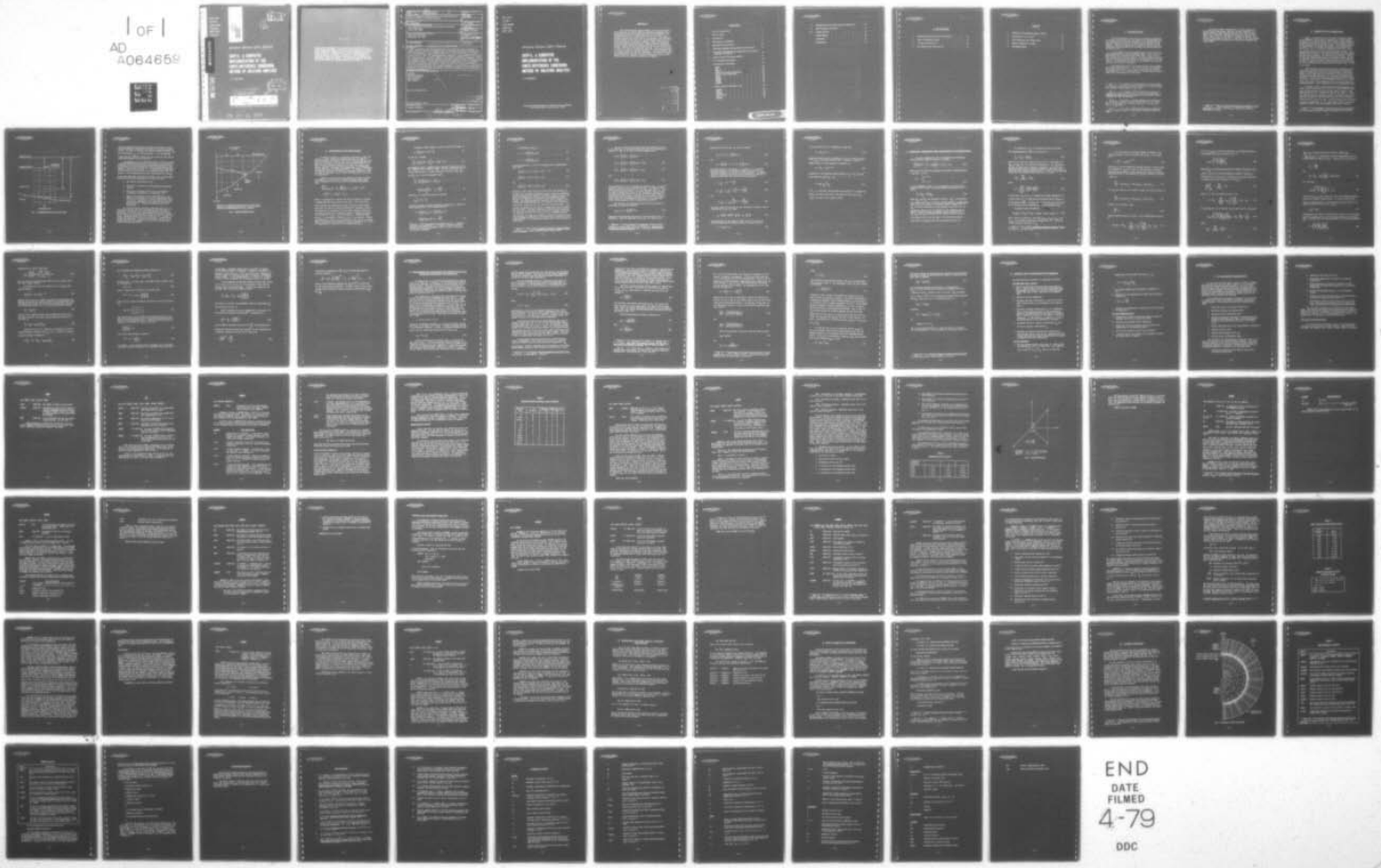
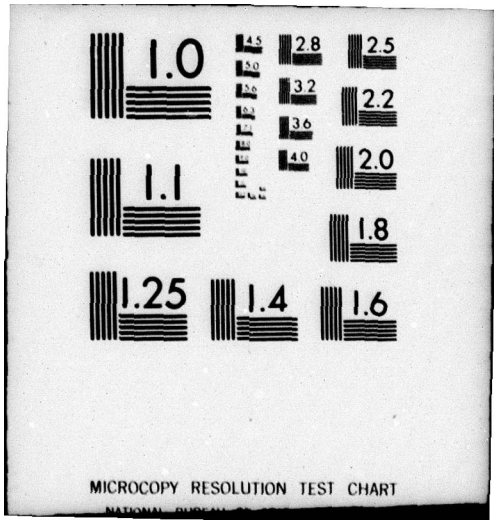
AD-A064 659 JOHNS HOPKINS UNIV LAUREL MD APPLIED PHYSICS LAB F/G 9/2
SHTP-E, A COMPUTER IMPLEMENTATION OF THE FINITE-DIFFERENCE EMBE--ETC(U)
MAY 78 J D RANDALL N00024-78-C-5384
UNCLASSIFIED APL/JHU-CP-067 DOE-ASMP-3060-16 NL

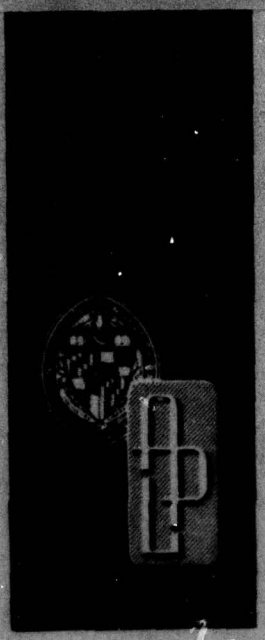
1 OF 1
AD A064659



END
DATE FILMED
4-79
DDC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3
7.1
MICROCOPY RESOLUTION TEST CHART

APL/JHU
CP 067
(DOE/ASMP
3060-16)
MAY 1978
Copy No.



