.72528174E+03
5	9.24834535E+02	7.48634072E-03	4.15804513E-06	2.72528174E+03
6	5.55372303E+02	9.35533982E-03	4.23193097E-06	2.76602741E+03
7	5.55215041E+02	1.39827288E-02	6.32460474E-06	2.76606131E+03
8	5.26268989E+02			
9	5.26235217E+02			
10	5.26214948E+02			
11	5.26212025E+02			
12	5.28768127E+02			
13	5.54660164E+02			
14	5.54660474E+02			
15	5.20031963E+02			
16	5.20031881E+02			
17	5.20223896E+02			
18	5.20374314E+02			
19	5.22980070E+02			
20	5.53271570E+02			
21	5.53427295E+02			
22	5.39630907E+02			
23	5.45263899E+02			
24	5.48696438E+02			
25	5.51019208E+02			
26	5.52112566E+02			
27	5.39636542E+02			
28	5.45215099E+02			
29	5.48287549E+02			
30	5.49860522E+02			
31	5.51035862E+02			

TIME = 50.0000 DELTA TIME = .2000

BLOCK NO	TEMPERATURE	CONVECTION Q	H * AREA	T(RECOVERY)
1	9.56475696E+02	4.60299639E-02	3.01076818E-05	2.48532020E+03
2	9.54227678E+02	2.24963031E-02	1.46929743E-05	2.48532020E+03
3	9.52922474E+02	2.21680723E-02	1.44662655E-05	2.48532020E+03
4	9.51494613E+02	1.09694394E-02	7.15168627E-06	2.48532020E+03
5	9.16493427E+02	4.47353455E-03	2.85151594E-06	2.48532020E+03
6	5.53669737E+02	5.71517421E-03	2.90326056E-06	2.52220618E+03
7	5.53512544E+02	8.54212277E-03	4.33891322E-06	2.52223659E+03
8	5.27551606E+02			
9	5.27511629E+02			
10	5.27488217E+02			
11	5.27486526E+02			
12	5.30310575E+02			
13	5.53119991E+02			
14	5.53113798E+02			
15	5.20045591E+02			
16	5.20045508E+02			
17	5.20318490E+02			
18	5.20491823E+02			
19	5.23542214E+02			
20	5.52118039E+02			
21	5.52230842E+02			
22	5.42245956E+02			
23	5.46368978E+02			
24	5.48843742E+02			
25	5.50511034E+02			
26	5.51296108E+02			
27	5.42253298E+02			
28	5.46337602E+02			
29	5.48555433E+02			
30	5.49687324E+02			
31	5.50532098E+02			

TIME = 55.0000 DELTA TIME = .2000

BLOCK NO	TEMPERATURE	CONVECTION Q	H * AREA	T(RECOVERY)
1	9.53416522E+02	2.52341872E-02	2.05612942E-05	2.18068300E+03
2	9.51170875E+02	1.23372672E-02	1.00342785E-05	2.18068300E+03
3	9.49865099E+02	1.21598618E-02	9.87949700E-06	2.18068300E+03
4	9.48367795E+02	6.01883792E-03	4.88417077E-06	2.18068300E+03
5	9.07505764E+02	2.48059567E-03	1.94835063E-06	2.18068300E+03
6	5.52495256E+02	3.29534637E-03	1.98516685E-06	2.21247984E+03
7	5.52339272E+02	4.92542583E-03	2.96682656E-06	2.21250569E+03
8	5.28820965E+02			
9	5.28774784E+02			
10	5.28748387E+02			
11	5.28748192E+02			
12	5.31777611E+02			
13	5.52053661E+02			
14	5.52043056E+02			
15	5.20061731E+02			
16	5.20061663E+02			
17	5.20423704E+02			
18	5.20615472E+02			
19	5.24079051E+02			
20	5.51307109E+02			
21	5.51390124E+02			
22	5.44131311E+02			
23	5.47126542E+02			
24	5.48924533E+02			
25	5.50137626E+02			
26	5.50708688E+02			
27	5.44139829E+02			
28	5.47106781E+02			
29	5.48717049E+02			
30	5.49539224E+02			
31	5.50152792E+02			

