NASA CONTRACTOR REPORT

NASA CR-2435

NASA CR-2435

LATERAL CONDUCTION EFFECTS ON HEAT-TRANSFER DATA OBTAINED WITH THE PHASE-CHANGE PAINT TECHNIQUE

by George Maise and Michael J. Rossi

Prepared by
GRUMMAN AEROSPACE CORPORATION
Bethpage, N.Y. 11714
for Langley Research Center

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • AUGUST 1974

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-2435		

4. Title and Subtitle	5. Report Date
LATERAL CONDUCTION EFFECTS ON HEAT-TRANSFER DATA OBTAINED WITH THE PHASE-CHANGE PAINT TECHNIQUE	August 1974
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
George Maise, Michael J. Rossi	
	10. Work Unit No.
9. Performing Organization Name and Address	
Grumman Aerospace Corporation Bethpage, New York 11714	11. Contract or Grant No. NAS-1-11818
	13. Type of Report and Period Covered Contractor Report
12. Sponsoring Agency Name and Address	
National Aeronautics and Space Admin. Washington, D.C. 20546	14. Sponsoring Agency Code

15. Supplementary Notes

Final report.

16. Abstract

A computerized tool "CAPE" (Conduction Analysis Program using Eigenvalues) has been developed to account for lateral heat conduction in wind tunnel models in the data reduction of the phase-change paint technique. The tool also accounts for the effects of finite thickness (thin wings) and surface curvature. A special reduction procedure using just one time of melt is also possible on leading edges. A novel iterative numerical scheme was used with discretized spatial coordinates but analytic integration in time to solve the inverse conduction problem involved in the data reduction.

A "yes-no" chart is provided which tells the test engineer when various corrections are large enough so that CAPE should be used.

The accuracy of the phase-change paint technique in the presence of finite thickness and lateral conduction is also investigated.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Heat Transfer Finite Slab Wind Tunnel Testing Phase-Change Paint Lateral Conduction	STAR Category 33

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	151	$5.00

* For sale by the National Technical Information Service, Springfield, Virginia 22151

FOREWORD

This study was carried out for NASA by Grumman's Advanced Development Office under the Shuttle Research and Technology Project (W. Ludwig, Manager; Dr. G. DaForno, Aerothermo Mgr.). The contract number is NAS1-11818, March 1972 to October 1973. Many of the results presented in this report were initially obtained under an in-house study.

M. Rossi developed the numerical method and the DETRAD subroutine package. Besides the authors' efforts, contributions were also provided by A. Jameson, who initially suggested the numerical approach and was consultant to the study, and by G. DaForno, who developed some points on the error amplification and the 'yes-no' chart, obtained the computer time minimization data, and worked out the extension to temperature-dependent properties and radiation. G. DaForno also assembled the results and wrote portions of the report.

LATERAL CONDUCTION EFFECTS ON HEAT-TRANSFER DATA OBTAINED WITH THE PHASE-CHANGE PAINT TECHNIQUE

By George Maise, Michael J. Rossi

Grumman Aerospace Corporation

SUMMARY

A computerized tool "CAPE" (Conduction Analysis Program using Eigenvalues) has been developed to account for lateral heat conduction in wind tunnel models in the data reduction of the phase-change paint technique. The tool also accounts for the effects of finite thickness (thin wings) and surface curvature. A special reduction procedure using just one time of melt is also possible on leading edges. A novel iterative numerical scheme was used with discretized spatial coordinates but analytic integration in time to solve the inverse conduction problem involved in the data reduction.

A "yes-no" chart is provided which tells the test engineer when various corrections are large enough so that CAPE should be used.

The accuracy of the phase-change paint technique in the presence of finite thickness and lateral conduction is also investigated.

TABLE OF CONTENTS

NOMENCLATURE

Symbols

a	=	half width of "top-hat" h distribution
a_{ij}	=	elements of coefficient matrix A
A	=	coefficient matrix
A_{ij}	=	conduction area between elements i and j
c_p	=	specific heat
f_i	=	departure of computed surface temperature at t_m from phase-change temperature, Eqn. (11)
G_{ij}	=	influence coefficient matrix, Eqn. (12) and (14)
h	=	heat transfer coefficient
$\bar{h}$	=	modified heat transfer coefficient, Eqn. (2)
H	=	enthalpy
k	=	thermal conductivity; also number of surface elements
$\bar{k}$	=	modified thermal conductivity Eqn. (3)
l	=	slab half thickness
M	=	Mach number
N	=	number of elements
p	=	pressure
Pr	=	Prandtl number
q	=	heat flux
Q	=	correction due to TD properties and $\epsilon \neq 0$
r	=	recovery factor
R	=	radius
s	=	distance along surface
t	=	time
T	=	temperature
u_e	=	velocity along body surface
v	=	injection velocity
V	=	matrix of eigenvectors of A
ΔV	=	volume of element
Δx	=	distance between elements (center to center)
α	=	angle of attack; also thermal diffusivity
ϵ	=	emissivity
λ	=	eigenvalue
Λ	=	matrix of eigenvalues, also sweep angle
μ	=	viscosity
ρ	=	density
τ_m	=	$\alpha t_m / l^2$, nondimensional melting time
θ	=	wedge angle

Symbols (cont.)

Subscripts

AW	=	adiabatic wall
bg	=	background
CP	=	constant material properties
e	=	edge
E	=	effective
i	=	i-th element
init	=	initial, $t = 0$
m	=	melting
r	=	row number of matrix
s	=	column number at matrix
sp	=	stagnation point
TD	=	temperature dependent
(o)	=	starting value (one-dimensional, semi-infinite)
(n)	=	n-th trial of h_i
o	=	stagnation point
1	=	surface 1 of slab-like geometry
2	=	surface 2 of slab-like geometry
∞	=	free stream

Notation

$$[V_{11}/V_{21}] = \begin{bmatrix} V_{11} \\ V_{21} \end{bmatrix}$$

INTRODUCTION

Subject

The subject of this study is the data reduction of the phase change paint technique, a well known technique (ref. 1 & 2) for obtaining the heat transfer coefficient from wind tunnel model tests. Briefly, the model of interest is covered with a paint that melts at a known temperature and then at each point on the model surface the melting time is measured, i.e., the time elapsed from the instant the heat transfer coefficient becomes steady. The heat transfer coefficients h are then deduced via a calculation of the model temperatures with a step-like time history for the h's.

Specifically, the situation studied here is that in which the spatial variations in the heat loads imposed on the model are so large that lateral conduction in the model must be accounted for to obtain accurate heat transfer coefficients. Another situation of the same type occurs when lateral conduction is caused by the model geometry; for example, lateral conduction is generated in a curved slab of variable thickness under spacially constant heat loads. In both situations we place a restriction that it is still possible to isolate a portion of the model where the problem is two dimensional. Two such cases that occur commonly are (1) a slab ('thin' or 'thick') impinged upon by a shock, where in the direction locally normal to the shock trace the problem can be considered two dimensional; and (2) a leading edge (l.e.) of a model wing or fin, where along cuts normal to the l.e. conduction is also essentially two-dimensional. Besides these two common occurrences, an arbitrary two-dimensional problem is also considered in this study as indicated in fig. 1.

Two types of data reduction problems can be posed for the situation in fig. 1:

1) the distribution of the times of melt is given on the surface subjected to heat loads and the corresponding h's are to be derived -- this is the typical problem in ref. 1 to 4. (Naturally the surfaces over which no data are given are to be taken as adiabatic, or if at infinity, at the initial temperature throughout the transient.)

2) the time of melt is given at one point on a leading edge and the entire h distribution is to be derived -- this is a somewhat novel problem proposed by R. A. Jones of the NASA Langley Research Center and suggested by the fact that on leading edges of very small radius (0.05 inches for the wing of a typical 1 ft. shuttle orbiter model), it is too difficult to determine experimentally the propagation of the melting line. At most, the minimum time of melt around the l.e. can be determined. Approximately, this minimum melt time should occur near the maximum heating point or near the stagnation point. Naturally, in order to reduce phase-change paint data giving only t_m at the stagnation point, the following information is needed:

   a) stagnation point location in the l.e. section considered, if $\alpha \neq 0$.

   b) pressure distribution around the l.e.

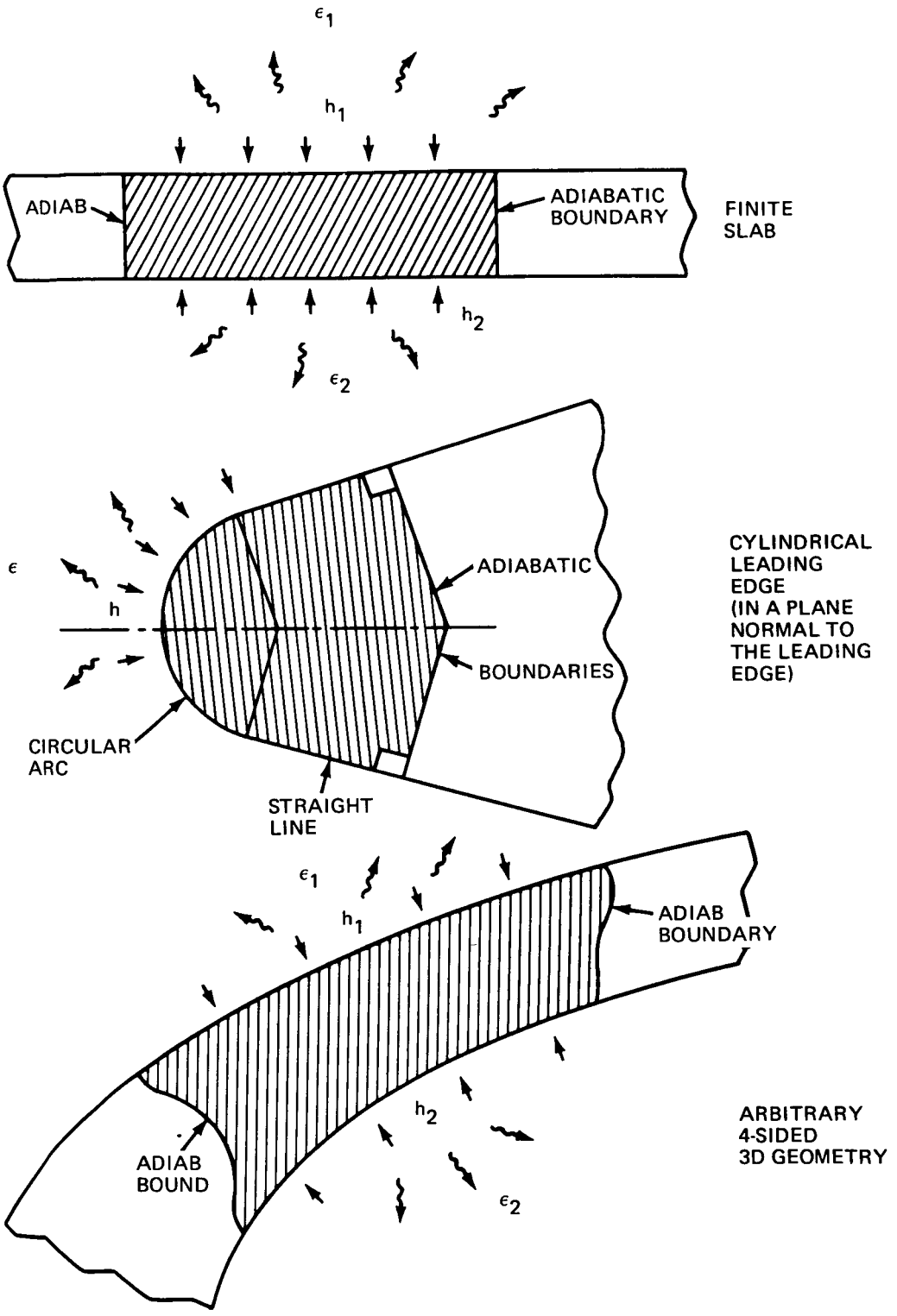


FIGURE 1. GEOMETRIES CONSIDERED IN THE STUDY

c) body entropy

d) T_{aw} distribution around the l.e., and naturally

e) a formula for the heat transfer coefficient distribution.

This information has to be obtained theoretically. To be realistic it is also imperative to use:

a) relatively simple formulae, preferably not involving table look-up or iterations.

b) infinite-cylinder approximation for the l.e. boundary layer.

Indeed, approximate formulae do exist that appear adequate for the task. However, this data reduction procedure has not yet been evaluated, as before this study there was no tool for carrying out the calculations. In what follows, this data reduction problem is referred to as 'special problem for the l.e.'

In both types of problems, the normal one and the special problem for the l.e., it is natural to include radiation from the model surface and model material properties variable with temperature. These two effects are grouped together because they both make the reduction non linear and therefore are handled in a straightforward manner only by numerical methods. For the current model materials (Stycast, other epoxies, etc.), both these effects are small. In particular the effects of temperature-dependent (TD) have been found to be very small both in one-dimensional and two-dimensional cases. Note that the TD solution should be compared to a constant property (CP) solution where the CP properties are properly chosen. The rational way is to evaluate k and c_p as the average of the values at T_{init} and T_m, as indicated in fig. 2. Then, during the time history from T_{init} and T_m the properties are correct on the average. Of course, at later times, when $T > T_m$, the properties are locally less accurate, but to have an effect, this inaccuracy would have to influence regions far away where the paint melts later. It is this influence that is minimal. For example, in a typical one-dimensional case with a Stycast model there was only 0.4% difference between h_{TD} and h_{CP}. Similar results were found in two dimensional cases (calculations were done in direct problems using AGTAP (ref. 5). In spite of the fact that they are small, that the k and c_p variations with temperature are not always available and the fact that the properties inhomogeneity can be more important, radiation and TD properties are here considered for completeness.

Problem

Two specific problems are attacked in this study, i.e.:

i) the development of a numerical method and relative computer code for reducing phase change paint data for the three geometries of fig. 1 and for both types of experimental inputs mentioned above;

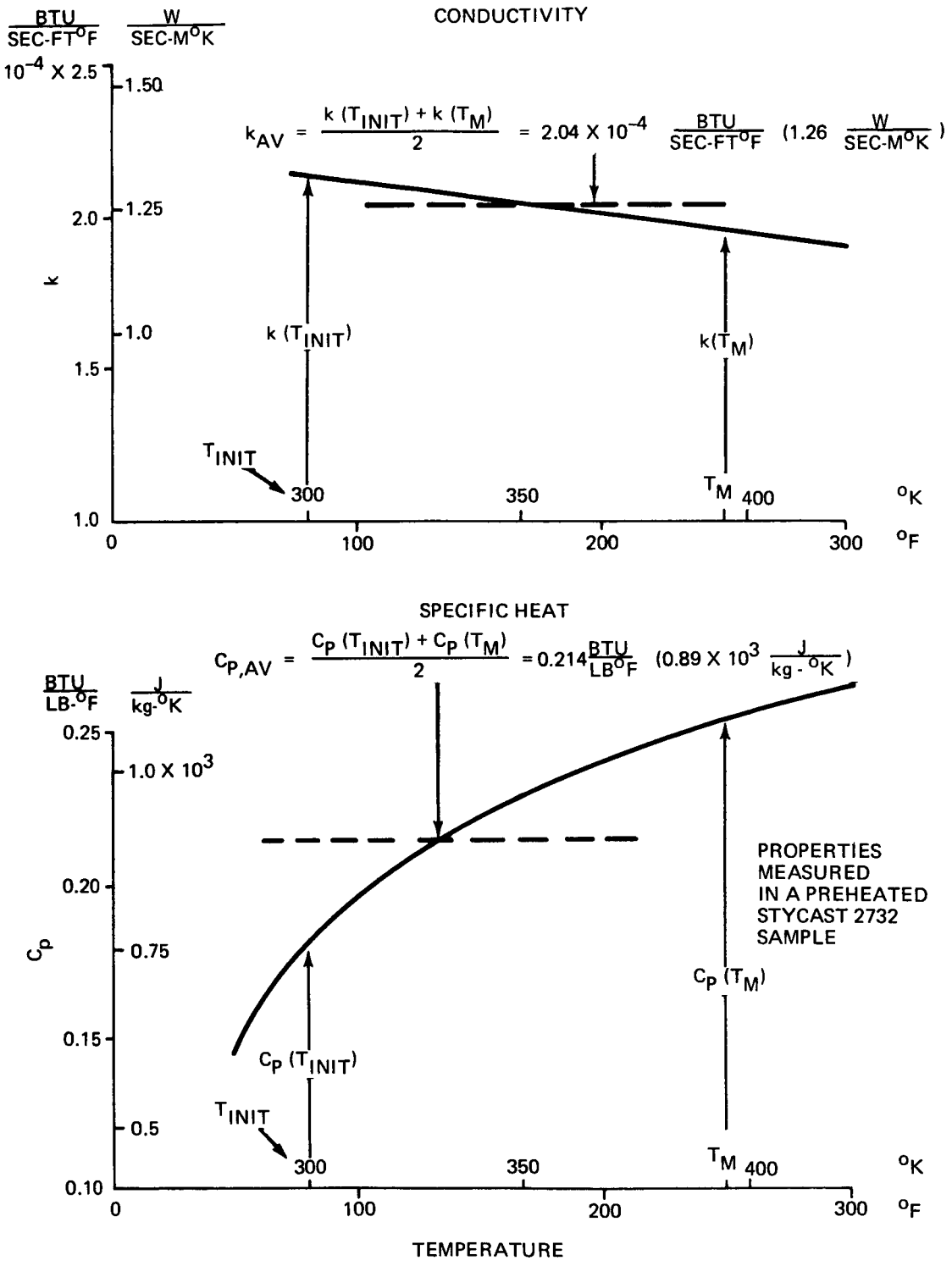


FIGURE 2. THE SIMPLEST WAY OF CHOOSING SUITABLE AVERAGE THERMAL PROPERTIES FOR THE MODEL MATERIAL

ii) the preparation of a 'yes-no' correction chart, that quickly tells the test engineer when the lateral conduction corrections are large enough (say 10% on h) to make reduction by 'semi-infinite slab' (ref. 1) or finite slab (ref. 3, 4) too inaccurate and therefore the use of the computer code above necessary.

The chart should, of course, be simple. Moreover, the computerized data reduction should allow for variable material properties and radiation. Naturally, the code itself should be easy and quick to use, especially as far as geometry input (e.g., grid lay-out, elements selection, conduction paths calculations).

Target Run Times

To produce a practical computer tool for the data reduction in the presence of (two-dimensional) lateral conduction, we set a maximum run time on the CDC 6600 on the order of 6 minutes for a typical slab case with h obtained to an accuracy around some 1 to 2% (which is an adequate value to impose on the numerical method, as it is an order of magnitude smaller than the absolute accuracy of the experimental h).

Prospects for Data Reduction in Three-Dimensional Geometries

Currently, the maximum service temperature of model materials such as Stycast imposes various limitations on the test conditions the model can be tested at, the maximum run time of the test and the number of times a model can be used. Metal, e.g., stainless steel models would eliminate these drawbacks of the epoxy models, at the price (among other things) of a much larger conductivity and therefore fully three-dimensional lateral conduction. Reducing data in the presence of 3D lateral conduction effects would require in the numerical method a large number of elements to describe the model (or a 3D portion of it). While not a part of this study, such a prospective extension of the numerical method should be kept in mind. As a target, one may perhaps put the maximum run time, on the CDC 6600, at 30'. Of course, the feasibility and practicality of such an elaborated data reduction would have to be judged on many more counts than the mere low computer time.

NUMERICAL METHOD FOR THE LATERAL CONDUCTION INVERSE PROBLEM

Possible Approaches

The most obvious approaches that could be taken, are finite difference methods (implicit or explicit) or finite element methods for the space variables and time, or a mixture of finite element for the space variable and finite differences for time. These are the methods used in the current computer tools for direct heat conduction problems, typically SINDA (ref. 6), AGTAP (ref. 5), etc. Along these lines the one-dimensional inverse problem has been also solved successfully (ref. 4.)

Unfortunately, none of these conventional approaches (finite difference or finite element, implicit or explicit) seems to offer much hope for a practical tool for two-dimensional inverse problems because of the extremely large machine times required. The basic reason is that, in iterating on the h_i, once a set of h_i is guessed, one needs influence coefficients of the type $\partial T_i/\partial h_j$ and to obtain these coefficients it is necessary to calculate a very large number of direct problems. To try to put the problem on quantitative, even if approximate, terms, figure 3 shows some comparisons of running times for explicit and implicit methods coupled with numerically determined influence coefficients (Newton's method). For the purpose of comparison, it was assumed that all methods converge in three iterations, which tends to put the method developed below at a disadvantage.*

The explicit method suffers from the fact that for physically small elements the critical time step beomces very small. Furthermore, if stainless steel models are to be considered, the critical time step is further reduced by a factor of 12 over that of Stycast with the corresponding increase in computer time. Thus, the interesting conclusion follows, from fig. 3, that the explicit method has to be ruled out.

*The curves in fig. 3 were determined as follows. For the explicit method, we ran a 121-element Stycast problem (heated on one side) in the direct mode with a thermal analyzer (ref. 5). To do the corresponding inverse problem with the thermal analyzer, the influence coefficients would be determined by perturbing all the h_i's in turn and rerunning the program. This would lead to a total running times of 21 minutes. The indicated proportionality of computer time with n^2 (n number of elements) would result if the number of surface elements to the total number of elements were kept in the ratio of 1 to 11. For the implicit method we timed one of our very efficient finite-element structural mechanics programs which resembles (mathematically) the implicit heat transfer problem. To do the corresponding inverse problem the same arguments that were used for the explicit method apply. A variation of computer time with n^2 was assumed because the implicit method involves matrix operations. Finally, for the method developed here, fig. 3 presents running times actually clocked for an early, nonoptimized version of both method and code.

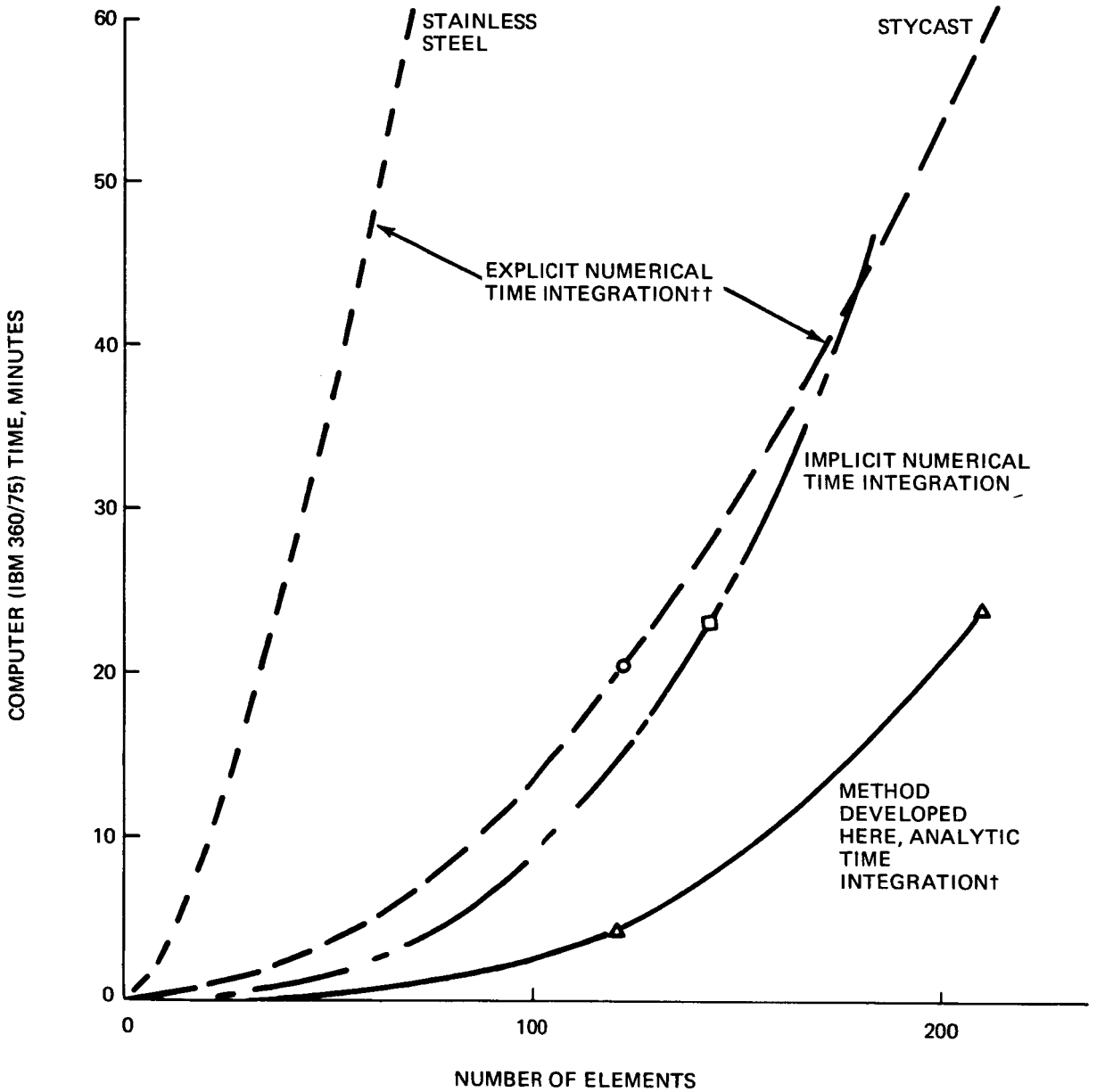


FIGURE 3. ESTIMATED COMPUTER TIME FOR TYPICAL INVERSE PROBLEMS VIA VARIOUS NUMERICAL METHODS

Computer time with the implicit method depends directly on the time step that is chosen. Figure 3 assumes that for accurate computations 50 time steps are required. The computer time could, however, be lowered at the sacrifice of accuracy in the time integration. However, the basic method does not possess the potential for significant reductions in computer time, and therefore, it is very doubtful that it could ever handle three-dimensional geometries within the 30' target time.

Method Developed Here

In order to deal with the difficulties just mentioned which are characteristic of the problem at hand, a different idea (suggested in this context by A. Jameson) was developed as follows.

Imagine to discretize as usual the structure, in lumps for example in the spirit of the finite element methods. Write the ordinary differential equation governing the temperature history in each element. The idea is now not to discretize the time variable and carry out the integration in time numerically, but instead to leave this variable continuous and carry out the integration analytically. The method therefore belongs to the class of the semi-descretized variables or the hybrid analytical-numerical methods. Hybrid methods have not yet received much attention, even though in the literature a few cases have been reported. Their potential seems to lie in their ability to incorporate the best aspects of both numerical and analytical methods. In this essential aspect the method developed here differs from the usual finite-element methods used in heat conduction.

The great advantage in not discretizing the time emerges when the inverse problems of phase change paint data are to be solved. These inverse problems are described by multipoint boundary value problems in time. Since the integration here is done analytically, the result is an explicit expression for the temperatures of all elements as functions of time and all the heat transfer coefficients, i.e.:

$$T_i = T_i(t,\ h_1,\ h_2 \ldots h_k)$$

When the $t_{m,i}$ and T_m are given, this equation has to be solved iteratively for the h_i's. An efficient iteration requires the influence coefficients $G_{ij} = \partial T_i / \partial h_j$. Two approaches are avilable to obtain these influence coefficients. First, since there is an analytical expression for T_i, the G_{ij} can be obtained directly by differentiating this expression with respect to all the h_i's. With any numerical scheme that discretizes time, the time integration must be repeated k times (k = number of surface elements) to obtain these influence coefficients. In other words, the usual methods, whether implicit or explicit, require k+1 integrations in time, while our scheme needs just one. As the number of points increases and also the melt times become larger (driven by the quest for higher accuracy in the data reduction, larger model conductivity and higher model service temperatures), k integrations in time are bound to cost, in terms of computer time and stor-

age, more than the calculation of an explicit expression for G_{ij}. The other approach is to calculate the G_{ij}, from the exact 'analytical' or 'infinitesimal' expression, from the approximate finite-difference expression:

$$G_{ij} \equiv \frac{\partial T_i}{\partial h_j} \simeq \frac{T_i(h_1, h_2, h_j + \Delta h_j \ldots . h_k) - T_i(h_1 h_2 \ldots h_k)}{\Delta h_j}$$

with a small but finite Δh_j. This requires one extra set of eigenvalue calculations per surface point. Therefore in this case we trade the $k+1$ time history calculations of the conventional methods with $k+1$ eigenvalue calculations (k of which have only one h perturbed).

In both cases, then, this unconventional analytical-numerical method permits the substitution of the numerical integration in time by one or more eigenvalue calculations and opens the possibility of taking advantage of the considerable amount of work done recently on fast eigenvalue calculations in the areas of aircraft control theory and structural analysis*. This gives latitude for considerable advances in computer time, storage and size of problems handled. Such latitude is not evident in the typical heat conduction methods.

Naturally, all this needs to be put on a quantitative basis. Figure 3 anticipates some indications of the machine time requirements for the method developed here. One point to be kept in mind is that these times are for a non-optimized version of method and code and that the method itself has considerable potential for time reduction.

Clearly, this hybrid analytical-numerical method can be carried out within the framework of finite element approaches or within the framework of finite difference approaches as far as the space variables are concerned. For the problem at hand, there may be only minor differences between these two approaches in the final discretized equations. We patterned our space discretization after the simplest version of the finite element approach, namely uniform distribution of each quantity within each element, in the belief that what is most important for large improvements in the numerics is an imaginative approach on the broad issues rather than sophistication in detailed matters. However, one point of the spatial discretization has been paid attention to, namely boundary elements are treated in a (very simple) way that assures--within the uniform distribution assumption -- maximum accuracy in imposing the boundary conditions, as will be seen below.

*The method developed here was transplanted from control theory, but most of the routines and eigenvalue-eigenvectors numerics originate from structural analysis.

Equations for the Basic Case

The key equations in our method are now presented. We give in this section the basic case of variable T_{AW}, no radiation, times of melt that are all given.

After the model's structure, or more likely, a portion of it is suitably subdivided into n elements, the temperature response of each element i can be represented by

$$\rho C_p \Delta V_i \frac{dT_i}{dt} = h_i A_i (T_{AW,i} - T_i) + \sum_{j \neq i} k \frac{A_{ij}}{\Delta X_{ij}} (T_j - T_i) \tag{1}$$

(h_i term suppressed for interior elements; j elements adjacent to i share with i the infinite area A_{ij}). The symbols are defined in the nomenclature. For simplicity, redefine new coefficients.

$$\bar{h}_i = h_i A_i \tag{2}$$

$$\bar{k}_{ij} = \frac{k A_{ij}}{\Delta X_{ij}} \tag{3}$$

The equation can be rearranged to read

$$\rho C_p \Delta V_i \frac{dT_i}{dt} = -(\bar{h}_i + \sum_{j \neq i} \bar{k}_{ij}) T_i + \sum_{j \neq i} \bar{k}_{ij} T_j + \bar{h}_i T_{AW,i} \tag{4}$$

These equations constitute a system of n first-order linear differential equations with constant coefficients. Because of the variable T_{AW}, these equations are inhomogeneous. Writing (4) in vector notation

$$M \frac{dT}{dt} = BT + F, \quad T(0) = T_{init}$$

or alternatively

$$\left.\begin{aligned} \frac{dT}{dt} &= M^{-1/2} A M^{1/2} T + M^{-1} F \\ T(0) &= T_{init} \end{aligned}\right\} \tag{5}$$

where M is the diagonal matrix made up with the $\rho C_p \Delta V_i$, $A \equiv M^{-1/2} B M^{-1/2}$ is a symmetric matrix where

$$b_{ii} = -(\bar{h}_i + \sum_{i \neq j} \bar{k}_{ij}), \quad \bar{h}_i = 0 \text{ if i is interior element} \tag{6}$$

$$b_{ij} = \bar{k}_{ij} \tag{7}$$

and F is a constant vector

$$F = \begin{bmatrix} \bar{h}_1 T_{AW,1} \\ \vdots \\ \bar{h}_i T_{AW,i} \\ \vdots \\ \bar{h}_k T_{AW,k} \end{bmatrix} \tag{8}$$

The solution to the initial value problem (5) is written in the usual fashion as

$$T = T_\infty + M^{-1/2} V e^{\Lambda t} V^T M^{1/2} (T_{init} - T_\infty) \tag{9}$$

Where the $t \to \infty$ temperature T_∞ is the solution of the RHS of (5) set equal to zero:

$$M^{-1/2} A M^{1/2} T_\infty + M^{-1} F = 0 \tag{9a}$$

and where Λ is a diagonal matrix formed with the eigenvalues λ_i of the matrix A, and V is the matrix of eigenvectors of matrix A, V^T is transposed of V.† Thus, once the eigenvalues and eigenvectors are determined with one of the standard subroutines, the temperature of any element at any time t is readily evaluated by eq. (9). For the problem at hand, only the temperatures of surface elements at $t_{m,i}$ are of interest.

Equation (9) solves the direct problem, since it gives the temperature distribution when T_{init} and $\bar{h}_i$ are given. The problem of interest is really one of determining the parameters $\bar{h}_i$ from a partial knowledge of the direct solution ($T = T_{m,i}$ at $t = t_{m,i}$ for each surface element). Therefore, the solution of the inverse problem is given by equations (9a) and (10).

$$\left.\begin{aligned} &T_{m,i} = \delta_i^T M^{-1/2} V e^{\Lambda t_{m,i}} V^T M^{1/2} (T_{init} - T_\infty) + T_\infty \\ &\Lambda,\ V \text{ and } T_\infty \text{ are functions of } \bar{h}_j \end{aligned}\right\} \tag{10}$$

Here δ_i^T is the i-th row of the unit matrix of n elements. Equation (10) is a system of transcendental equations in the unknown $\bar{h}_j$. By solving it one determines $\bar{h}_j$.

In order to solve eq. (10), an iteration in the $\bar{h}_j$ is used. As starting values for the iteration, the $\bar{h}_j$ are first computed using the one-dimensional semi-infinite slab theory (ref. 7). The temperature of the surface elements corresponding to the given $t_{m,i}$ are computed by eq. (9). This is of course the RHS of eq. (10). These temperatures should all equal T_m. Considering h a vector parameter, the errors in the temperature are thus

$$f_i(\bar{h}) = T_m - \delta_i^T T(t_{m,i}) \tag{11}$$

To obtain new values for the $\bar{h}_j$'s, use is made of Newton's method of iteration. It is necessary to evaluate the influence coefficients G_{ij}, (G in the matrix form), where

---

†If T_{AW} is constant in space, $T_\infty = T_{AW}$ is also constant. If $H \equiv 0$, $T_{\infty i} = (\sum_k^{1,n} m_k T_{init,k}) / (\sum_k^{1,n} m_k)$ is constant which of course, for T_{init} constant in space, becomes T_{init} the same constant. These results show that the discrete model is consistent.

$$G_{ij} = -\frac{\partial f_i}{\partial \bar{h}_j} = \delta_i^T \frac{\partial T(t_{m,i})}{\partial \bar{h}_j} \tag{12}$$

Since the $T(t_{m,i})$ are known, the derivatives, i.e., the G_{ij} can be calculated formally. The result is

$$G_{ij} = \delta_i^T \bar{M}^{1/2} V[m_j^{-1} Q_{ij}(t_{m,i}) V^T M^{1/2}(T_{init} - T_\infty) + (e^{\Lambda t_{m,i}} - I)\Lambda^{-1} V^T M^{-1/2} \Delta_{jj}(T_{AW} - T_\infty)] \tag{13}$$

$$Q_{jrs} = \begin{cases} V_{jr} V_{js} \dfrac{e^{\lambda_r t_{m,i}} - e^{\lambda_s t_{m,i}}}{\lambda_r - \lambda_s} & \lambda_r \neq \lambda_s \\ V_{jr} V_{js} t_{m,i} e^{\lambda_r t_{m,i}} & \lambda_r = \lambda_s \end{cases}$$

$$\delta_i = [0 \cdots 0 \overset{i}{1} 0 \cdots 0]$$

$$\Delta_{jj} = \begin{bmatrix} 0 \cdots \overset{j}{0} \cdots 0 \\ 0 \cdots 0 1 0 \cdots 0 \\ 0 \cdots 0 \cdots 0 \end{bmatrix} j$$

Although the expression for G_{ij} appears to be quite complicated, it is to be noted that V and λ_i are already available from the calculation of the $T(t_{m,i})$ (the RHS of eq. (10)), and the multiplication of terms in eq. (13) can be performed in such a way that matrix-by-matrix multiplication is avoided.

There is another possibility for calculating the G_{ij}. Rather than the infinitesimal expression (13), one could use the approximate finite-difference expression:

$$G_{ij} = \frac{\partial T_i}{\partial h_j} \cong \frac{T_i(\bar{h}_1, \bar{h}_2 \ldots \bar{h}_j + \Delta h_j \ldots \bar{h}_k) - T_i(\bar{h}_1, \bar{h}_2 \ldots \bar{h}_k)}{\Delta h_j} \tag{14}$$

with a small but finite Δh_j. This requires the evaluation of one extra set of eigenvalues and eigenvectors (for the perturbed sets of $\bar{h}_j$) for each i and the corresponding T_i at the time of melt $t_{m,i}$. Naturally there is no point in using a central difference that will require two extra sets of eigenvalues and eigenvectors. Later on, we will discuss the question of which of the expression (13) or (14) for the G_{ij} is most convenient.

Once, however, the G_{ij} are determined during each $\bar{h}_j$ in the following way.

For a change $\delta\bar{h}$ in the parameter vector, eq. (12) gives

$$\delta f = G\delta\bar{h} + 0\|\delta\bar{h}\|^2 \tag{15}$$

To make the errors f vanish at the next trial f should equal $-f$ from the previous trial. Ignoring the higher order terms, eq. (15) can be solved for the required change in $\bar{h}$:

$$\delta\bar{h} = -G^{-1}f \tag{16}$$

To carry out this step requires an inversion of the G matrix. It is to be noted that this matrix is typically much smaller (it is equal to the number of surface elements on which h is unknown) than the A matrix. The updated set of $\bar{h}_j$ can be formed as

$$\delta\bar{h}_j^{(1)} = \bar{h}_j^{(0)} + \delta\bar{h}_j \tag{17}$$

The above procedure is repeated until eq. (10) is satisfied. When this occurs, the desired h_j's have been determined.

As will be shown below, the iteration scheme just described is very powerful. For example, convergence to 0.1% error is generally achieved after three to four iterations, for the most severe case, when the times of melt at all surface points are given.

Temperature Dependent Properties and Radiation

Radiation and TD properties* can be simply included in the numerical method that we have developed. This is not quite obvious at first sight since the method depends upon the analytical solution for the ordinary differential equations and we are able to write the analytical solution only for problems linear in time. The usual radiation formulae or material properties variable with temperature make the problem non-linear.

The analysis can be extended to handle non-linearity in a manner that preserves the power of the numerical method. The extension amounts to "lag behind" in the non-linear terms so as to build up the exact solution through a succession of iterates, each of which is linear. The iterates are not new ones, rather the ones already necessary for h_j. The key point is that the analytical time integration for each element is maintained.

With radiation and variable material properties, the starting equation replacing (1) is:

$$\rho C_{pi}(T_i)\Delta V_i \frac{dT_i}{dt} = h_i A_i (T_{AW,i} - T_i) - \sigma\epsilon_i (T_i^4 - T_{BG,i}^4) A_i$$

$$+ \sum_{j \neq i} k_{ij}(T_i, T_j) \frac{A_{ij}}{\Delta X_{ij}} (T_j - T_i) \tag{18}$$

with the same convention as in (1) to suppress the first two terms in the RHS for interior elements and the same conventions for j.

---

*Naturally, radiation with arbitrary emissivity and background temperature for each surface; and also arbitrary $C_p(T)$ and $k(T)$, supplied, for example, in the usual tabular form. ρ is taken as constant.

Using constant but suitable values for c_p and k

$$\bar{c}_p = \frac{c_p(T_{init}) + c_p(T_m)}{2}$$

$$\bar{k} = \frac{k(T_{init}) + k(T_m)}{2}$$

the equation is now recast as

$$\rho \bar{c}_p \Delta V_i \frac{dT_i}{dt} = h_i A_i (T_{a\,w,i} - T_i) + \sum_{j \neq i} \bar{k} \frac{\Delta_{ij}}{\Delta X_{ij}} (T_j - T_i)$$
$$+ Q_i (T_i(t), T_j(t)) \tag{19}$$

$$Q_i = -\sigma \epsilon_i (T_i^4 - T_{BG,i}^4) A_i$$
$$+ \sum_{j \neq i} \left(\frac{k_{ij}}{\bar{k}} - 1 \right) \bar{k} \frac{A_{ij}}{\Delta X_{ij}} (T_j - T_i)$$
$$+ \left(1 - \frac{c_{pi}}{\bar{c}_p} \right) \rho \bar{c}_p \Delta V_i \frac{dT_i}{dt} \tag{20}$$

Naturally Q_i summarizes all the non-linearity in time and is made up of three effects as is clear from inspection of (20).

Proceeding as before, equation (5), (9), (10) and (13) are now replaced by (considering Q a known function and maintaining the same definition of T_∞):

$$\frac{dT}{dt} = M^{-1/2}\, AM^{1/2}\, T + M^{-1}\, F + Q \tag{6'}$$

$$T = M^{-1/2}\, V e^{\Lambda t}\, V^T M^{1/2} (T_{init} - T_\infty) + \int_0^t M^{-1/2}\, V e^{\Lambda (t-\tau)}\, V^T M^{1/2}\, Q(\tau)\, d\tau + T_\infty \tag{9'}$$

$$T_{M,i} = \delta_i^T M^{-1/2}\, V e^{\Lambda t_{m,i}}\, V^T M^{1/2} (T_{init} - T_\infty)$$
$$+ \int_0^{t_{m,i}} M^{-1/2}\, V e^{\Lambda (t_{m,i} - \tau)}\, V^T M^{1/2} Q_i(\tau) d\tau + T_\infty \tag{10'}$$

$$G_{ij} = \delta_i^T \Big[M^{-1/2}\, V Q_j\,(t_{m,i})\; V^T M^{1/2} (T_{init} - T_\infty)$$
$$+ M^{-1/2}\, V Q_j\,(t_{m,i})\; V^T M^{1/2} Q_i\,(t_{m,i})$$
$$+ M^{-1/2}\, V H\,(t_{mi})\; V \frac{\partial Q_i\,(t_{m,i})}{\partial h_j} \Big] \tag{13'}$$

$$\text{matrix H} \begin{cases} H_{rs} = 0 & r \neq s \\ H_{rr} = \dfrac{e^{\lambda_r t_{m,i}} - 1}{\lambda_r} & r = s \end{cases}$$

In these equations all the first terms are clearly those obtained in the absence of radiation and material properties variation. In particular, the equation solved for the inverse problem looks simply

$$T_{m,i} = (T_{m,i})_{\substack{\text{NO RAD,}\\ \text{CP}}} + \int_0^{t_{m,i}} M^{-1/2}\, V e^{\Lambda(t_{m,i}-\tau)}\, V^T M^{1/2} Q_i(\tau)\, d\tau \qquad (10'')$$

This equation that gives the $\bar{h}_j$ cannot be solved alone any more, but must be solved together with the integral equation (9′) that gives the $T_i(t)$. However, recalling that radiation and TD properties are not dominant effects, Q is relatively small and therefore an iteration in Q should rapidly converge. Therefore, in each $\bar{h}_j$ iteration, Q(t) can be taken approximately as the value computed a posteriori from the temperatures of the previous $\bar{h}_j$ iterate. In the first iteration, Q(t) = 0. Of course, as $\bar{h}_j$ converge to the exact values, so do the Q(t)s.

In this way linearity is maintained in each $\bar{h}_j$ iterate and the calculation method remains as indicated in the previous section. No new iteration for Q is needed. The only addition is the calculation of Q for equation (10″) and for this temperature history up to t_m is needed from equation (9′).

The integral in (10″) is discretized in the most straightforward fashion

$$T_{m,i} \simeq (T_{mi})_{Q=0} + \sum_{1}^{i,\, l_{max}} \frac{\Delta\tau}{2} [M^{-1/2}\, V e^{\Lambda(t_{m,i-\tau l+1})} V^T M^{1/2} Q_i(\tau_{l+1}) + M^{-1/2} V e^{\Lambda(t_{m,i-\tau 1})} V^T M^{1/2} Q_i(\tau_1)] \qquad (21)$$

$$\Delta\tau = \frac{t_{m,i}}{l_{max} - 1}$$

$$\tau_l = (l-1)\Delta\tau$$

l_{max} arbitrary

Equation (20) gives $Q_i(\tau_l)$ via the $T(\tau_l)$ from equation (9′), which is discretized just as (21).

No great sophistication is needed in the number of time intervals l_{max} nor in the discretization of the integral near $\tau = 0$ since the contribution of Q is altogether small.

In the same spirit, when using the analytical expression for the G_{ij}, we can retain (13) instead of (13′) even with radiation and TD properties. Experience with this method has been that only approximate values of the G_{ij} are needed to drive the $\bar{h}_j$ iterates to the solution in a few of iterations. Therefore, there is no need to add the small radiation correction to the G_{ij} whose only function is to drive the iteration.

Numerical results show that under normal conditions, radiation and TD properties do not affect the number of iterations and have very small effect on G_{ij}. However, the

extra calculations do cause an increase of computer time by a factor of about 2 for radiation and 3 for radiation plus TD properties. Naturally a method that integrates numerically in time would entail a small increase due to these two effects. However, since the effects appear to give a maximum change of the order of 1% in h_j for current situations, the present method allows to take advantage, for the majority of the applications, of the fastest version of the method. This advantage does not exist in the method requiring the numerical integration in time.

Geometry Discretization and Calculation Of Capacitances and Conductances

In this section we document the way in which the three types of geometries of interest (fig. 1) are discretized into elements and how capacitances and conductances are calculated. The calculations (grid lay-out and capacitances) are done in a standard way since they are automatized in the computer code so that the tedious labor of input preparation typical of multi-dimensional conduction codes is mostly eliminated.

The grid lay-out for the slab, the l.e. and the arbitrary four-sided two-dimensional geometry is illustrated schematically in figures 4, 5, and 6 respectively.

Some points to be noted are:

a) Control points. The temperature points are taken to be in the middle of the elements except for the surface elements, where the temperature points are taken on the surface. This procedure results in greater accuracy in surface temperature calculations, which is particularly important in phase-change paint applications. It eliminates the needs of extra-fine resolution near the surface.

b) Slab thermal capacitances and conductances are calculated as follows (see fig. 4):

$$C_i = \rho c_p \Delta x_i \Delta y_i \tag{22}$$

and

$$\bar{k}_{i,i+1} = k \frac{\Delta x_i}{\frac{1}{2}(\Delta y_i + \Delta y_{i+1})} \tag{23}$$

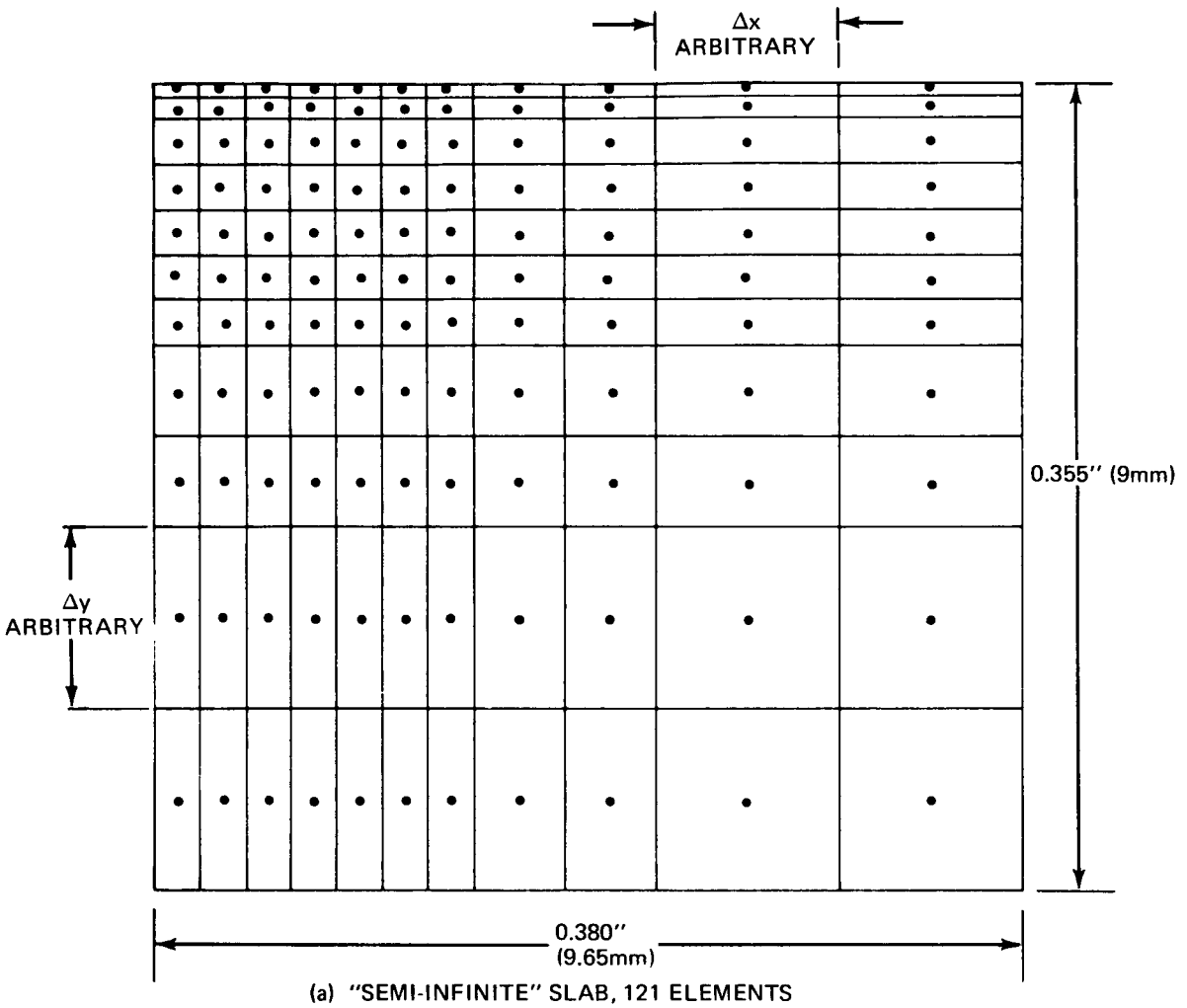
c) Leading Edge. Fig. 5 represents a geometry in a plane normal to the l.e. The cylindrical portion of the geometry is divided into elements by concentric circles and by rays. The wedge portion is divided into rectangular elements except near the centerline where the elements are trapezoidal. Conductances such as E-H, etc., see fig. 5, are taken to be zeros since these elements meet at a point. In the nose region conductances between elements are computed by the usual logarithmic relationships. For example:

$$\bar{k}_{A\text{-}D} = \frac{k}{\beta} \ln \frac{r_\delta}{r_\alpha} \tag{24}$$

and

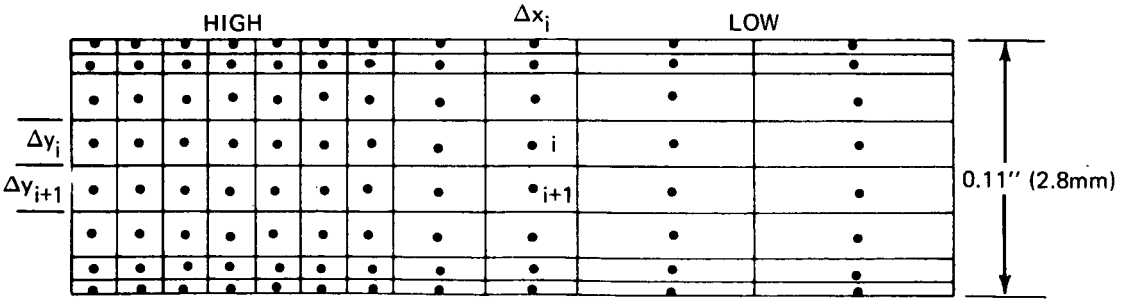
$$\bar{k}_{A\text{-}B} = k\beta \ln^{-1} \frac{r_\gamma}{r_\beta} \tag{25}$$

Δx, Δy ARBITRARY
NUMBER OF ELEMENTS IN BOTH DIRECTIONS ARBITRARY
2 EXAMPLES SHOWN WITH ACTUAL DIMENSIONS



(a) "SEMI-INFINITE" SLAB, 121 ELEMENTS

LATERAL GRADIENTS OF MELT TIMES:



(b) FINITE SLAB, 88 ELEMENTS

FIGURE 4. GRID LAYOUT FOR SLABS

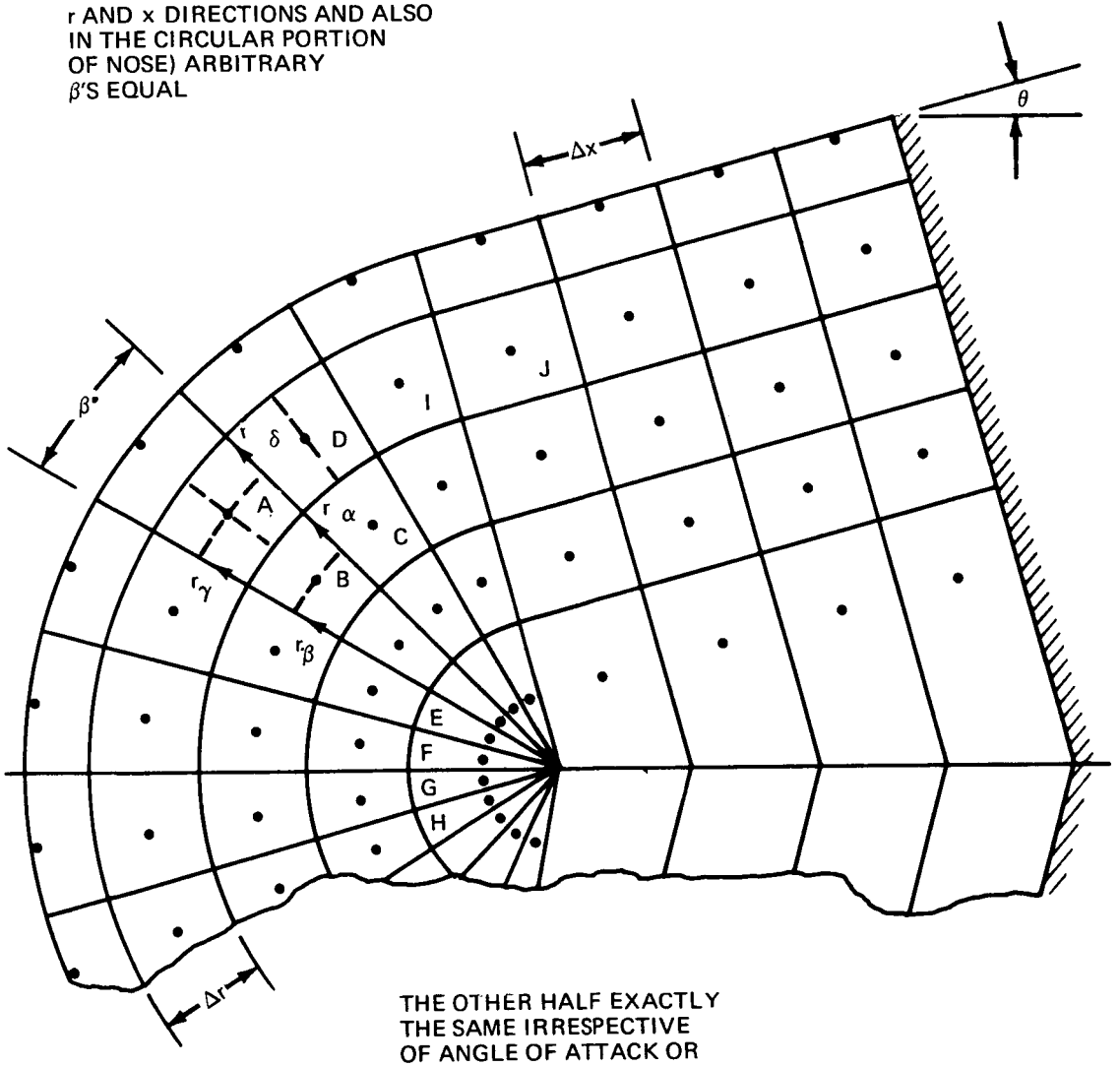


FIGURE 5. GRID LAYOUT FOR LEADING EDGES

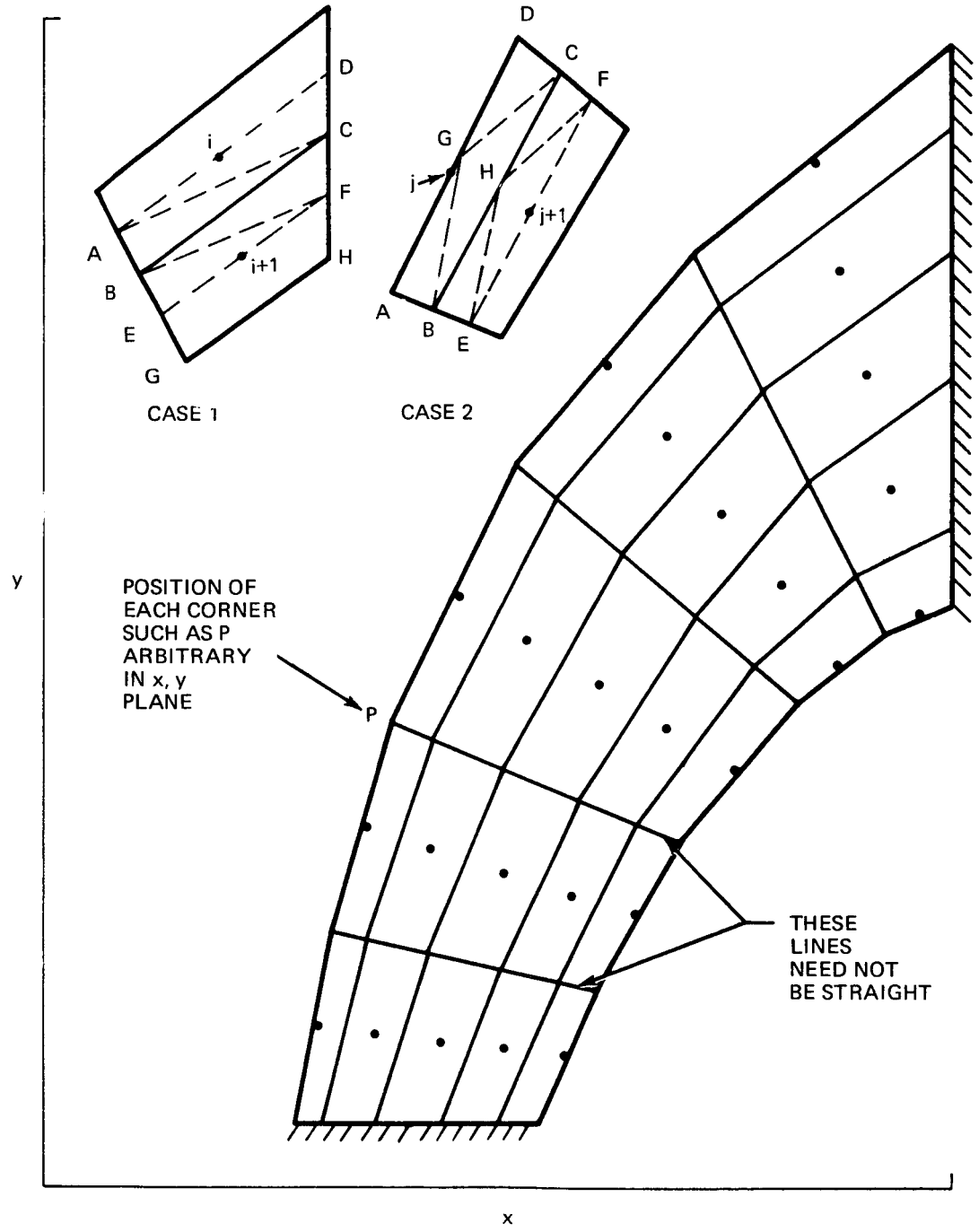


FIGURE 6. GRID LAYOUT FOR ARBITRARY GEOMETRIES.

One complication arises near the center of the circle. The conductance between adjacent elements such as F and G becomes infinite according to the above relationship. To circumvent this problem a small "hole" equal to 0.01 r_{CAP} is assumed for the center. The conductances between rectangular elements are computed in the same way as for the slab. For the conductances between dissimilar elements (e.g., I-J in fig. 5) are computed in two parts and added according to the conventional rules.

d) Arbitrary Four-sided Geometry. The geometry contour is assumed to be given pointwise. The grid lay-out is also specified pointwise as indicated in fig. 6. To compute by hand the conductances between these elements in this case would be an extremely tedious and time-consuming task. This task can be done automatically, when the coordinates (x, y) of the corners of the elements are given. To compute the conductances between arbitrary quadrilateral elements, use is made of the analysis given by Dusinberre (ref. 8). Dusinberre presents the relationships for computing conductances between triangular elements. Since all quadrilaterals can be subdivided into triangles, these relationships carry over to the problem at hand. Basically, two situations exist as shown in fig. 6. If the obtuse angles of the quadrilateral are on opposite corners, the resolution into acute triangles is accomplished by passing a diagonal through these corners (case 1). If the obtuse angles are adjacent, the quadrilateral must be divided into three triangles (case 2). The resulting conductances are given by the following equations:

case 1:

$$\bar{k}_{i,i+1} = \frac{1}{\frac{1}{\bar{k}_A} + \frac{1}{\bar{k}_B}} \tag{26}$$

where

$$\bar{k}_A = \tfrac{1}{2}\, k(\cot DAC + \cot ADC + \cot ABC + \cot ACB) \tag{27}$$

and

$$\bar{k}_B = \tfrac{1}{2}\, k(\cot CBF + \cot BCF + \cot BEF + \cot BFE) \tag{28}$$

case 2:

$$\bar{k}_{j,j+1} = \frac{1}{\frac{1}{\bar{k}_C} + \frac{1}{\bar{k}_D}} \tag{30}$$

and

$$\bar{k}_C = \tfrac{1}{2}\, k[\cot BAG + \cot AGB + \cot CDG + \cot DGC + \cot BCG + CBG]$$

$$k_D = \tfrac{1}{2} k [\cot EBH + \cot BHE + \cot FCH + \cot CHF + \cot HEF + \cot HFE] \tag{31}$$

The capacitance of the arbitrary quadrilateral element (for example i+1 in the insert of figure 6) is given by

$$C_{i+1} = \tfrac{1}{2} \rho c_p [(x_G - x_C)(y_H - y_B) - (x_H - x_B)(y_G - y_C)] \tag{32}$$

The code contains the logic to examine the element and decide on the appropriate subdivision into triangles. The code then proceeds to compute the capacitances and conductances as illustrated above.

Special Problem For The Leading Edge

As mentioned in the introduction, in this type of reduction problem one obtains from the model test just one melt time, the minimum melt time around the l.e., while the h distribution is obtained from theory.

The data reduction problem then is simplified to one of determining only one unknown, i.e., the magnitude of h at the stagnation point h_0. In performing the heat conduction analysis, the complete geometry must, of course, be treated, however, the iteration to convergence is performed on only h_0. The given time of melt is most conveniently chosen at the stagnation point in the neighborhood of which melting is expected to occur first.

The problem is solved as usual in a plane normal to the leading edge. The infinite-cylinder approximation is used. Approximations that are also accepted in order to obtain simple formulae are: perfect gas with an effective γ, constant Prandtl number, $\mu \propto T$ and 'cold walls'. The boundary layer is taken as laminar on the entire l.e.

The equation used for h/h_0 is the Lees' formula (see for example ref. 9; assumptions are cold wall and $\mu \propto T$):

$$\frac{h}{h_0} = \frac{F}{\sqrt{\left(\frac{du_e}{ds}\right)_0}} \tag{33}$$

where (the meaning of the symbols may be clarified by reference to figure 7):

$$F = \frac{\frac{p}{p_{02}} \frac{u_e}{u_\infty}}{2\left[\int_0^s \frac{p}{p_{02}} \frac{u_e}{u_\infty} ds\right]^{1/2}} \tag{34}$$

Of course this equation is written in a plane normal to the l.e.

To evaluate F quantitatively, one needs the distribution of pressure and velocity around the leading edge, and the position of the stagnation point. The pressure dis-

tribution is evaluated with the modified Newtonian law (between the stagnation and the two sonic points) joined smoothly to a Prandtl-Meyer expansion downstream of the sonic points. The modified Newtonian relation for the pressure coefficient is

$$C_p = C_{p0} \sin^2\phi \tag{35}$$

Expressing C_p in terms of pressure, the pressure is given explicitly by

$$p = p_\infty\left[\left(\frac{p_{02}}{p_\infty} - 1\right)\sin^2\phi + 1\right] \tag{36}$$

where p_{02} is the pressure behind the normal shock. From (36) the velocity at the edge of the boundary layer u_e follows by Bernoulli's equation:

$$u_e = \left\{\frac{2\gamma}{\gamma-1}\frac{p_{02}}{\rho_{02}}\left[1 - \left(\frac{p}{p_{02}}\right)^{\gamma-1/2}\right]\right\}^{1/2} \tag{37}$$

These expressions are used as long as u_e remains subsonic (see fig. 7). Beyond the sonic points the Prandtl-Meyer relations are used. The wall turning is related to the Prandtl-Meyer function in a very simple fashion. For expanding flow one has

$$\nu_2 = \nu_1 + |\,\phi_2 - \phi_1\,| \tag{38}$$

where ν is a unique function of Mach number

$$\nu(M) = \sqrt{\frac{\gamma+1}{\gamma-1}}\tan^{-1}\sqrt{\frac{\gamma-1}{\gamma+1}(M^2-1)} - \tan^{-1}\sqrt{M^2-1} \tag{39}$$

Thus as one follows around the contour of the leading edge, ν is given by eq. (38). From ν the Mach number is determined by a simple trial and error application on eq. (39). If one again assumes isentropic flow, all other flow properties e.g., p, u are readily related to their corresponding values at sonic conditions through the Mach number.

After p and u_e are determined around the body, F is evaluated by numerical integration. One can then substitute F into eq. (33), divide by $\sqrt{(du_e/ds)_0}$ and obtain the desired ratio of heat transfer coefficients. It is actually not necessary to evaluate $(du_e/ds)_0$. Since h/h_0 must equal 1.0 at the stagnation point, $\sqrt{(du_e/ds)_0}$ is simply equal to F at the stagnation point.

For the stagnation point position, the Newtonian rule is used, namely, that the stagnation point is where the "free-stream" velocity (i.e., the velocity vector in the plane normal to the l.e.) intersects the l.e. contour at 90°.

The distribution of the adiabatic wall temperature is also required. It is determined by the simple approximate relationship

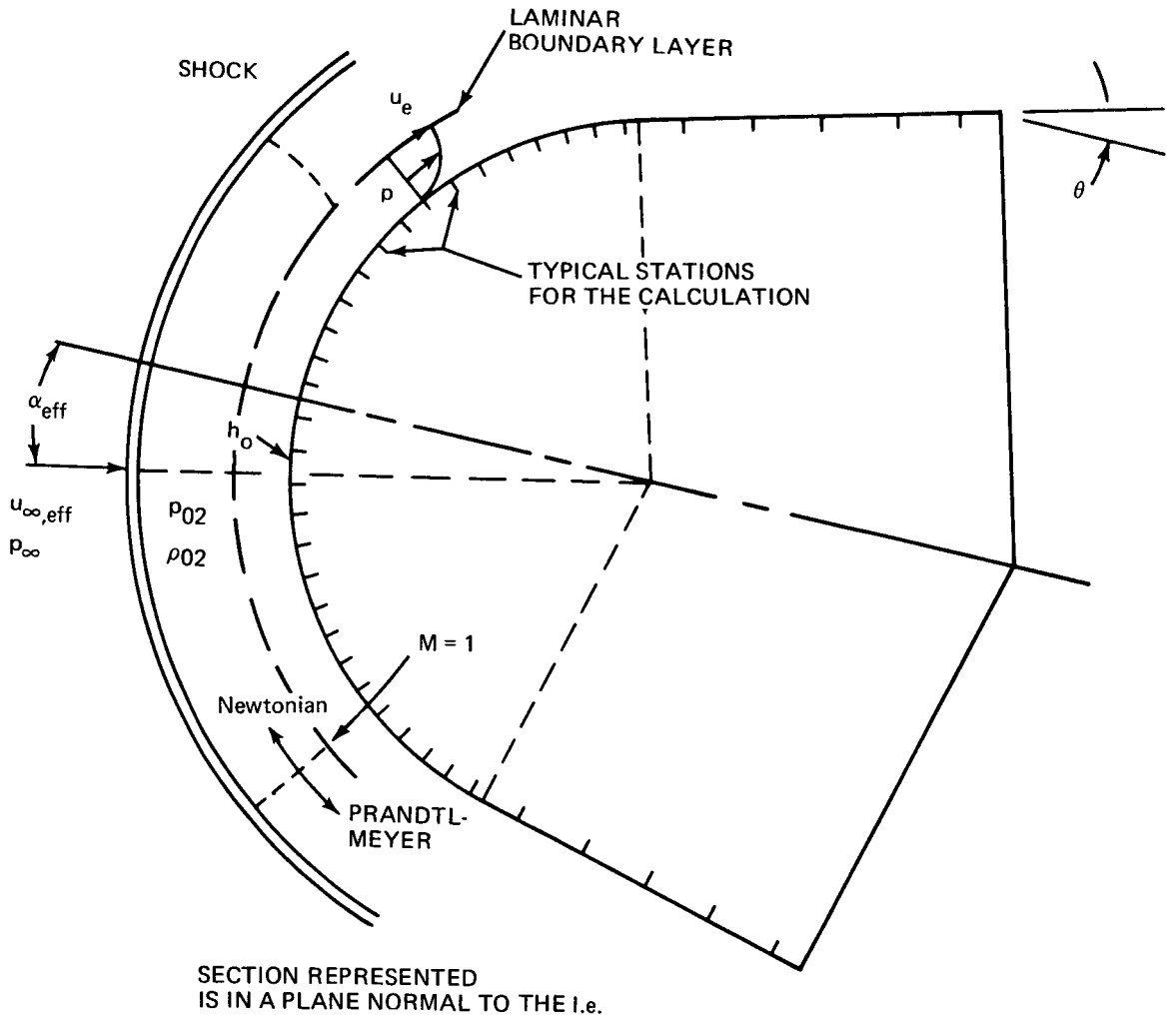


FIGURE 7. SCHEMATIC OF THE FLOW FIELD CALCULATION AROUND LEADING EDGE FOR THE SPECIAL PROBLEM

$$\sqrt{Pr} = \frac{T_{AW} - T_e}{T_0 - T_e} \tag{40}$$

or

$$T_{AW} = \sqrt{Pr}(T_0 - T_e) + T_e \tag{41}$$

where Pr is given.

In conclusion, if one wishes to make use of this special reduction procedure for l.e.'s, the additional input information needed consists of the free stream $M_{\infty,eff}, \gamma, p_\infty, T_0$, Pr, the gas constant R and the angle of attack. These quantities are to be understood given in a plane normal to the l.e.

The many approximations embodied in the formulae chosen raise the question of whether or not these formulae are adequate, even though it is almost unavoidable to keep the complexity of the problem within reasonable bounds. It is worthwhile therefore to state briefly how accurate the approximations are:

- Stagnation point and point of maximum heating can differ considerably so that the position of earliest melt need not be the correct position to start the integrals in equation 34. Angular displacement can be of the order of 20 to 30° on a swept leading edge at higher α (for example, ref. 10). Unfortunately, the the simple Newtonian rule for the stagnation point position is not accurate in such cases because the effective angle of attack

  $$\tan\alpha_E = \tan\alpha / \cos\Lambda \tag{42}$$

  is very high (for example for the wing l.e. of the NASA MSC 040A orbiter configuration during reentry, $\alpha_E \sim 50°$). Angular errors of the Newtonian formula can easily be 30° to 50° (ref. 10). We are not aware of a more accurate formula for determining the stagnation point position on leading edges.

- None of a dozen or so approximate methods for obtaining the pressure distribution around the l.e. is accurate in the shoulder region, including the typical, not-too-complex modified Newtonian plus Prandtl-Meyer. Typically C_p can be in error by 50% in the shoulder region. Pressure gradients cannot be used at all.

- When comparing with exact inviscid calculations (obtained with computer code of ref. 11), the entropy on the body is found to be well predicted by normal-shock entropy with effective free-stream Mach number. For a planar wing with dihedral ϕ, sweep Λ, angle of attack α

  $$M_{eff} = M_\infty \cos\Lambda_E$$

  $$\sin\Lambda_E = \sin\Lambda\cos\alpha + \sin\alpha\cos\Lambda\sin\phi$$

Refinements on these items should not be included in the method at this point. The approximations involved have to be kept in mind, but a realistic first step is to use formulae such as (33) and (41) carefully avoiding explicit pressure gradients. This will already represent a considerable step permitting the evaluation of this special data reduction method.

As far as the numerical method used to solve the equations, there are only minor modifications to the G_{ij} equations (eq. (13) and (14)) and to equation (16) for the updates of the h_{ij}. Since there is only one h_0, we have instead of (14)

$$G_{oo} = \frac{\partial T_0}{\partial h_0} \simeq \frac{T_0(\bar{h}_0 + \Delta h,\ \bar{h}_1 + \Delta h_1 \ldots\ldots \bar{h}_k + \Delta h_k) - T_0(\bar{h}_0,\ \bar{h}_1 \ldots h_k)}{\Delta h_0}$$

$$\Delta h_j = \frac{\bar{h}_j}{\bar{h}_0} \Delta h_0 \qquad (14'')$$

Similarly (11) and (16) are replaced by:

$$f_0 = T_M - T_0(t_{m,0}) \qquad (11'')$$

$$\delta \bar{h}_j = -\frac{\bar{h}_j}{\bar{h}_0} \frac{1}{G_{oo}} f_0 \qquad (16')$$

Minimization of Computer Time

Areas for Computer Time Minimization

In writing the equations of the method, some alternatives were left open in the way the calculation is carried out. The machine time required by each alternative will now be investigated and the most attractive alternative selected. Specifically, the areas in which there is still some freedom left are:

a) whether to calculate all the eigenvalues of the A matrix (see for example equation (10)) as opposed to calculate only the dominant ones and therefore save time in the eigenvalue-eigenvector (E&E) calculation;

b) whether to calculate all the terms in an equation such as (10) or limit the calculations to only the $e^{\lambda_i t}$ terms containing a dominant eigenvalue; here time is saved in the matrix operation to calculate the temperatures and the G_{ij};

c) what method to use for calculating the E&E's;

d) finally, how to calculate the G_{ij}, whether analytically through the explicit formula (13) or numerically through (14).

A few comments will illustrate these items.

As far as items a) and b) are concerned, this matrix reduction technique has been used successfully in vibration analyses (e.g., ref. 12, 13 and 14) where problems in-

volving several thousand elements are common. When applied to heat conduction, the idea amounts to representing the transient thermal behavior of the interior elements by a smaller number of "generalized" coordinates. For example, the temperature of each element i is normally written as the sum

$$T_i(t) = \sum_{j}^{1,n} P_{ij} e^{\lambda_j t}$$

The number of eigenvalues and, consequently, the number of terms to be added is exactly equal to the number of elements. But many of the terms in the sum are so small that they could be neglected without loss of accuracy in T_i. Table I gives for a typical case the error found by keeping only the largest terms. Clearly, sufficiently accurate values of T_i can be obtained by reducing the A matrix even by a factor of 10. The deletion of the remaining terms may thus result in a significant savings of computer time, partly because the subdominant eigenvalues need not be computed (item a) and partly because all the subsequent calculations involving the subdominant eigenvalues need not be carried out (item b). One problem, however, to be resolved with item b) is the effect of using only the dominant eigenvalues in obtaining the influence coefficients $G_{ij} = \partial T_i / \partial h_j$ when these coefficients are calculated through their analytical expression (13). Because of the differential nature of this expression is it not obvious that it will not depend in an essential way on the complete set of E&E's.

As far as item c) is concerned, standard and efficient methods exist for obtaining the E&E's (all or just a given number of diminant ones) of a real symmetric matrix, such as A. But of course, there exists the possibility of using methods that are particularly efficient for the special characteristics of the matrix A, which is sparse and banded. Another point to be kept in mind in choosing a method is that the eigenvalue calculations are repeated for each matrix A, i.e., each time the matrix A is changed during the iterations in the h_j. Therefore, iteration methods for the E&E's of each may be very attractive because there are already available good starting approximations for the E&E's, those of the previous A iterate. Of course, it is not within the scope of this study to embark in the development of new methods for E&E's calculations; the aim is rather to exploit the best state of the art.

Finally, as far as item d), the question of how to obtain the G_{ij}, whether through (13) or (14), depends not only upon which of the two equations takes up more time, but also whether or not equation (13) requires all the E & E's thereby affecting the best procedure selected under items a) and b).

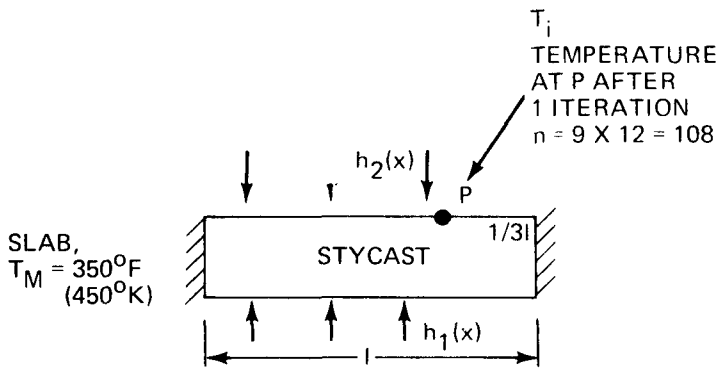
Items a) and d) were investigated through experimentation via a pilot computer code.* The problems taken as typical in the experimentation are: (i) a slab heated on two sides with constant adiabatic wall temperature and the special l.e. problem, (ii) with constant material properties and negligible radiation, and (iii) with a total number of elements of the order of 100. These are the most common current problems to which the method is intended and for which minimum computer time should be assured.

*The code which contained just a few double precision operations was exercised on the IBM 370/165. The results were spot-checked on a single precision code on the CDC 6400 with substantially the same results.

TABLE I. TYPICAL ERRORS INCURRED IN NEGLECTING SUBDOMINANT EIGENVALUES TERMS IN THE TEMPERATURES CALCULATIONS

NUMBER OF DOMINANT EIGENVALUES + EIGENVECTORS n_e ⟶

	2	3	4	5	10	20	30	MAX n_e = 108
T_M-T_i AT P								
^oR	-143.3	-158.7	-159.0	-159.0	-159.1	-159.1	-159.1	-159.0
(^oK)	(-79.6)	(-88.2)	(-88.4)	(-88.4)	(-88.5)	(-88.5)	(-88.5)	(-88.4)



Baseline Method and Its Computer Time Breakdown

In order to direct attention to the most time-consuming steps of the calculations, a baseline method is first selected and timed to serve as a reference. The baseline method consists of standard procedures for the items a) to d) above, namely:

a) all the E&E's of each A matrix are calculated

b) all terms $e^{\lambda_i t}$ are used in all matrix operations

c) the complete set of E&E's are obtained via a typical modern transformation method for real symmetric matrices, i.e., an n-step reduction to tri-diagonal form followed by the convergent Q-R iteration (ref. 15). Standard routines are used (ref. 16).

d) The analytical expression, eq. (13) is used for the G_{ij}.

Table II gives for the baseline method the breakdown of computer time for each of the four key steps which constitute one h iteration. This table reflects the medium-size problems, $n \sim 100$. The variation with n in the 50 to 100 range is shown in fig. 8. Note that in the baseline method, the machine time varies like n^3 and therefore little hope can be held that 3D problems can be handled in this way.

Time Savings Using Dominant E&E's in the Matrix Operations

As a first step in the computer-time minimization task, let us evaluate the potential of item b).

It follows from Table II and fig. 8, that no time savings can be accrued in the calculation of the temperature errors since the computer time is already negligible. There remains the possibility of dropping the subdominant-eigenvalues terms in the calculation (through formula (13)) of the G_{ij} which takes up most of the time. Unfortunately, it turns out that while it is quite possible in each h iterate to drop the subdominant terms in the temperature calculation, it is not possible to do so in the G_{ij} calculation through formula (13). To see this, let us consider first the temperature calculation in each h iterate.

We have already verified (Table I) that accurate values of the temperature can be obtained with only a handful of dominant E&E's. To examine the problem a little more formally, let us derive the same result formally from the equations as follows.

In each h iteration, for a set of h iterates, the temperatures at the time of melt are calculated from

$$T_i(t_i) = \sum_{j}^{1,n_e} P_{ij} e^{\lambda_j t_i}$$

TABLE II. COMPUTER TIME BREAKDOWN IN THE BASELINE NUMERICAL METHOD

METHOD	IBM 370/165 (FORTRAN G) TIME FOR ONE ITERATION IN h, SEC.				
	STEP 1 COMPUTE E&E'S	STEP 2 COMPUTE TEMPERATURE ERRORS, T_M-T_i	STEP 3 COMPUTE INFLUENCE COEFF., G_{ij}	STEP 4 COMPUTE UPDATED h_i	TOTAL
BASELINE (ALL E&E'S + ANALYTIC G_{ij})	~24	~0	~79	~0	~103

NOTES: IBM PILOT CODE
SLAB, HEATED ON 2 SIDES
TOTAL NUMBER OF ELEMENTS = 108, SURFACE ELEM = 2 X 12 = 24

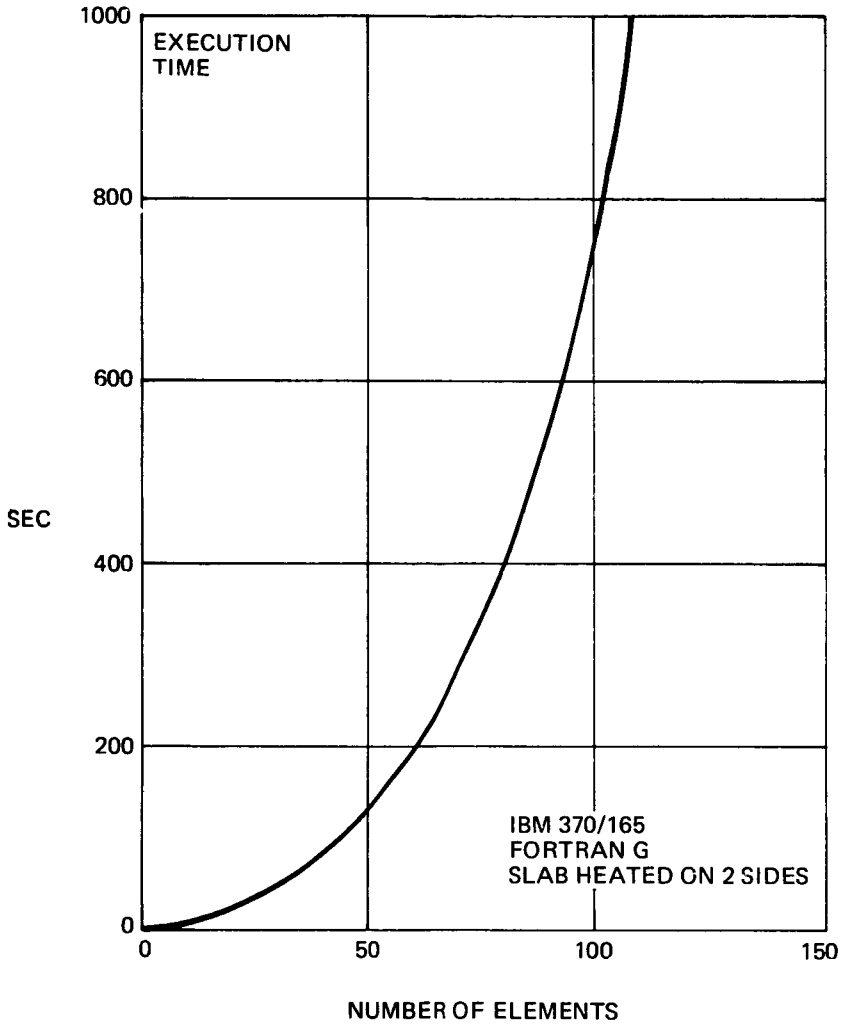


FIGURE 8. MACHINE TIME REQUIRED BY BASELINE VERSION OF THE NUMERICAL METHOD

where $T_i(t_i)$ is the temperature at the i-th surface element at the i-th melt time, t_i. P is a rectangular matrix and the λ's are the eigenvalues, assumed to be ordered so that $0 \geq \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$. Both P_{ij} and λ_j depend on h, the heat transfer coefficient for the k-th surface element. For any case which involves a nontrivial transient, $\lambda_1 < 0$, and $h_k > 0$ for at least one k; for the trivial case, $h_k = 0$, all k, $\lambda_1 = 0$, $P_{i1} = T_i(0)$, $P_{ij} = 0$, $j > 1$. For the latter trivial case clearly there is one dominant eigenvalue, $\lambda_1 = 0$; the practical cases of interest involve such relatively small positive values of h_k that there is a "cluster" of small dominant negative eigenvalues with a complementary cluster of fairly large negative eigenvalues. Thus, it usually suffices to sum over the first $n_e \ll n$ terms. We shall now derive the explicit dependence of the matrix P on the eigenvectors and show that the first n_e columns of P depend only on the n_e dominant eigenvectors. The equations which $T_i(t)$ satisfy are, in partitioned matrix form:

$$\begin{bmatrix} M_1 & 0 \\ 0 & M_2 \end{bmatrix} \begin{bmatrix} \dot{\overline{T}} \\ \dot{\overline{\tau}} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{21}^T \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} \overline{T} \\ \overline{\tau} \end{bmatrix} \; ; \; \begin{bmatrix} \overline{T}(0) \\ \overline{\tau} \end{bmatrix} = T_0 \begin{bmatrix} \overline{1} \\ \overline{1} \end{bmatrix}$$

where M_1, M_2 are positive diagonal matrices, $\overline{T}$ is the vector of surface temperatures, $\overline{\tau}$ that of internal temperature, A_{11}, A_{22} symmetric matrices, A_{21}^T the transpose of A_{21} and $\overline{1}$ a vector whose components are all ones of appropriate order. The k-th diagonal element of the A_{11} matrix is a linear function of h_k. The A matrix is singular for $h_k \equiv 0$ with $\overline{1}$ an eigenvector. The differential equation can be converted to the form:

$$\begin{bmatrix} \dot{\overline{T}} \\ \dot{\overline{\tau}} \end{bmatrix} = \begin{bmatrix} M_1^{-1/2} & 0 \\ 0 & M_2^{-1/2} \end{bmatrix} \left\{ \begin{bmatrix} M_1^{-1/2} & 0 \\ 0 & M_2^{-1/2} \end{bmatrix} \begin{bmatrix} A_{11} & A_{21}^T \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} M_1^{-1/2} & 0 \\ 0 & M_2^{-1/2} \end{bmatrix} \right\} \begin{bmatrix} M_1^{1/2} & \overline{T} \\ M_1^{1/2} & \overline{\tau} \end{bmatrix}$$

The solution for T(t) can then be written as:

$$\overline{T}(t) = T_0 M_1^{-1/2} [V_{11} V_{12}] \begin{bmatrix} e^{\Lambda_1 t} & 0 \\ 0 & e^{\Lambda_2 t} \end{bmatrix} \begin{bmatrix} V_{11}^T & V_{21}^T \\ V_{12}^T & V_{22} \end{bmatrix} \begin{bmatrix} M_1^{1/2} & \overline{1} \\ M_1^{1/2} & \overline{1} \end{bmatrix}$$

where Λ_1 is a diagonal matrix of dominant eigenvalues of $\{\}$, Λ_2 that of subdominant eigenvalues, and the corresponding matrix of eigenvectors is:

$$\begin{bmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{bmatrix}$$

Assuming $e^{\Lambda_2 t}$ is small enough to ignore for t = one of the melt times and considering the effect of the matrix multiplications:

$$\overline{T}(t) \simeq [M_1^{-1/2} V_{11}] [e^{\Lambda_1 t}] [V_{11}^T M_1^{1/2} \overline{1} + V_{21}^T M_2^{1/2} \overline{1}] = \hat{\overline{T}}(t)$$

Now, noticing that the premultiplication of a vector $\bar{v}$ by a diagonal matrix [D] is equivalent to the premultiplications of the vector $\overline{\text{col}\{[D]\}}$ by the diagonal matrix, $[\text{diag}\{\bar{v}\}]$ the result is:

$$\hat{T}(t) = [M_1^{-1/2} V_{11}] [\text{diag}\{V_{11}^T M_1^{1/2} \bar{I} + V_{21}^T M_2^{1/2} \bar{I}\}] \overline{\text{col}\{[e^{A_1 t}]\}}$$
$$= [P_1] \overline{\text{col}\{[e^{A_1 t}]\}}$$

where P_1 is a function of only $[V_{11}/V_{21}]$, the dominant eigenvectors. Therefore, we have shown even formally that the temperature equation admits an approximate (but as we have seen accurate) expression in terms of only the dominant eigenvalues and eigenvectors.*

Now the same method used for the temperature errors permits to answer in the negative the question as to whether the dominant eigenvalues approximation is accurate for the calculation of $\partial T_i/\partial h_k$.

Consider the following development.

$$\frac{\partial T_i(t_i)}{\partial h_k} = \sum_j^{1,n} \left(\frac{\partial P_{ij}}{\partial h_k} + t_i P_{ij} \frac{\partial \lambda_j}{\partial h_k} \right) e^{\lambda_j t_i}$$
$$= \sum_j^{1,n} R_{ijk} e^{\lambda_j t_i}$$

By similar reasoning the sum can usually be truncated at $n_e \ll n$ terms but the difficulty is in accurately evaluating R_{ijk}, $j \leqslant n_e$, as a function of only dominant eigenvalues and eigenvectors. Since R is a third order tensor, the analysis is more involved. In order to make the point we shall consider the explicit dependence of the infinitesimal influence coefficient on the eigenvalues and eigenvectors for the simple case:

i) 2 equal elements, 1 surface element T_1 and 1 interior element T_2, with $T_{aw_1} = T_{aw_2} = 2\ T_0$ and uniform initial temperature equal to T_0

ii) $M_1 = M_2 = \frac{1}{2}$

iii) 1 dominant eigenvalue and 2 subdominant $|\lambda_1| \ll |\lambda_2|$; the dominant vector is $[V_{11}/V_{21}]$ and the subdominant $[V_{12}/V_{22}]$.

The explicit solution of this problem is (temperatures are made nondimensional with the initial temperature, times with $\Delta x/\alpha$, the heat transfer coefficient with $2k/\Delta x$; Δx is the size of the element):

---

*By implication, then, also the direct heat conduction problem, when solved by the eigenvalue method, admits a simplification in terms of only the dominant eigenvalues and eigenvectors.

$$\begin{bmatrix}\dot{T}_1\\ \dot{T}_2\end{bmatrix}=\begin{bmatrix}-1-2h & 1\\ 1 & -2\end{bmatrix}\begin{bmatrix}T_1\\ T_2\end{bmatrix}+\begin{bmatrix}4h\\ 0\end{bmatrix};\ \begin{bmatrix}T_1(0)\\ T_2(0)\end{bmatrix}=\begin{bmatrix}1\\ 1\end{bmatrix}.$$

It follows:

$$\begin{bmatrix}T_1(\infty)\\ T_2(\infty)\end{bmatrix}=\begin{bmatrix}2\\ 2\end{bmatrix}.$$

Let

$$\begin{bmatrix}\tau_1\\ \tau_2\end{bmatrix}=-\begin{bmatrix}V_{11} & V_{21}\\ V_{12} & V_{22}\end{bmatrix}\begin{bmatrix}T_1-T_1(\infty)\\ T_2-T_2(\infty)\end{bmatrix};$$

where V is orthogonal so

$$\begin{bmatrix}T_1(t)\\ T_2(t)\end{bmatrix}=\begin{bmatrix}T_1(\infty)\\ T_2(\infty)\end{bmatrix}-\begin{bmatrix}V_{11} & V_{12}\\ V_{21} & V_{22}\end{bmatrix}\begin{bmatrix}\tau_1(t)\\ \tau_2(t)\end{bmatrix}$$

Then

$$\begin{bmatrix}\dot{\tau}_1\\ \dot{\tau}_2\end{bmatrix}=\begin{bmatrix}\lambda_1 & 0\\ 0 & \lambda_2\end{bmatrix}\begin{bmatrix}\tau_1\\ \tau_2\end{bmatrix};\ \begin{bmatrix}\tau_1(0)\\ \tau_2(0)\end{bmatrix}=\begin{bmatrix}V_{11}+V_{21}\\ V_{12}+V_{22}\end{bmatrix}$$

Since V can be chosen so that

$$\begin{bmatrix}V_{11} & V_{21}\\ V_{12} & V_{22}\end{bmatrix}\begin{bmatrix}-1-2h & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}V_{11} & V_{12}\\ V_{21} & V_{22}\end{bmatrix}=\begin{bmatrix}\lambda_1 & 0\\ 0 & \lambda_2\end{bmatrix}$$

where

$$\lambda_1=-(1+h-r);\ \lambda_2=-(1+h+r);\ r=\sqrt{1+h}$$

$$V_{11}=\sqrt{0.5/[1+h^2+hr]};\ V_{21}=V_{11}(h+r)$$

$$V_{12}=\sqrt{0.5/[1+h^2-hr]};\ V_{22}=V_{12}(h-r)$$

For every $h>0$, there is a $\hat{t}>0$ such that:

$$e^{\lambda_1 t}/e^{\lambda_2 t}=e^{(\lambda_1-\lambda_2)t}\gg 1$$

for $t>\hat{t}$. For those values of h and t, we say that λ_1 dominates λ_2. It follows that

$$T_1(t)=2-P_1\,e^{\lambda_1 t}-P_2\,e^{\lambda_2 t}$$

where

$$P_1=V_{11}(V_{11}+V_{21});\ P_2=V_{12}(V_{12}+V_{22})$$

Note that the dominant term can be evaluated from knowledge of the dominant eigenvalue, λ_1, and eigenvector, $[V_{11}/V_{21}]$, alone. Call this truncated approximation, $\hat{T}_1(t)$ where $\hat{T}_1(t) = 2 - P_1 e^{\lambda_1 t}$. It is this kind of approximation, retaining perhaps five terms out of one hundred, which is used in the difference quotient estimate of the derivative of the temperature at a surface element with respect to a heat transfer coefficient

$$\frac{\partial T_1}{\partial h} \cong \frac{\Delta T_1}{\Delta h} = \frac{T_1(h+\Delta h) - T_1(h)}{\Delta h} \cong \frac{\Delta \hat{T}_1}{\Delta h} = \frac{\hat{T}_1(h+\Delta h) - \hat{T}_1(h)}{\Delta h}$$

Note that there are two levels of error: (1) for the accuracy of the true difference quotient in estimating the derivative and (2) for the accuracy in the truncated approximation to the true difference quotient. For the success of the iteration for estimating heat transfer coefficients from melt times, the latter error is surely more serious. This is so since generalized secant methods tend to give convergence rates that are comparable to Newton's method.

In order to indicate the altogether different nature of the series for the analytical formula for $\partial T_1/\partial h$, consider the following expression that applies to Eq. (13):

$$\frac{\partial T_1}{\partial h}(t) = (R_{11}t + R_{12})\, e^{\lambda_1 t} + (R_{22}t - R_{12})\, e^{\lambda_2 t}$$

where

$$R_{11} = 2V_{11}^3 (V_{11} + V_{21})$$

$$R_{12} = 2V_{11}V_{12}(2V_{11}V_{12} + V_{12}V_{21} + V_{11}V_{22})/(\lambda_1 - \lambda_2)$$

$$R_{22} = 2V_{22}^2 (V_{12} + V_{22})$$

As in the expression for $T_1(t)$, the first term surely dominates; however, the coefficient has a component, R_{12}, which depends on all eigenvalues and eigenvectors. That this is generally true follows from a result in matrix perturbation theory (Ref. 20) that the derivative of any eigenvector depends on all eigenvalues and eigenvectors. R_{12} is in fact the derivative of P_1 which itself depends only on the dominant eigenvector, $[V_{11}/V_{21}]$. Define the dominant term in $\partial T/\partial h$ as $\partial \hat{T}/\partial h$ which is the limit of the dominant term in the difference quotient as $\Delta h \to 0$

$$\frac{\partial \hat{T}}{\partial h} = (R_{11}t + R_{12})\, e^{\lambda_1 t}$$

For the sake of investigating the effect of dropping all terms except those that depend on only the dominant eigenvalue and eigenvector define the truncated analytical derivative as:

$$\frac{\hat{\delta} T}{\delta h} = R_{11} t\, e^{\lambda_1 t}$$

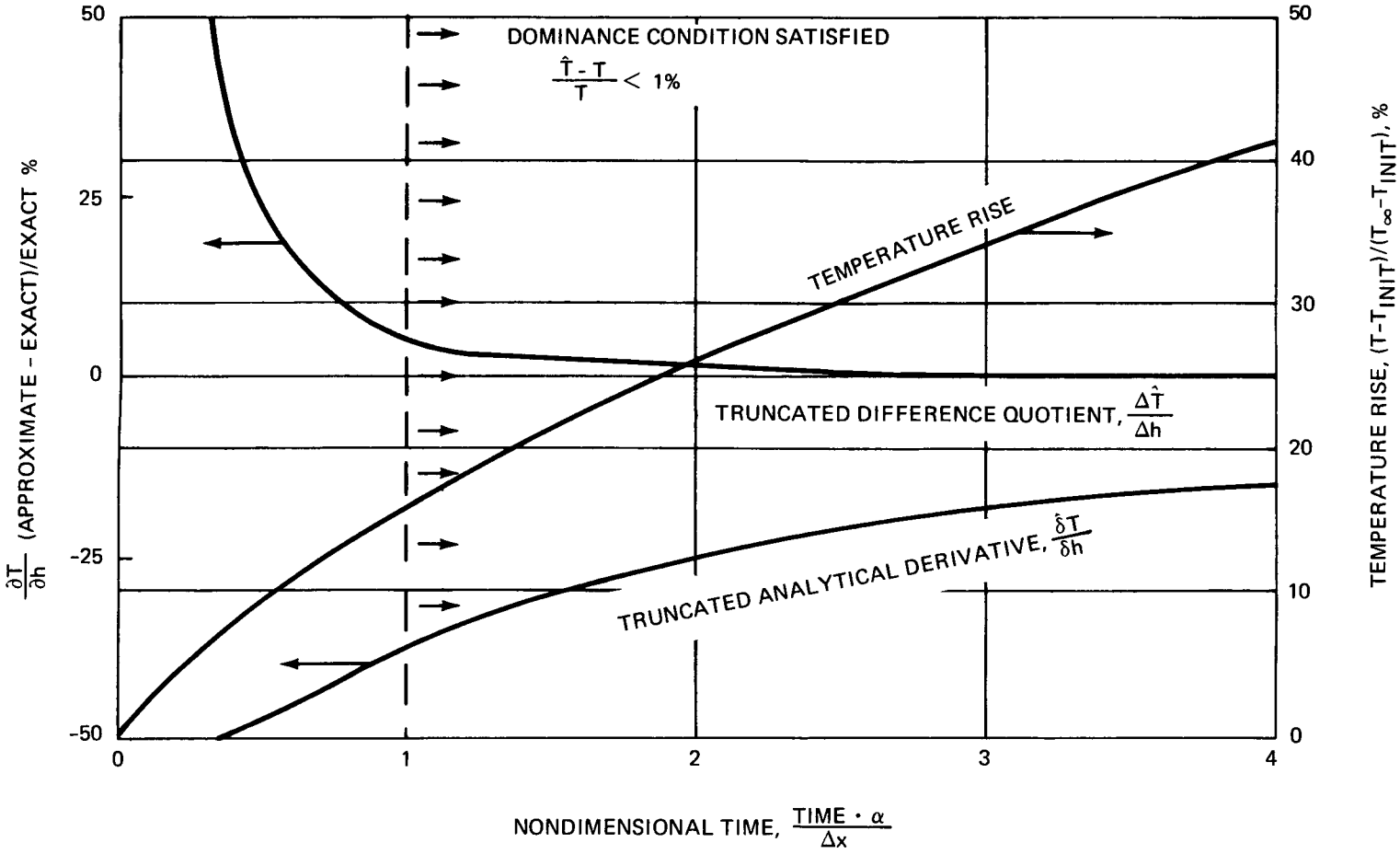


FIGURE 9. RELATIVE ERROR IN ESTIMATING $\frac{\partial T}{\partial h}$ with h = 0.125, Δh = 0.00125

Note that $\hat{\delta}T/\delta h$ is not the limit of the truncated difference quotient as $\Delta h \to 0$.

In order to illustrate the effectiveness of the truncated difference quotient by comparison with the truncated analytical derivatives, expand the above expressions in terms of h and t.

$$T_1(h,t) = 2 - \left[\frac{0.5(1+h+r)}{1+h^2+hr}\right] e^{-(1+h-r)t} - \left[\frac{0.5(1+h-r)}{1+h^2-hr}\right] e^{-(1+h+r)t}$$

$$\hat{T}_1(h,t) = 2 - \left[\frac{0.5(1+h+r)}{1+h^2+hr}\right] e^{-(1+h-r)t}$$

$$\frac{\Delta\hat{T}}{\Delta h} = \frac{\hat{T}(h+\Delta h) - \hat{T}(h)}{\Delta h}$$

$$\frac{\hat{\delta}T_1}{\delta h}(h,t) = 0.5\left[\frac{(1+h+r)t}{(1+h^2+hr)^2}\right] e^{-(1+h-r)t}$$

$$\frac{\partial\hat{T}}{\partial h}(h,t) = \frac{\hat{\delta}T_1}{\delta h}(h,t) + 0.5\left[\frac{1+h}{r^3}\right] e^{-(1+h-r)t}$$

$$\frac{\partial T_1}{\partial h}(h,t) = \frac{\partial\hat{T}}{\partial h}(h,t) + 0.5\left[\frac{(1+h-r)t}{(1+h^2-hr)^2} - \frac{1+h}{r^3}\right] e^{-(1+h+r)t}$$

where

$$r = \sqrt{1+h^2} \quad .$$

Figure 9 illustrates that the truncated difference quotient, $\Delta\hat{T}/\Delta h$, is a very good estimate of the true derivative, $\partial T/\partial h$, for values of h and t which satisfy the dominance condition that $\hat{T}$ is a very good approximation to T. The data for the illustration were chosen to correspond to a typical phase change test condition, i.e., $T_{init} = 76\,^\circ F$ ($298\,^\circ K$), $T_{AW} = 946\,^\circ F$ ($781\,^\circ K$), $T_{melt} = 300\,^\circ F$ ($422\,^\circ K$), $t_{melt} = 2$ sec. The value used for Δh ensures that $\Delta T/\Delta h$ estimates $\partial\hat{T}/\partial h$ within 0.5%, while $|\Delta\hat{T}|$ remains larger than 0.1%. The latter condition guarantees that the computed version of $\Delta\hat{T}/\Delta h$ will have three correct digits, for example, if the computed version of $\hat{T}$ has six correct digits.