TIME = 60.0000 DELTA TIME = .2000

BLOCK NO	TEMPERATURE	CONVECTION Q	H * AREA	T(RECOVERY)
1	9.49861258E+02	1.35467301E-02	1.48813255E-05	1.86017869E+03
2	9.47629523E+02	6.62731971E-03	7.26242479E-06	1.86017869E+03
3	9.46330504E+02	6.53441842E-03	7.15044194E-06	1.86017869E+03
4	9.44757685E+02	3.23604897E-03	3.53503900E-06	1.86017869E+03
5	8.98222504E+02	1.35736434E-03	1.41097620E-06	1.86022621E+03
6	5.51688435E+02	1.92125363E-03	1.43923995E-06	1.88659693E+03
7	5.51534572E+02	2.87168646E-03	2.15094152E-06	1.88661804E+03
8	5.30075230E+02			
9	5.30022889E+02			
10	5.29993704E+02			
11	5.29995139E+02			
12	5.33175511E+02			
13	5.51317975E+02			
14	5.51304381E+02			
15	5.20080356E+02			
16	5.20080318E+02			
17	5.20537013E+02			
18	5.20744101E+02			
19	5.24595467E+02			
20	5.50745677E+02			
21	5.50807700E+02			
22	5.45484457E+02			
23	5.47668693E+02			
24	5.48987602E+02			
25	5.49881218E+02			
26	5.50301582E+02			
27	5.45493794E+02			
28	5.47656991E+02			
29	5.48836299E+02			
30	5.49439587E+02			
31	5.49889916E+02			

TIME = 65.0000 DELTA TIME = .2000

BLOCK NO	TEMPERATURE	CONVECTION Q	H * AREA	T(RECOVERY)
1	9.46063933E+02	7.78468397E-03	1.10183328E-05	1.65258498E+03
2	9.43852008E+02	3.81103471E-03	5.37725049E-06	1.65258498E+03
3	9.42563488E+02	3.75911458E-03	5.29436733E-06	1.65258498E+03
4	9.40907180E+02	1.86279428E-03	2.61746859E-06	1.65258498E+03
5	8.88925718E+02	7.98653477E-04	1.04539191E-06	1.65290090E+03
6	5.51142888E+02	1.20046198E-03	1.06750902E-06	1.67568793E+03
7	5.50992936E+02	1.79435698E-03	1.59539108E-06	1.67570587E+03
8	5.31313433E+02			
9	5.31255001E+02			
10	5.31223247E+02			
11	5.31226326E+02			
12	5.34508042E+02			
13	5.50820793E+02			
14	5.50805434E+02			
15	5.20101437E+02			
16	5.20101447E+02			
17	5.20656565E+02			
18	5.20876930E+02			
19	5.25095145E+02			
20	5.50368219E+02			
21	5.50415501E+02			
22	5.46457872E+02			
23	5.48066235E+02			
24	5.49047438E+02			
25	5.49716682E+02			
26	5.50031117E+02			
27	5.46467776E+02			
28	5.48060139E+02			
29	5.48935279E+02			
30	5.49384364E+02			
31	5.49719735E+02			

TIME = 70.0000 DELTA TIME = .2000

BLOCK NO	TEMPERATURE	CONVECTION Q	H * AREA	T(RECOVERY)
1	9.42171222E+02	4.87552615E-03	8.84514682E-06	1.49338040E+03
2	9.39981759E+02	2.38885976E-03	4.31670692E-06	1.49338040E+03
3	9.38705422E+02	2.35747339E-03	4.25018878E-06	1.49338040E+03
4	9.36956861E+02	1.16919525E-03	2.10126848E-06	1.49338040E+03
5	8.79773454E+02	5.15700746E-04	8.39731228E-07	1.49389941E+03
6	5.50783710E+02	8.26391229E-04	8.58063812E-07	1.51387202E+03
7	5.50637859E+02	1.23524798E-03	1.28237579E-06	1.51388747E+03
8	5.32535173E+02			
9	5.32470728E+02			
10	5.32436635E+02			
11	5.32441260E+02			
12	5.35779183E+02			
13	5.50494711E+02			
14	5.50478266E+02			
15	5.20124941E+02			
16	5.20125018E+02			
17	5.20781111E+02			
18	5.21013409E+02			
19	5.25580823E+02			
20	5.50124510E+02			
21	5.50161452E+02			
22	5.47162790E+02			
23	5.48365025E+02			
24	5.49109194E+02			
25	5.49621349E+02			
26	5.49861561E+02			
27	5.47173083E+02			
28	5.48362823E+02			
29	5.49024267E+02			
30	5.49365149E+02			
31	5.49619866E+02			