The conclusion then is that item b), dropping the subdominant eigenvalues terms in the calculation of the temperature errors and the influence coefficient G_{ij}, does not offer potential for computer time savings since the temperature calculation takes up negligible time and the G_{ij} expression cannot be cast approximately in terms of only dominant eigenvalues.

Time Savings by Choosing an Appropriate Method for Calculation of the Eigenvalues and Eigenvectors

Next let us consider item c). The baseline method as indicated above uses, for the E&E's, a transformation-type method for real symmetric matrices. But an attrac-

tive alternative should be an iteration-type method because good starting approximation for the E&E's are available from the previous h iterations. Naturally a method to obtain a starting approximation must be provided for the zeroth h iteration, but this is less crucial a step since it occurs only once and the really important time savings should come from all the subsequent h iterations. The best iteration-type method for our application seems to be the Jennings algorithm (Ref. 17 & 18) since (i) it can take advantage easily of the sparseness and bandedness of the A matrix, (ii) it is just about the only method for dominant E&E's, (iii) it is very efficient if the starting approximation is good -- which is a very useful feature in our case when the h iterations provide us with increasingly better starting approximations of the E&E's.

Therefore, a specially adapted version of Jennings' algorithm has been used. Due to the distribution of eigenvalues for the heat equation, the implicit form of the inverse matrix is used for the eigenvector recursion or power step. The Jennings' algorithm used does not include a refinement, suggested by Clint and Jennings (ref. 18), called "Jacobi eigenvalue reduction."

Jennings' method has been adapted by coding the matrix multiplications to take full advantage of the sparse and banded form of the real symmetric matrix which arises in this problem. Multiplications which would produce zero results are thus omitted. This should result in a considerable time savings in the eigenvalue-eigenvector refinement over that of the standard Jennings' algorithm as coded by Vachris (ref. 19), for example. The Jennings' algorithm turns out to produce considerable time savings compared to the base line method as fig. 10 shows.

The semi-discrete form of the heat equation results in a matrix for which the most important eigenvalues are the smallest ones. Jennings' method requires dominant eigenvalues to be the largest ones. There are two transformations which could accomplish this. The simpler one to apply involves shifting all the eigenvalues by a uniform amount. This, unfortunately, results in inherently slow convergence of Jennings' method since the ratio of the largest unimportant (shifted) eigenvalue to the smallest important one is around 0.99. The approach which we have implemented results in faster convergence with values of the above ratio of around 0.4. This method utilizes a special efficient algorithm for inverting a banded, positive definite, symmetric matrix and is fully described in ref. 20.

In conclusion, item c, the use of an optimum method such as the Jennings' algorithm provides interesting time savings when only the dominant E&E's are required. Simultaneously, it follows also that item a), whether it is possible to save machine time by not calculating the subdominant eigenvalues, is answered in the affirmative.

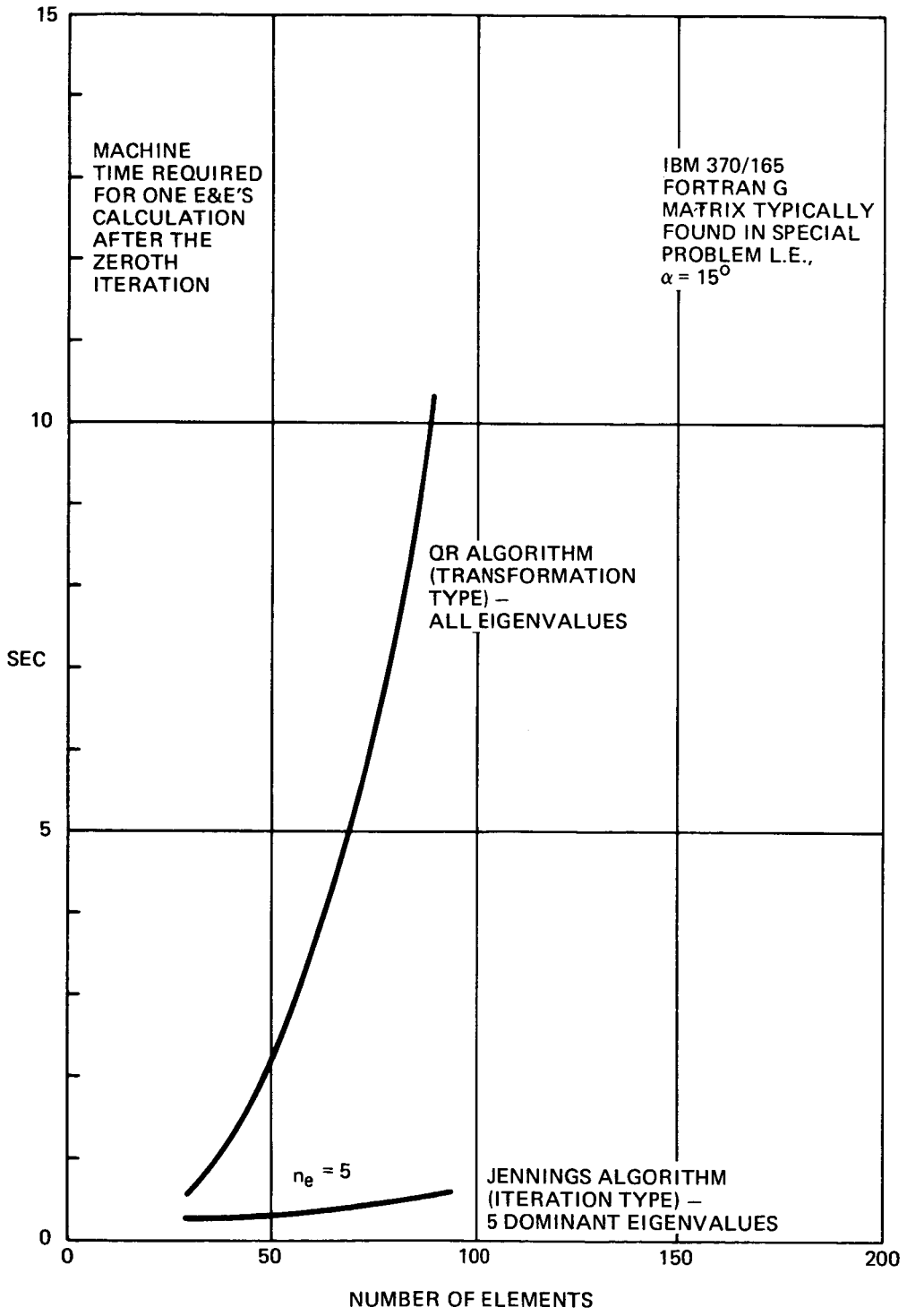


FIGURE 10. MACHINE TIME REQUIRED BY TWO OF THE BEST METHODS FOR ALL OR THE DOMINANT EIGENVALUES AND EIGENVECTORS

Incidentally, a different adaptation of Jennings' algorithm to problems similar to our is reported by Rutishauser (ref. 21 and 22).* However, Rutishauser's adaptation is very complicated and contains various sophisticated features. The machine time quoted for a 70 element problem should correspond to some 30 sec. on the IBM 370/165. Naturally, if this adaptation were applied to our problem, it would result in running times that are larger than the QR algorithm in the baseline method.

Minimum Computer Time Alternatives

Summarizing the discussion up to now, we have eliminated item b) and we have selected the optimum method for item c). Therefore, the alternatives remaining are a), whether to calculate all or dominant eigenvalues, and d) whether to calculate G_{ij} analytically or numerically. These alternatives form the matrix given in Table III. The main conclusion so far is that since the G_{ij} take up most of the computer time in the baseline method, minimum computer time will be achieved by (i) calculating only the dominant eigenvalues; (ii) calculating the G_{ij} through the numerical method. Indeed, timing of the alternatives gives the results already indicated in Table III.

The machine time required by the numerical G_{ij} and the dominant eigenvalues depends upon the number of dominant eigenvalues and naturally decreases drastically with that number. The optimum number is the one that minimizes the CPU time maintaining a good accuracy in the results. As expected, it turns out that this number is very small (fig. 11) and the number of iterations unchanged at constant accuracy (fig. 12). It is very satisfying that both accuracy and the number of iteration remains substantially constant for a wide range of attractive values, so that a universal working value can be easily selected.

*Initially, Bauer (ref. 23) generalized Jennings basic idea to nonsymmetric matrices. Rutishauser (ref. 21 & 22) subsequently specialized and refined what he refers to as Bauer's simultaneous iteration method to the symmetric case. Rutishauser (ref. 22, p. 221) discusses an example which has all of the properties of our problem. The system matrix is banded, sparse, and definite, the dominant eigenvalues are those which are smaller in absolute value and the relative magnitudes of the shifted eigenvalues to their now smaller immediate neighbors is very close to 1. For Rutishauser's example the ratio is 0.999 while for our example it is 0.99. The shifting is required in order that the working matrix have, as its dominant eigenvalues, the largest ones in absolute value. Of course, the alternative to shifting is inverting by the Cholesky factorization method. The expense of the factorization and subsequent multiplications by now dense, banded matrices in Rutishauser's case more than compensated for the slow convergence and poor final accuracy of the shifted result.

TABLE III. VARIANTS OF THE NUMERICAL METHOD THAT HAVE BEEN STUDIED

			INFLUENCE COEFFICIENTS G_{ij}	
			(α) ANALYTIC	(β) NUMERICAL
E&E'S	ALL (A)	EXACT IN EACH h ITERATION (A1)	BASELINE	MUCH SLOWER THAN A1α
		EXACT IN ZEROTH h ITERATION + APPROXIMATE SUBCOM IN SUBSEQUENT h ITERATIONS (A2)	SLOWER THAN A2β	ABOUT THE SAME AS B1β
	ONLY DOMINANT (B)	IN ZEROTH ITERATION, E&E'S FROM TRANSFORMATION METHOD (B1)	IMPOSSIBLE	ABOUT SAME TIME AS B2β FOR n ~ 100
		IN ZEROTH ITERATION, E&E'S FROM JENNINGS ALGORITHM (B2)		BEST • FOR STORAGE • FOR ACCURACY

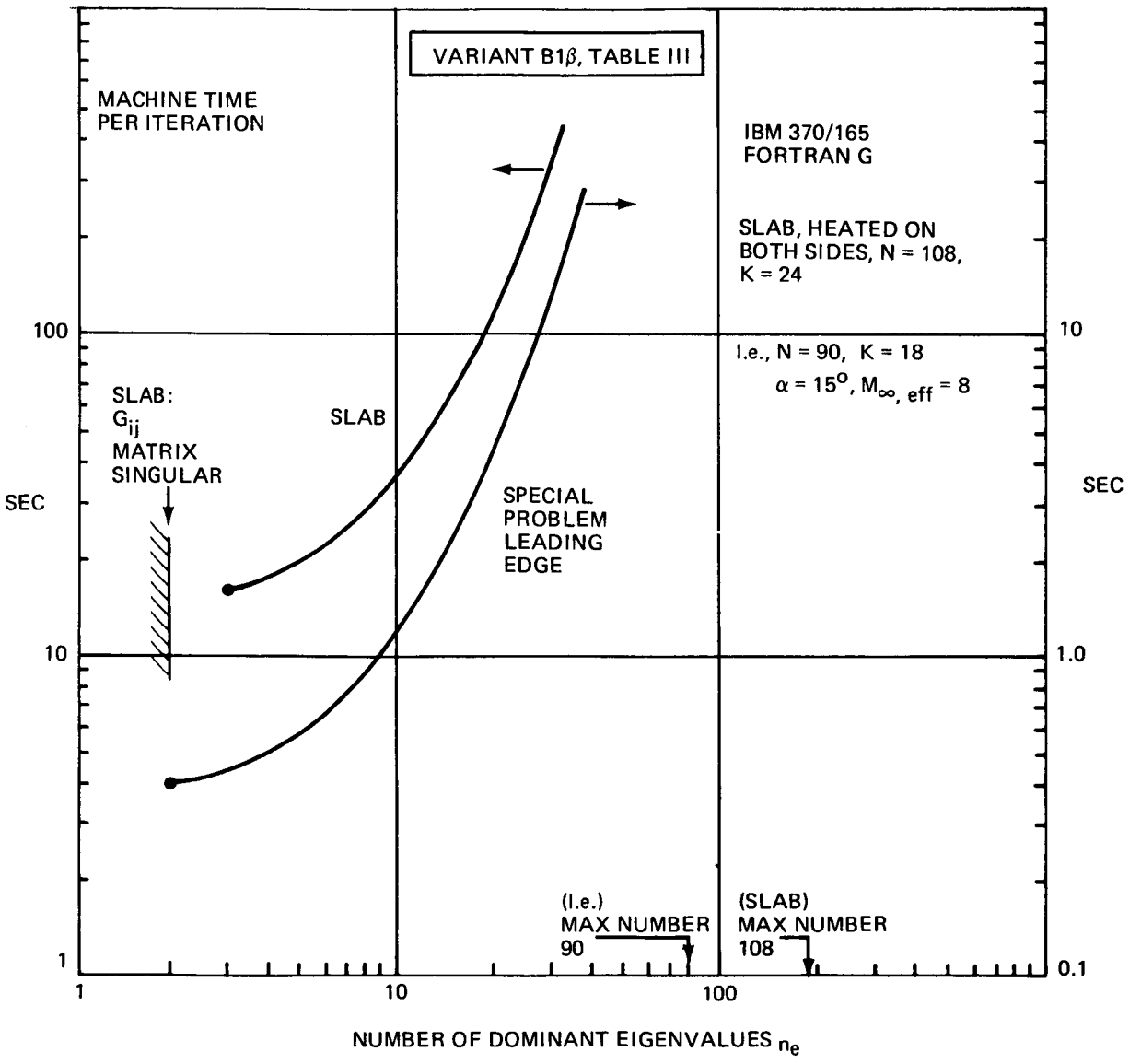


FIGURE 11. OPTIMIZING FOR THE NO. OF DOMINANT EIGENVALUES WITH NUMERICAL INFLUENCE COEFFICIENTS G_{ij}

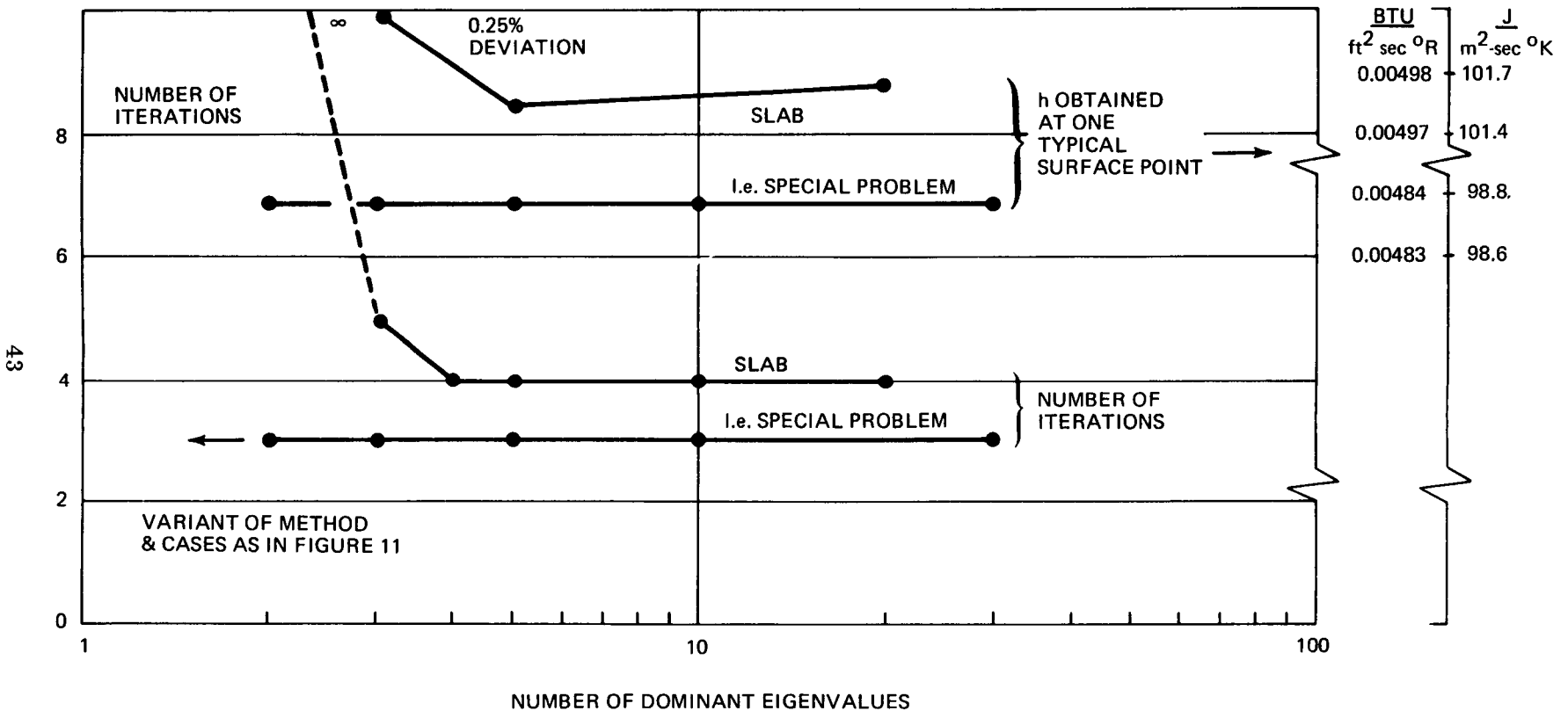


FIGURE 12. NUMBER OF ITERATIONS AND ACCURACY USING A LOW NUMBER OF DOMINANT EIGENVALUES AND NUMERICAL INFLUENCE COEFFICIENTS g_{ij}

The machine time required by the alternative A2α in the matrix (Table III) depends also on the number of dominant E&E's selected. Again the selection can be optimized, see fig. 13 and 14.

The machine time required by the alternative A2β) is the same as B2β) since the difference -- for the same number of dominant eigenvalues -- is just whether or not all the E&E's are used in the matrix operations and we have already seen that the difference is negligible.

The optimum machine time in each alternative is compared in fig. 15. The conclusion is that numerical G_{ij} and only dominant E&E's represent the minimum machine time alternative and therefore the one used for the operational computer code. The variation with the number of elements of the optimum alternative is shown in fig. 16. The important result in this figure is that the selected alternative not only has the lowest machine time, but also its machine time varies as n^2 against approximately n^3 of the baseline method. This is a remarkable feature of the method developed here that permits to run relatively large problems.

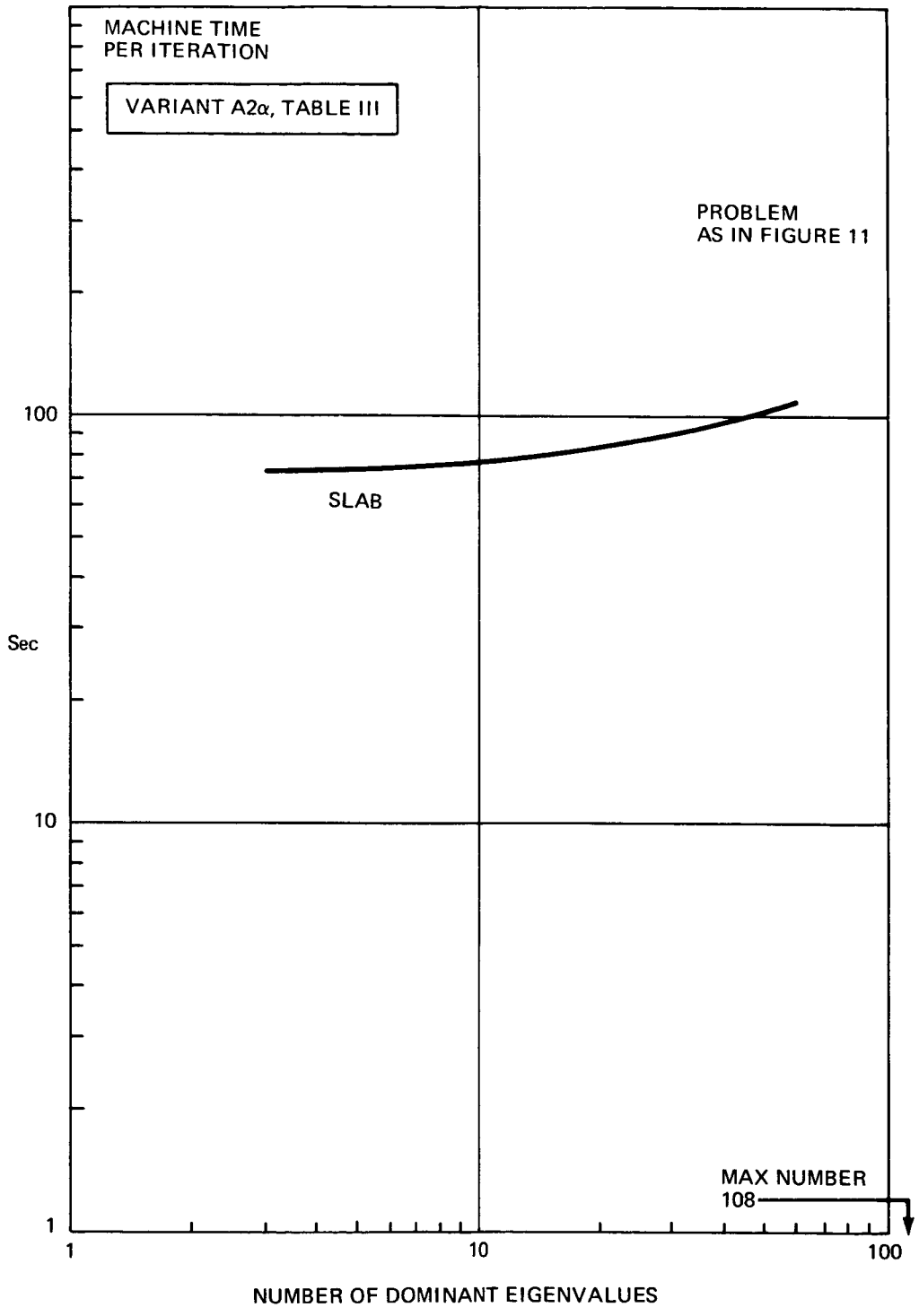


FIGURE 13. OPTIMIZING FOR THE NUMBER OF EIGENVALUES WITH ANALYTIC INFLUENCE COEFFICIENTS G_{ij}

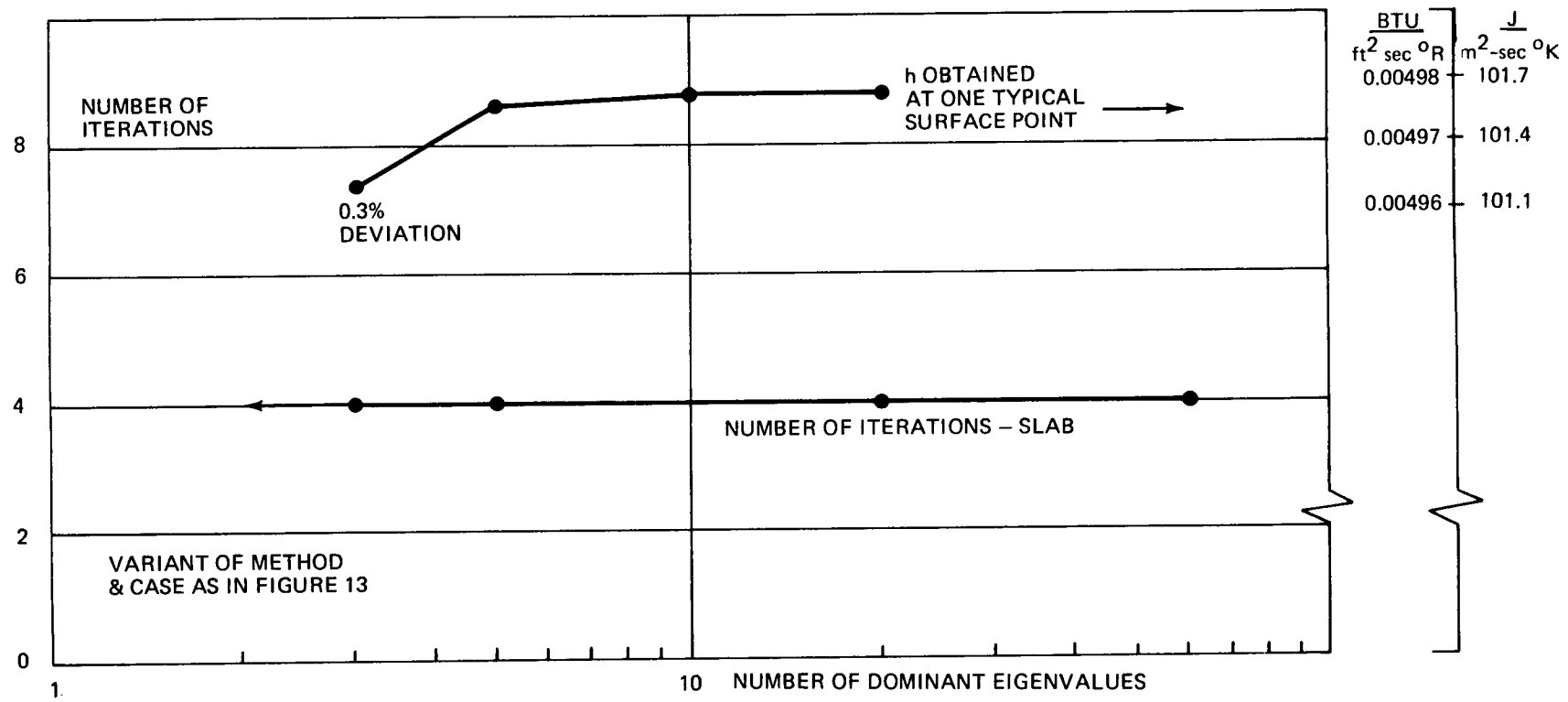


FIGURE 14. NUMBER OF ITERATIONS AND ACCURACY USING A LOW NUMBER OF DOMINANT EIGENVALUES AND ANALYTIC INFLUENCE COEFFICIENTS G_{ij}

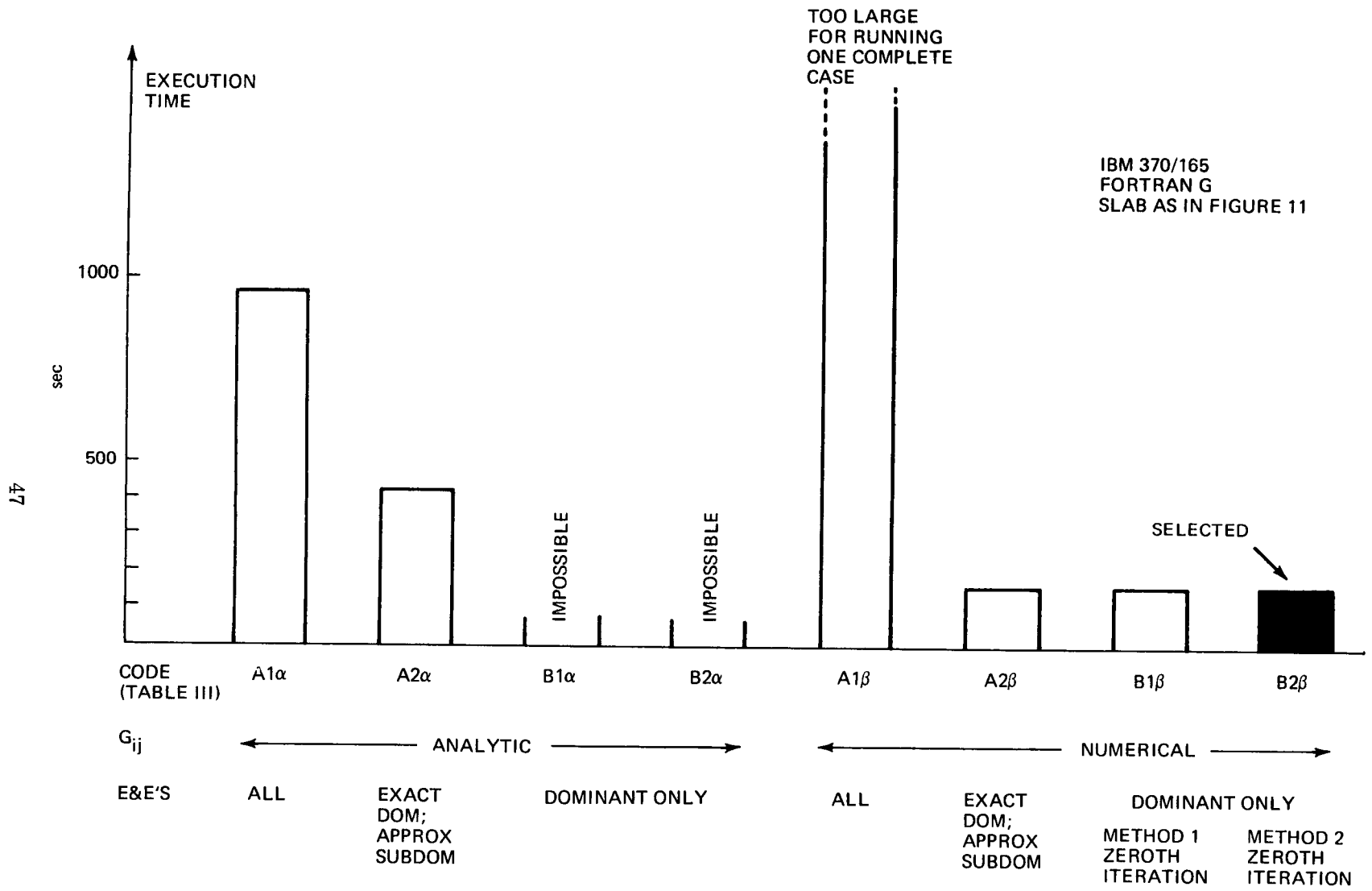


FIGURE 15. COMPARISON OF THE SMALLEST COMPUTER TIME OF EACH VARIANT OF THE NUMERICAL METHOD

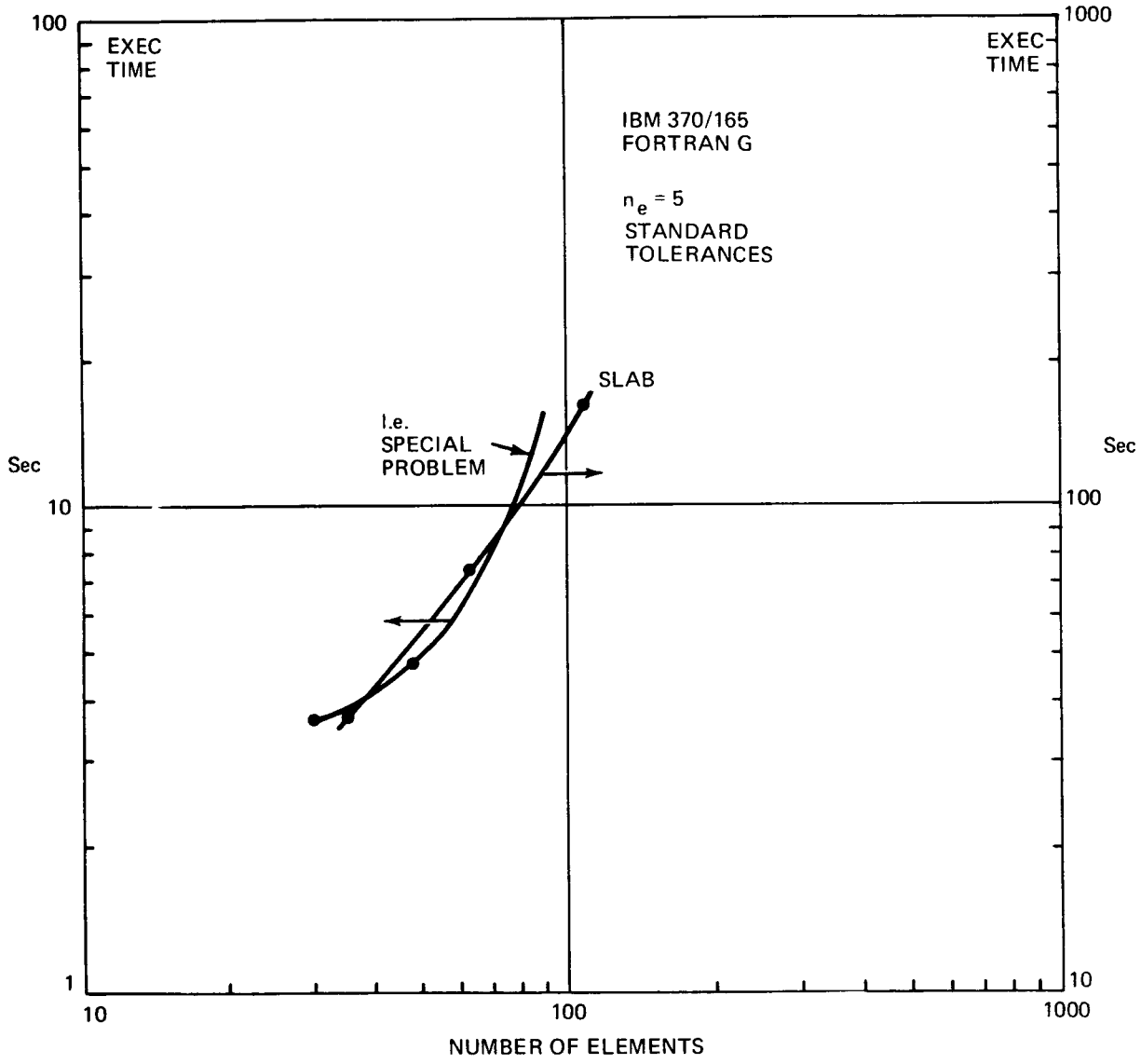


FIGURE 16. MACHINE TIME REQUIRED BY OPTIMIZED NUMERICAL METHOD

Accuracy and Computer times of the Numerical Method Developed

The method presented, even when run with a minimal number of dominant E&E's, is very accurate. This is brought out in a typical case by fig. 17, where the results of the same problem are presented with increasing number of points.

Note that the three values used in fig. 17 for the total number of points range from a very low number ($n \sim 30$), to an average number ($n \sim 100$). The experience to date suggest that $n \sim 100$ is more than adequate for typical cases and target accuracies of h_j of 1 to 2%. Unfortunately, it is not easy to compare the accuracy of this method with others, because no other inverse method (with lateral conduction) appears to be available and nor any exact analytical solution has been tabulated.

A strong indication of the soundness of the method is the manner of convergence during the h_j iteration. Fig. 18 shows the decrease in error (from a final result) at each iteration.

As far as tolerances used to declare the iteration converged, there is no point in requiring very tight tolerances in light of the soundness of the method. Fig. 19 shows that with 0.5-1% error, the number of iterations can be reduced to typically 4 for slab-like problems and 2 for 1.e. special problems.

The computer code has been checked out by running numerous test cases. In this section we show the results for the sample problem supplied by NASA Langley Research Center. The computed heat transfer coefficients, fig. 19a, match the original values within $\pm 5\%$. For this case the phase-change temperature was considered variable and the time of melt was a constant 3.6 seconds. The code was also checked in the more conventional mode of constant phase-change temperature with a variable time of melt. To obtain the necessary input information, the NASA supplied run was first duplicated on our thermal analyzer (direct problem). By plotting the temperature histories, the times of melt corresponding to $T_{cp} = 1000\,^{\circ}R$ were read off this curve. This information, along with the resulting h values from the code is shown in figure 19b. The accuracy in h is again within $\pm 5\%$.

A quantitative indication of the potential machine time reduction exploited during the optimization effort carried out in this study is given by fig. 20. This is a measure of the method's potential that was announced at the outset.

Finally, a point of the maximum importance is the fact that this method appears to require machine time proportional to n^2 (n is the number of elements) rather than n^3 that appears unavoidable for implicit methods with time discretized. This is very important if one insists on or needs to handle problems in the $n \sim 200$ range or above.

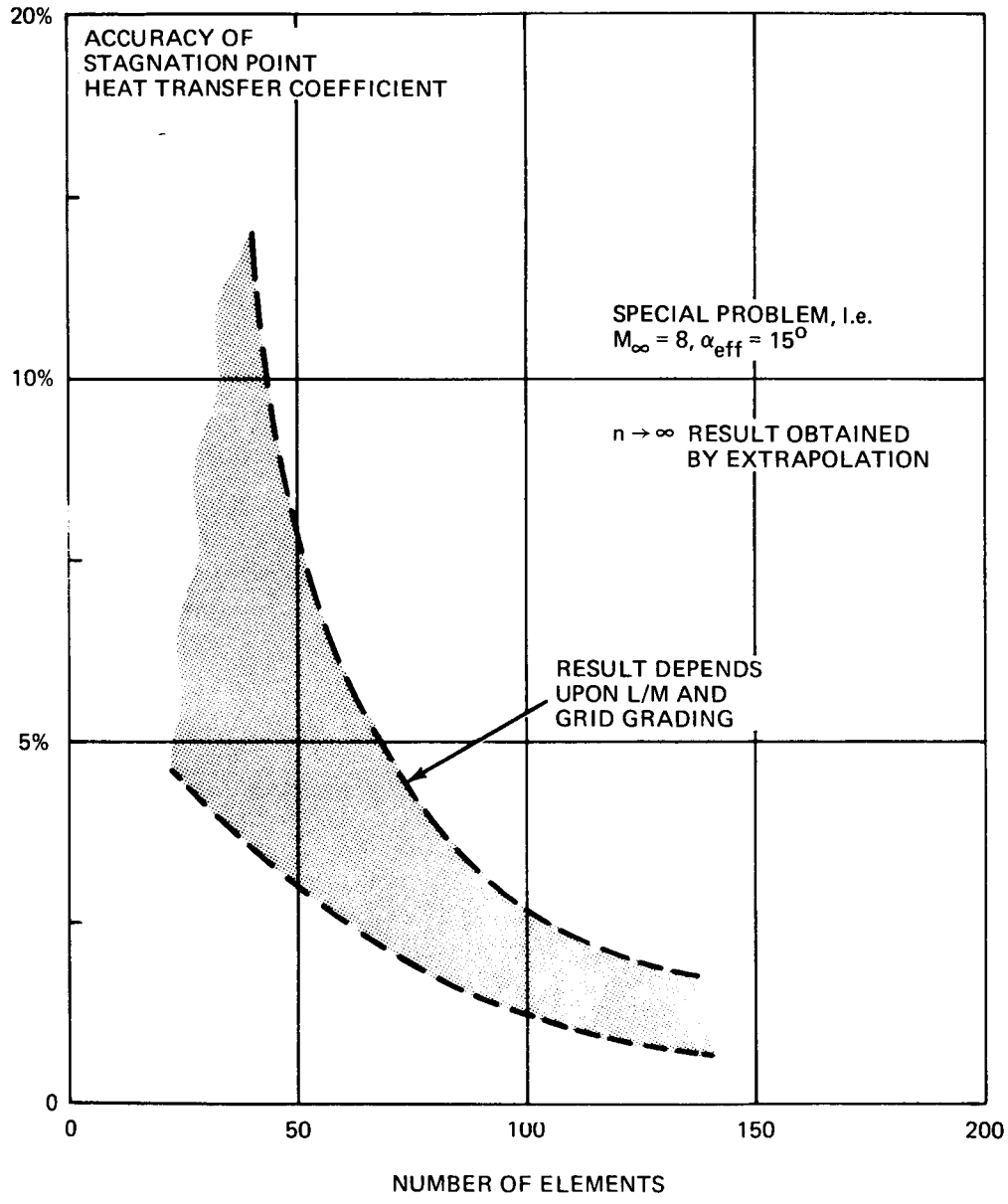


FIGURE 17. ACCURACY OF HEAT TRANSFER COEFFICIENT VERSUS NUMBER OF ELEMENTS

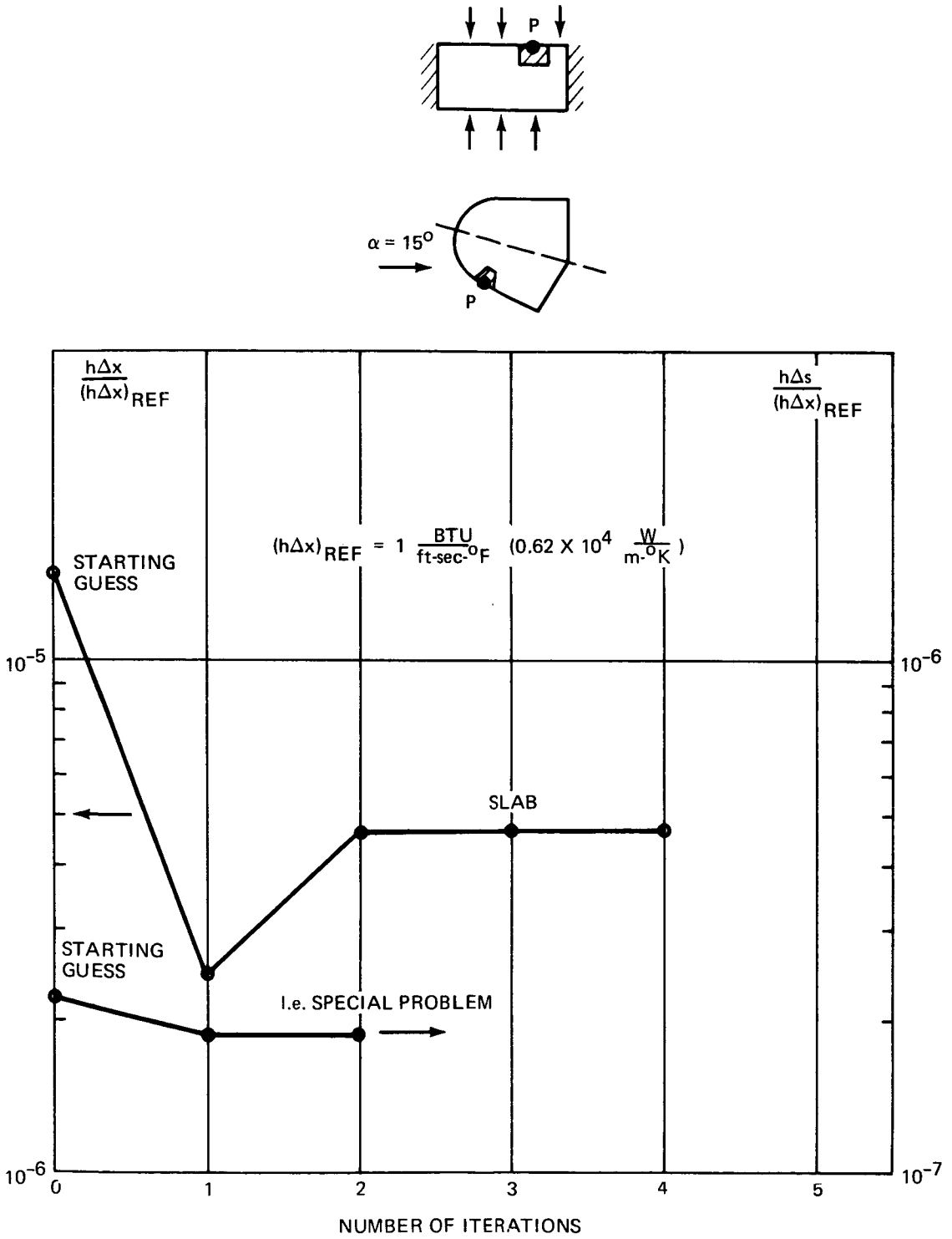


FIGURE 18. TYPICAL BEHAVIOR OF THE HEAT TRANSFER COEFFICIENT DURING CONVERGENCE TO SOLUTION

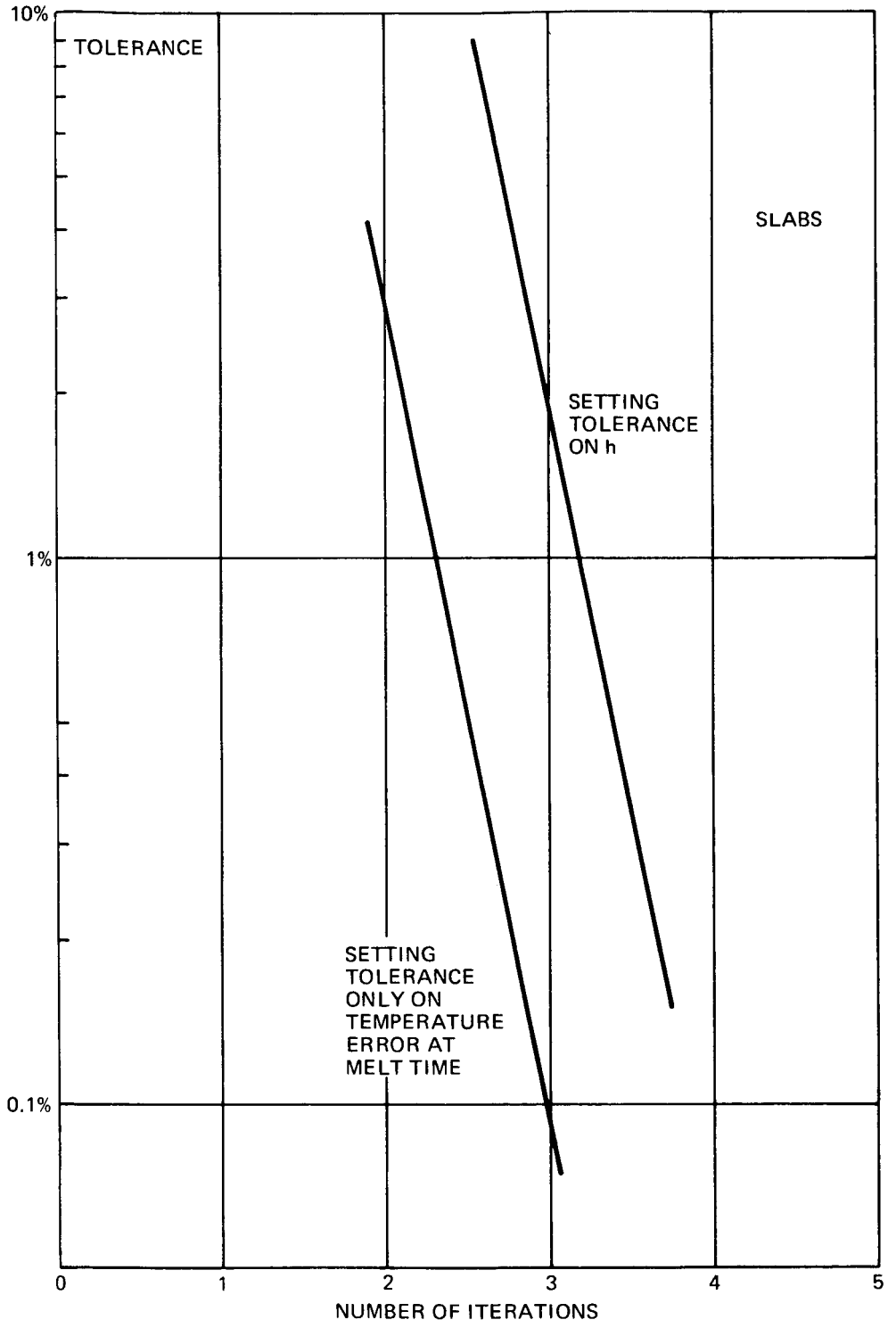


FIGURE 19. TYPICAL NUMBER OF ITERATIONS REQUIRED BY A GIVEN TOLERANCE IN THE ITERATION

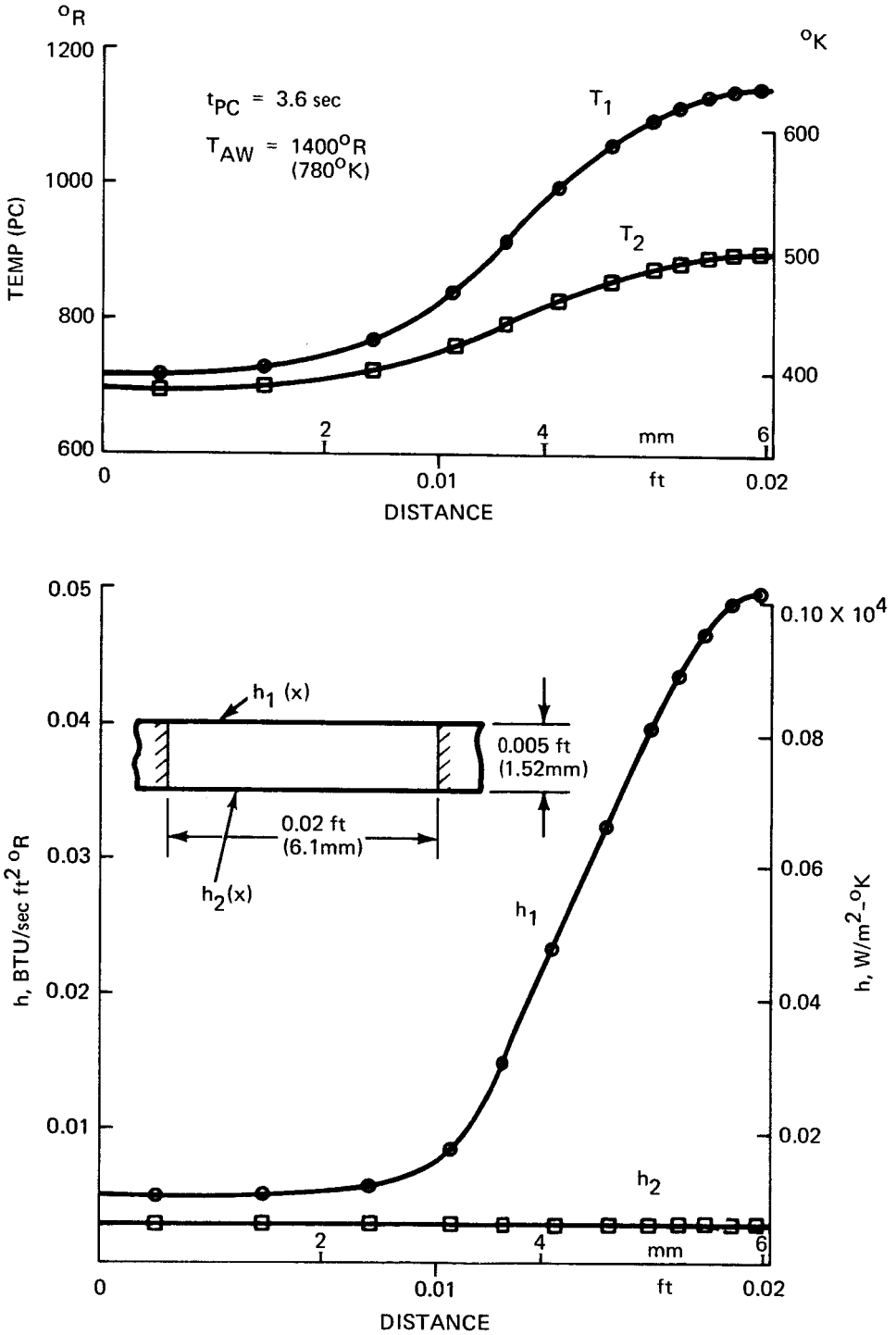


FIGURE 19a. SAMPLE RESULTS FOR DEMONSTRATION CASE (SUPPLIED BY NASA), MELT TIME CONSTANT

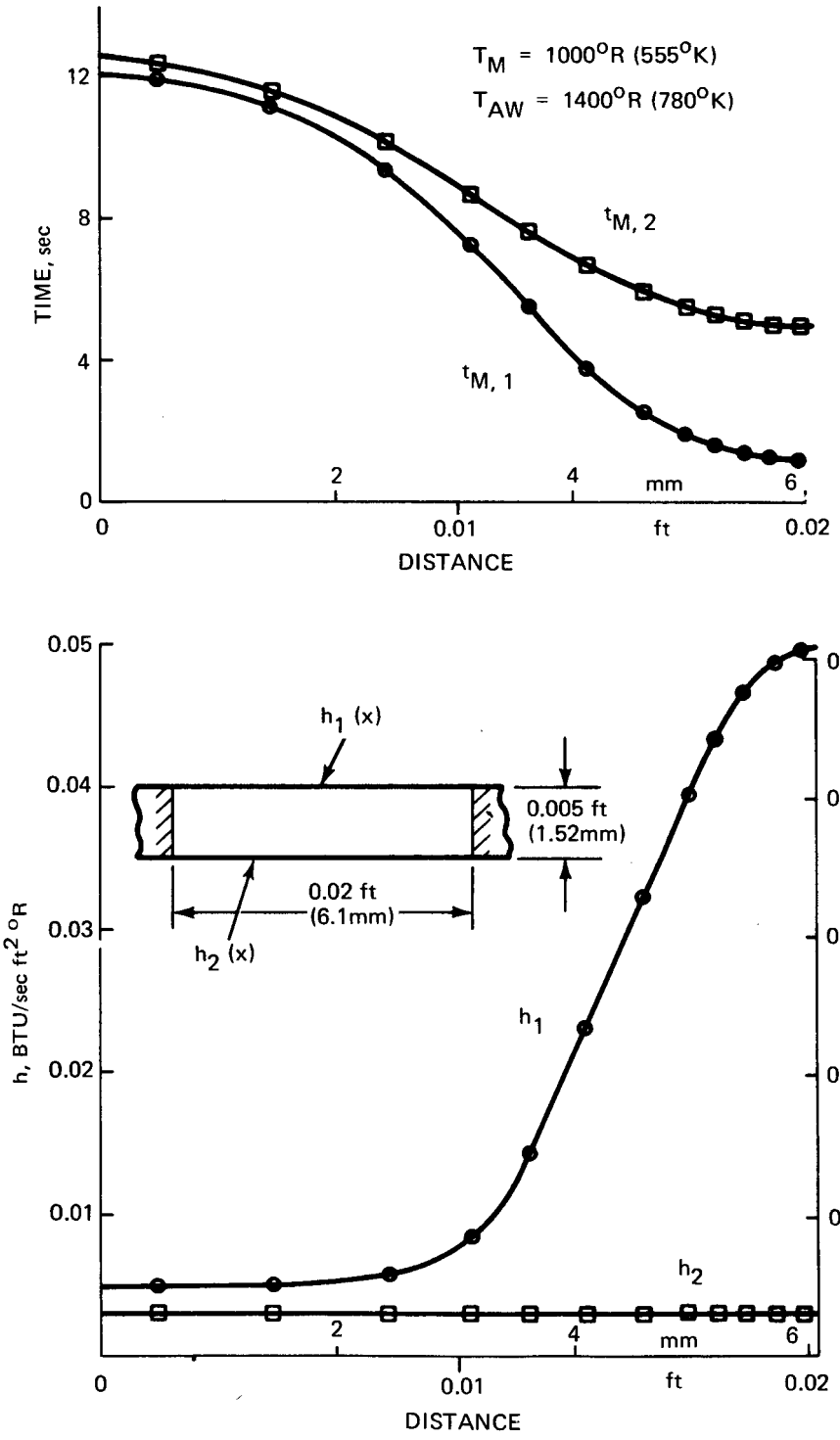


FIGURE 19b. SAMPLE RESULTS FOR SLAB, CONSTANT PHASE – CHANGE TEMPERATURE

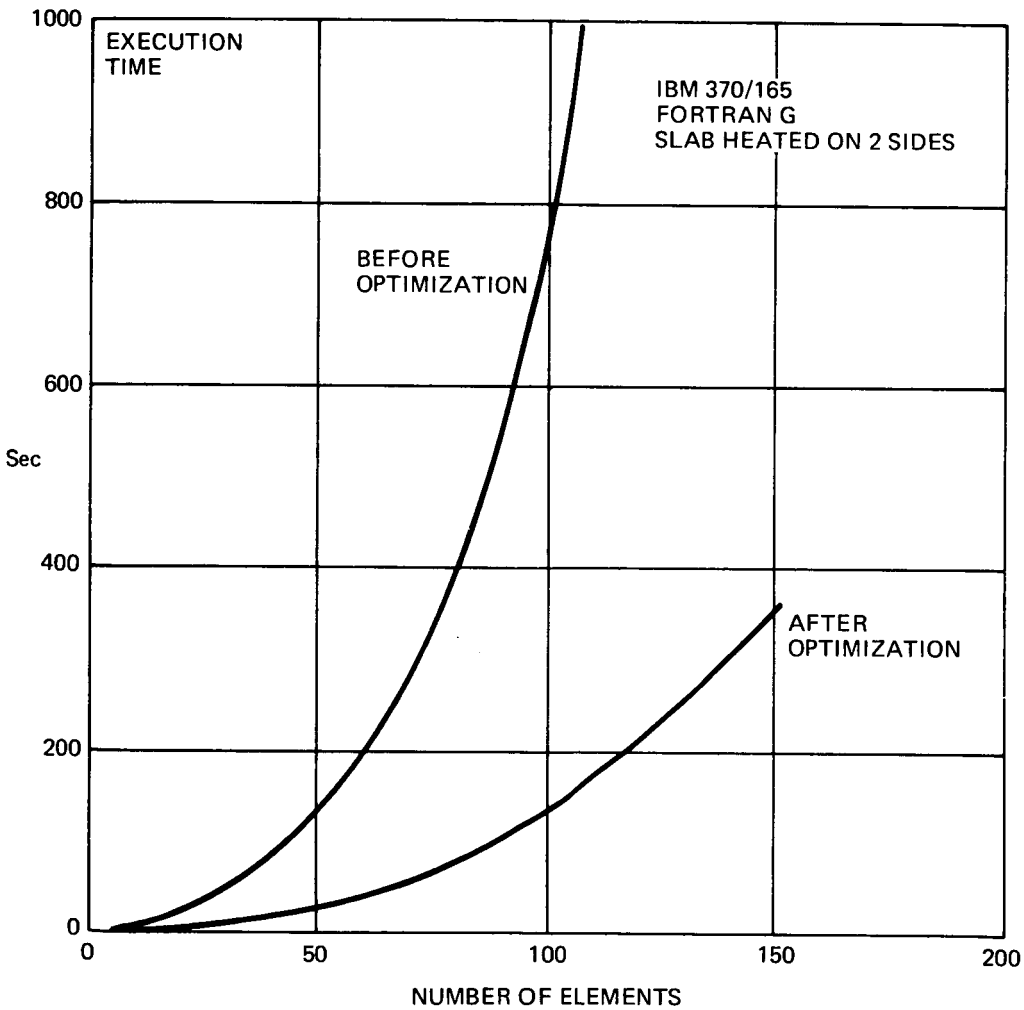


FIGURE 20. COMPARISON OF COMPUTER TIME REQUIRED BY THE NUMERICAL METHOD BEFORE AND AFTER THE OPTIMIZATION

Prospectives for Further Improvements

In spite of the considerable improvements realized during the optimization effort carried out, it is quite likely that considerable more savings on machine time and storage can be obtained.

First of all, the G_{ij} calculation seems not to be necessary for every h_j iteration. Just doing it every two h_j iteration, will cut the machine time by some 40% (assuming even number of iterations and the machine time breakdown per iteration shown in table II).

Moreover, the Newton iteration is the simplest and most straightforward procedure, but it is probable that a more sophisticated procedure may cut the machine time per iteration. This should be worth another 25% savings.

Finally, the present method could be applied also to direct heat transfer problems. Of course, there would be a considerable reduction in computer time over the inverse method. A rough estimate of this reduction is as follows. Currently, a typical slab-like inverse problem uses about $(k+1)$ I calculations of E&E's where k is the number of surface elements and I is the number of iterations. A direct problem requires just one calculation of E&E's. However, the calculations of E&E's in inverse problems accelerate as convergence is approached; this is particularly true of the k calculations needed for the G_{ij}. Probably then the ratio CPU time, direct versus inverse problem, is more likely equal to the ratio of iterations in E&E's calculations via the Jennings algorithm or $i_2/(ki_0+i_2)$ I where i_0 is the number of Jennings iterations during the G_{ij} calculation, i_1 that for each I iteration and i_2 that for a calculation of the E&E's starting with poor zeroth-order guesses. Typical values found have been: $i_0=2$, $i_1=4$, $i_2=20$, $I=4$. Therefore, the expectation is that for a typical slab-like 2-sided problem where $k\sim 20$, the direct method will require -- with the method as it stands -- only some 11% of the CPU time of the corresponding inverse problem.

TYPICAL RESULTS ON LATERAL CONDUCTION EFFECTS

The computer tool developed (the code is described in the appendices; its name is CAPE (Conduction Analysis Program using Eigenvalues)) makes it possible to obtain quantitative results on lateral conduction effects on the data reduction. While the emphasis in this study was on developing the computer tool, in this section we briefly present some of the typical results that have been obtained.

The first question is naturally, how large are the differences in the h obtained with lateral conduction and without? Figure 21 gives the comparison in a representative case, a slab of the small size found for example on the fin of orbiter models of 1 ft. length. As one would expect, lateral conduction returns h's with higher peaks.

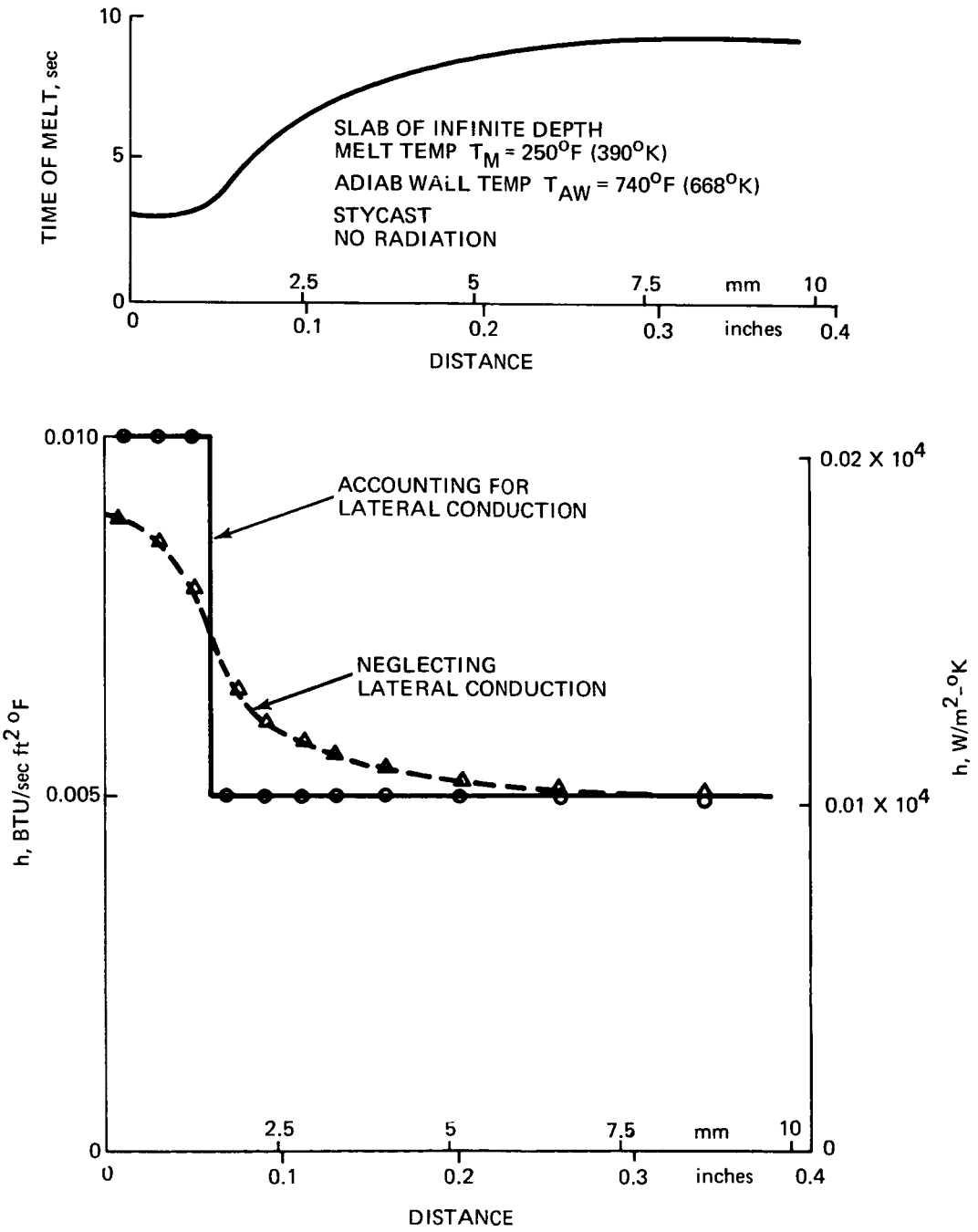


FIGURE 21. EFFECT OF LATERAL CONDUCTION IN THE DATA REDUCTION FOR A SLAB OF INFINITE DEPTH

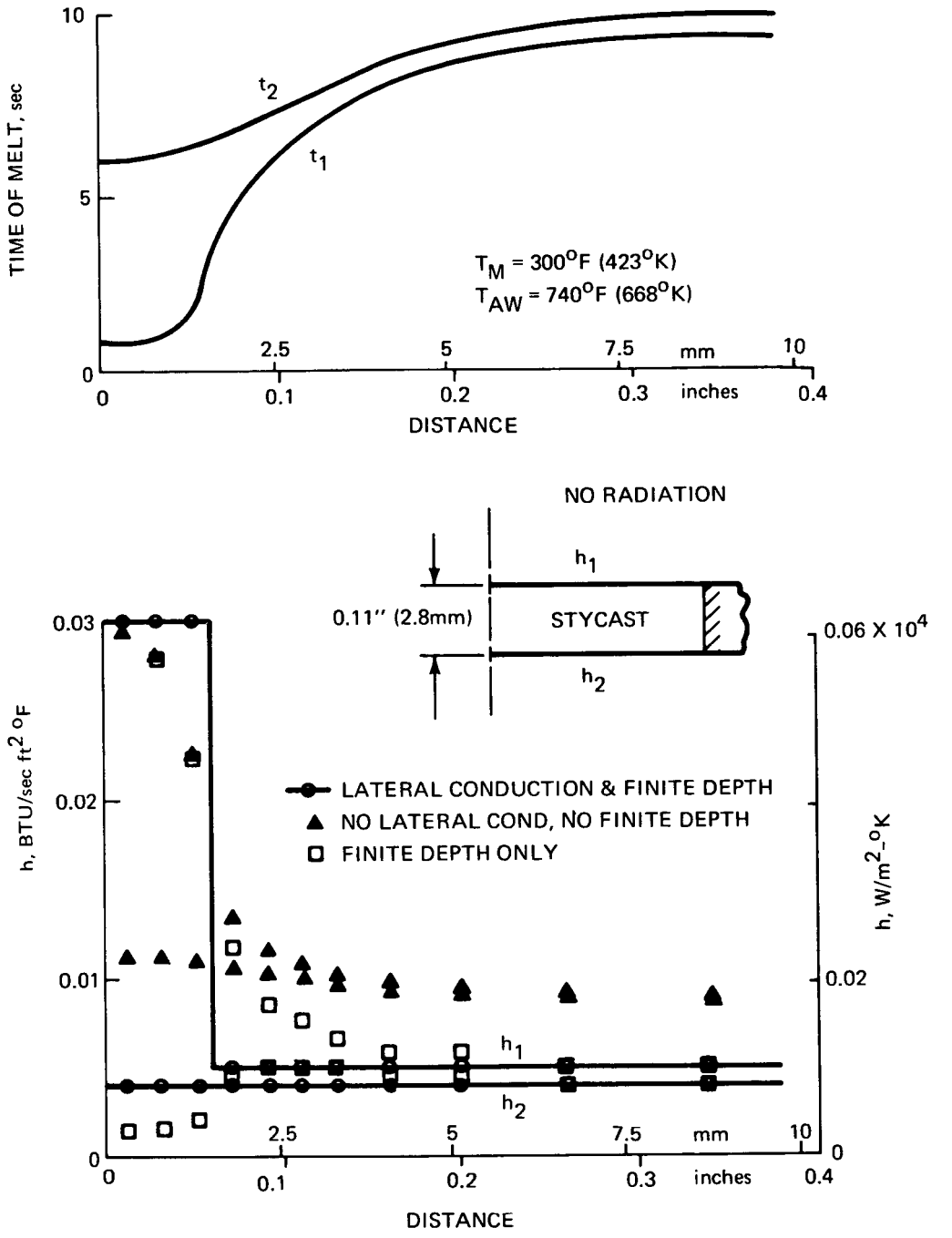


FIGURE 22. EXAMPLE OF COMPLEX INTERPLAY BETWEEN LATERAL CONDUCTION AND FINITE THICKNESS EFFECTS IN SLAB HEATED FROM TWO SIDES

Lateral conduction plays a somewhat more complicated role in the vicinity of a large h(x) gradient, when the slab is thin and in fact heated on both sides as in figure 22.

When the h(x) gradient is not so large, the effects are smaller and qualitatively obvious, see figure 23 where the finite thickness is obtained via the charts of Hunt et al (ref. 4)

In wing l.e.'s of the size typical of orbiter models, the lateral conduction and finite-thickness effects are both large. It is no surprise that the results are rather different than by semi-infinite slab reduction, the only procedure available before this study. Figure 24 gives a good idea of the typical errors one would incur if one were to neglect lateral conduction and finite thickness in such small models.* In this case, it is not obvious how to separate the two effects, lateral conduction from finite thickness, in the nose region. We believe that the basic reason for the large differences in h over the wedge portion is the finite thickness effect. This effect on the wedge portion is very important as $\alpha t/l^2$ is roughly 3.5 and the yes-no chart indicates that corrections are definitely needed. A finite slab calculation over the wedge should eliminate a major portion, but not the entire discrepancy. Lateral conduction effects should still be non-negligible over the wedge. An assessment is obtained by considering that the heat input near the nose reaches the wedge within the times of melt, as the length reached is about 0.017 ft and the wedge portion extends from 0.004 to 0.008 ft.

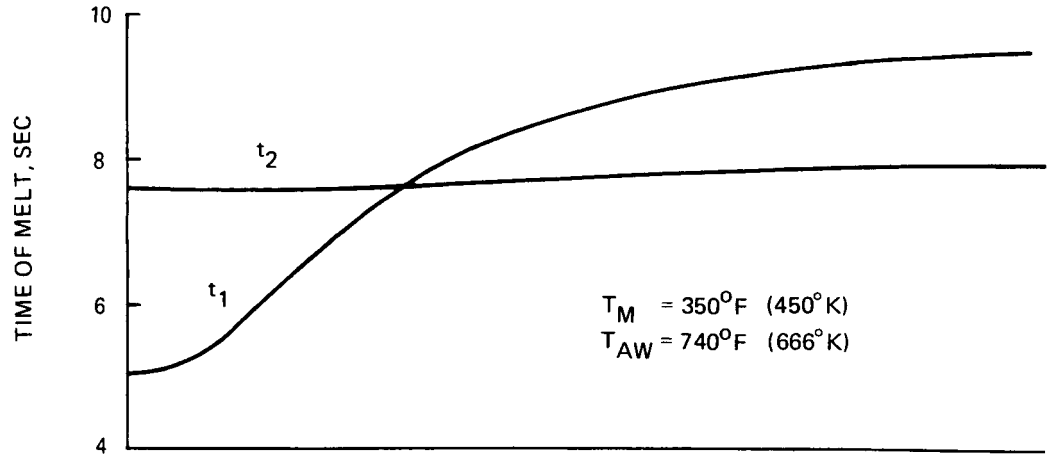
Radiation, as expected, is negligible in typical situations involving Stycast. Fig. 25 gives typical quantitative results on this matter.

Properties variation with temperature in this case of Stycast are also negligible, see fig. 26, which represents an extreme, if somewhat artificial case, in that the melt temperatures were relatively high and the c_p variation with temperature was somewhat pronounced, simulating a behavior found in some previously heated samples of pressure-cast Stycast 2762. The properties used were measured by Revenko and Hansen of the Grumman Aerospace Corporation.

Finally, in exercising this data-reduction computer tool in the presence of significant lateral conduction, we found sometimes large sensitivity of the h's from the (inevitable) errors in measuring the melt times. This problem, that turns out to be a basic problem of the phase-change-paint technique in the presence of either finite thickness or lateral conduction, is discussed separately below.

---

*Incidentally, it should be kept in mind that such lateral conduction errors are not peculiar to the phase-change technique, as severe effects are also found with thin-skin thermocouples in regions of large lateral heating gradients.



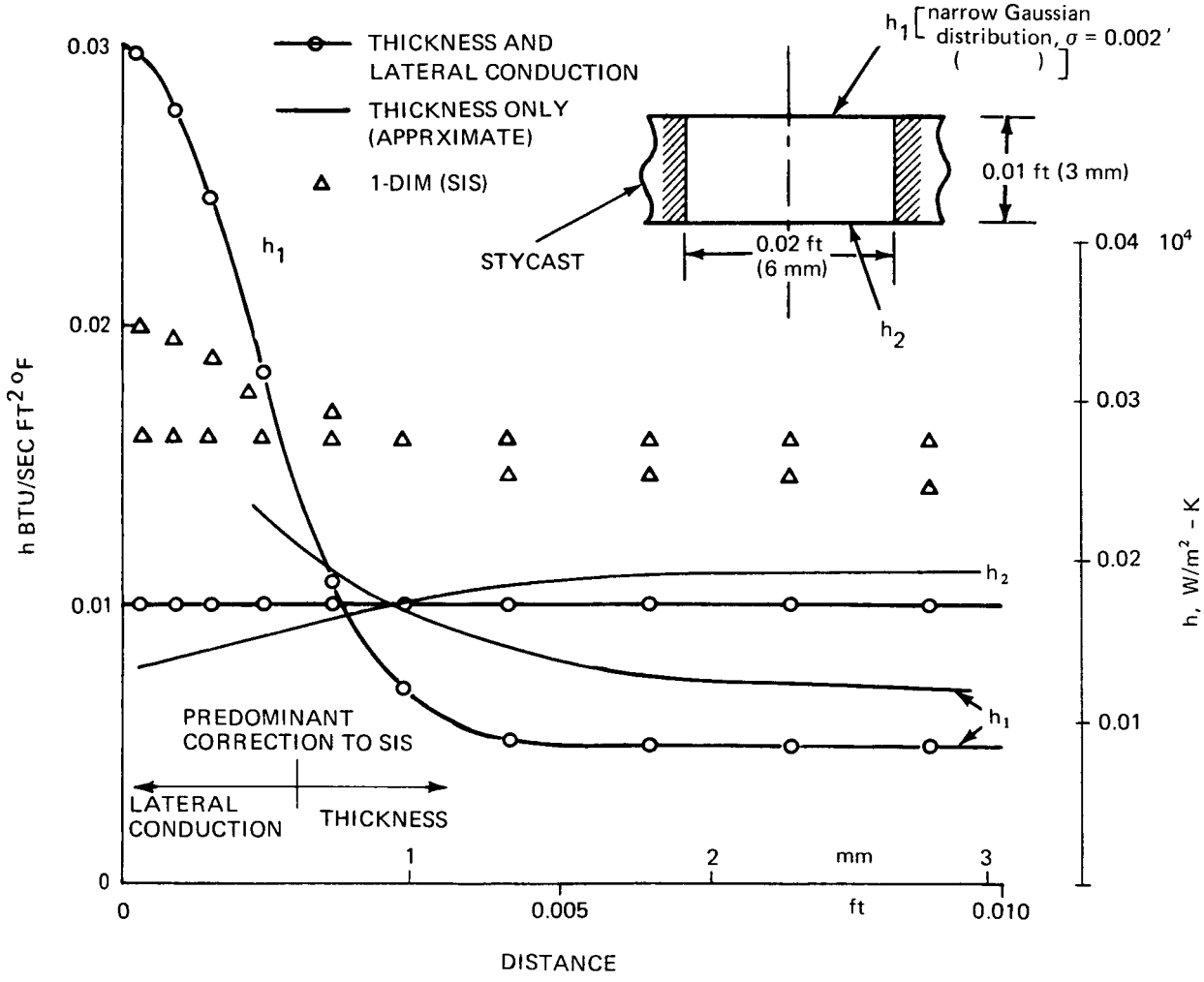


Fig. 23 Example of Simple Interplay Between Lateral Conduction and Finite Thickness In a Slab on Two Sides

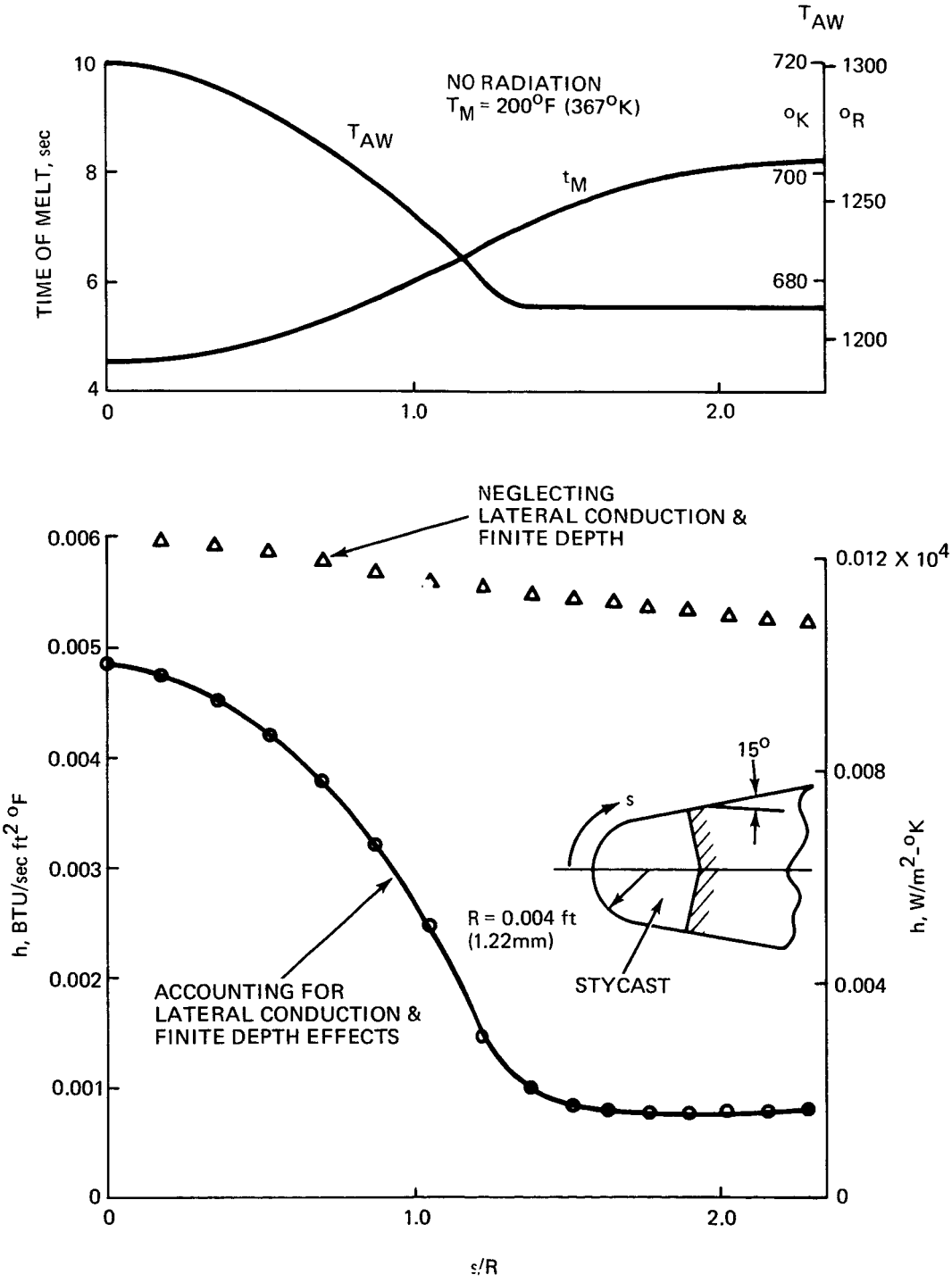


FIGURE 24. COMBINED EFFECT OF LATERAL CONDUCTION AND FINITE DEPTH IN THE DATA REDUCTION FOR A TYPICAL LEADING EDGE

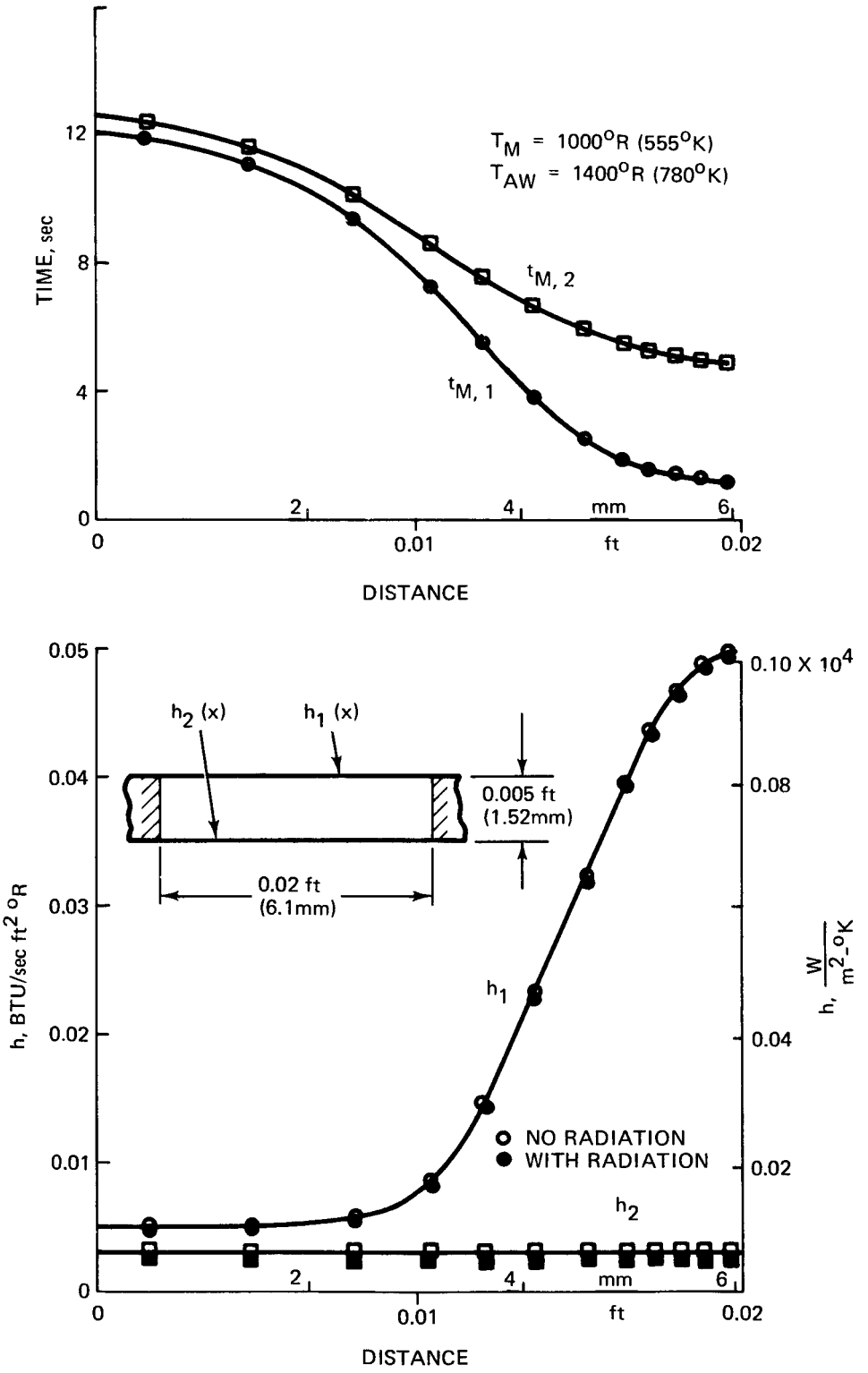


FIGURE 25. SAMPLE RESULTS FOR THE EFFECT OF RADIATION

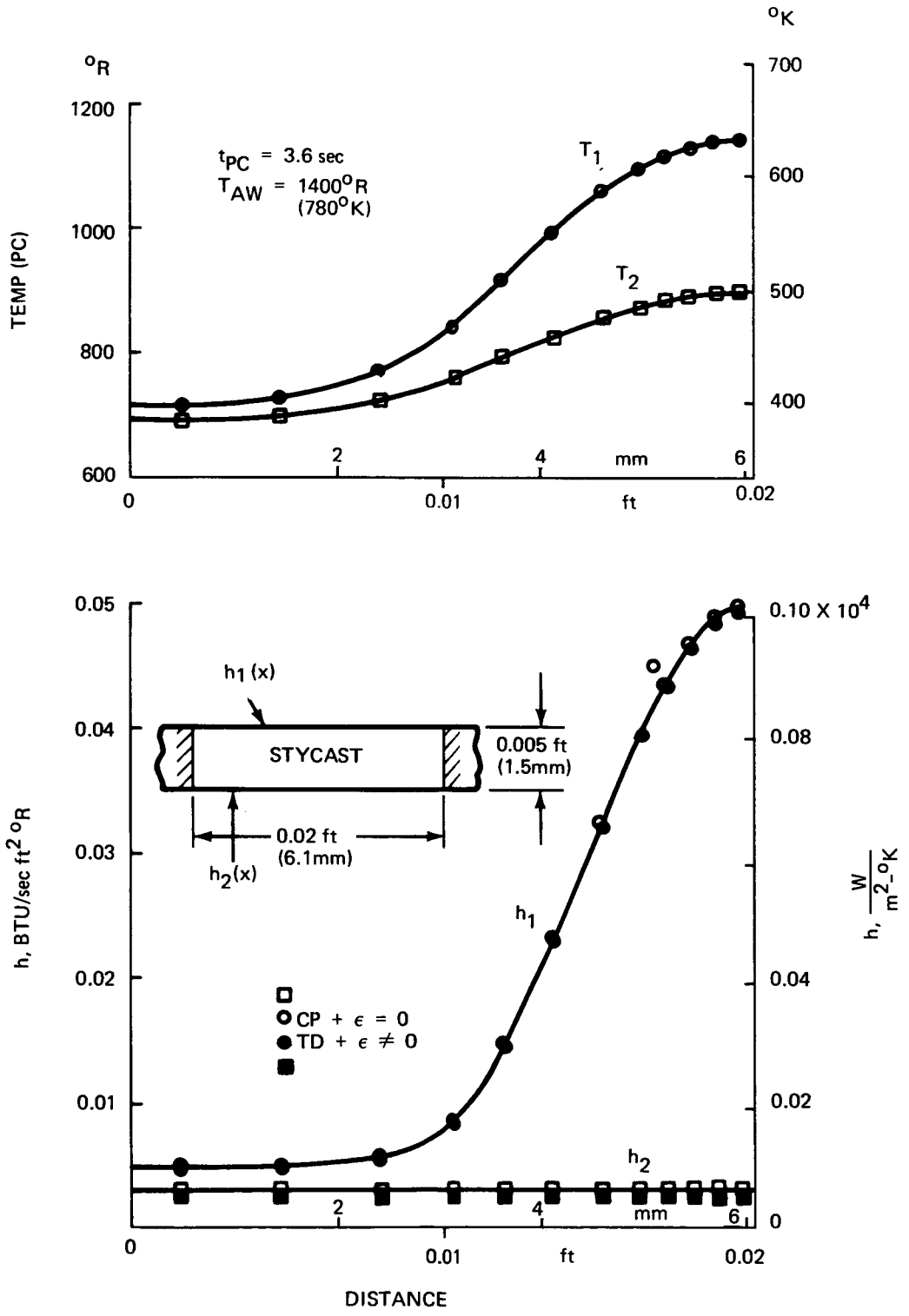


FIGURE 26. SAMPLE RESULTS FOR THE EFFECTS OF TEMPERATURE DEPENDENT PROPERTIES AND RADIATION

WHEN IS THE COMPUTER TOOL NECESSARY TO REDUCE THE DATA?

Parameters Characterizing the Correction to 'Semi-Infinite-Slab' Formula

The problem, of course, is one of providing a single set of "yes-no" charts that will quickly tell the test engineer whether or not he needs to reduce the data via the computer tool that accounts for lateral conduction. More generally, the "yes-no" chart should tell whether the error over the 'semi-infinite slab' reduction (ref. 1) becomes excessive, say 10%, because of (1) lateral conduction, (2) surface curvature, and (3) finite thickness.

In order to maintain simplicity and generality in the chart, we isolated four non-dimensional parameters that globally characterize the three effects. The approach is as follows.

The criterion on whether a slab is thick or thin is given by the approximate rule (Jones and Hunt, ref. 1).

$$\alpha t/l^2 = \begin{cases} < 0.2 \text{ semi-infinite slab} \\ > 0.2 \text{ thin slab} \end{cases}$$

If the slab is thin the corrections of Hunt and Pitts (ref. 3, 4) are required. This criterion, of course, is strictly valid only when other corrections are negligible, but it indicates the key parameter characterizing thickness corrections.

A criterion for curvature can be readily obtained by comparing transient heating of slabs and cylinders under otherwise comparable conditions. The solutions for these geometries are available in the standard heat transfer literature for many types of boundary conditions, (e.g., ref. 24, fig. 4-6 and 4-7). Using the solutions for given and constant heat flux, it turns out that the difference in heat transfer coefficients for the slab and cylinder exceeds 10% when $\alpha t/R_c^2$ is greater than 0.12. This establishes the limit

$$\alpha t/R_c^2 \begin{cases} < 0.12 \text{ no curvature correction} \\ > 0.12 \text{ curvature corrections needed} \end{cases}$$

This identifies a synthetic non-dimensional parameter for curvature corrections. The parameter has the obvious meaning of the distance of heat propagation compared to relevant length, the radius of curvature. Incidentally, at first glance it might appear that curvature corrections are more restrictive than thickness corrections. Actually, this is rarely the case. Except for the leading edges, the curvature is generally much greater than the thickness. Consequently, $\alpha t/R_c^2$ would have a low value and curvature corrections would be negligible.

The situation of the lateral conduction is a little more complicated. The gradients in the heat-transfer coefficients can be essentially of two types, as shown in the sketch of fig. 27. When the gradients are very steep, we have a 'step' or a 'top-hat'. These two cases, even though severe, represent a useful reference.

In seeking synthetic parameters, analytical solutions to the problem of variable heat flux again provide a precious guide. Analytical solutions for a 'top-hat' of given heat flux, are available for both the two-dimensional (strip) and three dimensional (spot) cases on a semi-infinite slab (ref. 7). The corresponding temperature distributions are reinterpreted in terms of required corrections to the semi-infinite slab heat transfer coefficient in fig. 28. It is seen that the effect of lateral conduction on the heat transfer coefficient is "correlatable" by the single parameter $a/\sqrt{\alpha t}$. This parameter can be put in the form $\alpha t/a^2$ completely similar to the parameter characterizing the thickness of the slab. The only difference is that the length scale is the width of the "top-hat" instead of the thickness of the slab. As might be expected, the lateral conduction effects for the spot are greater than for the strip*. If one takes errors in q at the center of the hat in excess of 10% as significant, one can determine the values of the $a/\sqrt{\alpha t}$ parameter at which this occurs:

	$a/\sqrt{\alpha t}$	$\alpha t/a^2$
strip	1.5	0.445
spot	2.0	0.25

The parameter $\alpha t/a^2$ characterizes the errors, since the point at which one is most interested to find h is precisely the center of the hat. In other words, the distribution of the heat input carries explicitly the relevant length scale.

The situation is a little different for the step distribution because here there is no length associated with the heat input. This is a reminder that in the immediate region of the discontinuity, lateral conduction is always important. But as the discontinuity is a gross schematization of a large gradient, the existence of lateral conduction at the discontinuity is just a defect of the schematization. Appropriately, in the case of the 'top-hat', the position for judging whether or not there are significant lateral conduction effects was not at the discontinuity, but at the center of the hat under the presumption that the length a is not much smaller than the length scales over which the problem is examined. It follows that the meaning of the parameter emerging (from fig. 28) is the usual one, a diffusion length $\sqrt{\alpha t}/1.45$ becoming about equal to the semi-width of the hat and thereby affecting the conditions at the center of the hat. In the same fashion, for the step, lateral conduction will be important at distances less than $\sqrt{\alpha t}/1.45$. The question is whether such distances are of the order we are considering. If the slab is of width w, lengths about 0.1w must surely be of interest. In other words, some ten time-of-melt measurements are conceivable over the width w. This fixes the

*The spot problem is axisymmetric and cannot be handled by the code as it stands.

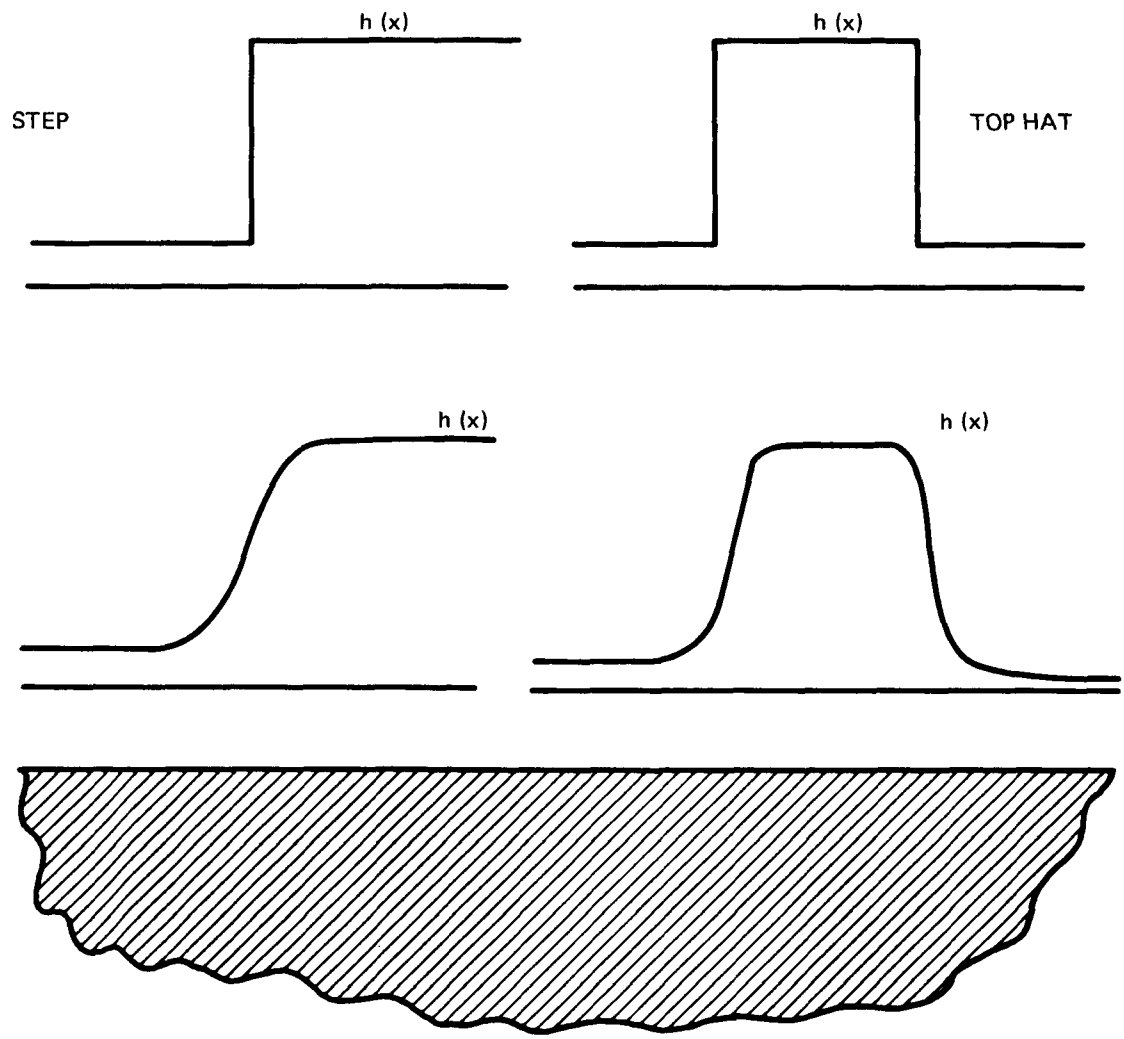


FIGURE 27. SKETCH OF THE TWO ESSENTIAL TYPES OF h DISTRIBUTION

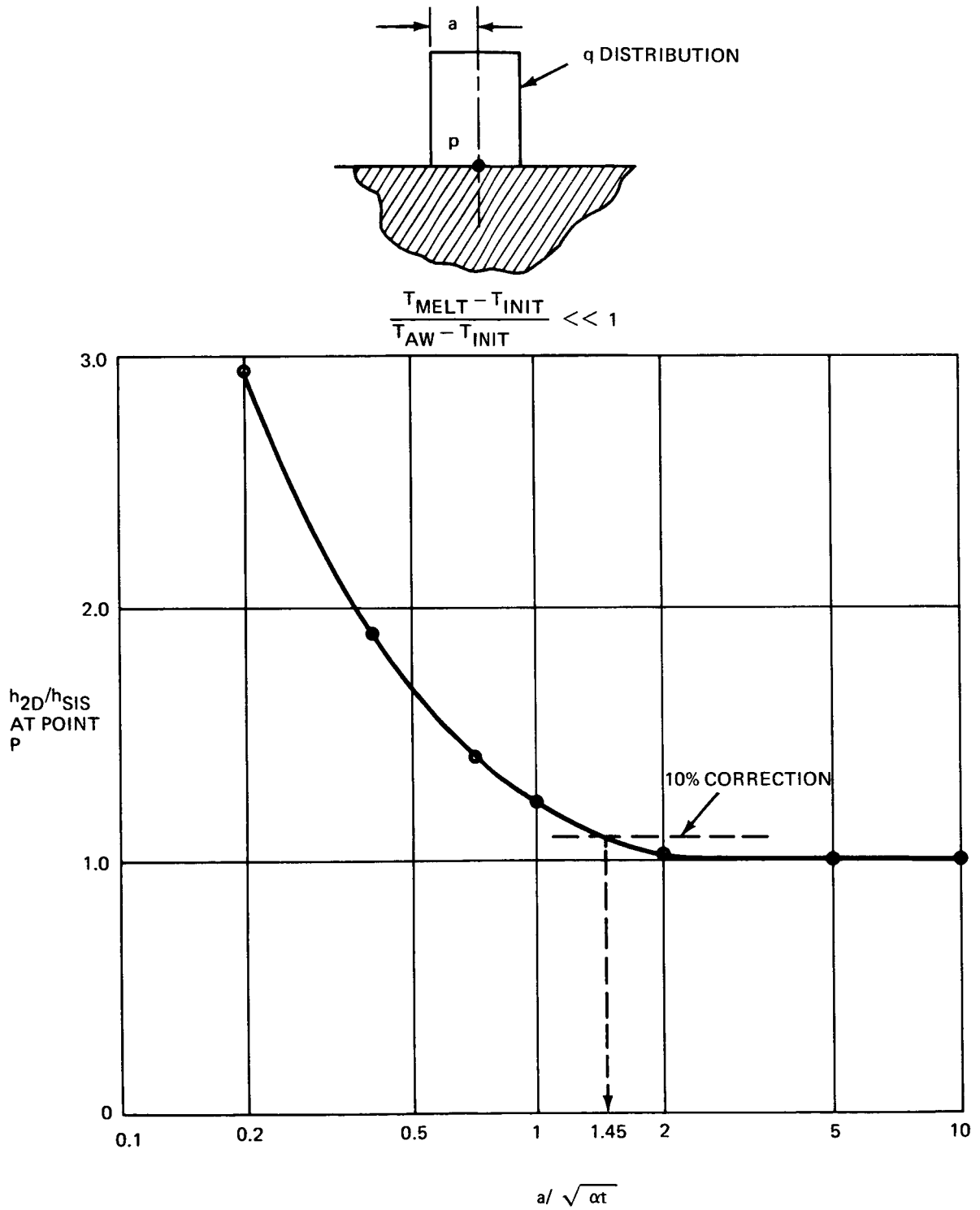


FIGURE 28. CORRECTIONS DUE TO LATERAL CONDUCTION FOR "TOP-HAT" HEATING DISTRIBUTIONS

minimum length scale of interest. Therefore, to be noticeable, lateral conduction in the case of the hat must diffuse at least

$$\frac{\sqrt{\alpha t}}{1.45} \geq 0.1\ w$$

or

$$10^2 \frac{\alpha t}{w^2} \geq 0.47$$

This is the appropriate parameter for step distributions.

'Yes-No' Charts

The limits for lateral conduction can now be combined with the criteria for thickness and curvature in a single "yes-no" chart, fig. 29. This chart maps out the region where different corrections are needed. As special case the chart incorporates the criterion for thickness in the absence of lateral conduction and curvature (Jones and Hunt, ref. 1) and therefore indicates the ranges where the corrections of Hunt and Pitts (ref. 4) may be applied.

The yes-no chart should also be used in deciding where to place the adiabatic boundary in l.e. problems. The distance of this boundary from the stagnation point should at least be greater than the value obtained by $\alpha t/l^2 = 0.2$. The results of fig. 24 do not satisfy this criterion by a factor of 2.