TIME = 75.0000 DELTA TIME = .2000

BLOCK NO	TEMPERATURE	CONVECTION Q	H * AREA	T(RECOVERY)
1	9.38259716E+02	3.68889175E-03	7.56206659E-06	1.42607501E+03
2	9.36093700E+02	1.80829587E-03	3.69054048E-06	1.42607501E+03
3	9.34830159E+02	1.78502710E-03	3.63368103E-06	1.42607501E+03
4	9.32980424E+02	8.85841469E-04	1.79649400E-06	1.42607501E+03
5	8.70836806E+02	3.99359233E-04	7.18331817E-07	1.42679051E+03
6	5.50555984E+02	6.57119643E-04	7.34195725E-07	1.44557569E+03
7	5.50414297E+02	9.82236067E-04	1.09725511E-06	1.44558996E+03
8	5.33740344E+02			
9	5.33669970E+02			
10	5.33633774E+02			
11	5.33639740E+02			
12	5.36993138E+02			
13	5.50290144E+02			
14	5.50273083E+02			
15	5.20150837E+02			
16	5.20151001E+02			
17	5.20909655E+02			
18	5.21153143E+02			
19	5.26054519E+02			
20	5.49976810E+02			
21	5.50006517E+02			
22	5.47678916E+02			
23	5.48596084E+02			
24	5.49174447E+02			
25	5.49576902E+02			
26	5.49765252E+02			
27	5.47689475E+02			
28	5.48596581E+02			
29	5.49108436E+02			
30	5.49373638E+02			
31	5.49571948E+02			

TIME = 80.0000 DELTA TIME = .2000

BLOCK NO	TEMPERATURE	CONVECTION Q	H * AREA	T(RECOVERY)
1	9.34359996E+02	2.73789351E-03	6.43010219E-06	1.36015315E+03
2	9.32217739E+02	1.34291236E-03	3.13811924E-06	1.36015315E+03
3	9.30967180E+02	1.32609012E-03	3.08977974E-06	1.36015315E+03
4	9.29007613E+02	6.58622828E-04	1.52761137E-06	1.36015315E+03
5	8.62139864E+02	3.04912024E-04	6.11171405E-07	1.36103759E+03
6	5.50416560E+02	5.17531288E-04	6.24840661E-07	1.37867778E+03
7	5.50279027E+02	7.73591002E-04	9.33824108E-07	1.37869093E+03
8	5.34928998E+02			
9	5.34852780E+02			
10	5.34814718E+02			
11	5.34821724E+02			
12	5.38153499E+02			
13	5.50168987E+02			
14	5.50151630E+02			
15	5.20179092E+02			
16	5.20179362E+02			
17	5.21041547E+02			
18	5.21295839E+02			
19	5.26517699E+02			
20	5.49896193E+02			
21	5.49920829E+02			
22	5.48062817E+02			
23	5.48780481E+02			
24	5.49242920E+02			
25	5.49568616E+02			
26	5.49720688E+02			
27	5.48073554E+02			
28	5.48782853E+02			
29	5.49190118E+02			
30	5.49402400E+02			
31	5.49561272E+02			

TIME = 85.0000 DELTA TIME = .2000

BLOCK NO	TEMPERATURE	CONVECTION Q	H * AREA	T(RECOVERY)
1	9.30479055E+02	1.98481952E-03	5.43583921E-06	1.29561486E+03
2	9.28360715E+02	9.74287472E-04	2.65289715E-06	1.29561486E+03
3	9.27123219E+02	9.62514929E-04	2.61204007E-06	1.29561486E+03
4	9.25045625E+02	4.78565917E-04	1.29143456E-06	1.29561486E+03
5	8.53682513E+02	2.29007374E-04	5.16993415E-07	1.29664246E+03
6	5.50335989E+02	4.03331774E-04	5.28719453E-07	1.31318247E+03
7	5.50202554E+02	6.02894219E-04	7.90171081E-07	1.31319456E+03
8	5.36101222E+02			
9	5.36019247E+02			
10	5.35979558E+02			
11	5.35987215E+02			
12	5.39263286E+02			
13	5.50103414E+02			
14	5.50085975E+02			
15	5.20209673E+02			
16	5.20210070E+02			
17	5.21176310E+02			
18	5.21441278E+02			
19	5.26971396E+02			
20	5.49860389E+02			
21	5.49881434E+02			
22	5.48354200E+02			
23	5.48931983E+02			
24	5.49312820E+02			
25	5.49584255E+02			
26	5.49710712E+02			
27	5.48365055E+02			
28	5.48935677E+02			
29	5.49269386E+02			
30	5.49444388E+02			
31	5.49575474E+02			

CONCLUDING REMARKS

A general transient heat-transfer program (D1244) developed at the Langley Research Center is described. This program has been used frequently in the thermal design of space vehicles and spacecraft which are subject to aerodynamic heating.

The governing equations and finite-difference method for their numerical solution are given. Complete program listing and flow diagrams are included, and the program inputs are described in detail.

The input data and computer results from two sample heating examples are included as check cases for program D1244 users.

Langley Research Center,
National Aeronautics and Space Administration,
Hampton, Va., April 17, 1970.