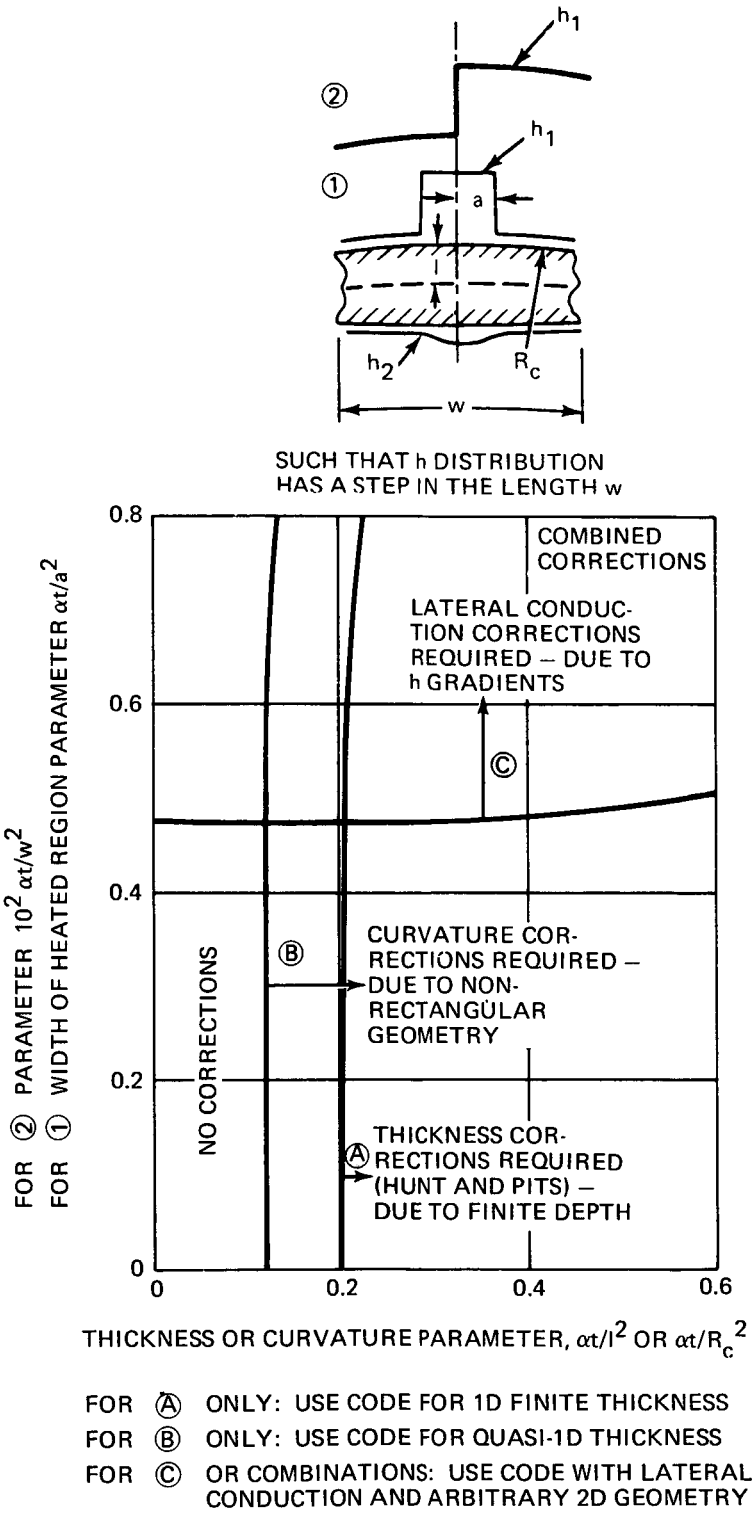


FIGURE 29. "YES-NO" CHART FOR DETERMINING WHEN CORRECTIONS TO ONE-DIMENSIONAL SEMI-INFINITE REDUCTION ARE REQUIRED

ACCURACY OF THE PHASE-CHANGE PAINT TECHNIQUE IN THE PRESENCE OF FINITE THICKNESS AND LATERAL CONDUCTION EFFECTS

Effects of Inaccuracies in the Tunnel Input

In a comparison between results from different computer tools such as fig. 30, where the direct and inverse problems were run with two different computer codes, some waviness was observed in the h distribution from CAPE. We have never encountered this behavior when CAPE was used to solve both the direct and inverse problems. One can speculate that the waviness results from slight inaccuracies in plotting the temperature, cross plotting the times of melt and reading off the appropriate values for input to CAPE. Since these functions were performed by eye, some loss in accuracy is inevitable. If we deliberately perturb the time of melt distribution, in a simulated data reduction, we find the same behavior, as indicated in fig. 31. These perturbations are imperceptible on the time of melt curve; however, they resulted in pushing the h distribution outside the ±5% error limits. Thus, any waviness in the time of melt distribution appears to be greatly amplified in the h distribution. This is particularly serious where points are closely spaced. The same behavior is observed if, rather than controlled perturbations of the melt times at specific locations, random melt time perturbations of given rms are generated by computer as in fig. 32.

An essential point that is to be stressed is that this happens only in regions where lateral conduction effects are important or, more precisely, when dh/dx are large enough to make them so. See, in fig. 32, how well h is recovered when dh/dx is not inordinately large. The problem disappears, if the melt times are reduced with the semi-infinite slab approximation - except of course that the h estimates are wrong. The 'waving' of the solution, and particularly the higher amplitudes obtained by refining the grid have a simple physical interpretation.

One can readily imagine what would occur when the time of melt distribution is obtained from tunnel measurements with all the inherent uncertainties. Therefore, the problem cannot be ignored. In fact, it appears of the greatest importance if one is to be able to reduce the tunnel data.

Smoothing of Tunnel Inputs

The natural solution to the problem is to 'smooth' the raw time-of-melt data prior to use in CAPE or any other inverse tool, at least when lateral conduction is non-negligible and is to be taken into account. Naturally, smoothing of raw-data is nothing new in wind-tunnel data reduction and is included in many automatic data reduction systems.

As tool for smoothing 'noisy' melt time data, we can borrow a recently developed

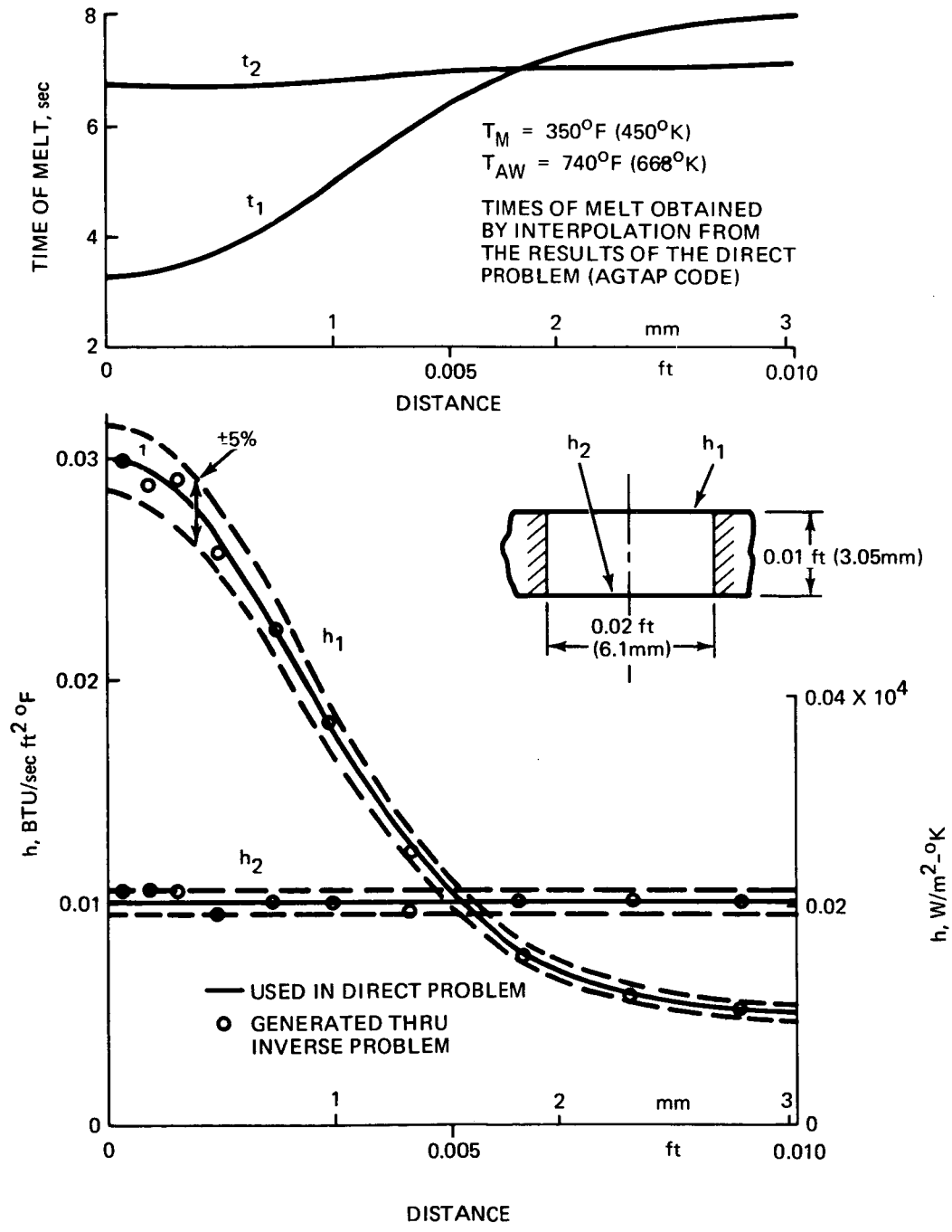


FIGURE 30. WIGGLES GENERATED IN THE INVERSE PROBLEM WHEN THE INPUT DATA ARE OBTAINED BY ANOTHER HEAT CONDUCTION CODE

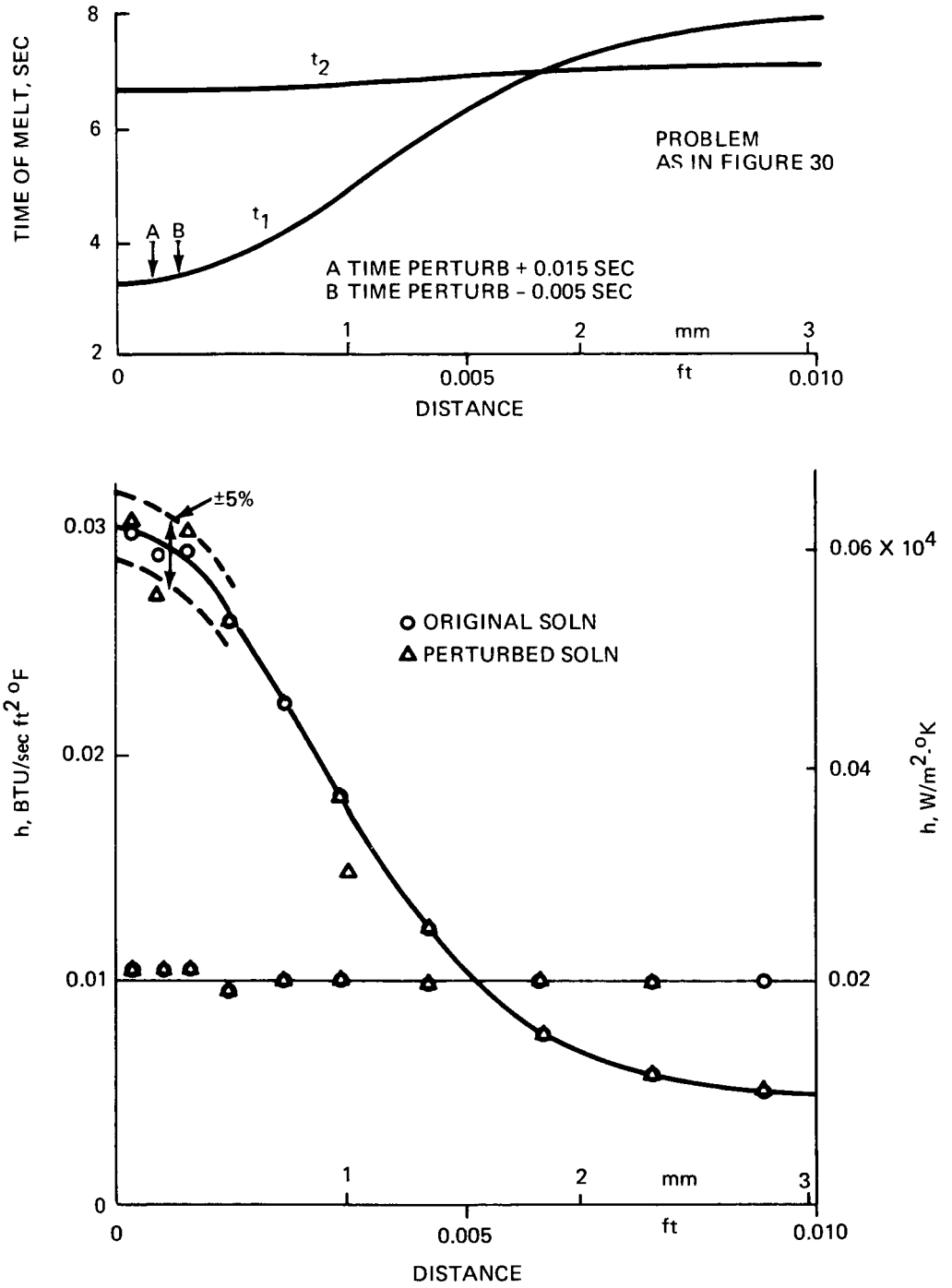


FIGURE 31. SENSITIVITY OF INVERSE SOLUTION TO DELIBERATE PERTURBATION OF THE TIMES OF MELT

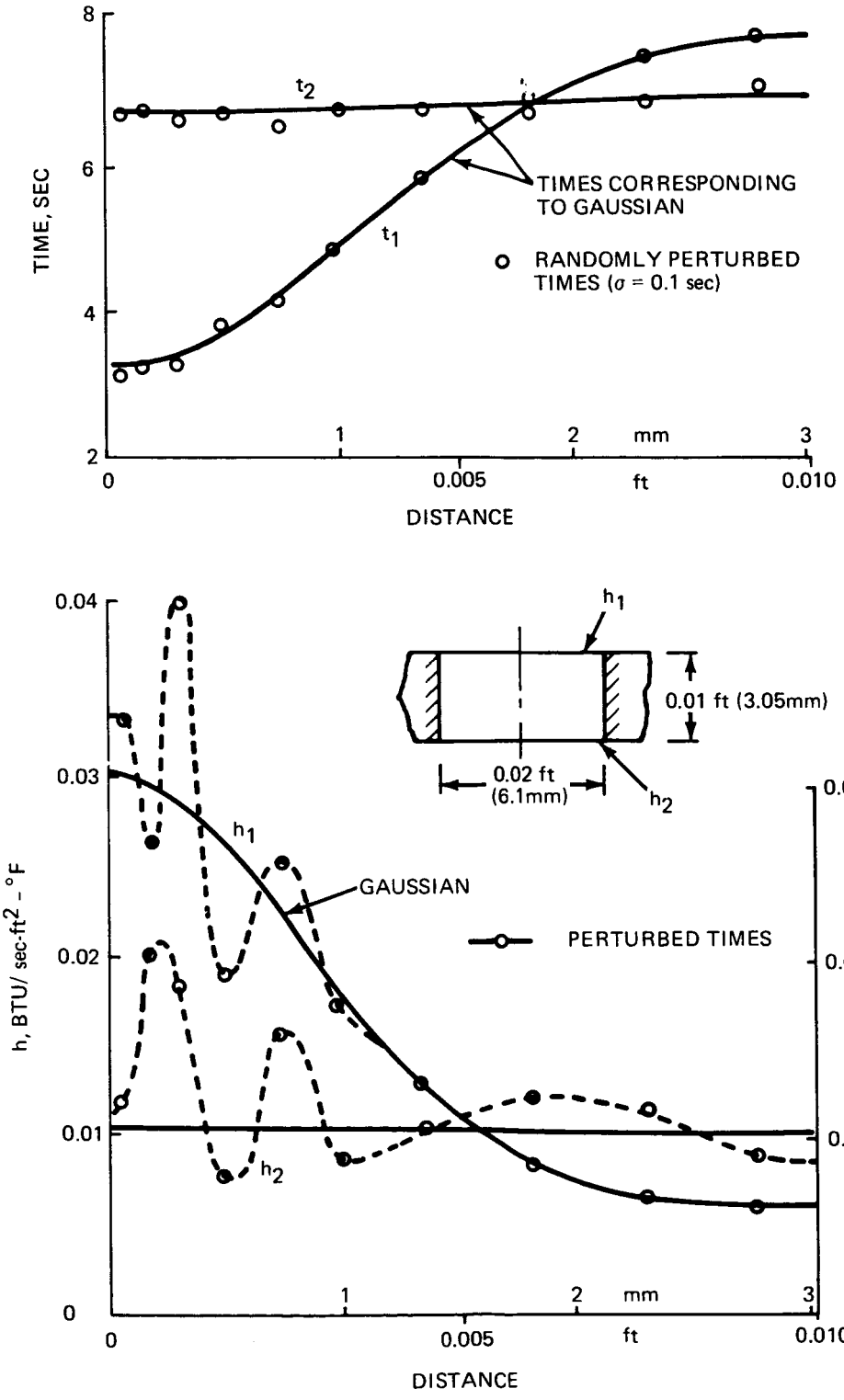


FIG. 32 – DATA REDUCTION WITH RANDOMLY PERTURBED MELT TIMES

technique, the 'smoothing spline' (ref. 25, 26). In essence, the method allows one to choose an arbitrary degree of smoothness by specifying a parameter R. When $R = 0$ there is no smoothing and the method produces a spline fit through all the data. When $R = 1$ one obtains ultimate smoothing, i.e., a straight line. For intermediate values of R, fig. 33 shows some examples of various degrees of smoothing. The right amount of smoothing is to be found by trial and error.

In detail, the smoothing spline consists of the following. Given a set of raw data g_i associated with a coordinate s_i, $i = m$ to n, in the interval for $s\{0, 1\}$, one computes a set of $f_i(s)$, a third-order spline for each interval. The four parameters in each $f_i(s)$ are chosen so as to minimize the expression:

$$R_1 \sum_{i}^{m,n} \int_{s_i}^{s_{i+1}} f''^2 ds + R_2 \sum_{i}^{m,n} \left(\frac{g_i - f_i(s_i)}{Q} \right)^2$$

where $R_1 = 1 - R_2$, $R_2 = (1 - R)^4$ and Q are chosen parameters that control the degree of smoothing. $R_2 = 0$ gives the least square straight line fit, while $R_1 = 0$ gives the normal spline.

The technique has been applied to the randomly perturbed times of melt of fig. 32. The results were not favorable as shown in fig. 34. Indeed the smoothing process reduces the waviness in the h results, but this does not prevent unacceptable deviations of the h values near the peak. The degree of smoothness in fig. 34, $R \simeq 0.3$, is just about the maximum values that one can reasonably use, since as R increases the times of melt are modified to the point of changing the problem, see fig. 35. Naturally, in going from $R = 0.1$ to $R = 0.3$ there are definite improvements, as fig. 36 shows, but the results remain unsatisfactory.

If, instead of smoothing, the melt times are just fitted and the fit is used in the data reduction, the results are equally unsatisfying, apparently because the fit is poor at the ends of the interval. This is a facet of the problem that should be looked at more closely as perhaps there is a chance for at least better results than fig. 36.

It becomes clear that the problem is of deeper nature and is not solved by straightforward treatment of the raw data before reduction.

Interpretation of the Difficulties Encountered

The difficulties experienced lead us to explore the fundamental problem of the sensitivity of the phase-change paint technique to errors in the times of melt. Clearly, this is the crucial point emerged above, i.e., the fact that small variations in the times of melt give large variation of h resulting from the inverse problem.

A basic condition for a sound experimental technique requires that the errors in the

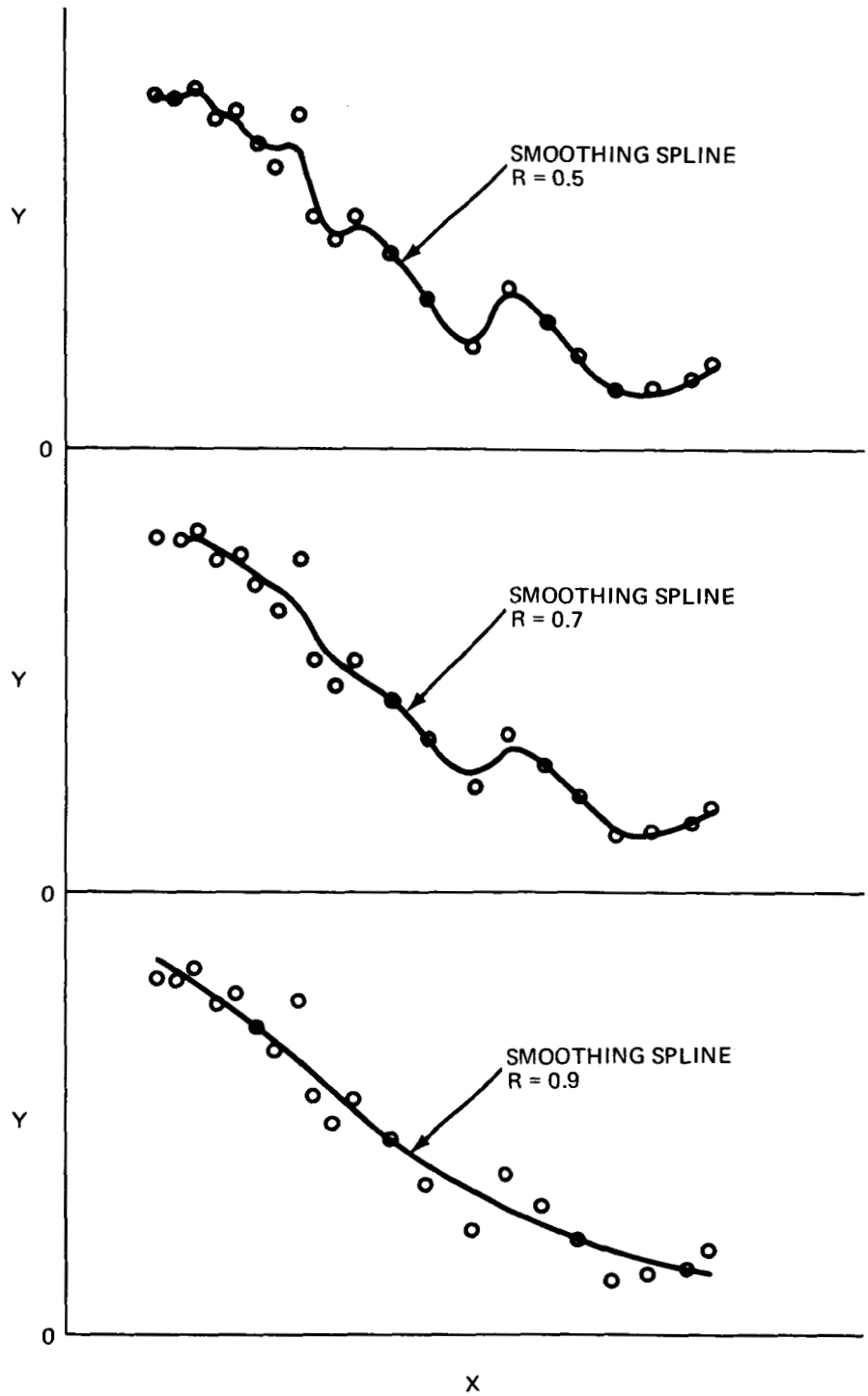


FIGURE 33. SMOOTHING SPLINE FOR TREATMENT OF "NOISY" EXPERIMENTAL DATA

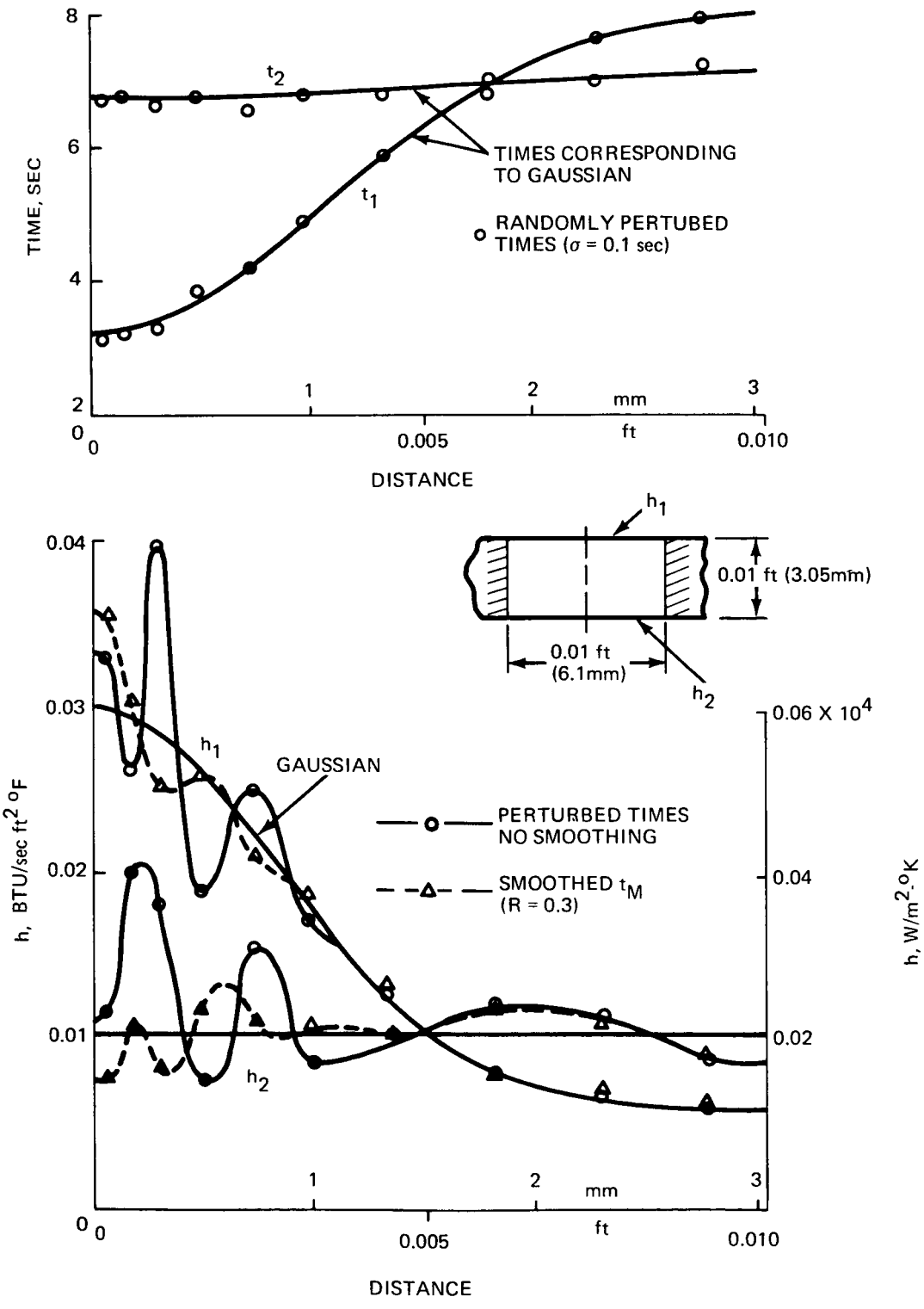


FIGURE 34. DATA REDUCTION AFTER STRAIGHTFORWARD SMOOTHING OF THE MELT TIMES

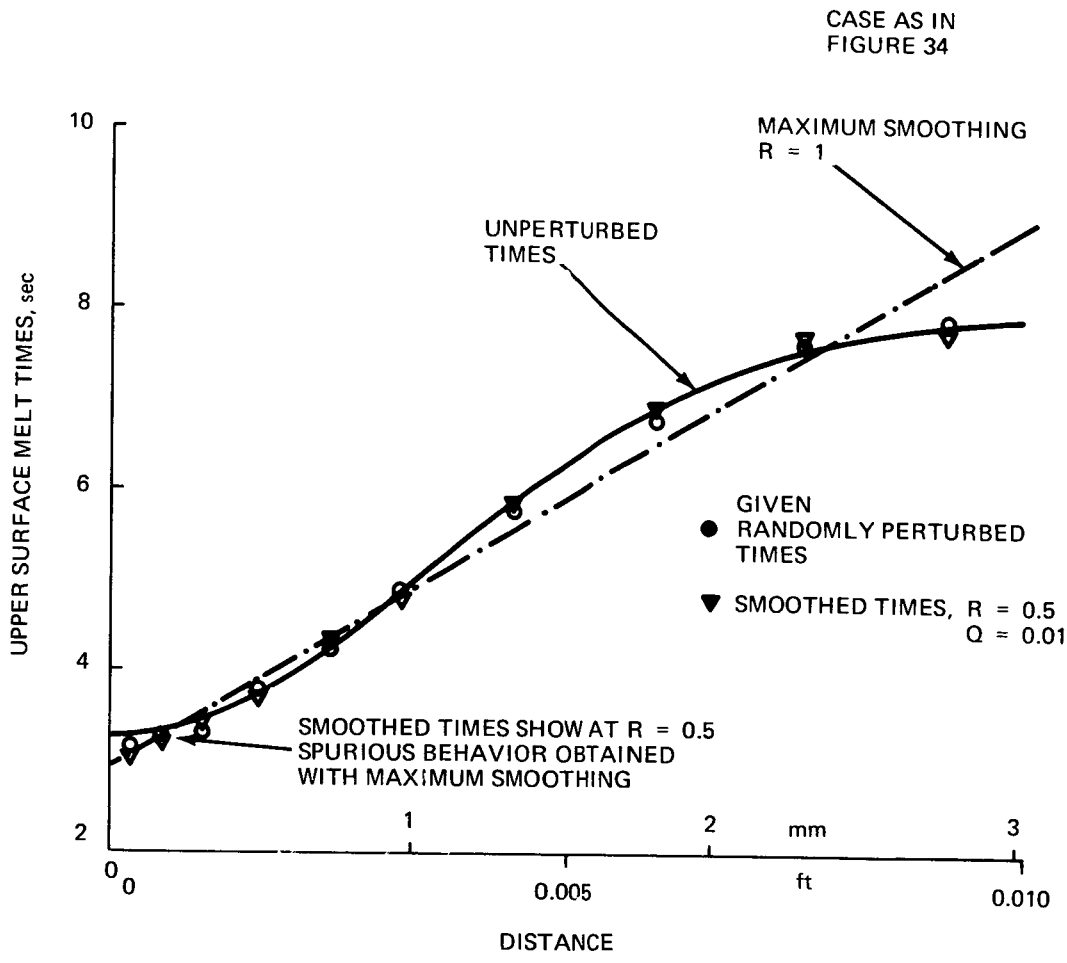


FIGURE 35. EXCESSIVE SMOOTHING MODIFIES INTRINSIC CHARACTERISTICS OF GIVEN SET OF MELT TIMES.

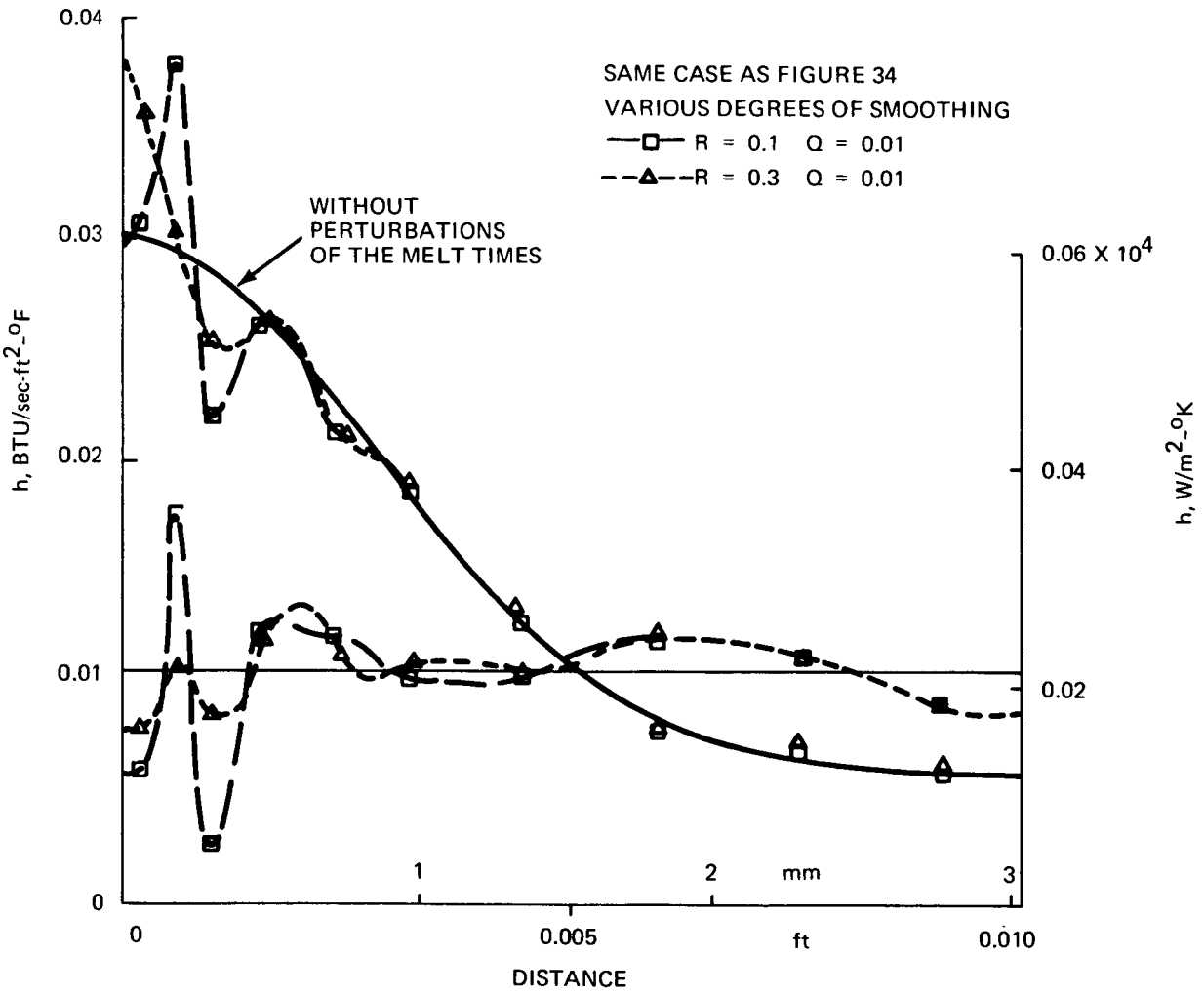


FIGURE 36. EFFECT OF VARIOUS DEGREES OF SMOOTHING OF THE MELT TIMES

measured t_m are not magnified into much larger errors of the results of h, i.e., the error amplification

$$\frac{t_j}{h_i}\frac{\partial h_i}{\partial t_j} = 0(1)$$

so that the % error in h_j is about the % error in the t_i. As a typical value for accurate and sound techniques, we can take error amplification less or equal to 2.

The question is what values have the $(t_j/h_i)(\partial h_i/\partial t_j)$ in the typical applications? The sensitivity found suggests that the $(t_j/h_i)(\partial h_i/\partial t_j)$ are unduly large. This can be checked by calculating them.

Error Amplification in the Semi-Infinite Slab Reduction

In situations where the semi-infinite slab reduction is accurate, the calculation of the error amplification is straightforward. By differentiating the explicit expression for h (t_m) (ref. 9) it follows that

$$-\frac{t_{m,j}}{h_i}\frac{\partial h_i}{\partial t_{m,j}} = \begin{cases} \frac{1}{2} \text{ for } i = j \\ 0 \text{ for } i \neq j \end{cases}$$

Therefore, as one would have suspected from the general success of the phase-change paint technique, the technique is sound and the errors are not amplified and most obviously, are not propagated.

Error Amplification in the Presence of Finite-Thickness Effects

In the presence of important finite thickness effects, but no lateral conduction, it is obvious that

$$\frac{t_{m,j}}{h_i}\frac{\partial h_i}{\partial t_{m,j}} = 0$$

if $j \neq i$ and not on the 'other side' of i. Therefore, at each surface point i, there are two influence coefficients (calling 1 and 2 the two sides of the slab):

$$A_{11} = \frac{t_{m1}}{h_1}\frac{\partial h_1}{\partial t_{m1}}$$

$$A_{12} = \frac{t_{m2}}{h_1}\frac{\partial h_1}{\partial t_{m2}}$$

Naturally, the same considerations are valid for the 'other side' of point i. These coefficients must be obtained numerically, for example with CAPE or with the tool described in ref. 4. This has been done for all slabs heated on two sides with same time of melt and same T_{aw} on both sides. In this case there is only one independent non-dimensional parameter, $\tau_{m1} = \alpha t_{m1}/l^2$. Therefore, A_{11} and A_{12} depend only on this parameter. Of course, as $\tau_{m1} \rightarrow 0$, we recover the semi-infinite slab and therefore $A_{11} \rightarrow -\frac{1}{2}$ and $A_{12} \rightarrow 0$. For arbitrary τ_{m1}, A_{11} and A_{12} are the universal functions given in fig. 37.

The important result in fig. 37 is that at large τ_{m1} A_{11} and A_{12} are very large. Therefore, the waviness encountered in the data reduction is the result of large error amplification intrinsic in the physics of the dependence of the h on the t_m when finite-thickness effects are important.

There follows the conclusion that at very large t_m and in the absence of lateral conduction, the phase-change paint technique is unsound since it does not satisfy the condition that A_{11} and A_{12} be less than, say, 2. Moreover, using this limit, one can deduce from fig. 37 that the range of applicability in the presence of finite-thickness effects is below about $\tau_m = 2.5$.

One relatively common situation in which one may encounter τ_m of the order of 2 is the trailing edge of a thin fin section such as the one studied in ref. 4.

Error Amplification in the Presence of Lateral Conduction.

Strong lateral conduction must also result in large error amplification. This is suggested by the 'waviness' experienced in the data reduction process and, more fundamentally, by the physical idea behind the error amplification. Therefore, along lines similar to the finite-thickness case, one can determine the limits of applicability of the phase-change paint technique in the presence of strong lateral conduction. Also, it seems probable that the potential of stainless steel models or data reduction on the presence of fully three-dimensional lateral conduction should be seriously questioned.

These matters have not been looked into, but deserve a close quantitative study.

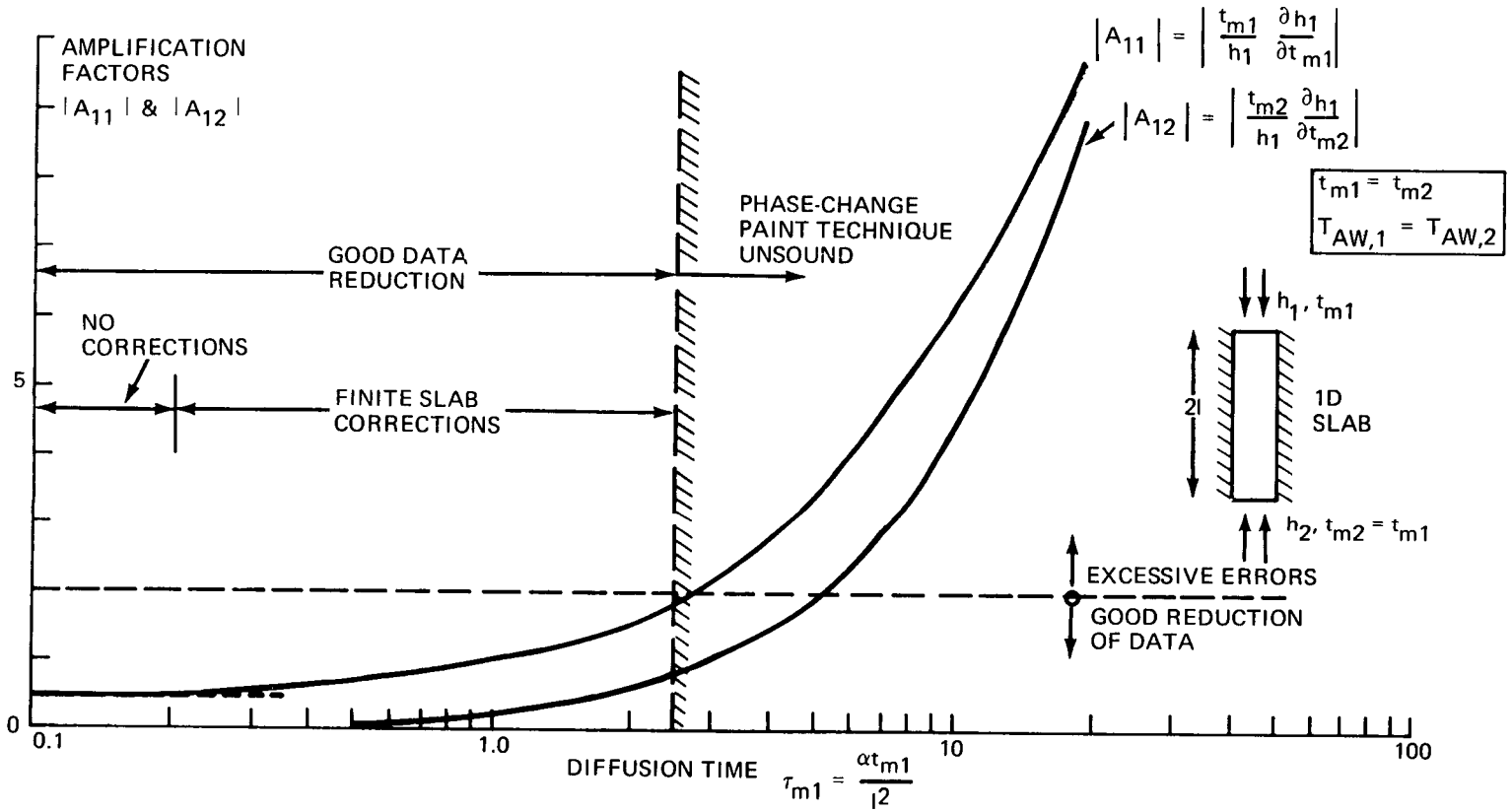


FIGURE 37. AMPLIFICATION FACTORS AND RANGE OF GOOD DATA REDUCTION FOR ONE DIMENSIONAL FINITE-THICKNESS SLAB WITH EQUAL TIMES OF MELT

CONCLUSIONS

The main conclusions of this study are:

1. A single and synthetic chart has been developed that permits one to quickly decide when lateral conduction and finite-thickness effects have to be accounted for in the data reduction for the three geometries of interest.

2. A computer tool has been developed to carry out the data reduction for slabs, l.e. and arbitrary two-dimensional geometries. For typical slab problems, the machine time is about 6 min. on the CDC 6600, with an accuracy of 1 to 2% on the h calculation. The labor needed for the preparation of the input data is 15 to 30 minutes per case.

3. The computer tool can also handle a special data reduction problem for l.e.'s, where only one time of melt is supplied. It is now possible to evaluate, by appropriate experimentation, this type of data reduction. The machine time for this problem is about 2 min. on the CDC 6600.

4. Lateral conduction is found to have considerable effects on the h value returned especially for l.e. of wings and fins of sizes typical 1 ft models of complete vehicles.

5. The inaccuracies in time-of-melt data produce considerably amplified inaccuracies in the h when strong finite-thickness or lateral conduction effects are present. Straightforward smoothing of the raw data does not help. This behavior appears to limit the range of applicability of the phase change paint technique. For finite-thickness effects, the limit is roughly $\alpha t_m/l^2 \sim 2.5$ where α is the thermal diffusivity, t_m the time of melt and l the thickness of the slab.

6. As byproduct of the effort, an interesting numerical method has been developed that applies to heat conduction ideas evolved in control theory and structural analysis. The method can be applied also to direct problems of unsteady heat conduction with rather low computer time estimated, in two dimensions and with constant properties, at about 11% of the corresponding inverse problem.

REFERENCES

1. Jones, R. A. and Hunt, J. L.: Use of Fusible Temperature Indicators for Obtaining Quantitative Aerodynamic Heat Transfer Data, NASA TR-R-230 (February 1966).

2. Jones, R. A. and Hunt, J. L.: Use of Temperature Sensitive Coatings for Obtaining Quantitative Aerodynamic Heat-Transfer Data, AIAA J., 2, 1964, 1354 - 1356.

3. Hunt, J. L. and Pitts, J. I.: Thin Wing Corrections for Phase-Change Heat-Transfer Data, J. Spacecraft, 8, 1971, 1228 - 1230.

4. Hunt, J. L., Pitts, J. I., and Richie, C. B.: Application of Phase-Change Technique to Thin Sections with Heating on Both Surfaces, NASA TN D-7193 (August 1973).

5. Kutilek, J. (Ed.): Thermal Model Handbook, Apollo Lunar Module (LM), Vol. III, Propulsion Systems, Grumman Aerospace Corp., April 1970.

6. Smith, James P.: SINDA User's Manual, TRW Systems Group Report 14960-H001-RO-00, April 1971.

7. Carslaw, H. S. and Jaeger, J. C.: Conduction of Heat in Solids, Oxford University Press, London 1959.

8. Dusinberre, G. M.: Heat-Transfer Calculations by Finite Differences, International Textbook Co., Scranton 1961 Hunt & Pits (pp. 1228-1229).

9. Dorrance, W. H.: Viscous Hypersonic Flow, McGraw-Hill, 1962.

10. Nagel, A. R., et al: Analysis of Hypersonic Pressure and Heat Transfer Tests on Delta Wings with Laminar and Turbulent Boundary Layers, NASA CR 535, 1966.

11. Moretti, G., Grossman, B., Marconi, F., A Complete Numerical Technical for the Calculation of three dimensional inviscid Supersonic flows, AIAA Paper No. 72-192 (1972).

12. Jennings, A. and Orr, D. R. L.: Application of Simultaneous Iteration Method to Undamped Vibration Problems, Int. J. Numer. Methods in Engngr., 3, 1971, 13 - 24.

13. Ramamurti, V.: Application of Simultaneous Iteration Method to Torsional Vibration Problems, J. Sound Vibration 29, 1973, 331 - 340.

14. Goldberg, S.: Matrix Reduction Procedure for Vibration Analysis, Grumman Aerospace Corp., ADR-02-01-72.1, April 1972.

15. Martin, R. S., Reinsch, C. and Wilkinson, J. H.: Householder's Tridiagonalization of a Symmetric Matrix, Numer. Math., 11, 1968, 181 - 195.

16. Anon., Mathematics Software Library (GRUMLIB), User's Manual, Vol. I and II, Grumman Data Systems, 1973 (revised).

17. Jennings, A.: A Direct Iteration Method of Obtaining Latent Roots and Vectors of a Symmetric Matrix, Proc. Cambridge Phil. Soc., 63, 1967, 755 - 765.

18. Clint, M. and Jennings, A.: The Evaluation of Eigenvalues and Eigenvectors of Real Symmetric Matrices by Simultaneous Iteration, Computer J., 13, 1970, 76 - 80.

19. Vachris, A.: The Real Symmetric Eigenvalue Problem, Polytechnic Institute of Brooklyn, Master's Thesis, June 1969.

20. Wilkinson, J. H., The Algebraic Eigenvalue Problem, Oxford Univ. Press, 1965.

21. Rutishauser, H.: Computational Aspects of F. L. Bauer's Simultaneous Iteration Method, Numer. Math. 13, 4-13, 1969.

22. Rutishauser, H.: Simultaneous Iteration Method for Symmetric Matrices, Numer. Math. 16, 1970, 205-223.

23. Bauer, F. L.: Das Verfahren der Treppeniteration and verwandte Verfahren zur Loesung algebraischer Eigenwertprobleme, Z. angew. Math. Physik, 8, 1957, 214 - 235.

24. Eckert, E. R. G. and Drake, R. M.: Heat and Mass Transfer, 2nd Ed., McGraw-Hill, New York, 1959.

25. Reisch, C. H.: Smoothing by Spline Functions, Numer. Math., 10, 1967, 177-183.

26. Berghaus, D. G. and Cannon, J. P.: Obtaining Derivatives for Experimental Data, etc., Exp. Mech., 2, 1973, 38-42.

APPENDIX A

USER ORIENTED DOCUMENTATION OF THE CODE

This appendix contains the amount of information strictly necessary to the engineer to be able to use the code. The details of the code structure are set out in Appendix B. The code named CAPE (Conduction Analysis Program using Eigenvalues) solves the two type of inverse problem mentioned in the introduction, using the method described in the body of the report. CAPE is programmed in Fortran IV for the CDC 6600. A version with a few double precision statements is also available for the IBM 360/175.

The geometries handled by the code are those indicated in Figures 4, 5 and 6. For slab-like geometries, CAPE can also handle one-dimensional finite slabs, quasi-one dimensional arbitrary geometry cases and semi-infinite (in depth) two-dimensional geometries. The calculation of h via the 'semi-infinite-slab' (SIS) formulae is also done in every case and printed out.

The grid lay-outs are given in figures 4, 5 and 6. In order to operate the code, the first step is to assemble the input information, i.e.,

a) establish dimensions of the slab, afterbody angle for the l.e., and geometry definition of the arbitrary four-sided geometry;

b) decide for one of the two types of problems, the one where all the times of melt are given or, for the l.e., the special option where only the minimum melt time around the l.e. is given; establish, for slab-like problems, whether both sides are heated or just one.

c) tabulate the time(s) of melt together with the position on the surface(s) at which they are known.

d) secure the model material properties, i.e. density, conductivity and specific heat, whether variable with temperature or not; if the material is Stycast, CAPE can be instructed to automatically select constant, but appropriately chosen properties.

e) tabulate the adiabatic wall temperature on the model surface(s), whether constant or not.

f) tabulate the melt temperature(s), whether variable on the surface(s) or not.

g) if a l.e. geometry uses the option of a single time of melt given, obtain the tunnel gas conditions, $M_{\infty,eff}$, γ, p_∞, T_{tot}, R, Pr, α_{eff} (the tunnel gas is treated as having constant: γ, gas constant, Prandtl number, etc.). $M_{\infty,eff}$ and α_{eff} are the effective values in cuts normal to the l.e.

h) decide whether or not the experimental times of melt need 'smoothing' of the inevitable scatter.

The input cards that need to be prepared for one case are described in table A-1 together with the definition of the symbols in table A-2 and Fig. A-1, A-2 and A-3.

The output printed out by CAPE for each case is described in table A-3 (the input data printout and the initial calculations), and in table A-4 (the results). This is the normal printout containing the information needed by the user.

Cases can be stacked at will.

CAPE uses no tapes and no disks. It does use a single overlay consisting of a root region and three primary levels. The standard score 3.2 overlay feature is used to prepare the overlay. Three additional main programs control each of the primary levels, the names being MPCP, MDETRAD and MOUT. CAPE is made up of the main program and 33 routines.

Minimum storage required depends upon the size of the problem, i.e., the number of elements of the problem. This is given in figure A-4. Typically, good accuracy is obtained with $N \sim 100$, for example, $N = L \times M = 10 \times 9 = 90$ for the l.e., and $N = M \times L = 12 \times 9 = 108$ for a slab-like geometry. For this problem, the storage required by CAPE is 60_8k. CAPE includes a feature that enables the user to keep the storage to a minimum for the N selected without having to change dimension cards. Appropriate dimensions are set automatically by the code but the total length in decimals has to be input into the code. The length can be read from fig. A-4 and the following two cards of the main program have to be set as follows:

```
DIMENSION S(L)

CALL SIZE (S, L)
```

where L is the length in decimals read from fig. A-4. Naturally, if one is not interested in the storage savings that go with this feature, one can just set L, once and for all, for the typical $N \sim 100$ problems. When stacking cases, L must be chosen so as to accommodate the largest case of the stack.

A guide on the computer time required on the CDC 6600 is given in fig. A-5. Naturally, the computer time depends somewhat on the problem, so that deviations from fig. A-5 are possible.

Some advice is appropriate on the choice of a few parameters:

1) for the total number of points, it seems that $N \sim 100$ gives accuracies of the order of 1 or 2% in h and is therefore considered satisfactory since it is about one order of magnitude more accurate than the final result, i.e., the experimental h.

2) In order to maintain reasonable balance in accuracy between the two directions in a slab, typically one should use numbers of elements $L \times M$ such as 9 (in depth) $\times$ 12 (on the surface), 11×19, etc. Analogously, for the l.e., typically MCAP = 10, M = 9, L = 10 (see figures A-1, A-2 and A-3).