APPENDIX A

LINEARIZATION OF THE RADIATION-OUT TERM

For rapidly changing wall temperatures, particularly when the radiation-out heat flux is significant, the computations are more accurate if the fourth power of temperature used in the radiation-out term is included as an unknown in the solution for T_i rather than using $\left(T_i'\right)^4$ at the beginning of the time step. The computational procedure for the solution to the temperatures of the blocks is based on a system of linear equations; thus, the inclusion of a T_i^4 term is not feasible. However, as a first approximation, the $\left(T_i'\right)^4$ term can be updated in the radiation-out heat flux term.

Let $\dot{Q}_{rad\text{-}out} = \epsilon\sigma A\overline{T}_w^4$ where $\overline{T}_w$ is an average temperature given by

$$\overline{T}_w \equiv \frac{T_i' + T_i}{2}$$

since $\Delta T_w = T_i - T_i'$; then

$$\overline{T}_w = T_i' + \frac{\Delta T_w}{2} = T_i'\left(1 + \frac{\Delta T_w}{2T_i'}\right)$$

and

$$\overline{T}_w^4 = \left(T_i'\right)^4\left(1 + \frac{\Delta T_w}{2T_i'}\right)^4 = \left(T_i'\right)^4(1 + X)^4$$

where $X \equiv \frac{\Delta T_w}{2T_i'}$. Expanding $(1 + X)^4$ in a series and neglecting terms higher than first order yields

$$\begin{aligned}\overline{T}_w^4 &= \left(T_i'\right)^4(1 + X)^4 \approx \left(T_i'\right)^4(1 + 4X) = \left(T_i'\right)^4\left(1 + \frac{2\Delta T_w}{T_i'}\right)\\ &= \left(T_i'\right)^3\left[T_i' + 2\left(T_i - T_i'\right)\right]\\ &= 2\left(T_i'\right)^3 T_i - \left(T_i'\right)^4\end{aligned}$$

and

$$\dot{Q}_{rad\text{-}out} \approx \epsilon\sigma A\left[2\left(T_i'\right)^3 T_i - \left(T_i'\right)^4\right]$$

Since T_i is raised to the first power, it can be included in the linear solution of the equations.

APPENDIX B

DEVELOPMENT OF SIBULKIN EQUATION FOR MACHINE COMPUTATION

Sibulkin's expression for the heat-transfer coefficient (ref. 5) is:

$$N_{Nu,D} = 0.763 N_{Pr}{}^{0.4} D\left(\frac{\beta\rho}{\mu}\right)^{0.5} \tag{B1}$$

At the stagnation point equation (B1) becomes

$$h = 0.763 N_{Pr}{}^{0.4} k_s\left(\frac{\beta\rho}{\mu}\right)_s^{0.5} \tag{B2}$$

By introducing free-stream conditions and assuming that N_{Pr} is a constant, equation (B2) becomes

$$h = \frac{0.763}{\sqrt{2}\, N_{Pr}{}^{0.6}}\left(\frac{\beta D}{V_\infty}\right)^{0.5}\left(\frac{\mu_s \rho_s}{\mu_\infty \rho_\infty}\right)^{0.5} \frac{g\rho_\infty V_\infty c_{p,s}}{\sqrt{r_n}\sqrt{N_{Re,\infty}/ft}} \tag{B3}$$

Upon assuming that

$$\mu \propto T^{0.76} \tag{B4}$$

and including the equation of state

$$\frac{\rho_s}{\rho_\infty} = \frac{p_s}{p_\infty}\frac{T_\infty}{T_s} \tag{B5}$$

equation (B3) becomes

$$h = \frac{0.763}{\sqrt{2}\, N_{Pr}{}^{0.6}}\left(\frac{\beta D}{V_\infty}\right)^{1/2}\left[\left(\frac{T_\infty}{T_s}\right)^{0.24}\frac{p_s}{p_\infty}\right]^{1/2} \frac{g\rho_\infty V_\infty c_{p,s}}{\sqrt{r_n}\sqrt{N_{Re,\infty}/ft}} \tag{B6}$$

From Van Driest (ref. 3) for Newtonian flow,

$$\frac{\beta D}{V_\infty} = \left\{ \frac{8\left[(\gamma - 1)M_\infty^2 + 2\right]}{(\gamma + 1)M_\infty^2} \left[1 + \frac{\gamma - 1}{2} \frac{(\gamma - 1)M_\infty^2 + 2}{2\gamma M_\infty^2 - (\gamma - 1)}\right]^{-\frac{1}{\gamma-1}} \right\}^{1/2} \tag{B7}$$

Now

$$\frac{T_s}{T_\infty} = 1 + \frac{\gamma - 1}{2} M_\infty^2 \tag{B8}$$

and

$$\frac{p_s}{p_\infty} = \left[\frac{(\gamma + 1)M_\infty^2}{2}\right]^{\frac{\gamma}{\gamma-1}} \left[\frac{\gamma + 1}{2\gamma M_\infty^2 - (\gamma - 1)}\right]^{\frac{1}{\gamma-1}} \tag{B9}$$

For $\gamma = 1.4$ and $N_{Pr} = 0.72$, equation (B6) becomes

$$h = f\left(M_\infty\right) \frac{g\rho_\infty V_\infty c_{p,s}}{\sqrt{r_n}\sqrt{N_{Re,\infty}/ft}} \tag{B10}$$

where

$$f\left(M_\infty\right) = \frac{0.763}{\sqrt{2}\, N_{Pr}^{0.6}} \left(\frac{\beta D}{V_\infty}\right)^{1/2} \left[\left(\frac{T_\infty}{T_s}\right)^{0.24} \frac{p_s}{p_\infty}\right]^{1/2} \tag{B11}$$

Substituting equations (B7), (B8), and (B9) into equation (B11) and approximating the resulting expression by a second-order polynomial in M_∞ over the range $M_\infty = 0$ to $M_\infty = 5$ yields

$$f\left(M_\infty\right) \equiv B = 0.800 + 0.541M_\infty - 0.00574M_\infty^2 \tag{B12}$$

Thus,

$$h = \frac{Bg\rho_\infty V_\infty \left(c_p\right)_{Perfect}}{\sqrt{r_n}\sqrt{N_{Re,\infty}/ft}} \tag{B13}$$

where $c_{p,s}$ is replaced by $\left(c_p\right)_{Perfect}$.

APPENDIX C

LANGLEY LIBRARY SUBROUTINES

Subroutine DISCOT

Language: FORTRAN

Purpose: DISCOT performs single or double interpolation for continuous or discontinuous functions.

Given a table of some function y with two independent variables, x and z, this subroutine performs K_xth- and K_zth-order interpolation to calculate the dependent variable. In this subroutine all single-line functions are read in as two separate arrays and all multi-line functions are read in as three separate arrays; that is,

$$x_i \qquad (i = 1, 2, \ldots, L)$$

$$y_j \qquad (j = 1, 2, \ldots, M)$$

$$z_k \qquad (k = 1, 2, \ldots, N)$$

Use: CALL DISCOT (XA, ZA, TABX, TABY, TABZ, NC, NY, NZ, ANS)

XA	The x argument
ZA	The z argument (may be the same name as x on single lines)
TABX	A one-dimensional array of x values
TABY	A one-dimensional array of y values
TABZ	A one-dimensional array of z values
NC	A control word that consists of a sign (+ or -) and three digits. The control word is formed as follows:

(1) If NX = NY, the sign is negative. If NX ≠ NY, then NX is computed by DISCOT as NX = NY/NZ, and the sign is positive and may be omitted if desired.

(2) A one in the hundreds position of the word indicates that no extrapolation occurs above z_{max}. With a zero in this position, extrapolation occurs when $z > z_{max}$. The zero may be omitted if desired.

(3) A digit (1 to 7) in the tens position of the word indicates the order of interpolation in the x-direction.

(4) A digit (1 to 7) in the units position of the word indicates the order of interpolation in the z-direction.

NY The number of points in y array

NZ The number of points in z array

ANS The dependent variable y

The following programs will illustrate various ways to use DISCOT:

Case I: Given $y = f(x)$

NY = 50
NX (number of points in x array) = NY
Extrapolation when $z > z_{max}$
Second-order interpolation in x-direction
No interpolation in z-direction
Control word = -020

```
  DIMENSION TABX (50), TABY (50)
1 FORMAT (8E 9.5)
  READ (5,1) TABX, TABY
  READ (5,1) XA
  CALL DISCOT (XA, XA, TABX, TABY, TABY, -020, 50, 0, ANS)
```

CASE II: Given $y = f(x,z)$

NY = 800
NZ = 10
NX = NY/NZ (computed by DISCOT)
Extrapolation when $z > z_{max}$
Linear interpolation in x-direction
Linear interpolation in z-direction
Control word = 11

```
  DIMENSION TABX (800), TABY (800), TABZ (10)
1 FORMAT (8E 9.5)
  READ (5,1) TABX, TABY, TABZ
  READ (5,1) XA, ZA
  CALL DISCOT (XA, ZA, TABX, TABY, TABZ, 11, 800, 10, ANS)
```