TABLE A-1. INPUT DATA FORMAT FOR CAPE

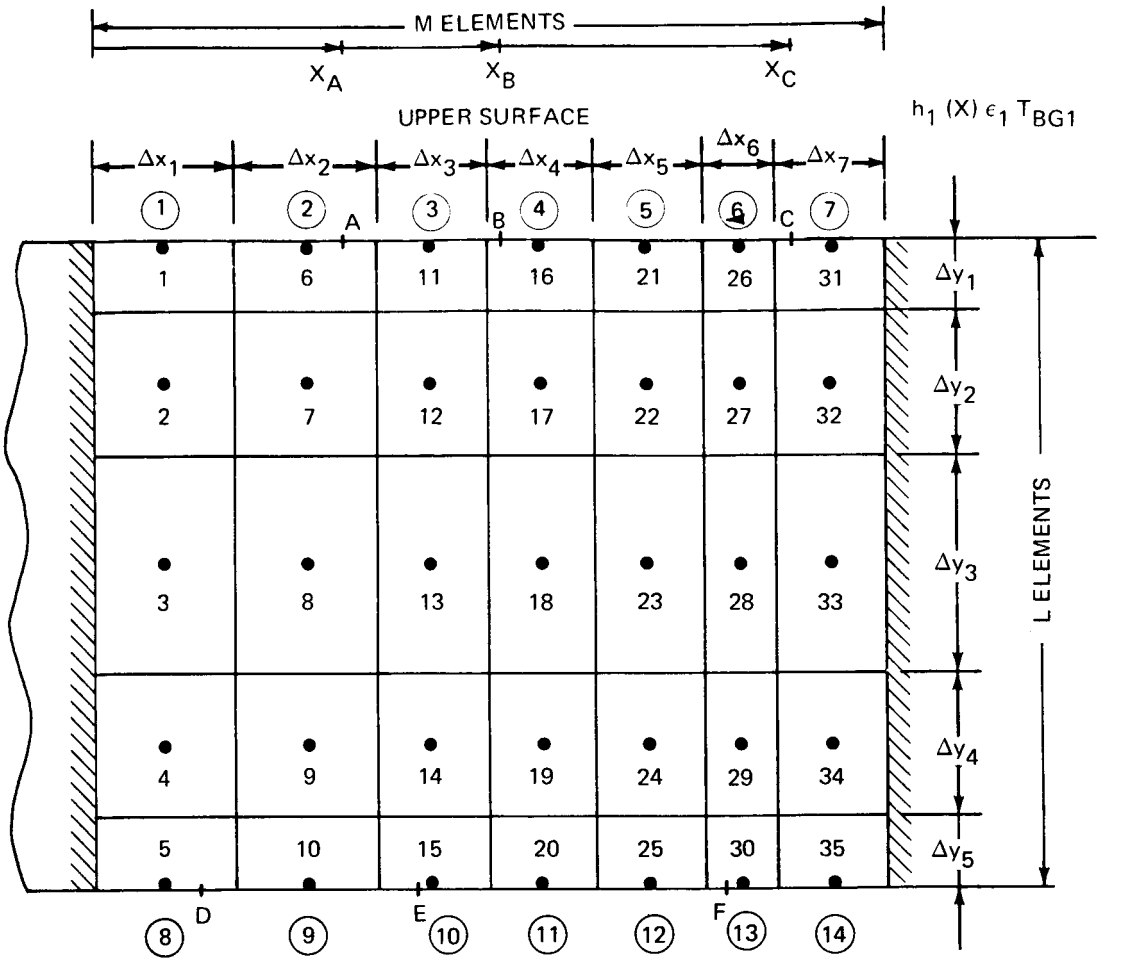
Card No.	Data	Format	Remarks
1	NSIDES, L, M, NE, LLL	(4I5)	NSIDES = 2 for l.e.; for the l.e., L must be even; For the choice of NE, L, M, LLL see suggested values
2	LABEL	(5A10)	Any alphameric identification
3	KEY IŌ	(5X,5X, I5)	Use tolerances preset by code.
4	JGEO, JMAT	(I5,5x, I5)	
5	TO, TAW, TM, EPS1, TPS2, TBG1, TBG2	(7F10.5)	a) If the adiabatic temperature or the paint melt temperature vary on the surface(s) where h is to be computed, leave TAW or TM blank b) For l.e.s, EPS1 = EPS2, TBG1 - TBG2
6,6a,6b etc.	TAW(1), . . TAW(K)	(8F10.5)	Needed only if the adiabatic wall temperature vary on the surface where the hs are to be computed (more than 1 card if K > 8).and also if JGEO is not 2. Omit cards otherwise.
7,7a,7b,etc.	TM(1), . . . TM(K)	(8F10.5)	Needed only if paint melt temperature varies on surface where h is to be computed. Omit cards otherwise.
8	NMAT, RHO, CONDAV, CPAV	(I10, 3F10.5)	a) Needed only if arbitrary properties are inputted (JMAT = -1). Omit card otherwise. b) If material properties vary with temperature, CONDAV and CPAV are left blank. c) If material properties are constant with temperature, NMAT is left blank.
9,9a,9b, etc.	TMAT(1), . . . TMAT(NMAT)	(8F10.5)	Cards needed only if properties are temperature dependent; omit otherwise.
10,10a,10b,etc.	CONDT(1), . . . CONDT(NMAT)	(8F10.5)	
11,11a,11b,etc.	CPT(1), . . . CPT(NMAT)	(8F10.5)	
12,12a,12b,etc.	DELX(1), . . . DELX(M)	(8F10.5)	Cards needed only for slab geometry (JGEO=0); omit otherwise.
13, 13a, 13b, etc;	DELY(1), . . . (DELY(L)	(8F10.5)	
14	MCAP, THETA, ALPHA	(I10,2F10.5)	Cards needed only for leading edge geometry (JGEO=1 and 2); omit otherwise.
15,15a,15b,etc.	DELX(1), . . . DELX(NWEDGE)	(8F10.5)	
16,16a,16b,etc.	DELR(1), . . . DELR(L/2)	(8F10.5)	
17,17a,17b,etc.	X(1), Y(1), . . , X(NN), Y(NN)	(8F10.5)	Needed only for arbitrary 2D geometries (JGEO= -1); omit cards otherwise
18	JSIDES, KPTS	(2I5)	KPTS is always even
19,19a,19b,etc.	X(1), . . ., X(KPTS)	(8F10.5)	
20,20a,20b,etc.	T(1), . . . T(KPTS)	(8F10.5)	
21	RR, QQ	(2F10.5	If smoothing not needed use RR = 0.0, QQ = 0.001
22	EMINF, GAM, PINF, TTOT, RGAS, PR	(6F10.5)	Card needed only with special option for l.e. (JGEO=2); omit otherwise

TABLE A-2. NOMENCLATURE

Symbol		Units
ALPHA	Effective angle of attack of l.e., positive as in Figure A-2 ('effective' means in a cut normal to the model l.e.)	degrees
CONDAV	Average thermal conductivity of model in the temperature range of interest	BTU/sec ft^{O}F
CONDT(I)	Thermal conductivity of model (function of temp.) at the temperature TMAT(I), I = 1, NMAT	BTU/sec ft^{O}F
CPAV	Average specific heat of model in the temperature range of interest	BTU/lb ^{O}F
CPT(I)	Specific heat of model at temperature TMAT(I) (function of temperature)	BTU/lb ^{O}F
DELR(I)	Spatial increments in radial direction for l.e. as indicated in Figure A-2; I = 1, L/2	feet
DELX(I)	Spatial increments in x direction of slab geometry as indicated in Figure A-1; also for the l.e. as in Figure A-2; I = 1, M	feet
DELY(I)	Spatial increments in y direction of slab geometry as indicated in Figure A-1; I = 1, L	feet
EMINF	Effective free stream Mach number	
EPS1	Emissivity of "upper" surface of slab-like geometry or l.e. surface	
EPS2	Emissivity of "lower" surface of slab-like geometry or of the l.e. surface	
GAM	Constant ratio of specific heats of tunnel gas	
JGEO	Geometry index; selects geometry and, for the l.e., the type of reduction problem, as follows: = -1, arbitrary four-sided geometry; = 0, slab; = 1, l.e. with all melt times given; = 2, l.e. with special option with minimum melt time given	
JMAT	Model material index; selects model material as follows: = -1 arbitrary; = 0, Stycast with constant properties, automatically averaged over appropriate temperature range; = 2, properties used in NASA submitted problem	
JSIDES	Same as NSIDES	
K	Number of surface elements on which the hs are to be calculated; = NSIDES x M for four sided geometries; 2 x M for l.e.	
KEY10	Index that selects tolerances: KEY 10 = 0, tolerances preset in code	
KPTS	Number of positions on the model surface(s) at which the times of melt are given ($\leqslant$ K). Same position number on upper and lower surfaces	
L	Number of elements through the material (see figures A-1, A-2, A-3)	
LLL	Number of time steps for the time integration of the temperature correction due to temperature dependent properties or radiation; if the material properties are constant and radiation is neglected, use LLL = 5	
M	(for slab-like geometry) number of elements along each surface on which the hs must be calculated (if 2 surfaces, must be same for both); (for l.e.) number of elements on half l.e. surface — see Figure A-2	
MCAP	Number of elements into which nose of l.e. is subdivided (see Figure A-2); must be even	
NE	Number of dominant eigenvalues (substancially less than N)	

TABLE A-2. NOMENCLATURE (CONTINUED)

Symbol		Units
NMAT	Number of points in thermal properties tables	
NN	Number of points to specify corners of elements (arbitrary geometry – see Figure A-3)	
NSIDES	Number of sides (1 or 2) in the slab that are heated (a side not heated is taken as adiabatic)	
NWEDGE	Number of elements on the wedge portion of I.e. (see Figure A-2)	
PINF	Free stream pressure of tunnel gas	psf
PR	(Constant) Prandtl number of tunnel gas	
QQ	Spline fitting parameter	
RGAS	Gas constant of tunnel gas	ft-lb/slug ^{o}R
RHO	Density of model material (constant with temperature)	lb/ft^3
RR	Spline fitting parameter (= 0 no smoothing; = 1 straight line)	
T, T(I)	Time(s) of melt; I = 1, KPTS	seconds
TAW	Adiabatic wall temperature, constant on the surface(s) where hs are to be calculated	^{o}R
TAW(I)	Adiabatic wall temperature, variable or just different constants on upper and lower surface; I = 1, K	^{o}R
TBG1	Background temperature for radiation from 'upper' surface of slab-like geometry or from the I.e. surface	^{o}R
TBG2	Background temperature of radiation from 'lower' surface of slab-like geometry or from the I.e. surface	^{o}R
THETA	Wedge half angle of I.e. geometry as indicated in Figure A-2	
TM	Melting temperature of the paint, if constant on the surface(s)	^{o}R
TM(I)	Melting temperature of the paint, if variable on the surface(s) or even constant and different between two surfaces; I = 1, K	^{o}R
TMAT(I)	Temperatures in thermal properties table, i.e. temperatures at which a value of CONDT(I) and CPT(I) is given; I = 1, NMAT	^{o}R
TTOT	Total temperature of tunnel gas	^{o}R
TO	Initial temperature of model	^{o}R
X (in cards 19)	arc lengths at which the times of melt are given (see Figures A-1, A-2, A-3)	feet
X (in cards 17)	X coordinate of corners of elements (arbitrary geometry, see Figure A-3)	feet
Y	Y coordinate of corners of elements (arbitrary geometry, see Figure A-3)	feet



EXAMPLE SHOWN:

M = 7, L = 5, N = 35

T ADIAB WALL, MELT TEMPER } ORDERED (1) → (14)

TIMES OF MELT, THEIR LOCATIONS $x_A x_B$ } ORDERED (A) → (F), KPTS = 6

FIGURE A-1. INDEXES AND INPUTS FOR SLABS

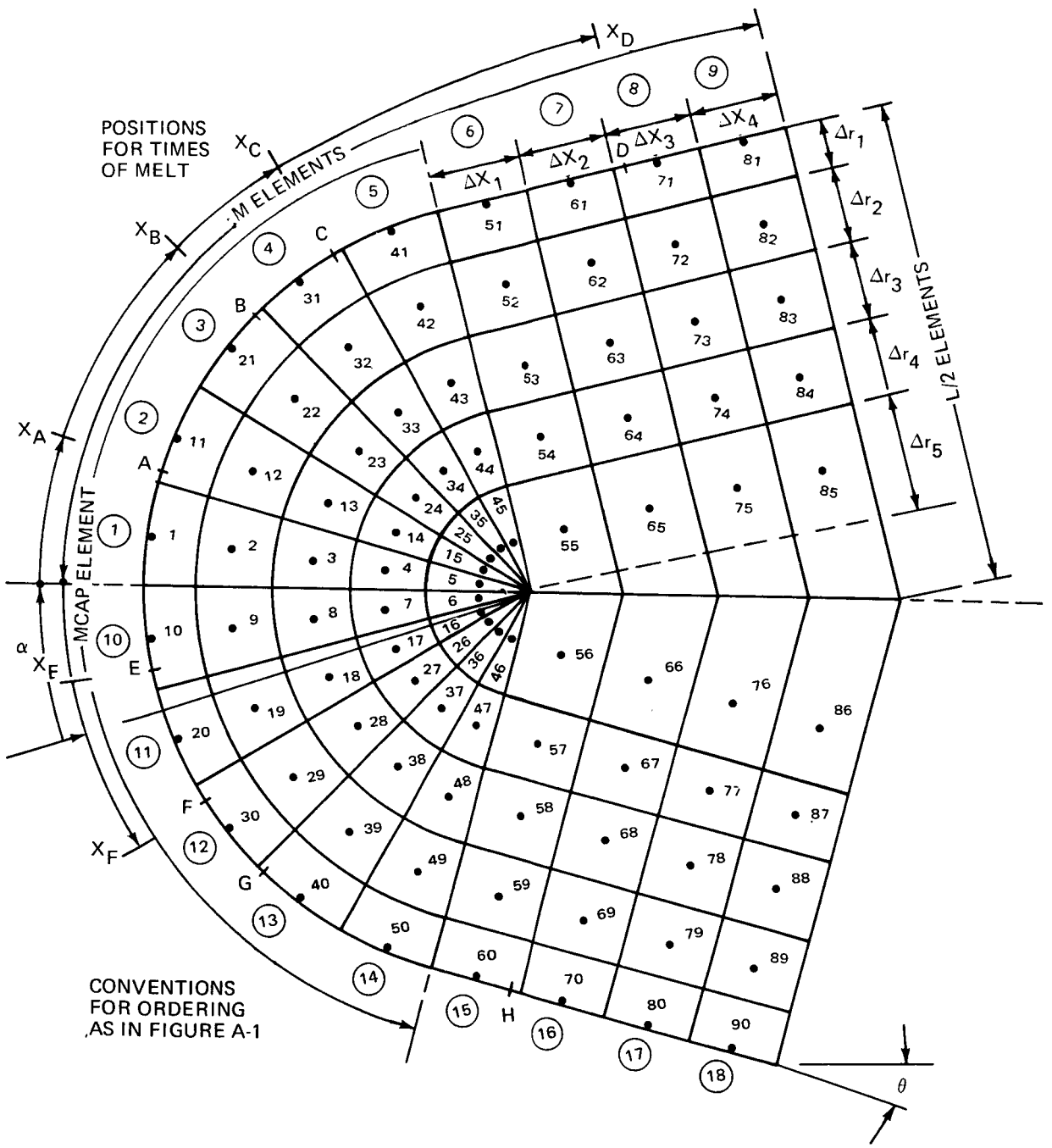


FIGURE A-2. INDEXES AND INPUTS FOR LEADING EDGES

EXAMPLE SHOWN:

M = 5, L = 5, N = 35
NO. OF CORNERS OF ELEM = (M + 1) (L + 1) = 36

COORDINATES OF CORNERS OF ELEMENTS (P_1, P_2, . . .), ORDERED AS SHOWN, DEFINE GEOMETRY

T ADIAB WALL, MELT TEMP } ORDERED (1) → (10)

TIMES OF MELT INPUTTED AS FUNCTIONS OF DISTANCE X ALONG SURFACE

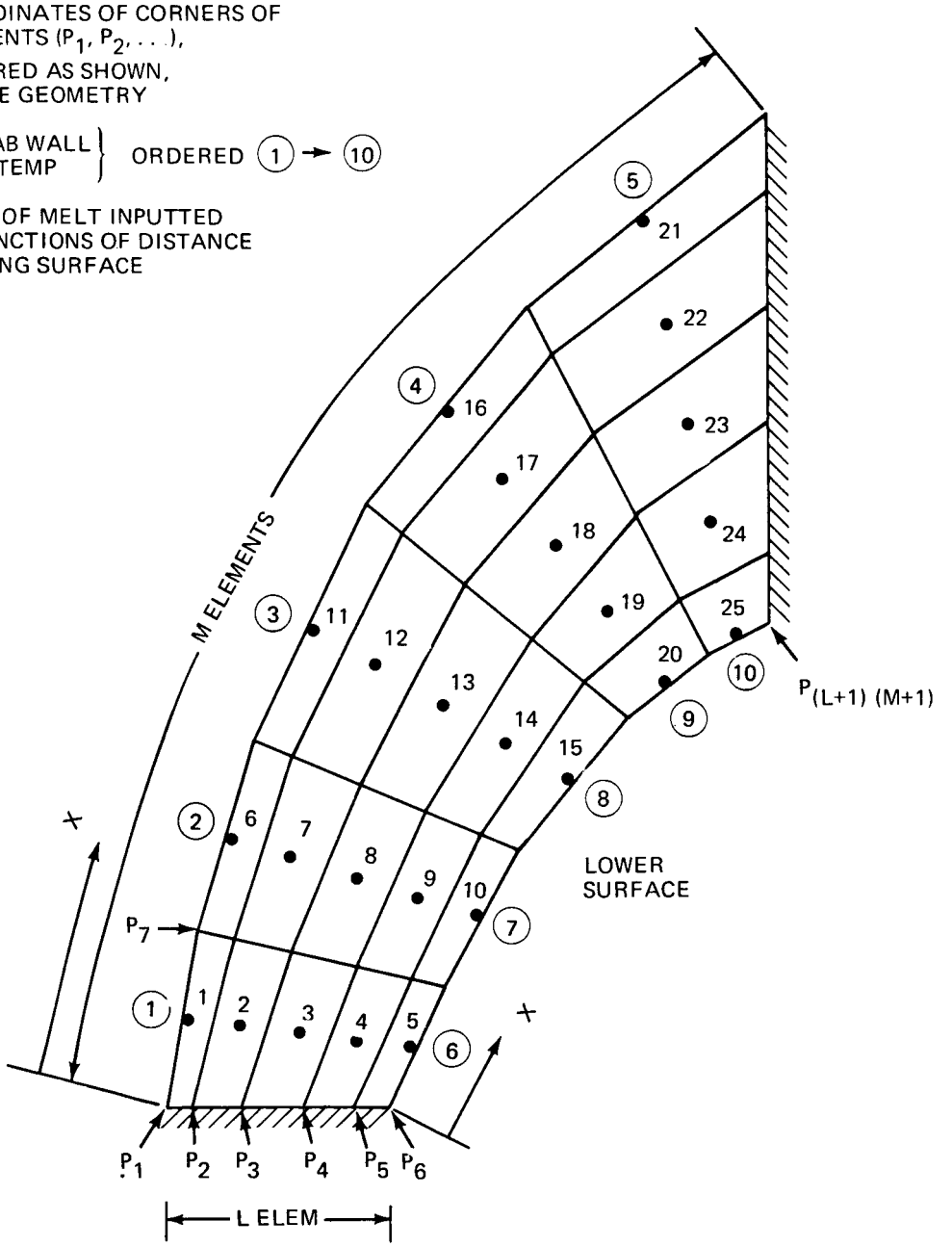


FIGURE A-3. INDEXES AND INPUTS FOR ARBITRARY GEOMETRIES

TABLE A-3. OUTPUT FORMAT OF INPUT DATA AND PRELIMINARY CALCULATIONS

3 .542 THE HEAT TRANSFER PROBLEM ON A RECTANGLE**ELEMENTS NUMBERED COLUMNWISE--IO=2,DIR=INV OPTS PART R

INPUT DATA

SIZE 2 9 12 1 5

NSIDES L M NE LLL

2 9 12

NUMBER OF EIGENVALUES NE

N

24 SURFACE ELEMENTS-- 9 ROWS BY 12 COLUMNS GIVES 1?8 ELEMENTS 14 DOMINANT MODES...RESOLUTIES 15689 WORDS OF MEMORY

1	325	433	541	649	674	698	722	746	854	878	962	12566	12674	12782	12890
12998	13574	13682	13737	13761	13785	13809	13833	13857	13881	13905	13929	13953	13977	14601	14025
14049	14073	14097	14121	14145	14169	14277	14385	14493	14601	14709	14718	15693			

ECONONOMIZE...REDUCE DIMENSION OF S AND VALUE OF MWORDS FROM 20000 TOWARDS 15693

QUANTITY L:
VALUE ACTUALLY NEEDED
VALUE USED

LABEL INFORMATION $T_{M,}\ ^{o}R$ E_1 E_2 $T_{BG1,}\ ^{o}R$ $T_{BG2,}\ ^{o}R$

0 2

540.0000014**0.00000 0.00000 0.00000 0.00000 540.00000 540.00000

$T_{init.}\ ^{o}R$ $T_{aw,}\ ^{o}R$ IF CONSTANT

COND = .00012 RHO = 14?.00000 CP = .22000

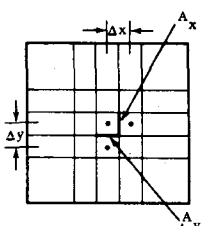
CONSTANT OR AVERAGE MATERIAL PROPERTIES:
CONDUCTIVITY BTU/SEC-FT-^{o}R
DENSITY LB/FT^3
SPECIFIC HEAT BTU/LB-^{o}R

N	SMI	CAx	CAY
1	2?1.45574	.00001	.00077
2	142.45072	.00002	.00064
3	120.39283	.00003	.00055
4	120.39283	.00003	.00055
5	120.39283	.00003	.00055
6	120.39283	.00003	.00055
7	120.39283	.00003	.00064
8	142.45072	.00002	.00077
9	2?1.45574	.00001	0.00000
10	2?1.45574	.00001	.00077
11	142.45072	.00002	.00064
12	120.39283	.00003	.00055
13	120.39283	.00003	.00055
14	120.39283	.00003	.00055
15	120.39283	.00003	.00055
16	120.39283	.00003	.00064
17	142.45072	.00002	.00077
18	2?1.45574	.00001	0.00000
19	2?1.45574	.00001	.00077
20	142.45072	.00002	.00064
21	120.39283	.00004	.00055
22	120.39283	.00004	.00055
23	120.39283	.00004	.00055
24	120.39283	.00004	.00055
25	120.39283	.00004	.00064
26	142.45072	.00002	.00077
27	2?1.45574	.00001	0.00000
28	284.90144	.00002	.00038
29	2?1.45574	.00004	.00032
30	170.26118	.00005	.00027
	170.26118	.00005	.00027
	170.26118	.00005	.00027
	170.26118	.00005	.00027
		.00005	

FOR EACH ELEMENT OF GEOMETRY, NUMBERED AS IN FIG A-1, A-2, A-3

$$CAX(I) = \frac{k\,A_x}{\Delta X}$$

$$CAT(I) = \frac{k\,A_y}{\Delta Y}$$



CAX=0 last column
CAY=0 last row

	284.90144	0.00000	.00019
102	284.90144	0.00000	.00016
103	240.78567	0.00000	.00014
104	240.78567	0.00000	.00014
105	240.78567	0.00000	.00014
106	240.78567	0.00000	.00016
107	284.90144	0.00000	.00019
108	402.91148	0.00000	0.00000

SMOOTHING OF EXPERIMENTAL TIMES-OF-MELT

R = 0.00000 ← SMOOTHING FACTOR FOR UPPER SURFACE

1st, 2nd, 3rd DERIVATIVES OF SMOOTHED TIMES OF MELT

X	G	F	FP	FPP	FPPP
.001600	3.600000	3.600000	0.000000	0.000000	0.000000
.004800	3.600000	3.600000	0.000000	0.000000	0.000000
.008000	3.600000	3.600000	0.000000	0.000000	0.000000
.010400	3.600000	3.600000	0.000000	0.000000	0.000000
.012800	3.600000	3.600000	0.000000	0.000000	0.000000
.013600	3.600000	3.600000	0.000000	0.000000	0.000000
.015200	3.600000	3.600000	0.000000	0.000000	0.000000
.016400	3.600000	3.600000	0.000000	0.000000	0.000000
.017200	3.600000	3.600000	0.000000	0.000000	0.000000
.018000	3.600000	3.600000	0.000000	0.000000	0.000000
.018800	3.600000	3.600000	0.000000	0.000000	0.000000
.019600	3.600000	3.600000	0.000000	0.000000	0.000000

POSITION AT WHICH TIMES OF MELT ARE GIVEN

GIVEN TIMES OF MELT

TIMES OF MELT USED

UPPER SURFACE

R = 0.00000 ← SMOOTHING FACTOR FOR LOWER SURFACE

Y	G	F	FP	FPP	FPPP
.001600	3.600000	3.600000	0.000000	0.000000	0.000000
.004800	3.600000	3.600000	0.000000	0.000000	0.000000
.008000	3.600000	3.600000	0.000000	0.000000	0.000000
.010400	3.600000	3.600000	0.000000	0.000000	0.000000
.012800	3.600000	3.600000	0.000000	0.000000	0.000000
.013600	3.600000	3.600000	0.000000	0.000000	0.000000
.015200	3.600000	3.600000	0.000000	0.000000	0.000000
.016400	3.600000	3.600000	0.000000	0.000000	0.000000
.017200	3.600000	3.600000	0.000000	0.000000	0.000000
.018000	3.600000	3.600000	0.000000	0.000000	0.000000
.018800	3.600000	3.600000	0.000000	0.000000	0.000000
.019600	3.600000	3.600000	0.000000	0.000000	0.000000

LOWER SURFACE

ALL SYMBOLS AS IN FIG. A-1, A-2, A-3

TABLE A-4. FORMAT OF OUTPUT OF FINAL RESULTS (SHEET 1 OF 2)
a) FOR A SLAB

INTERPRETATION OF PHASE-CHANGE PAINT DATA

GEOMETRY: SLAB HEATED ON TWO SIDES
LENGTH = .020 FEET THICKNESS = .005 FEET
INITIAL TEMPERATURE = 540.00 DEG R
EMISSIVITY(1) = 0.000 ← UPPER SURFACE
EMISSIVITY(2) = 0.000
BACKGROUND TEMPERATURE(1) = 540.00 DEG R
BACKGROUND TEMPERATURE(2) = 540.00 DEG R

ELEM	X FEET	TIME(PC) SEC	TEMP(PC) DEG R	TEMP(AW) DEG R	H(SIS) B/S*F2*DF	F(CORR) RATIO $\frac{h}{h_{SIS}}$	H HEAT TRANSFER COEFFICIENT B/S*F2*DF
			UPPER SURFACE				
1	.0016	3.6000	719.5000	1400.0000	.0071258	.69871	.0049789
2	.0048	3.6000	729.0000	1400.0000	.0075861	.65056	.0049352
3	.0080	3.6000	767.0000	1400.0000	.0095437	.59919	.0057185
4	.0104	3.6000	840.0000	1400.0000	.0139010	.60592	.0084230
5	.0120	3.6000	914.0000	1400.0000	.0194120	.75985	.0147503
6	.0136	3.6000	992.0000	1400.0000	.0270204	.86085	.0232606
7	.0152	3.6000	1058.5000	1400.0000	.0358432	.89689	.0321476
8	.0164	3.6000	1097.5000	1400.0000	.0426141	.92960	.0396140
9	.0172	3.6000	1117.5000	1400.0000	.0467515	.92788	.0433799
10	.0180	3.6000	1132.5000	1400.0000	.0502244	.93225	.0468218
11	.0188	3.6000	1142.0000	1400.0000	.0526212	.92816	.0488411
12	.0196	3.6000	1146.5000	1400.0000	.0538116	.92352	.0496962
			LOWER SURFACE				
1	.0016	3.6000	692.0000	1400.0000	.0058467	.51267	.0029974
2	.0048	3.6000	700.0000	1400.0000	.0062084	.47743	.0029641
3	.0080	3.6000	726.0000	1400.0000	.0074396	.42179	.0031380
4	.0104	3.6000	762.0000	1400.0000	.0092757	.31407	.0029132
5	.0120	3.6000	792.0000	1400.0000	.0109366	.26358	.0028827
6	.0136	3.6000	825.0000	1400.0000	.0129326	.23035	.0029790
7	.0152	3.6000	856.0000	1400.0000	.0149884	.20489	.0030710
8	.0164	3.6000	874.5000	1400.0000	.0163084	.19193	.0031300
9	.0172	3.6000	884.5000	1400.0000	.0170544	.18109	.0030884
10	.0180	3.6000	892.0000	1400.0000	.0176356	.17011	.0030001
11	.0188	3.6000	897.0000	1400.0000	.0180279	.15862	.0028596
12	.0196	3.6000	900.0000	1400.0000	.0182664	.16677	.0030462

NOTE: X COORDINATE,
ORDERING OF ELEMENTS,
SYMBOLS AS IN FIG. A-1.

TABLE A-4. FORMAT OF OUTPUT OF FINAL RESULTS (SHEET 2 OF 2)
b) FOR A I.e. SPECIAL PROBLEM

INTERPRETATION OF PHASE-CHANGE PAINT DATA

GEOMETRY: CYLINDRICAL LEADING EDGE FOLLOWED BY WEDGE
RADIUS R = .004 FEET S = .004 FEET AFTERBODY LENGTH

THETA = 15.000 DEGREES ALPHA = 15.000

LEES DISTRIBUTION USED FOR HEAT TRANSFER COEFFICIENT
MACH NUMBER = 8.000 GAMMA = 1.400
INITIAL TEMPERATURE = 540.00 DEG R
EMISSIVITY(1) = 0.000
EMISSIVITY(2) = 0.000
BACKGROUND TEMPERATURE(1) = 540.00 DEG R
BACKGROUND TEMPERATURE(2) = 540.00 DEG R

{ THESE LINES APPEAR ONLY FOR SPECIAL PROBLEM

ELEM	X FEET	TIME(PC) SEC	TEMP(PC) DEG R	TEMP(AW) DEG R	H(SIS) B/S*F2*DF	RATIO $\frac{h}{h_{SIS}}$ F(CORR)	HEAT TRANSFER COEFFICIENT H B/S*F2*DF
			UPPER SURFACE				
1	.0005	4.5501	660.0000	1284.6706	.0061361	.66977	.0041098
2	.0016	4.7502	660.0000	1265.1762	.0061918	.48776	.0030201
3	.0026	5.2005	660.0000	1232.4878	.0062427	.25981	.0016219
4	.0037	5.8007	660.0000	1205.6169	.0061907	.12706	.0007866
5	.0047	6.4510	660.0000	1180.4754	.0061425	.05206	.0003198
6	.0062	7.4963	660.0000	1168.7040	.0058256	.03309	.0001928
7	.0072	7.8821	660.0000	1168.7040	.0056817	.03359	.0001908
8	.0082	8.0959	660.0000	1168.7040	.0056064	.03370	.0001889
9	.0092	8.2651	660.0000	1168.7040	.0055489	.03371	.0001871
			LOWER SURFACE				
1	.0005	4.5501	660.0000	1296.1852	.0060290	.80194	.0048349
2	.0016	4.7502	660.0000	1300.0000	.0058669	.82409	.0048349
3	.0026	5.2005	660.0000	1296.1852	.0056401	.85724	.0048349
4	.0037	5.8007	660.0000	1284.6706	.0054368	.75606	.0041098
5	.0047	6.4510	660.0000	1265.1762	.0053160	.56822	.0030201
6	.0062	7.4963	660.0000	1246.8929	.0050779	.39646	.0020132
7	.0072	7.8821	660.0000	1246.8929	.0049545	.36683	.0018174
8	.0082	8.0959	660.0000	1246.8929	.0048899	.34142	.0016695
9	.0092	8.2651	660.0000	1246.8929	.0048406	.32077	.0015527

NOTE: ARC LENGTH X,
ORDERING OF ELEMENTS,
SYMBOLS AS IN FIG. A-2

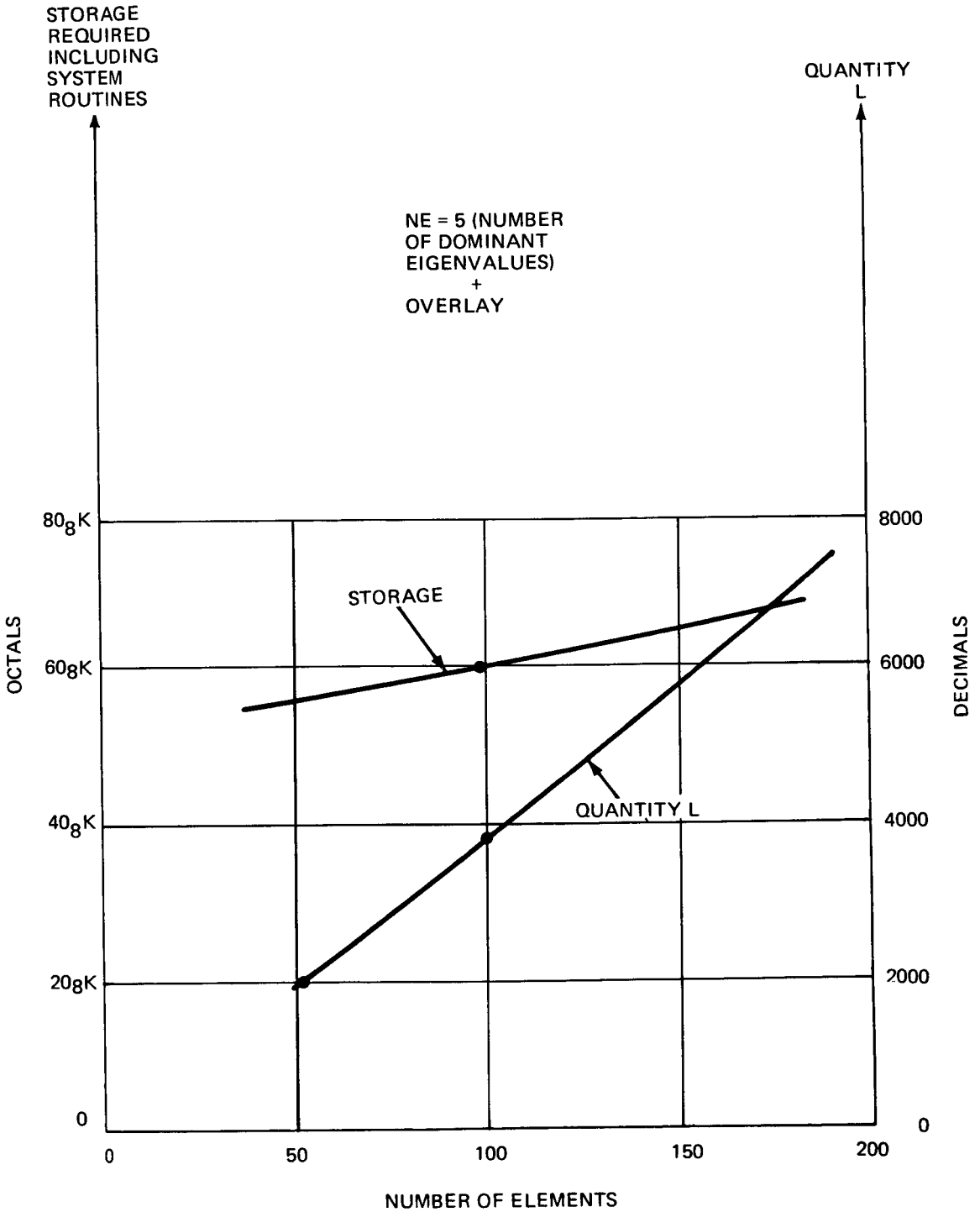


FIGURE A-4. STORAGE REQUIRED BY CAPE

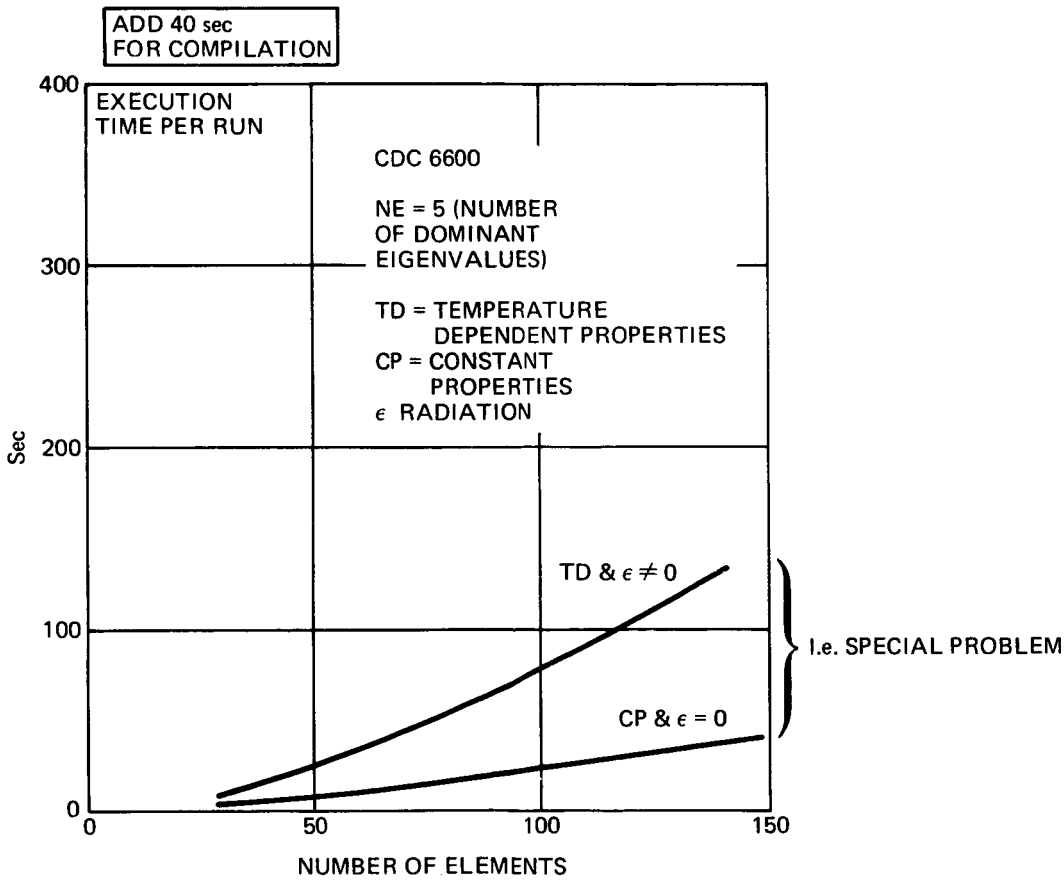


FIGURE A-5. COMPUTER TIME REQUIRED BY CAPE

3) for the number of dominant eigenvalues NE, use a standard NE = 5. Absolute safety is afforded by NE = 10, but the CPU time savings between 5 and 10 are some 50% and are therefore attractive.

4) Since the material properties variations with temperature do not have but a tiny effect, great sophistication in the table of the properties is not warranted. Use, for example, NMAT = 5.

5) LLL = 5 has been found adequate.

6) The tolerances set automatically by the code if KEYIO = 0, are typical, perhaps on the tight side, namely at all surface points, simultaneously the h's differ by less than 1% between iterations and the temperature errors, $T_{melt} - T_{computed}$, is less than 1 °R.

Other points worth of attention are the following:

7) Each element in the geometry must be attributed the ordering indicated in figures A-1, A-2 and A-3; it is in this order that such arrays as DELX, DELY, DELR, TAW, TM, X, Y are to be structured. For the slab, one first inputs the entire set of (say) TAW etc. for the upper surface, followed by the lower surface. For the l.e., similarly first the 'upper' side of fig. A-2, sequentially followed by the lower side. Similarly with the X(KPTS) and T (KPTS) arrays. However, for DELX and DELR, both in the slab and in the l.e., because of symmetry, just one set -- say for the upper surface -- is to be inputted. When inputting the lower surface of a l.e., start with the point on the axis of symmetry (which therefore is considered a double point).

8) To do a one dimensional finite-slab geometry or a quasi-one dimensional arbitrary geometry take M = 2 respectively in the slab and the arbitrary geometry; in other words, 'double' the given input data in the M direction.

9) To do a semi-infinite slab or a semi-infinite arbitrary four sides geometry, take the 'lower' surface in fig. A-1 and A-3 far enough into the material so that there the initial temperature is substantially unchanged up to the maximum melt time; then prescribe NSIDE = 1 and the code will treat the 'lower' surface as adiabatic.

10) In l.e. cases where the input data are symmetric and $\alpha = 0$, the code works out the entire problem disregarding the symmetry; there is no artifice to take advantage of the symmetry. On a slab or a four-sided geometry, to take advantage of symmetry of inputs, either in the lateral or in the depth direction, just input one-half of the problem.

11) Appendix B gives the input cards and printouts for two check cases.

APPENDIX B

PROGRAMMER-ORIENTED DOCUMENTATION OF THE CODE

In this appendix, the details of CAPE organization and structure are assembled. A simplified overall flow diagram of CAPE is given in fig. B-1.

CAPE is organized in a main program and 33 subroutines. The list of the subroutines is given in table B-1 together with the function of each subroutine and the subroutine from which it is called.

The overall logic of the program can now be organized in terms of the subroutines as shown in fig. B-2.

The flow charts for each subroutine are given in sheets B1 - B29. (four routines ERFCC, COTAN, ARCOS, and OUT are omitted since they consist of obvious few lines of coding). The object deck of each subroutine includes comment cards for each of the operations described in the flow chart so that correspondence between the instructions and the flow chart can be done at once.

The set of input cards for two check cases are presented in sheet B-30 while the respective outputs are given in table A-4.

The standard output has been described in Appendix A. For detailed output that maps the iterations and eigenvalues and also times the main steps of the calculation, the flags LTE and MON must be set equal to six in a statement card at the very beginning in subroutine PCP SIZE. All the detailed output refers to the iterations performed in the subroutine DETRAD and the subroutines that are called from it.

Self-explanatory diagnostic messages are printed out for the two main failure modes (that are intrinsic to the method, rather than to errors in input preparation and integrity of the code). One mode is a failure in the inversion if the matrix G_{ij} is singular. The other mode is the failure of the Jennings algorithm to converge, within the maximum number of iterations, to the requested number of dominant eigenvalues and eigenvectors. The latter mode of failure has never been encountered in normal runs. Presumably the corrective action would be to increase the allowed number of iterations, fixed by the index NIJ that is set in statement at the outset of subroutine SIZE. The former failure mode has been encountered once while exploring the extreme values for NE, the number of dominant eigenvalues and eigenvectors. It was found that one slab-like case generated a singular G_{ij} matrix for the extreme value of NE equal to 2. The corrective action is, of course, to use a reasonable number of NE, from 5 to 10.

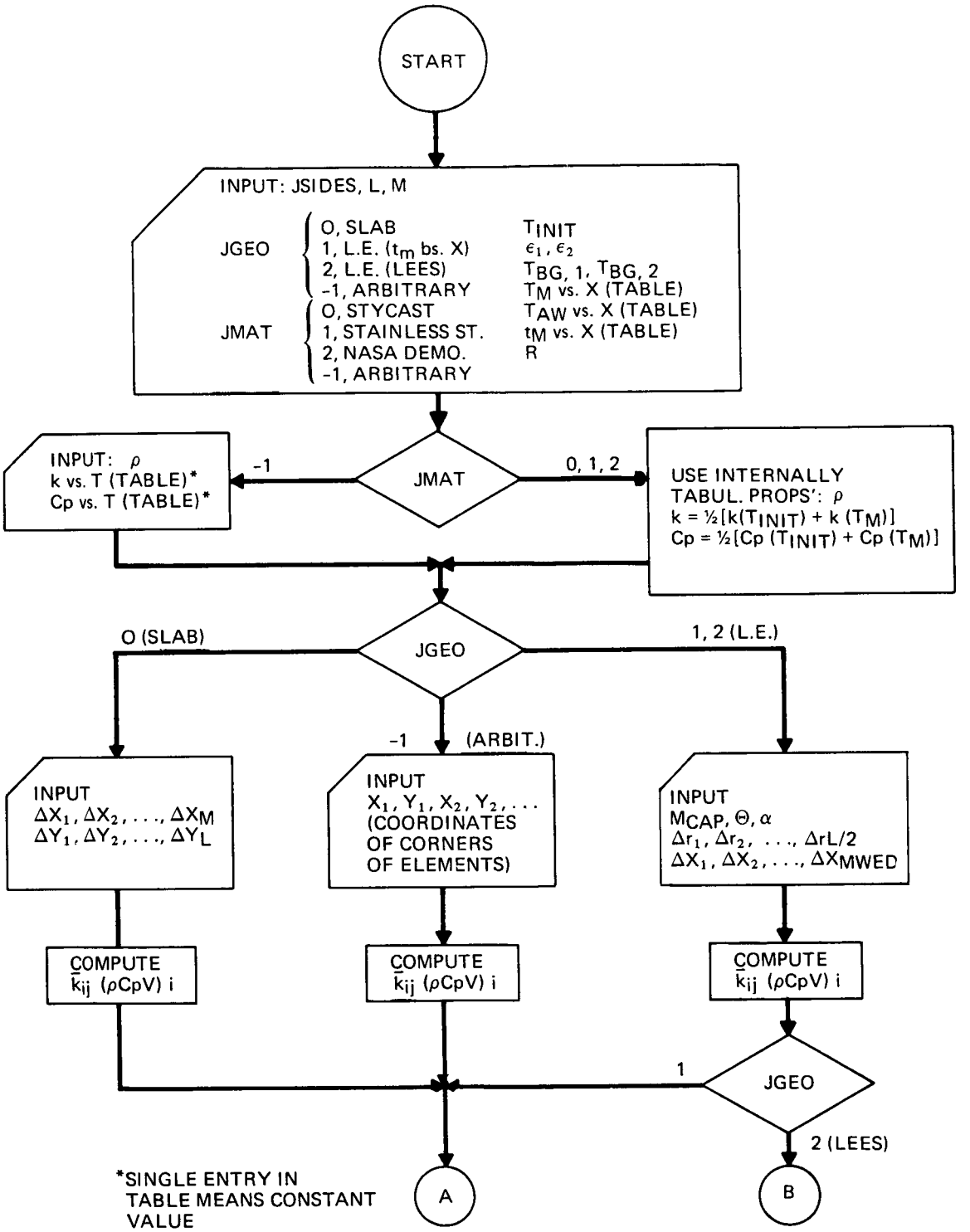


FIGURE B-1. OVERALL FLOW OF LOGIC IN CAPE (SHEET 1 OF 3)

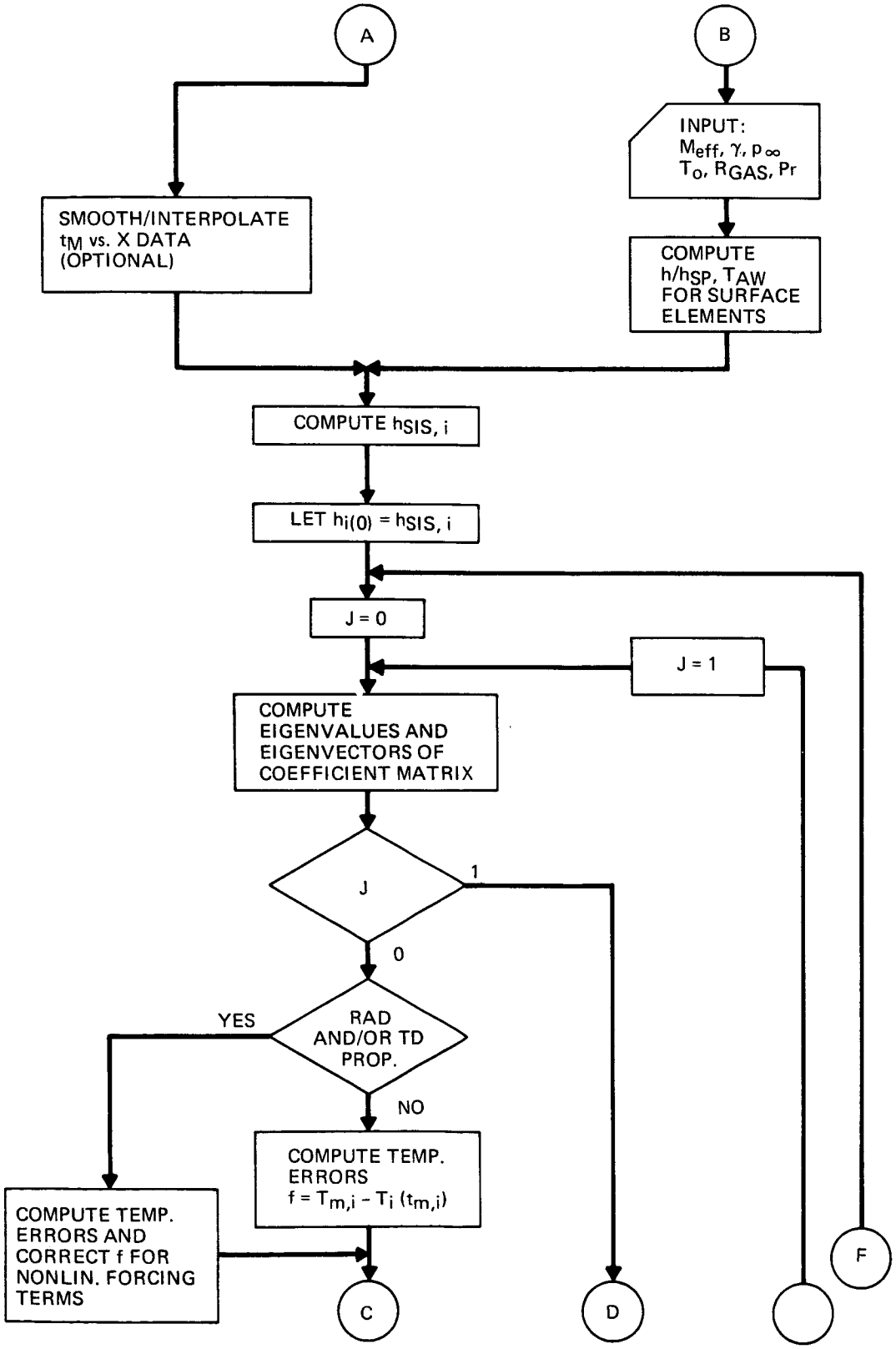


FIGURE B-1. OVERALL FLOW OF LOGIC IN CAPE (SHEET 2 OF 3)

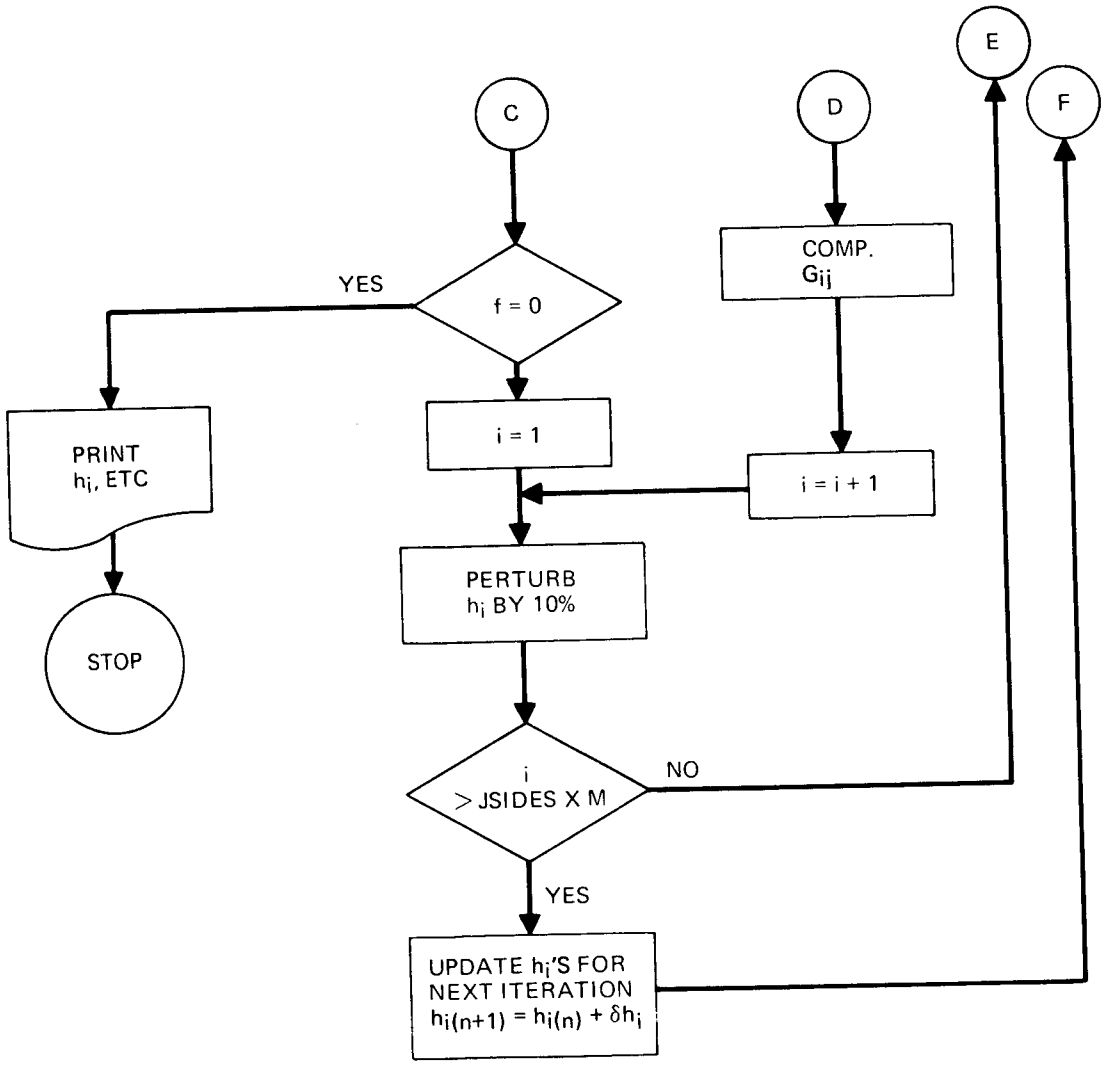
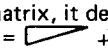


FIGURE B-1. OVERALL FLOW OF LOGIC IN CAPE (SHEET 3 OF 3)

TABLE B-1. SUBROUTINES USED IN CAPE

Name	Called by	Main Purpose
SIZE	MAIN PROG	Computes storage locations needed. Compares to number requested.
PCP	SIZE	Reads main input, output initial data. Performs preliminary calculations. Controls geometry, material properties and other options.
SLAB	PCP	Computes volumes and conduction shape factors for slab geometry.
LEAD	PCP	Computes volumes and conduction shape factors for leading edge geometries.
ARBIT	PCP	Orients individual elements of the arbitrary geometry for computations in ELEM.
ELEM	ARBIT	Computes volumes and conduction shape factors between two arbitrary quadrilateral elements of the arbitrary geometry.
SMOOTH	PCP	Sets up calculations for DATFT and normalizes the range for the smoothing fit.
SMOFIT	PCP	Finds value from table by smoothing-spline interpolation.
DATFT	SMOOTH	Determines Taylor series coefficients for smoothing spline fit.
LINFIT	PCP, DETRAD	Finds value from a table by linear interpolation.
ATTACK	PCP	Finds element number closest to stagnation point and renumbers elements as required by LEES.
LEES	ATTACK	Computes ratios h/h_{sp} and T_{AW} variation around leading edge for the special l.e. problem.
DETRAD	SIZE	Main routines for the inverse problem calculation. Calls eigenvalue and matrix routines. Perturbs h to generate influence coefficients. Performs iteration on h. Returns the h values.
IJEN	DETRAD	Obtains dominant eigenvectors and eigenvalues of a given matrix (using Jennings method, i.e. by simultaneous vector iteration).
EIGVC	DETRAD	Prepares approximate guesses for the eigenvectors to start the Jennings algorithm iteration for the zeroth h iteration.
BFACS	DETRAD, IJEN	Factorizes a banded positive-definite matrix.
BSOLS	DETRAD, IJEN	Using the factors of a given banded positive-definite matrix A as generated by BFACS solves for X the system AX = Y.
ORNML	IJEN	Carries out the standard Gram-Schmidt orthonormalization of a group of vectors.
HETRAC	DETRAD	Sets up coefficient matrix (of conductances) in compact form.
RVORDR	IJEN	Re-orders estimated eigenvalues according to magnitude.
AORDER	IJEN, RVORDER	Sets up permutation indices needed for ordering the eigenvalues.
DISPLA	Various	Prints information, mainly debug special output, in array form.
LUSOL	RIMEQF	Given the results of RDET, substitutes the solution for right hand side.
PART	Various	Prints debug output information and intermediate timing of calculation.