CASE III: Given $y = f(x,z)$

NY = 800
NZ = 10
NX = NY
Extrapolation when $z > z_{max}$
Seventh-order interpolation in x-direction

Third-order interpolation in z-direction

Control word = -73

```
  DIMENSION TABX (800), TABY (800), TABZ (10)
1 FORMAT (8E 9.5)
  READ (5,1) TABX, TABY, TABZ
  READ (5,1) XA, ZA
  CALL DISCOT (XA, ZA, TABX, TABY, TABZ, -73, 800, 10, ANS)
```

Case IV: Same as Case III with no extrapolation above z_{max}. Control word = -173

```
  CALL DISCOT (XA, ZA, TABX, TABY, TABZ, -173, 800, 10, ANS)
```

Restrictions: See rule (5c) of section "METHOD" for restrictions on tabulating arrays and discontinuous functions. The order of interpolation in the x- and z-directions may be from 1 to 7. The following subprograms are used by DISCOT: UNS, DISSER, LAGRAN.

Method: Lagrange's interpolation formula is used in both the x- and z-directions for interpolation. This method is explained in detail in reference (a) of this subroutine. For a search in either the x- or z-direction, the following rules are observed:

(1) If $x < x_1$, the routine chooses the following points for extrapolation:

$$x_1, x_2, \ldots, x_{k+1} \quad \text{and} \quad y_1, y_2, \ldots, y_{k+1}$$

(2) If $x > x_n$, the routine chooses the following points for extrapolation:

$$x_{n-k}, x_{n-k+1}, \ldots, x_n \quad \text{and} \quad y_{n-k}, y_{n-k+1}, \ldots, y_n$$

(3) If $x \leqq x_n$ the routine chooses the following points for interpolation:

When k is odd,

$$x_{i-\frac{k+1}{2}}, x_{i-\frac{k+1}{2}+1}, \ldots, x_{i-\frac{k+1}{2}+k}$$

and

$$y_{i-\frac{k+1}{2}}, y_{i-\frac{k+1}{2}+1}, \ldots, y_{i-\frac{k+1}{2}+k}$$

When k is even,

$$x_{i-\frac{k}{2}}, x_{i-\frac{k}{2}+1}, \ldots, x_{i-\frac{k}{2}+k}$$

and

$$y_{i-\frac{k}{2}}, y_{i-\frac{k}{2}+1}, \ldots, y_{i-\frac{k}{2}+k}$$

(4) If any of the subscripts in rule (3) become negative or greater than n (number of points), rules (1) and (2) apply. When discontinuous functions are tabulated, the independent variable at the point of discontinuity is repeated.

(5) The subroutine will automatically examine the points selected before interpolation and if there is a discontinuity, the following rules apply. Let x_d and x_{d+1} be the point of discontinuity.

(a) If $x \leqq x_d$, points previously chosen are modified for interpolation as shown:

$$x_{d-k}, x_{d-k+1}, \ldots, x_d \quad \text{and} \quad y_{d-k}, y_{d-k+1}, \ldots, y_d$$

(b) If $x > x_d$, points previously chosen are modified for interpolation as shown:

$$x_{d+1}, x_{d+2}, \ldots, x_{d+k} \quad \text{and} \quad y_{d+1}, y_{d+2}, \ldots, y_{d+k}$$

(c) When tabulating discontinuous functions, there must always be $k + 1$ points above and below the discontinuity in order to get proper interpolation.

(6) When tabulating arrays for this subroutine, both independent variables must be in ascending order.

(7) In some engineering programs with many tables, it is quite desirable to read in one array of x values that could be used for all lines of a multi-line function or different functions. Even though this situation is not always applicable, the subroutine has been written to handle it. This procedure not only saves much time in preparing tabular data, but also can save many locations previously used when every y coordinate had to have a corresponding x coordinate. Another additional feature that may be useful is the possibility of a multi-line function with no extrapolation above the top line.

Accuracy: A function of the order of interpolation used.

Reference: (a) Nielsen, Kaj L.: Methods in Numerical Analysis. The Macmillan Co., c.1956.

Storage: 555_8 locations

Subprograms used: UNS 40_8 locations

DISSER 110_8 locations

LAGRAN 55_8 locations

Subroutine date: August 1, 1968

Subroutine SIMEQ

Language: FORTRAN

Purpose: SIMEQ solves the matrix equation $AX = B$ where A is a square coefficient matrix and B is a matrix of constant vectors. The solution to a set of simultaneous equations and the determinant may be obtained. If the user wants the determinant only, use DETEV for savings in time and storage.