TABLE B-1. SUBROUTINES USED IN CAPE (CONTINUED)

Name	Called by	Main Purpose
RDET	RIMEQF	Given a matrix, it decomposes it into two triangular arrays, a lower and an upper A =  + 
RIMEQF	DETRAD	Solves system of simultaneous linear equations of type AX = Y for X.
SCAPRO	Various	Computes scalar products of two vectors and adds a value to the result.
SWITCH	DISPLA	Converts columns of a matrix to rows or visa versa.
SCAPR2	ORNML	Computes scalar product of two vectors (without adding a given number to the result)
ERFCC	PCP	Computes complementary error function of a given argument.
COTAN	ELEM	Computes the cotangent of an angle.
ARCOS	ELEM	Computes the arc cos of an angle.
OUT	SIZE	Prints final output of problem.

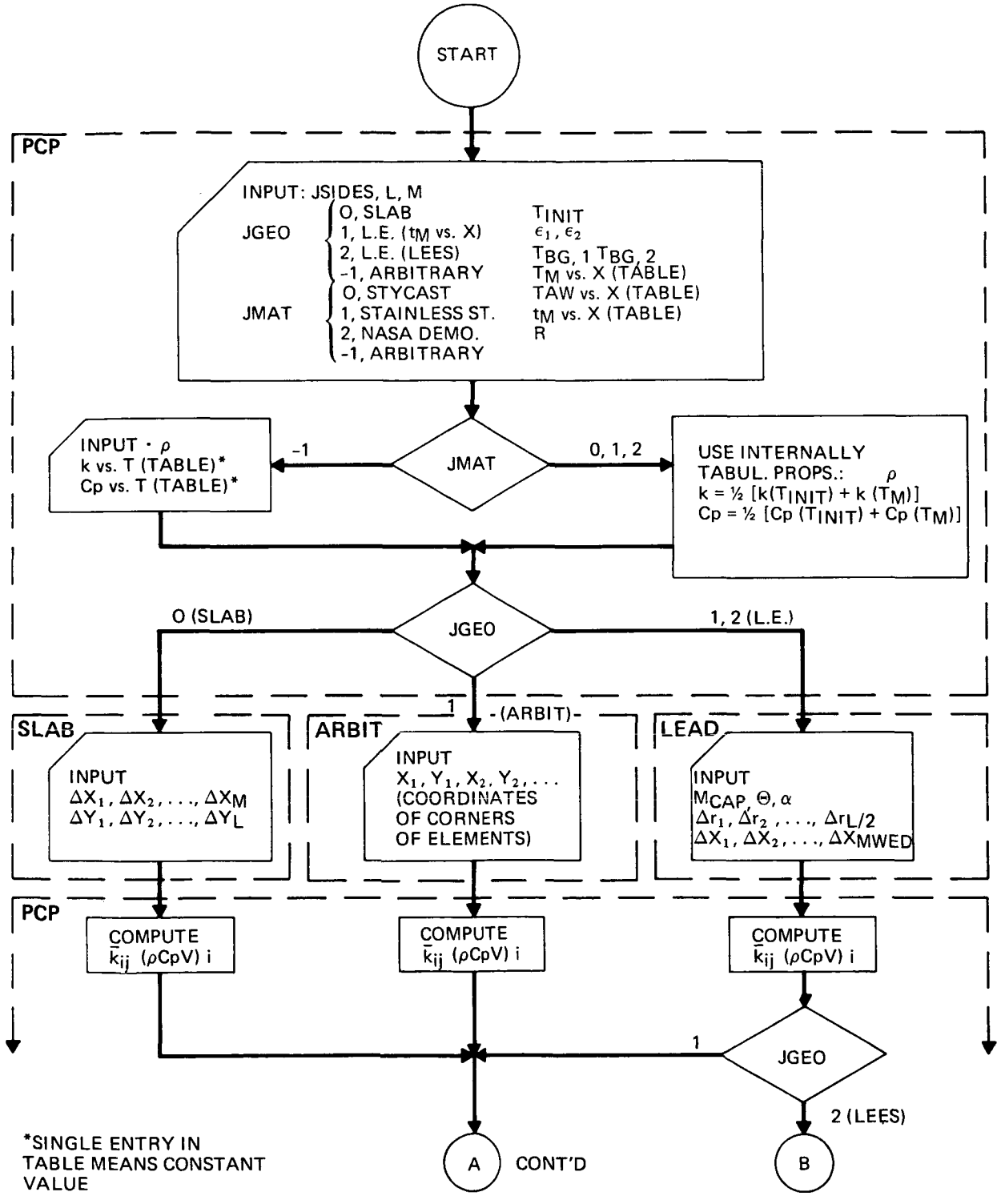


FIG. B-2 OVERALL FLOW OF LOGIC ON CAPE IDENTIFIED WITH MOST IMPORTANT ROUTINES. (SHEET 1 OF 3)

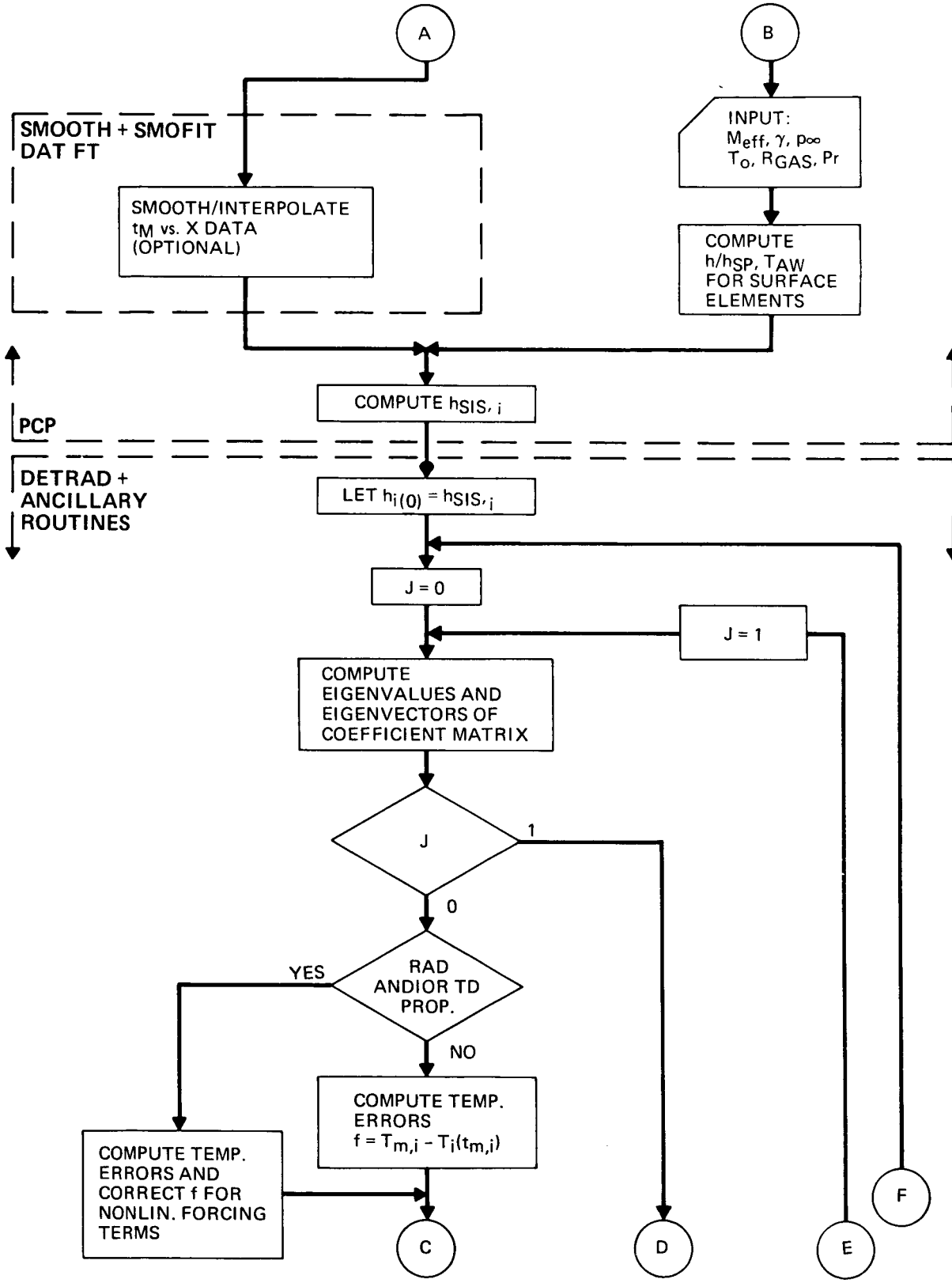


FIGURE B-2. OVERALL FLOW OF LOGIC ON CAPE IDENTIFIED WITH MOST IMPORTANT ROUTINES. (SHEET 2 OF 3)

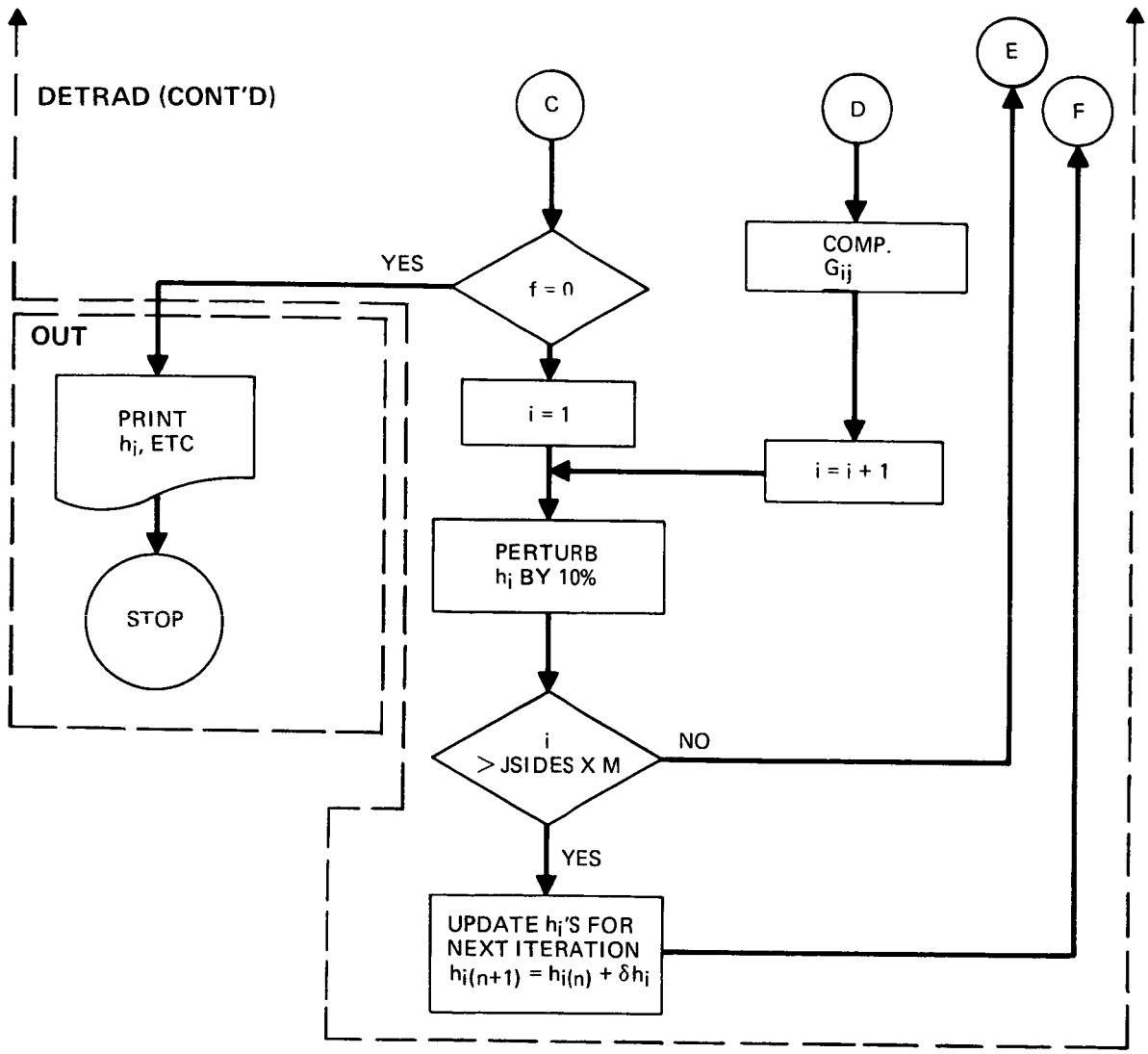
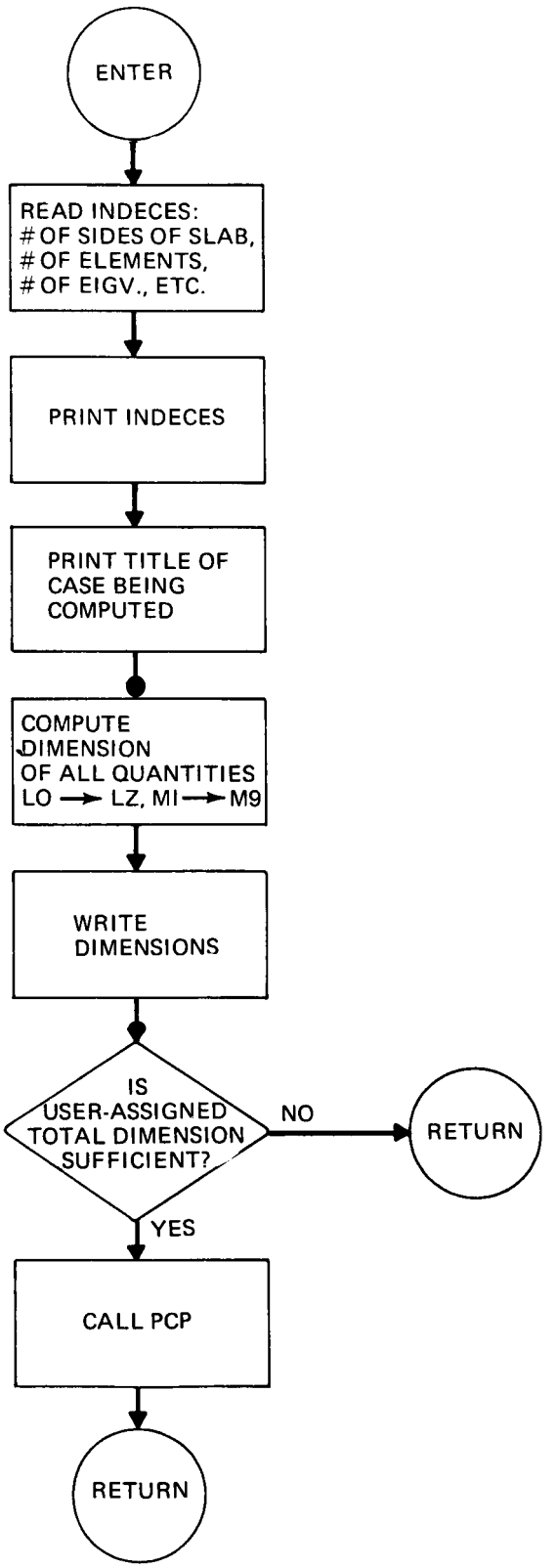
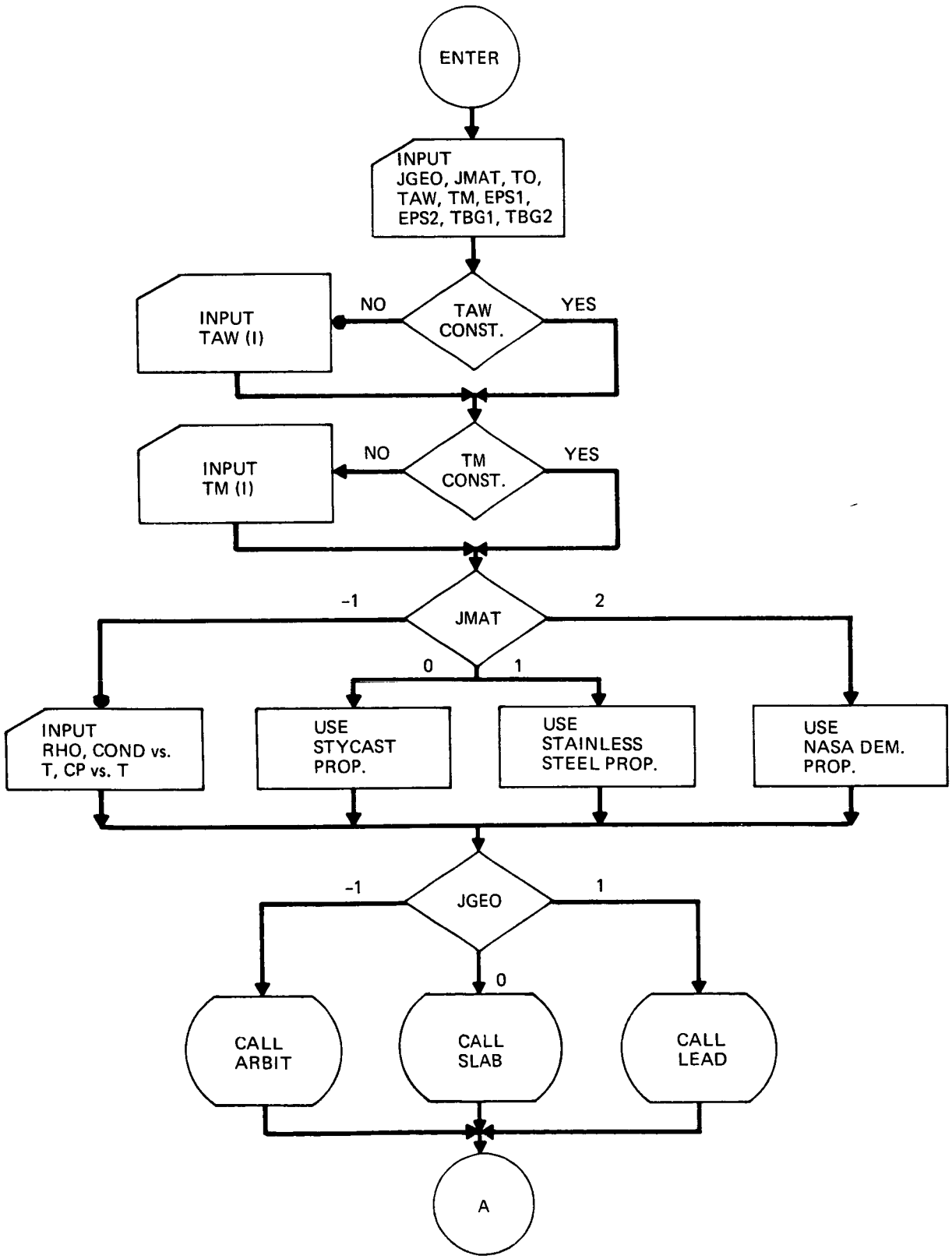


FIGURE B-2. OVERALL FLOW OF LOGIC ON CAPE IDENTIFIED WITH MOST IMPORTANT ROUTINES. (SHEET 3 OF 3)

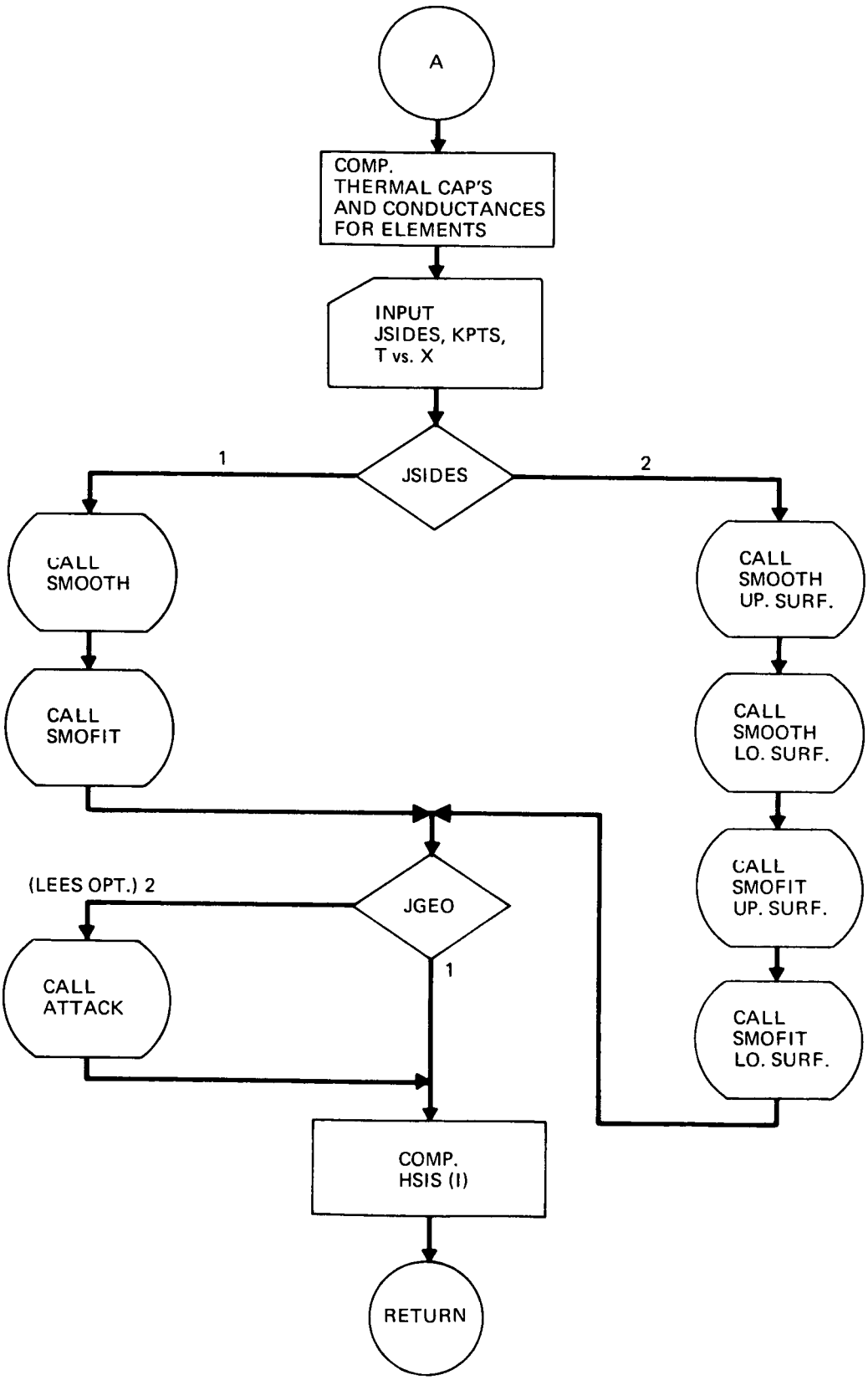
SHEET B-1 SUBROUTINE SIZE FLOW CHART



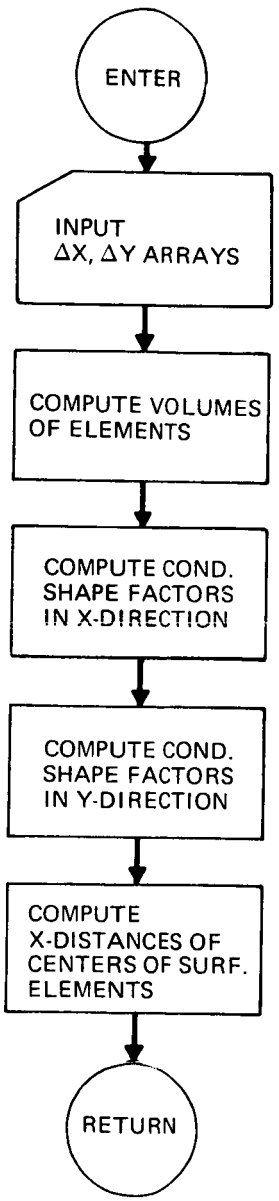
SHEET B-2 (SHEET 1 OF 2) SUBROUTINE PCP FLOW CHART



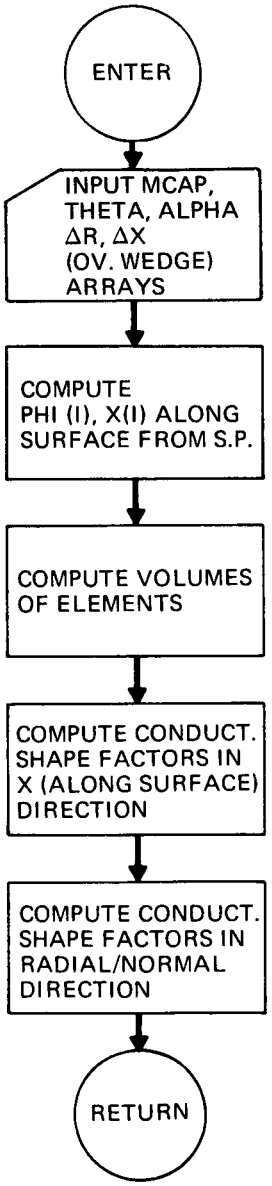
SHEET B-2 (SHEET 2 OF 2) PCP FLOW CHART



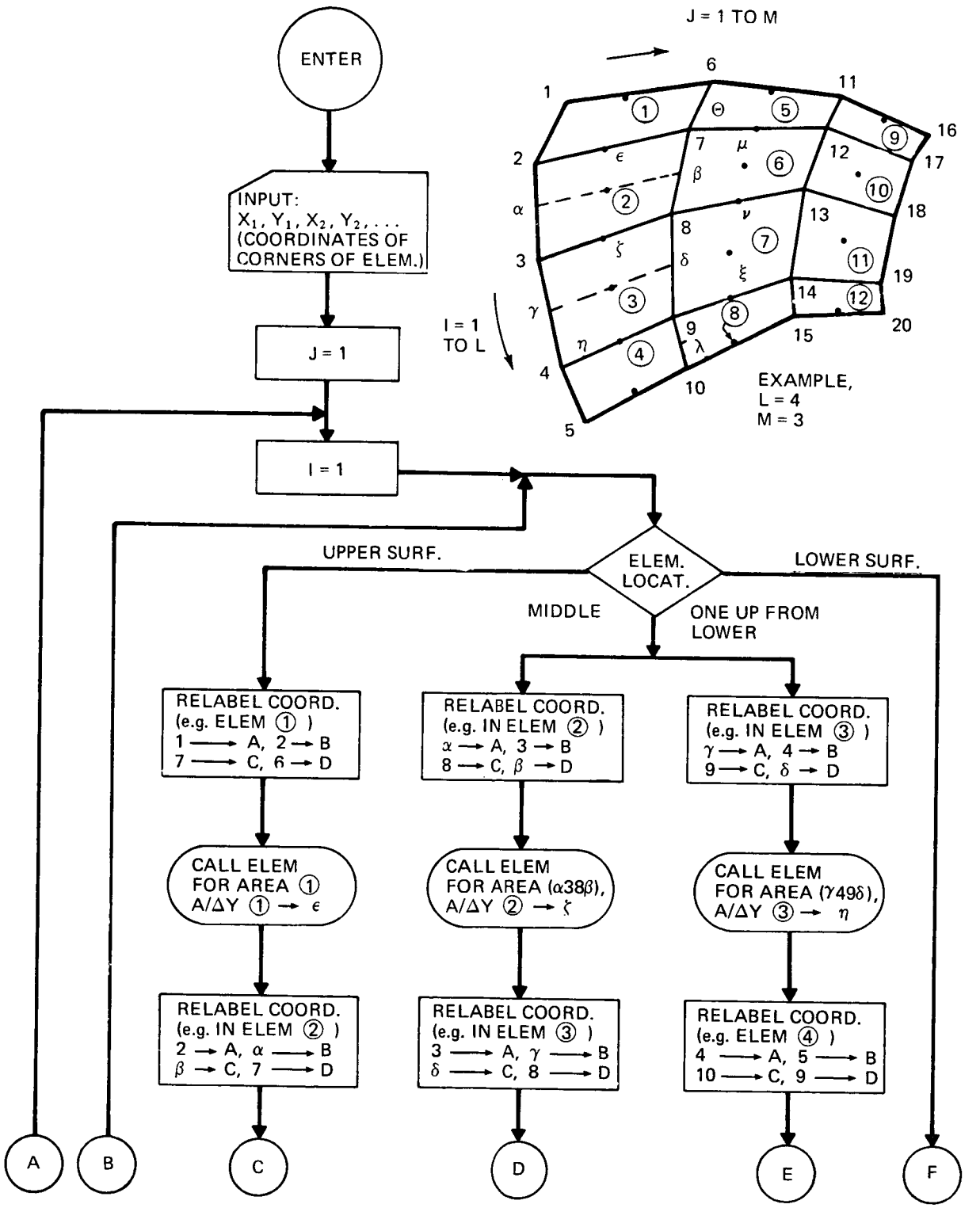
SHEET B-3 SUBROUTINE SLAB FLOW CHARTS



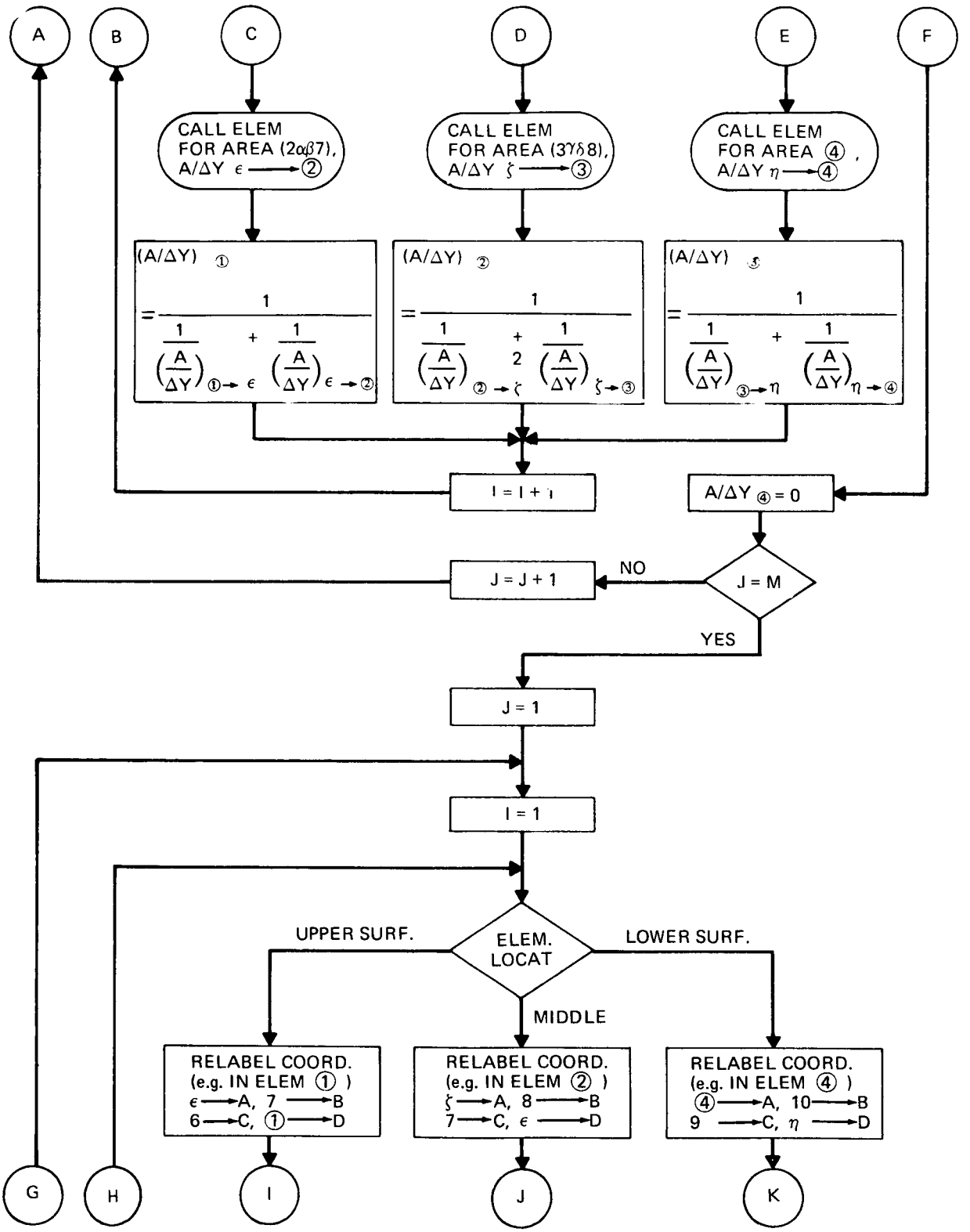
SHEET B-4 SUBROUTINE LEAD FLOW CHART



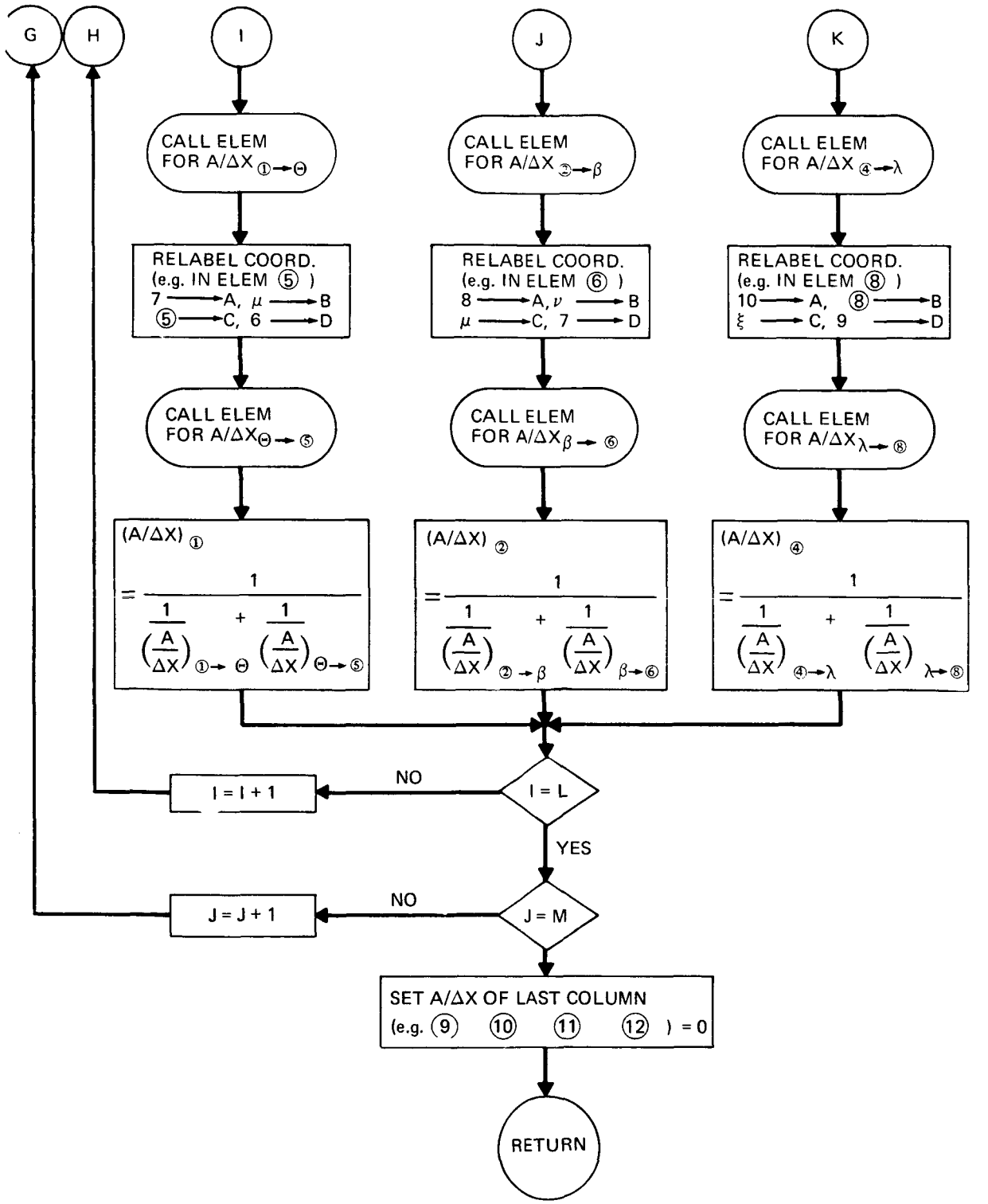
SHEET B-5 (SHEET 1 OF 3) SUBROUTINE ARBIT FLOW CHART



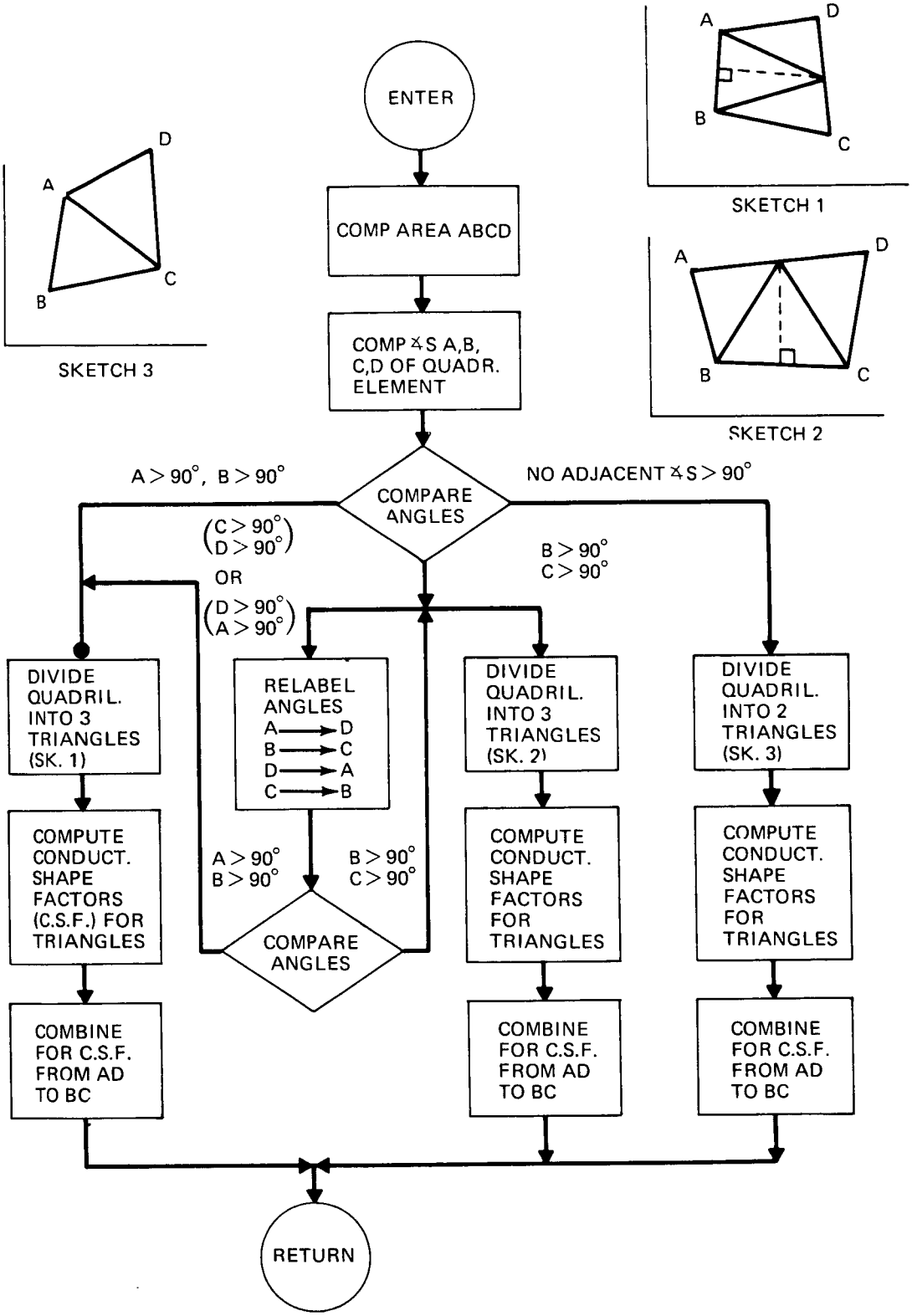
SHEET B-5 (SHEET 2 OF 3) ARBIT FLOW CHART



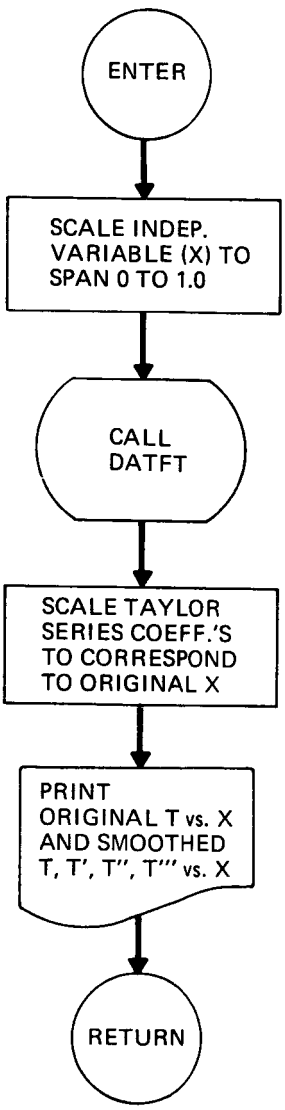
SHEET B-5 (SHEET 3 OF 3) ARBIT FLOW CHART



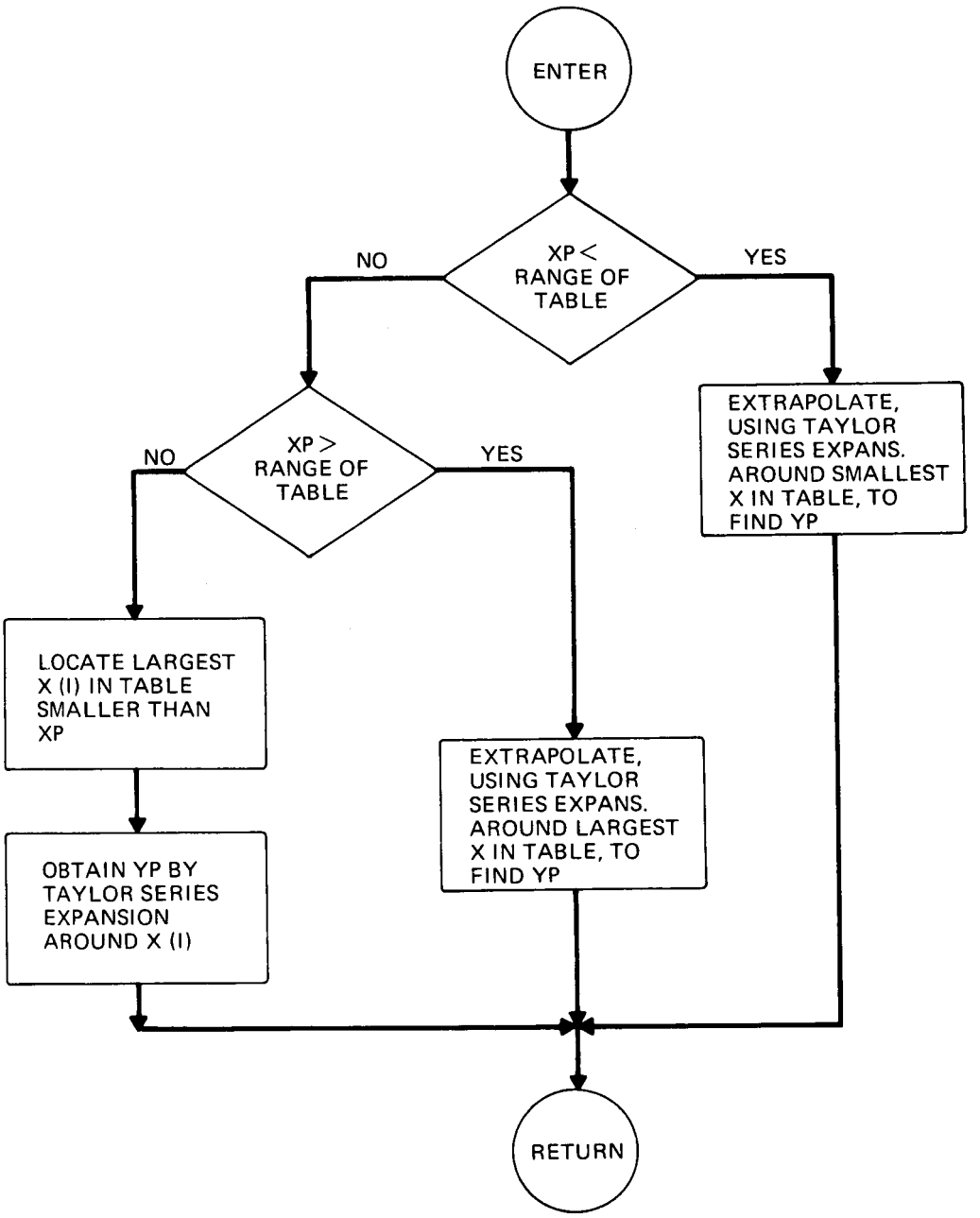
SHEET B-6 SUBROUTINE ELEM FLOW CHART



SHEET B-7 SUBROUTINE SMOOTH FLOW CHART



SHEET B-8 SUBROUTINE SMOFIT FLOW CHART



SHEET B-9 SUBROUTINE DATFT DESCRIPTION (SHEET 1 OF 2)

Smoothing Spline Routine

TITLE: DATFT

AUTHOR: Antony Jameson, Grumman Aerospace Corporation

DATE: January 1972

APPLICABLE COMPUTERS: IBM 1130, 360-370 series, CDC 6600 series

SOURCE LANGUAGE: IBM 1130 FORTRAN

PURPOSE: To generate the smoothest possible curve that will pass within specified tolerances of a given set of data points. By reducing the tolerances to zero, the curve can be made to pass through the data points. By increasing the tolerances sufficiently the smoothest curve becomes a straight-line, least-squares fit to the data.