Use: CALL SIMEQ (A, N, B, M, DETERM, IPIVOT, NMAX, ISCALE)

A	A two-dimensional array of the coefficients
N	The order of A; $1 \leqq N \leqq NMAX$
B	A two-dimensional array of the constant vectors B. On return to calling program, X is stored in B
M	The number of column vectors in B
DETERM	Gives the value of the determinant by the following formula: $DET(A) = 10^{100\ ISCALE}(DETERM)$
IPIVOT	A one-dimensional array of temporary storage used by the routine
NMAX	The maximum order of A as stated in dimension statement of calling program
ISCALE	A scale factor computed by subroutine to keep results of computation within the floating-point word size of the computer

Restrictions: Arrays A, B, and IPIVOT are dimensioned with variable dimensions in the subroutine. The maximum size of these arrays must be specified in a DIMENSION statement of the calling program as: A (NMAX, NMAX), B (NMAX, M), IPIVOT (NMAX). The original matrices, A and B, are destroyed. They must be saved by the user if there is further need for them. The determinant is set to zero for a singular matrix.

Method: Jordan's method is used through a succession of elementary transformations: $l_n, l_{n-1}, \ldots, l_1$. If these transformations are applied to a matrix B of constant vectors, the result is X where $AX = B$. Each transformation is selected so that the largest element is used in the pivotal position.

Accuracy: Total pivotal strategy is used to minimize the rounding errors; however, the accuracy of the final results depends upon how well-conditioned the original matrix is.

Reference: (a) Fox, L.: An Introduction to Numerical Linear Algebra. Oxford Univ. Press, c.1965.

Storage: 432_8 locations

Subroutine date: August 1, 1968

Subroutine FTLUP

Language: FORTRAN

Purpose: Computes $y = F(x)$ from a table of values using first- or second-order interpolation. An option to give y a constant value for any x is also provided.

Use: CALL FTLUP (X, Y, M, N, VARI, VARD)

X — The name of the independent variable x.

Y — The name of the dependent variable $y = F(x)$

M — The order of interpolation (an integer)

- $M = 0$ for y a constant. VARD(I) corresponds to VARI(I) for $I = 1, 2, \ldots, N$. For $M = 0$ or $N \leqq 1$, $y = F(VARI(1))$ for any value of x. The program extrapolates.
- $M = 1$ or 2. First or second order if VARI is strictly increasing (not equal)
- $M = -1$ or -2. First or second order if VARI is strictly decreasing (not equal)

N — The number of points in the table (an integer).

VARI — The name of a one-dimensional array which contains the N values of the independent variable.

VARD — The name of a one-dimensional array which contains the N values of the dependent variable.

Restrictions: All the numbers must be floating point. The values of the independent variable x in the table must be strictly increasing or strictly decreasing. The following arrays must be dimensioned by the calling program as indicated: VARI(N), VARD(N).

Accuracy: A function of the order of interpolation used.

References:

(a) Nielsen, Kaj L.: Methods in Numerical Analysis. The Macmillan Co., c.1956, pp. 87-91.

(b) Milne, William Edmund: Numerical Calculus. Princeton Univ. Press, c.1949, pp. 69-73.

Storage: 430_8 locations

Error condition: If the VARI values are not in order, the subroutine will print TABLE BELOW OUT OF ORDER FOR FTLUP AT POSITION xxx TABLE IS STORED IN LOCATION xxxxxx (absolute). It then prints the contents of VARI and VARD, and STOPS the program.

Subroutine date: September 12, 1969

Subroutine AT62

Language: FORTRAN

Purpose: AT62 approximates the U.S. Standard Atmosphere, 1962 (ref. (a) of this subroutine). Computes density in slugs/ft^3, pressure in lb/ft^2, temperature in degrees Kelvin, and the velocity of sound in ft/sec at any geometric altitude z in the range $-16\,500$ feet $\leqq z \leqq 2\,320\,000$ feet.

Use: CALL AT62 (Z, ANS)

Z	Geometric altitude in feet	
ANS	A one-dimensional array that contains the results.	
	ANS(1)	Density in slugs/ft^3
	ANS(2)	Pressure in lb/ft^2
	ANS(3)	Temperature in degrees Kelvin
	ANS(4)	Velocity of sound in ft/sec

Restrictions: Range: For altitudes below -16 500 feet the values of density, pressure, temperature, and velocity of sound are not valid. The concept of the velocity of sound in the atmosphere becomes essentially meaningless at altitudes in excess of 300 000 feet. To point out this limitation, the velocity of sound at altitudes above 300 000 feet is set equal to the velocity of sound at 300 000 feet. For altitudes above 2 320 000 feet, density, pressure, and temperature are set equal to their respective values at 2 320 000 feet.