METHOD: To construct a smoothing spline:
Let (y_i, x_i), $i = 1, 2, \ldots, n$, be the coordinates of the data points. Then construct the curve $f(x)$ that minimizes

$$I = R_1 \int_{x_1}^{x_n} (f''(x))^2 \, dx + R_2 \sum_{i=1}^{n} \left(\frac{y_1 - f_i}{Q_i}\right)^2 \tag{1}$$

where

$$f_i = f(x_i)$$

and

$$Q_i = \text{tolerance for } i^{th} \text{ point}$$

Here the second derivative $f''(x)$ is used as a measure of curvature.

The minimizing curve is a piecewise cubic curve that can be expressed over the interval from x_i to x_{i+1} in the form

$$f(x) = f_i + (x - x_i) f_i' + \frac{(x - x_i)^2}{2} f_i'' + \frac{(x - x_i)^3}{6} f_i''' \tag{2}$$

where

f_1', f_i'', and f_i''' are the first, second, and third derivatives at x_i.

The coefficients f_i', f_i'', and f_i''' are calculated by a method similar to that described by Reinsch (Num. Math. 10, 1967, 177-183.). This requires the solution of a set of linear equations with a sparse matrix containing five diagonals. The number of computer operations is directly proportional to the number of n data points. This is an advantage compared with some other regression techniques where the number of the operations may vary as n^3.

SHEET B-9 SUBROUTINE DATFT DESCRIPTION (SHEET 2 OF 2)

METHOD: (Continued) In the limit $R_1 \rightarrow 0$ the minimizing curve reduces to a spline passing through the points. In the limit $R_2 \rightarrow 0$ the minimizing curve reduces to a straight line giving a weighted least squares fit to the data. In the routine R_1 and R_2 are determined by a 'smoothing parameter' R through the relations

$$R_2 = (1 - R)^4 \ , \ R_1 = 1 - R_2$$

so that R = 0 yields a pure spline (zero smoothing) and $R_1 = 1$ gives a straight line (full smoothing).

The tolerances can be varied from point to point by choice of the weighting parameters Q_i. Smaller values of Q_i should be used in regions where the curve is expected to have high curvature, to prevent the smoothing procedure replacing the true curve by one of larger radius passing outside the data points.

LIMITATION: The curve that minimizes I has zero curvature at the end points:

$$f''(x_1) = f''(x_n) = 0$$

If the true curve is known not to have zero curvature at either or possibly both of the end points, this will lead to a systematic error near the end point where the violation occurs. The minimization problem, Eq. (1), with free end conditions ought then to be replaced by a minimization problem with appropriate constraints at the end points. The existing routine has no provision for this.

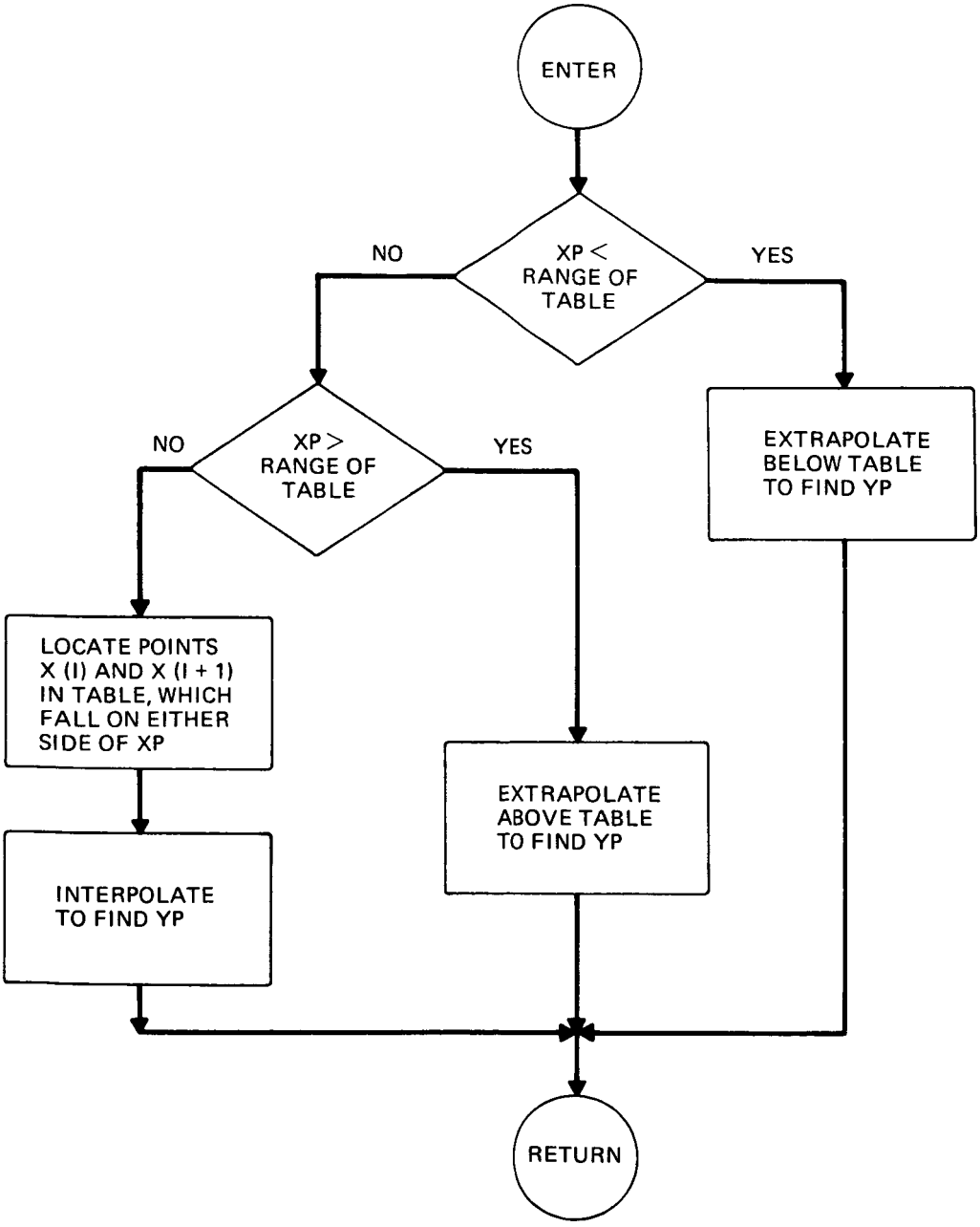
USAGE: CALL DATFT (M, N, S, F, Q, R, A, B, C, D)

The input data consists of the arrays F(I) of values of the dependent variable, S(I) of values of the independent variable, and Q(I) of the tolerances to be allowed. S(I) must be a monotone increasing or decreasing array. It is also necessary to supply values for the indices M and N, and the smoothing parameter R. The routine generates a fit over the interval from S(M) to S(N). R should be a number between 0.0 and 1.0; R = 0.0 gives zero smoothing (pure spline); R = 1.0 gives full smoothing (straight line least squares fit).

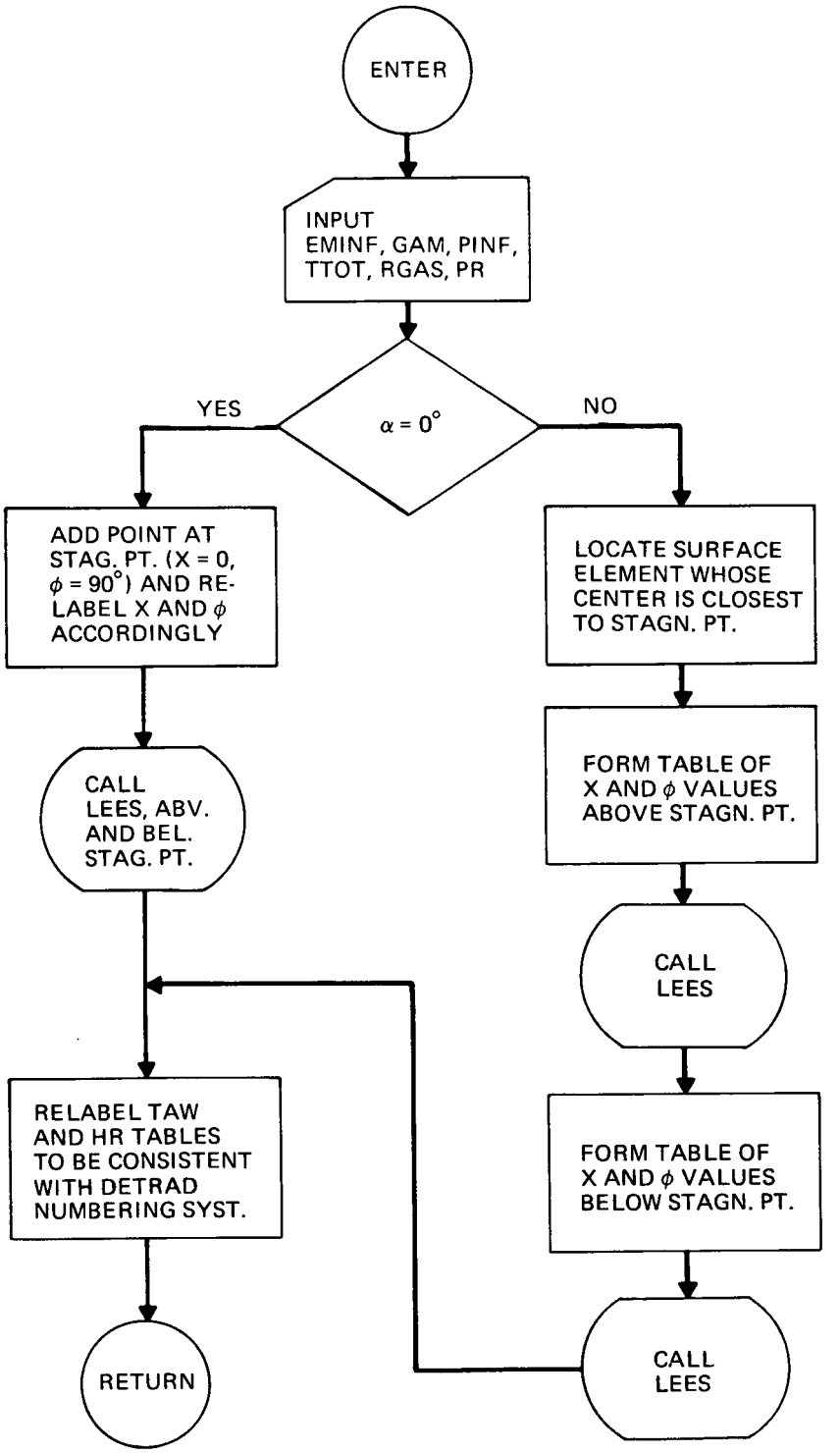
The routine returns the arrays A(I), B(I), C(I), and D(I) defined from I = M to I = N. A(I) are the fitted values at S(I). B(I), C(I) and D(I) are the first, second, and third derivatives to be used in evaluating the cubic curve over the interval from S(I) to S(I+1) according to the formula, Eq. (2).

SUBROUTINES REQUIRED: None.

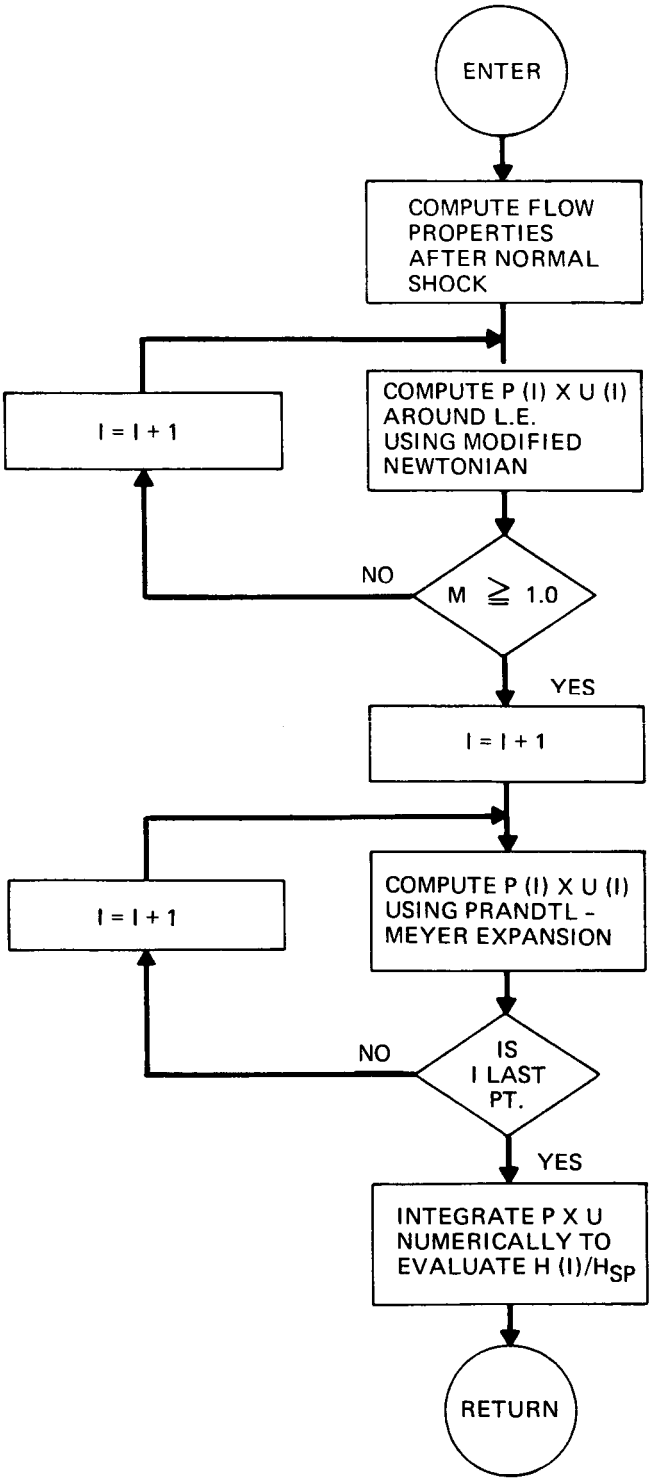
SHEET B-10 SUBROUTINE LINFIT FLOW CHART



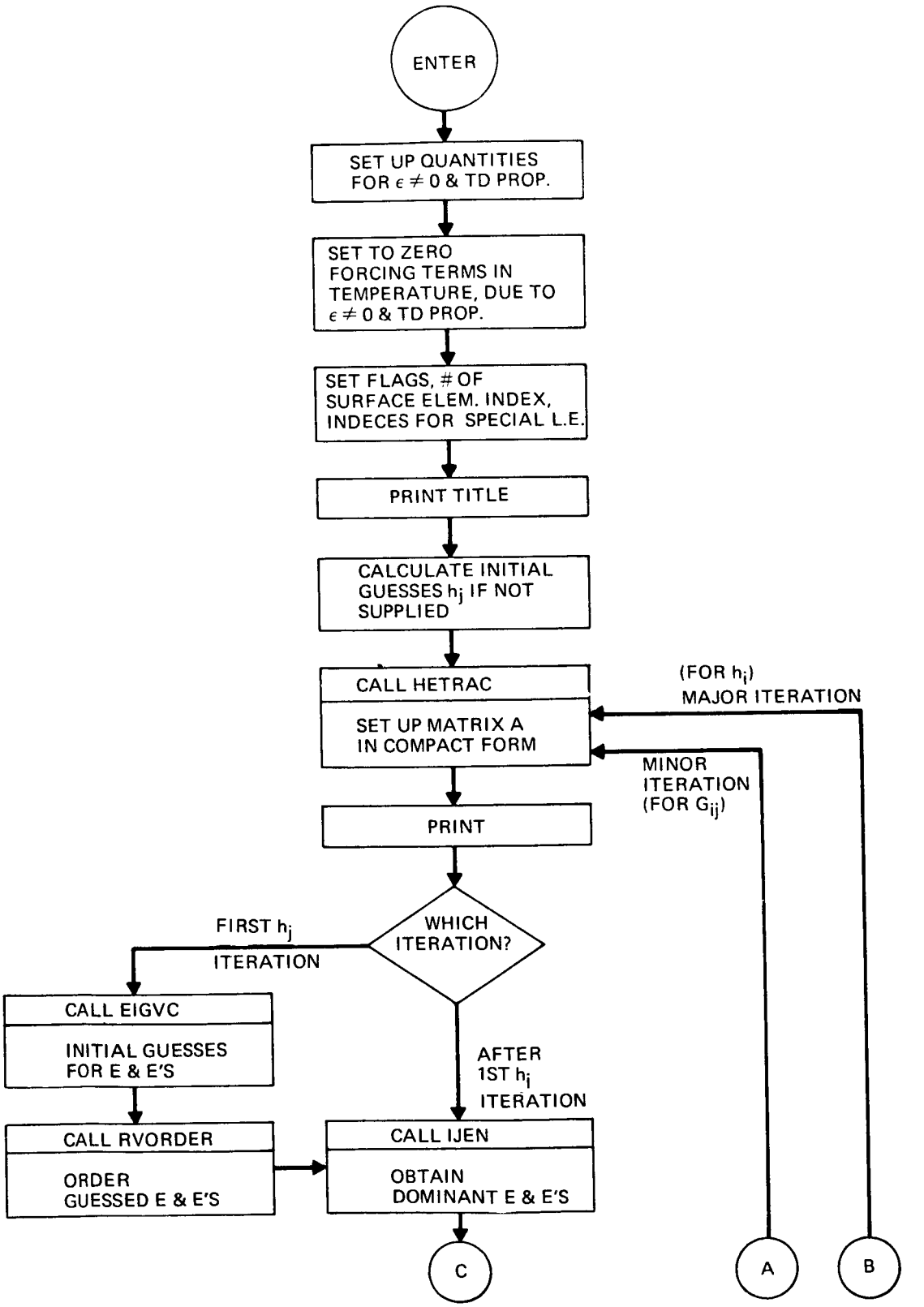
SHEET B-11 SUBROUTINE ATTACK FLOW CHART



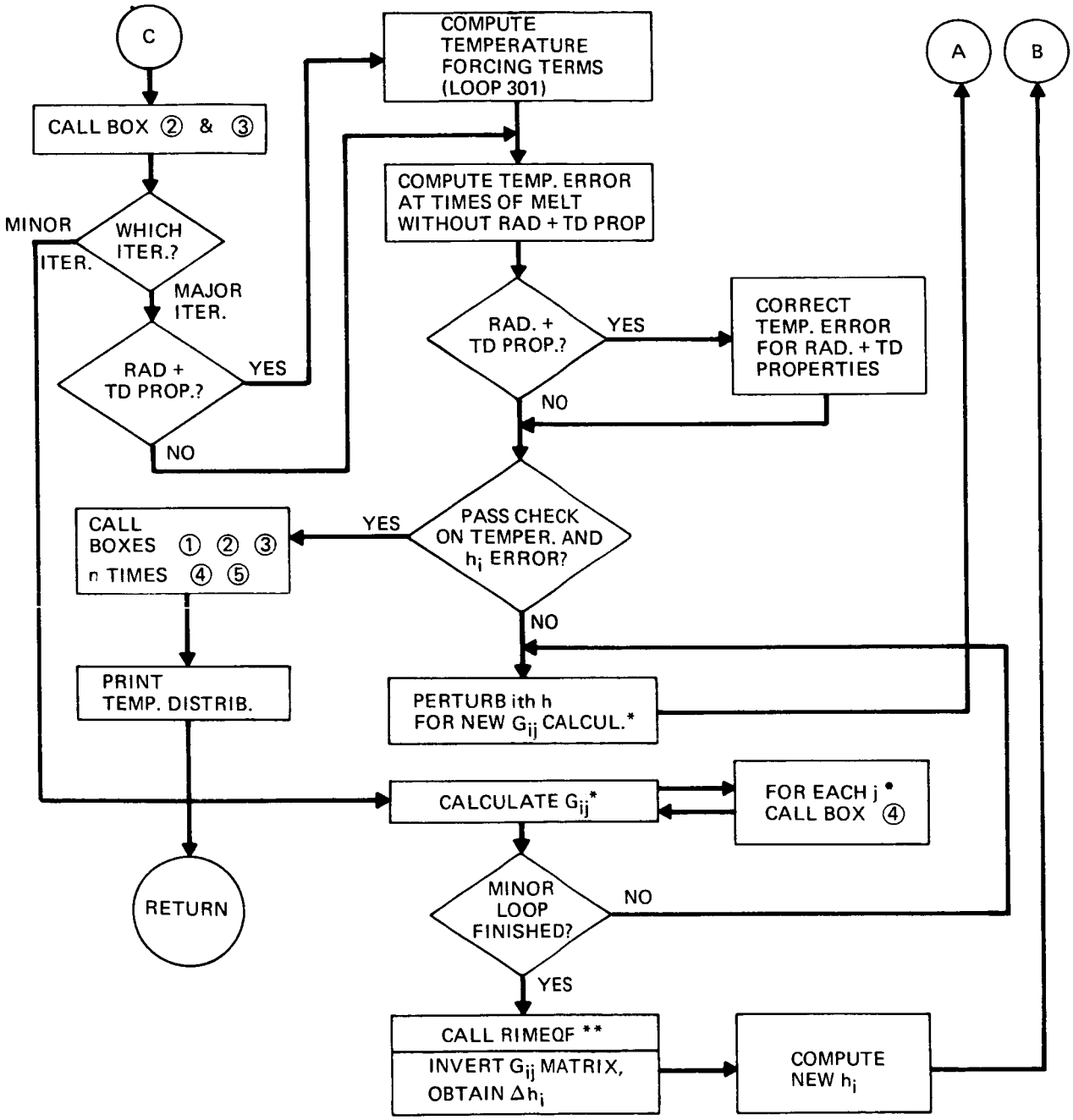
SHEET B-12 SUBROUTINE LEES FLOW CHART



SHEET B-13 SUBROUTINE DETRAD FLOW CHART



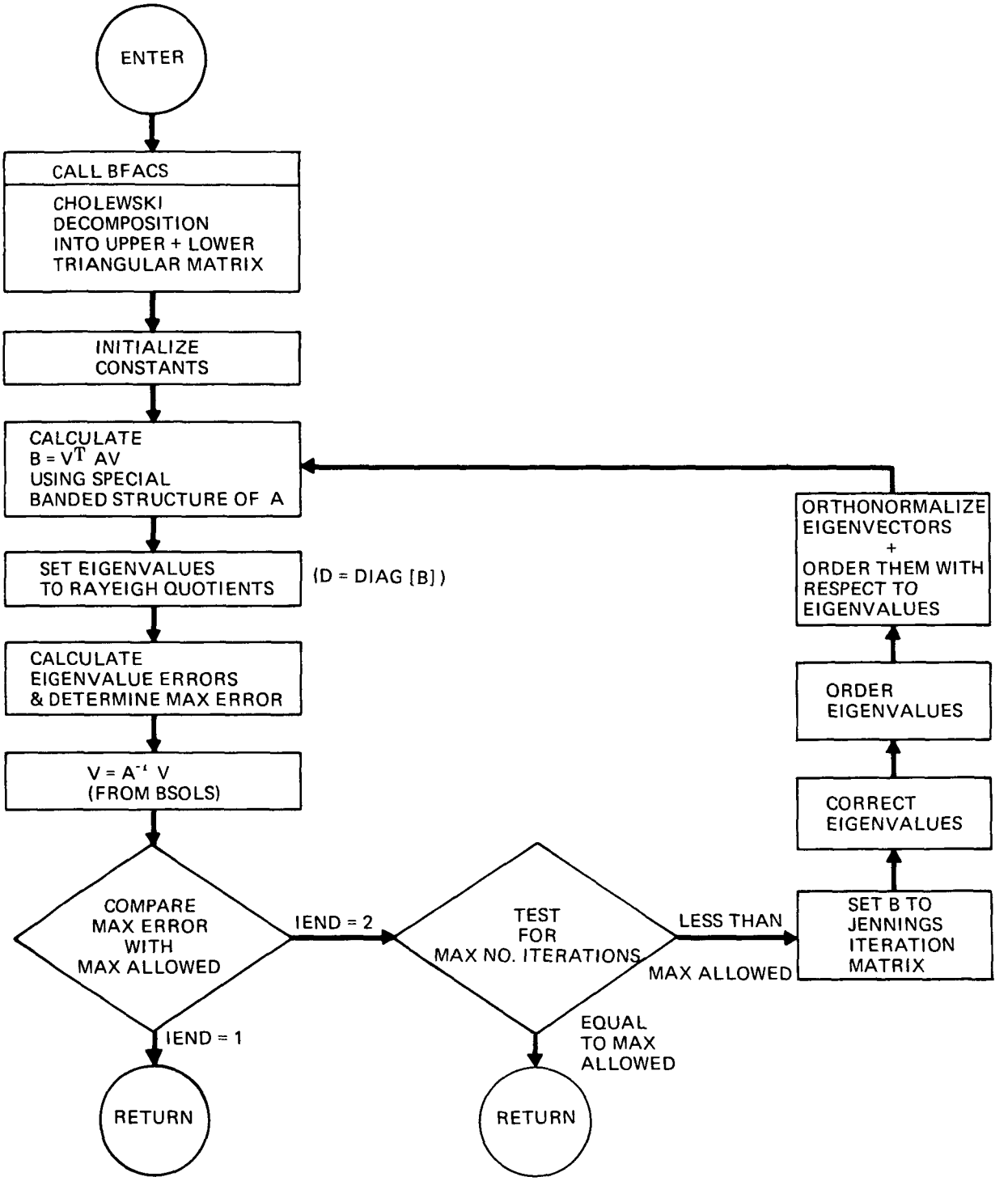
SUBROUTINE DETRAD (CONT'D)



BOX ①	②	③	④	⑤
80 PICKS MAX MELT TIME	81-84 CALCULATE STEADY STATE TEMP T_∞ (NO CALCULATION IF TAW UNIFORM)	39-90 CALCULATE VECTOR B = $V^T M^{1/2}(T_{init} - T_\infty)$	91-100 CALCULATE SUM = $(T_i - T_\infty)\sqrt{M_i}$ FOR ELEM. i	101 CALCULATE FINAL TEMP. ERROR FOR kth ELEM.

I.e. SPECIAL PROBLEM { * ONLY FOR STAGNATION POINT
** 1 X 1 MATRIX DONE EXPLICITLY WITHOUT RIMEQF

SHEET B-14 SUBROUTINE IJEN FLOW CHART

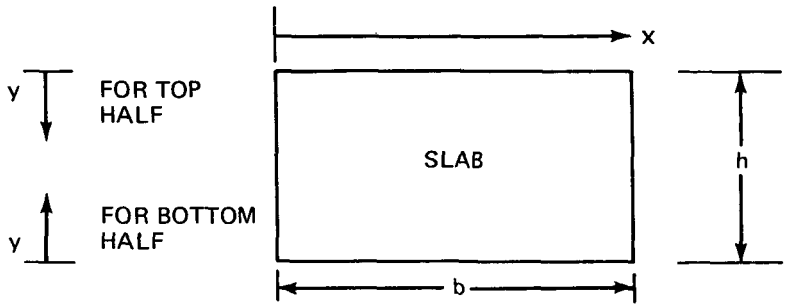


SHEET B-15 SUBROUTINE EIGVC DESCRIPTION

EIGVC computes guesses of the eigenvalues, eigenvectors and associated permutation index that are necessary to start the iteration in the Jennings method to calculate eigenvalues and eigenvectors. The formulae used for these guesses are:

$$\text{ith eigenvalue} \quad R \simeq -i$$

$$\text{ith eigenvector} \quad A \simeq e^{+y/(h/2)} \sin\left\{\frac{\pi}{2} + \pi\frac{x}{b}(i-1)\right\}$$



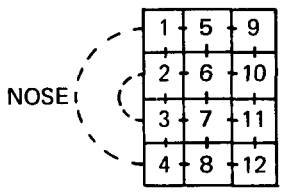
SHEET B-16 SUBROUTINE BFACS DESCRIPTION (SHEET 1 OF 2)

Given a special, L-banded, positive or negative definite symmetric N-th order matrix, S, decompose it into the product:

$$S = U^T D^{-1} U$$

where U is an L-banded nonsingular upper triangular matrix with unit diagonal elements, and D is a nonsingular diagonal matrix.

The S matrix is inputted as elements of three one-dimensional arrays, A, B, E. The N elements of A are the main diagonal elements of S, the leading (N−L) elements of E are the L-th super- (and sub-) diagonal elements of S. The N-1 elements of B are super- (and sub-) diagonal elements of S. *The trailing L/2 elements of E (optionally) define main cross diagonal elements of the upper L-th order submatrix of S. In general, later definitions override earlier one, e.g. if L = 1, B (not E) defines the super diagonal elements of S. In the case for which BFACS is intended, most elements inside the band are zero. The cross diagonal is installed only if the argument, α, is not zero. The S matrix is topologically equivalent to conduction paths in a slab (leading edge if $\alpha \neq 0$); consider the N = 12, L = 4 example:



Note: $b_4 = b_8 = 0$ for the conduction problem generally but must be explicitly made 0 for BFACS

Note: If nose paths are included e_{11}, e_{12} are used ($\alpha \neq 0$).

	1	2	3	4	5	6	7	8	9	10	11	12
1	a_1	b_1	0	e_{12}	e_1	0	0	0	0	0	0	0
2	b_1	a_2	e_{11}	0	0	e_2	0	0	0	0	0	0
3	0	e_{11}	a_3	b_3	0	0	e_3	0	0	0	0	0
4	e_{12}	0	b_3	a_4	b_4	0	0	e_4	0	0	0	0
5	e_1	0	0	b_4	a_5	b_5	0	0	e_5	0	0	0
6	0	e_2	0	0	b_5	a_6	b_6	0	0	e_6	0	0
7	0	0	e_3	0	0	b_6	a_7	b_7	0	0	e_7	0
8	0	0	0	e_4	0	0	b_7	a_8	b_8	0	0	e_8
9	0	0	0	0	e_5	0	0	b_8	a_9	b_9	0	0
10	0	0	0	0	0	e_6	0	0	b_9	a_{10}	b_{10}	0
11	0	0	0	0	0	0	e_7	0	0	b_{10}	a_{11}	b_{11}
12	0	0	0	0	0	0	0	e_8	0	0	b_{11}	a_{12}

SHEET B-16 SUBROUTINE BFACS DESCRIPTION (SHEET 2 OF 2)

To take advantage of both symmetry and the band form U is stored in a rectangular array of size MID by L, where MID $\geq$ N; the bottom row is used as scratch storage, $r_1, r_2, \ldots, r_L$, and the unused bottom triangle is zeroed out (for convenience in printing only). U appears as:

STORED ARRAY

	1	2	3	4
1	u_{11}	u_{12}	u_{13}	u_{14}
2	u_{21}	u_{22}	u_{23}	u_{24}
3	u_{31}	u_{32}	u_{33}	u_{34}
4	u_{41}	u_{42}	u_{43}	u_{44}
5	u_{51}	u_{52}	u_{53}	u_{54}
6	u_{61}	u_{62}	u_{63}	u_{64}
7	u_{71}	u_{72}	u_{73}	u_{74}
8	u_{81}	u_{82}	u_{83}	u_{84}
9	u_{91}	u_{92}	u_{93}	0
10	$u_{10,1}$	$u_{10,2}$	0	0
11	$u_{11,1}$	0	0	0
12	r_1	r_2	r_3	r_4

NON ZERO ELEMENTS OF U MATRIX

	1	2	3	4	5	6	7	8	9	10	11	12
1	1	u_{11}	u_{12}	u_{13}	u_{14}							
2		1	u_{21}	u_{22}	u_{23}	u_{24}						
3			1	u_{31}	u_{32}	u_{33}	u_{34}					
4				1	u_{41}	u_{42}	u_{43}	u_{44}				
5					1	u_{51}	u_{52}	u_{53}	u_{54}			
6						1	u_{61}	u_{62}	u_{63}	u_{64}		
7							1	u_{71}	u_{72}	u_{73}	u_{74}	
8								1	u_{81}	u_{82}	u_{83}	u_{84}
9									1	u_{91}	u_{92}	u_{93}
10										1	$u_{10,1}$	$u_{10,2}$
11											1	$u_{11,1}$
12												1

The N elements of D^{-1} are stored in an N-array. For the usual case of S being either positive-or negative-definite, these elements are all positive or all negative, respectively. However the routine will "work" provided only that the leading N principle minors are non-zero. For details see the following article which guarantees high accuracy only for the definite cases of usual interest: "Symmetric Decomposition of Positive Definite Band Matrices", R.S. Martin, J.H. Wilkinson, C. I/4, LINEAR ALGEBRA – HANDBOOK FOR AUTOMATIC COMPUTATION, VOLUME II, Springer-Verlag, 1971.

SHEET B-17 SUBROUTINE BSOLS DESCRIPTION

Given the product form decomposition of an L-banded symmetric matrix, $S = U^T D^{-1} U$, as calculated by the BFACS subroutine, BSOLS solves a system of N linear equations with M right hand sides:
$S \{Y_1\ Y_2 \cdots YM\} = \{Y_1\ Y_2 \cdots YM\}$

The routine sumply carries out the standard forward substitution phase:

$$z = U^{-T} y$$

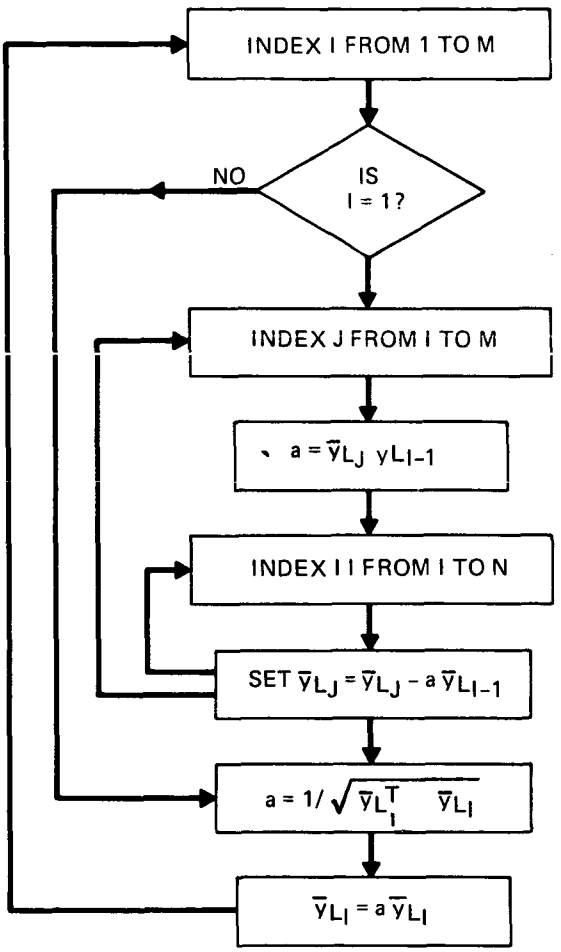
followed by the standard backward substitution phase:

$$x = U^{-1} D\ z$$

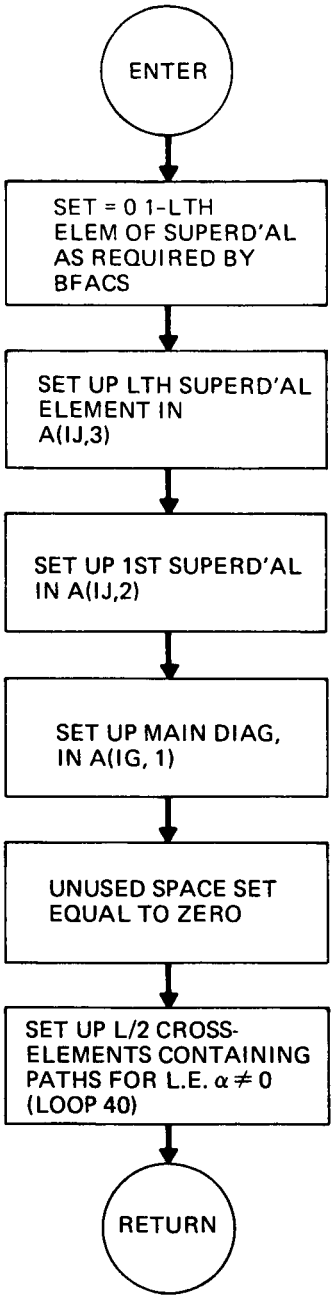
The only unusual aspect is the rather unorthodox storage scheme which is described in the documentation for subroutine BFACS. This scheme is necessary to exploit the banded symmetric form of S in the most efficient way in terms of computer memory. For details see: R.S. Martin, J.H. Wilkinson, "Symmetric Decomposition of Positive Definite Bank Matrices", in: Linear Algebra–Handbook for Automatic Computation, Volume II, C. I/4, Springer-Verlug, 1971

SHEET B-18 SUBROUTINE ORNML

Classical Gram-Schmidt orthonormalization of a set of M linearly independent vectors $\overline{y}_1, \overline{y}_2 \cdots \overline{y}$ in the order $L_1, L_2, L_3, \ldots, L_M$

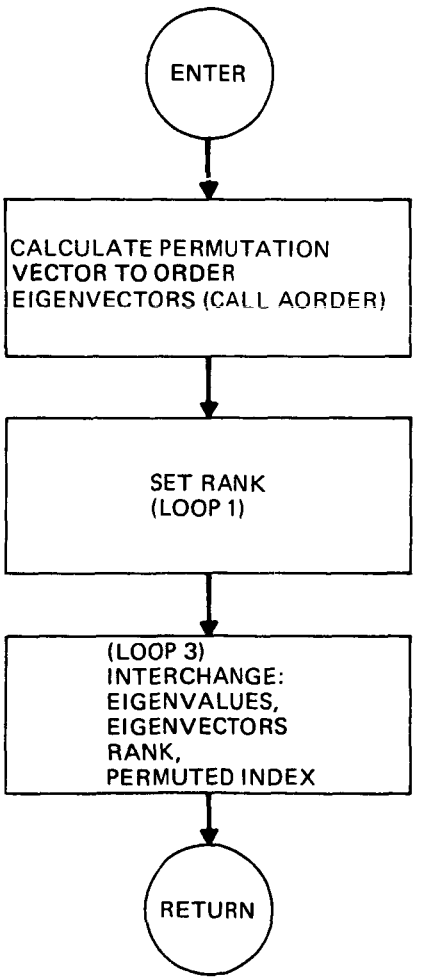


SHEET B-19 SUBROUTINE HETRAC FLOW CHART



SHEET B-20 SUBROUTINE RVORDER

QUANTITY	SYMBOL	INPUT/OUTPUT	DIMENSION
EIGENVALUES	R	IN + OUT	R(MM)
EIGENVECTORS	V	IN + OUT	V(MID, MM)
RANK VECTOR	K	OUT	K(MM)
PERMUTATION VECTOR	L	OUT	L(MM)
EIGENVECTOR DIMENSION	N	IN	–
NUMBER OF EIGENVECTORS	MM	IN	
DIMENSION OF ARRAY USED TO STORE EIGENVECTORS	MID	IN	



SHEET B-21 SUBROUTINE AORDER DESCRIPTION

PURPOSE: ORDER A SET OF REAL NUMBERS

CALLING SEQUENCE:

CALL AORDER (A, N, IPERM)

	NAME	DIMENSION	DESCRIPTION
INPUT	A	A(N)	ELEMENTS TO BE ORDERED
	N		NUMBER OF ELEMENTS = I N I N > 0 INCREASING ORDER N < 0 DECREASING ORDER
OUTPUT	IPERM	IPERM(N)	ORDER VECTOR - SPECIFIES THE SEQUENCE OF ELEMENT INDEX NUMBERS WHICH WILL PRESENT A AS AN ORDERED SET, i.e. DO 100 I = 1, N 100 WRITE (6,1) A (IPERM(I)) 1 FORMAT (F 10.5) WILL LIST A AS AN ORDERED ARRAY

AORDER CALLS NO OTHER SUBROUTINES

SHEET B-22 SUBROUTINE DISPLA DESCRIPTION

TITLE: DISPLA – Prints scalars, vectors, rectangular matrices, packed symmetric matrices, and Hessenberg matrices.

AUTHOR: M. J. Rossi

DATE: September 1973

APPLICABLE COMPUTERS: IBM 360/370; CDC 6000 SERIES

SOURCE LANGUAGE: FORTRAN IV

PURPOSE: To simplify printing of mathematical types of data structures in an easily read format which allows titles and index labels.

METHOD: FORTRAN looping and write statements which indexes and addresses arrays according to their type.

USAGE: Call DISPLA (X, NFILE, TITLE, KAR, KIND, NROWS, NCOLS, MID).

X – Input – Array of one or more values to be printed
NFILE – Input – FORTRAN unit for printing.
TITLE – Input – Vector of KAR characters used as title.
KAR – Input – Number of characters in above string.
KIND – Input – Type of mathematical data structure:
= 0 scalar (or vector printed on one line with no index)
= 1 vector of |NROWS| elements, indexed
= 2 Rectangular |NROWS| by NCOLS matrix – Dimension (MID, *)
= 3 Packed Symmetric matrix of order |NROWS|

– $\begin{bmatrix} 1 & 2 & 4 \\ 2 & 3 & 5 \\ 4 & 5 & 6 \end{bmatrix}$ – lower triangular partial rows if NROWS positive

– $\begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 6 \end{bmatrix}$ – lower triangular partial columns if NROWS negative

= 4 – Transposed Hessenberg matrix of order NROWS – Dimension (MID, MID)

NROWS – Input – Number of elements if KIND = 0 or 1
– Number of rows if KIND = 2
– Matrix order if KIND = 3 or 4
NCOLS – Input – Number of columns if KIND = 2
– Ignored otherwise
MID – Input – Matrix Dimension if KIND = 2 or 4
– Ignored otherwise

SUBROUTINE REQUIRED: SWITCH

SHEET B-23 SUBROUTINE LUSOL DESCRIPTION

Given the factorized product form of $A = P \bullet L \bullet U$, LUSOL solves a linear system $A\,x = y$. The solution vector is obtained in two steps: (1) $z = (L^{-1}\ (P^{T}\ y))$, and then (2) $x = U^{-1}\ z$. A good discussion of the details may be found in the following article: "Solution of Real and Complex Systems of Linear Equations," H. J. Bowdler, R. S. Martin, G. Peters, and J. H. Wilkinson, C. I/7, pp. 93-110, Linear Algebra – Handbook for Automatic Computation, Volume II, Springer-Verlag, 1971.

SHEET B-24 SUBROUTINE PART DESCRIPTION

TITLE: PART – Prints standard 120 character labels at the top of the next page.

AUTHOR: M. J. Rossi

DATE: September 1973

APPLICABLE COMPUTERS: IBM 360/370; CDC 6000 SERIES

SOURCE LANGUAGE: FORTRAN IV with 2 Assembler Language Subordinate Subroutines.

PURPOSE: To make it convenient to produce standard printed labels with "part" numbers, date, and running CPU time on the top of the next page. Also, prints short line on next line with just CPU time for intermediate timing.

METHOD: On the first printing entry for a given computer run the Date Subroutine is invoked and an 8-character field of an internal word is stored with the date in the form: "KK/LL/MM", where KK is the index number for the month, LL is the day of the month, and MM is the last 2 digits of the year, e.g., March 15, 1973 3/15/73. Also, at this time, the SECOND subroutine is invoked to both establish the zero time point and to set the units to hundreds of a second. Then the first printed page heading is given with a zero time and a PART number 1 reported. Subsequent printing calls will give the time as: NN.II.JJ where NN is the number of minutes elapsed, II is the number of seconds, and JJ is the number of hundredths of seconds. The PART number is incremented by one for each printing call. There are two fields of alphameric information for the full printing mode which are under control of the user: (1) The first is a 40 character LABEL field which is set upon calling PART in the non-printing mode, (2) The second is a 48 character field which is supplied on a full printing call. There is also a partial printing mode which simply results in the appearance on the next line of an 8 character field of user supplied TITLE along with running CPU time.

USAGE: Call PART ('XX. . .X', L)
'XX. . .X' – Input = Alphameric string of either 8, 40, or 48 characters depending on the value of L.
L – Input – FORTRAN unit for printing, if positive
– If zero, simply sets 40 character LABEL field and returns
– If negative, prints 8 character TITLE – 'XX. . .X' – and CPU time on next line and increments PART number.
– If positive, prints DATE, TIME, 40 character LABEL, 48 character TITLE, Part Number and spacers with standard notation.

SUBROUTINES REQUIRED: DATE, SECOND

SHEET B-25 SUBROUTINE RDET DESCRIPTION

Given a square matrix, A, stored in a Fortran double array, decompose it into the product:

A = P * L * U

where P is a permutation matrix, L is lower triangular and U is upper triangular. The algorithm includes implicit row scaling and partial pivoting while providing a test for singularity of A. For details, see reference given for subroutine LUSOL.

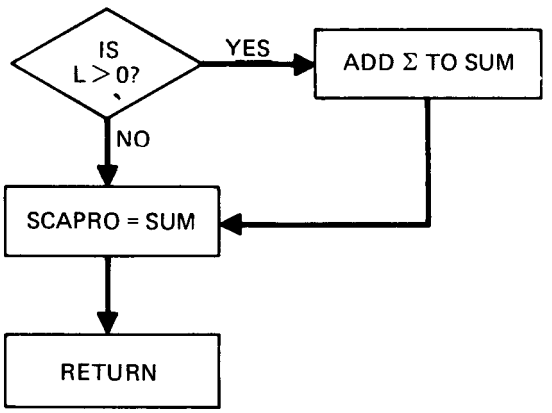
SHEET B-26 SUBROUTINE RIMEQF

Solves the system of linear equations, A x = y. The first step involves decomposing A as the product: A = P * L * U which is done by a call to subroutine RDET. This is followed by the standard method of forward substitution and then backward substitution performed by a call to subroutine LUSOL. This is equivalent to solving a pair of triangular systems using L and then U. The total procedure may be viewed as a variant of Gaussian elimination as described in more detail in the reference given for subroutine LUSOL.

SHEET B-27 SUBROUTINE SCAPRO

SCAPRO adds to a quantity the inner product of two vectors X, Y stored as equally spaced words in Fortran arrays.

$$SUM = SUM + \sum_{J=1}^{L} X(IX \cdot (J-1) + 1) * Y(IY) \cdot (J-1) + 1)$$



SHEET B-28 SUBROUTINE SWITCH DESCRIPTION

PROGRAM TITLE: Utility routine for re-arrangement of certain triangular arrays
SUBROUTINE NAME: SWITCH INDEX: 12.6.0.1
ANALYST: F. Nolan
PROGRAMMER: F. Nolan DATE: June 15, 1967
DOCUMENTATION AUTHOR: F. Nolan DATE: June 20, 1973
SOURCE LANGUAGE: FORTRAN IV
APPLICABLE COMPUTERS: IBM Systems 360, 370; CDC 6000 series
REVISION: DATE:

PURPOSE: To provide a convenient conversion between two common arrangements for the storage of triangular (and symmetric) matrices.

ANALYTIC DESCRIPTION: The routine makes systematic use of transpositions, i.e., interchanges of two array elements. It is a well known result in permutation theory that every permutation can be represented as a product of transpositions.

PROGRAM DESCRIPTION: There is no loss of generality in assuming that the input matrix is of lower triangular form. It is natural to store such matrices by row or by column. Both arrangements are illustrated for a matrix of order 5. The understanding is that the (4,3) element, for example, is assigned position 9 using row storage, and position 11 using column storage.

Row Storage					Column Storage				
1					1				
2	3				2	6			
4	5	6			3	7	10		
7	8	9	10		4	8	11	13	
11	12	13	14	15	5	9	12	14	15

Given a lower triangular or symmetric matrix, stored in either fashion, SWITCH can re-arrange it to the other form. The re-arrangement is carried out "in place" in the sense that no auxiliary array is required. For an input matrix of order m, the transformation is performed in approximately $\frac{1}{4}m^2$ transpositions. There are no rounding errors.

PROGRAM RESTRICTIONS: The matrix must be of order at least 3.

INPUT PARAMETERS:

FORTRAN Name	Description
A	Singly-dimensioned real array containing the matrix to be re-arranged.
M	Order of matrix A. If M is given positive, conversion is from row to column storage. If M is given negative, conversion is from column to row storage.

OUTPUT PARAMETERS:

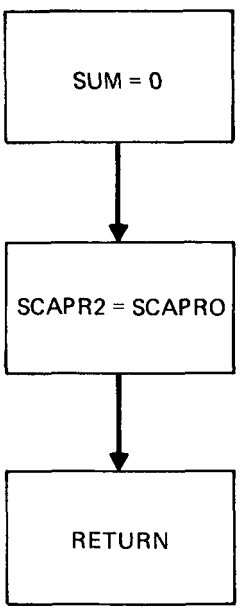
A	Matrix in re-arranged order.

CALLING SEQUENCE:

```
CALL SWITCH (A, M)
```

SHEET B-29 SUBROUTINE SCAPR2

SCAPR2 CALCULATES THE INNER
PRODUCT OF TWO VECTORS STORED
AS EQUALLY SPACED WORDS IN FORTRAN ARRAYS.



SHEET B-30 INPUT DATA LISTING FOR TWO CHECK CASES

```
00000000011111111112222222222333333333344444444445555555555666666666677777777777
01234567890123456789012345678901234567890123456789012345678901234567890123456789

    2   10    9   10    5
LABEL INFORMATION   LE LEES ALPHA#15 CP
   -1    0    0
    2    0    1
540.0      0.0        660.0      0.0        0.0        540.0      540.0
        10   15.0    15.0
0.001      0.001      0.001      0.001
0.0004     0.0008     0.0008     0.0008     0.0012
    2   18
0.000523   0.001570   0.002617   0.003664   0.004711   0.005735   0.006735   0.007733
0.008733   0.000523   0.001570   0.002617   0.003664   0.004711   0.005735   0.006735
0.007733   0.008733
4.55       4.75       5.20       5.80       6.45       7.20       7.72       8.00
8.18       4.55       4.75       5.20       5.80       6.45       7.20       7.72
8.00       8.18
0.0        0.001
8.0        1.4        125.0      1300.0     53.53045   0.72
```

```
00000000011111111112222222222333333333344444444445555555555666666666677777777777
01234567890123456789012345678901234567890123456789012345678901234567890123456789

    2    9   12   10    5
LABEL INFORMATION   SLAB TWO SIDES CP
   -1    0    0
    0    0    2
540.0      1400.0     0.0        0.0        0.0        540.0      540.0
719.5      729.0      767.0      840.0      914.0      992.0      1058.5     1097.5
1117.5     1132.5     1142.0     1146.5     692.0      700.0      726.0      762.0
792.0      825.0      856.0      874.5      884.5      892.0      897.0      900.0
0.0032     0.0032     0.0032     0.0016     0.0016     0.0016     0.0016     0.0008
0.0008     0.0008     0.0008     0.0008
0.00025    0.0005     0.0007     0.0007     0.0007     0.0007     0.0007     0.0005
0.00025
    2   24
0.0016     0.0048     0.0080     0.0104     0.0120     0.0136     0.0152     0.0164
0.0172     0.0180     0.0188     0.0196     0.0016     0.0048     0.0080     0.0104
0.0120     0.0136     0.0152     0.0164     0.0172     0.0180     0.0188     0.0196
3.6        3.6        3.6        3.6        3.6        3.6        3.6        3.6
3.6        3.6        3.6        3.6        3.6        3.6        3.6        3.6
3.6        3.6        3.6        3.6        3.6        3.6        3.6        3.6
0.0        0.001
```

U.S. GOVERNMENT PRINTING OFFICE: 1974—635-042/6