Method: The equations and techniques are identical to those used in computing the U.S. Standard Atmosphere, 1962 (ref. (a) of this subroutine)

Accuracy: The tables in the referenced publication were computed with an IBM 7094 using some double-precision arithmetic. In converting the routine for the CDC 6000 computers, all double-precision arithmetic was eliminated. Accordingly, there may be slight differences between the results of the converted subroutine and the tables.

Reference: (a) Anon.: U.S. Standard Atmosphere, 1962. NASA, U.S. Air Force, and U.S. Weather Bur., Dec. 1962.

Storage: 1654_8 locations

Subroutine date: August 1, 1968

REFERENCES

1. Arpaci, Vedat S.: Conduction Heat Transfer. Addison-Wesley Pub. Co., c.1966.
2. Anon.: U.S. Standard Atmosphere, 1962. NASA, U.S. Air Force and U.S. Weather Bur., Dec. 1962.
3. Van Driest, E. R.: Investigation of Laminar Boundary Layer in Compressible Fluids Using the Crocco Method. NACA TN 2597, 1952.
4. Van Driest, E. R.: The Problem of Aerodynamic Heating. Aeronaut. Eng. Rev., vol. 15, no. 10, Oct. 1956, pp. 26-41.
5. Sibulkin, M.: Heat Transfer Near the Forward Stagnation Point of a Body of Revolution. J. Aeron. Sci. (Readers' Forum), vol. 19, no. 8, Aug. 1952, pp. 570-571.
6. Detra, R. W.; Kemp, N. H.; and Riddell, F. R.: Addendum to "Heat Transfer to Satellite Vehicles Re-entering the Atmosphere." Jet Propulsion, vol. 27, no. 12, Dec. 1957, pp. 1256-1257.
7. Cohen, Nathaniel B.: Boundary-Layer Similar Solutions and Correlation Equations for Laminar Heat-Transfer Distribution in Equilibrium Air at Velocities Up to 41,100 Feet Per Second. NASA TR R-118, 1961.
8. Kreith, Frank: Principles of Heat Transfer. Int. Textbook Co. (Scranton, Pa.), c.1958.
9. Ames Research Staff: Equations, Tables, and Charts for Compressible Flow. NACA Rep. 1135, 1953. (Supersedes NACA TN 1428.)
10. Truitt, Robert Wesley: Fundamentals of Aerodynamic Heating. The Ronald Press Co., c.1960.
11. Lee, Dorothy B.; and Faget, Maxime A.: Charts Adapted From Van Driest's Turbulent Flat-Plate Theory for Determining Values of Turbulent Aerodynamic Friction and Heat-Transfer Coefficients. NACA TN 3811, 1956.
12. Eckhert, Ernst R. G.: Survey on Heat Transfer at High Speeds. ARL 189, U.S. Air Force, Dec. 1961.
13. Lees, Lester: Laminar Heat Transfer Over Blunt-Nosed Bodies at Hypersonic Flight Speeds. Jet Propulsion, vol. 26, no. 4, Apr. 1956, pp. 259-269, 274.
14. Hansen, C. Frederick: Approximations for the Thermodynamic and Transport Properties of High-Temperature Air. NASA TR R-50, 1959. (Supersedes NACA TN 4150.)
15. Lewis, Clark H.; and Burgess, E. G., III: Altitude-Velocity Tables and Charts for Imperfect Air. AEDC-TDR-64-214, U.S. Air Force, Jan. 1965. (Available from DDC as AD 454078.)

16. Zoby, Ernest V.; and Sullivan, Edward M.: Effects of Corner Radius on Stagnation-Point Velocity Gradients on Blunt Axisymmetric Bodies. NASA TM X-1067, 1965.

17. Eggers, A. J., Jr.; Hansen, C. Frederick; and Cunningham, Bernard E.: Stagnation-Point Heat Transfer to Blunt Shapes in Hypersonic Flight, Including Effects of Yaw. NACA TN 4229, 1958.

18. Howard, Floyd G.: Single-Thermocouple Method for Determining Heat Flux to a Thermally Thick Wall. NASA TN D-4737, 1968.

NASA-Langley, 1970 —— 33 L-6850

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
WASHINGTON, D.C. 20546

OFFICIAL BUSINESS

FIRST CLASS MAIL

POSTAGE AND FEES PAI
NATIONAL AERONAUTICS
SPACE ADMINISTRATIO

POSTMASTER: If Undeliverable (Section 1
Postal Manual) Do Not Re

"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."

— NATIONAL AERONAUTICS AND SPACE ACT OF 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include conference proceedings, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Technology Surveys.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION DIVISION

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Washington, D.C. 20546