LEVEL II

Aerospace Nuclear Safety Program

SHTP-E, A COMPUTER IMPLEMENTATION OF THE FINITE-DIFFERENCE EMBEDDING METHOD OF ABLATION ANALYSIS

J. D. RANDALL

THE JOHNS HOPKINS UNIVERSITY • APPLIED PHYSICS LABORATORY

NOTICE

BIBLIOGRAPHIC DATA SHEET	1. Report No. APL/JHU-CR-067	2.	3. Recipient's Accession No.
4. Title and Subtitle: SHTP-E, A Computer Implementation of the Finite-Difference Embedding Method of Ablation Analysis.			5. Report Date: May 1978
			6.
7. Author(s): J. D. Randall			8. Performing Organization Rept. No. CP 067
9. Performing Organization Name and Address: The Johns Hopkins University Applied Physics Laboratory, Johns Hopkins Rd., Laurel, MD 20810			10. Project/Task/Work Unit No. Task Z510
			11. Contract/Grant No. N00024-78-C-5384
12. Sponsoring Organization Name and Address: Department of Energy, Germantown, MD 20545			13. Type of Report & Period Covered: Technical Memorandum
			14.

15. Supplementary Notes: DOE; ASMP-3060-16; 93 p.

16. Abstract:

PL/I procedures have been developed that use finite-difference techniques to analyze ablation problems by embedding them in inverse heat-conduction problems with no moving boundaries. The procedures form a set of subroutines that can be called from a problem-oriented main program written by the user. The procedures include provisions for one-, two-, or three-dimensional conduction, parallel modes of heat transfer, thermal contact, choices of implicit- and explicit-difference techniques, temperature-dependent and directional thermal properties, radiation relief, aerodynamic heating, chemical ablation, and material removal from combinations of flat, cylindrical, and spherical surfaces. This report is meant to serve as a source of underlying theory not covered elsewhere and as a user's manual for PL/I procedures. Also included are useful debugging aids and external identifiers, a directory of Applied Physics Laboratory computer libraries pertaining to the PL/I procedures, and an illustrative problem as an example.

17. Key Words and Document Analysis. 17a. Descriptors

Ablation
Aerodynamic Heating
Finite difference
Heat conduction
Heat transfer

17b. Identifiers/Open-Ended Terms

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

17c. COSATI Field/Group 20/13

18. Availability Statement: Release unlimited	19. Security Class (This Report) UNCLASSIFIED	21. No. of Pages
	20. Security Class (This Page) UNCLASSIFIED	22. Price

031 650

FORM NTIS-35 (REV. 3-72) THIS FORM MAY BE REPRODUCED USCOMM-DC 14952-P72

APL/JHU
CP 067
(DOE/ASMP
3060-16)
MAY 1978

Aerospace Nuclear Safety Program

SHTP-E, A COMPUTER IMPLEMENTATION OF THE FINITE-DIFFERENCE EMBEDDING METHOD OF ABLATION ANALYSIS

J. D. RANDALL

THE JOHNS HOPKINS UNIVERSITY ■ APPLIED PHYSICS LABORATORY
Johns Hopkins Road, Laurel, Maryland 20810

THE JOHNS HOPKINS UNIVERSITY
APPLIED PHYSICS LABORATORY
LAUREL, MARYLAND

ABSTRACT

PL/I procedures have been developed that use finite-difference techniques to analyze ablation problems by embedding them in inverse-heat-conduction problems with no moving boundaries. The procedures form a set of subroutines that can be called from a problem-oriented main program written by the user. The procedures include provisions for one-, two-, or three-dimensional conduction, parallel modes of heat transfer, thermal contact, choices of implicit and explicit difference techniques, temperature-dependent and directional thermal properties, radiation relief, aerodynamic heating, chemical ablation, and material removal from combinations of flat, cylindrical, and spherical surfaces. This report is meant to serve as a source of underlying theory not covered elsewhere and as a user's manual for the PL/I procedures. Also included are useful debugging aids and external identifiers, a directory of Applied Physics Laboratory computer libraries pertaining to the PL/I procedures, and an illustrative problem as an example.

CONTENTS

ILLUSTRATIONS

TABLES

1. INTRODUCTION

The work described is the result of an effort to establish an accurate finite-difference (FD) method of multidimensional ablation analysis that can be used economically in a general-purpose computer program applicable to reentry ablation problems of long duration. Details of comparisons of various FD ablation analysis methods and the justification for selecting the one discussed here are given in Ref. 1.

PL/I procedures for incorporating the FD version of the embedding method (EM) of ablation analysis (Refs. 1 and 2) into the BBE Standard Heat Transfer Program (SHTP) (Refs. 3 and 4) have been written and successfully tested. The new procedures described here are collectively called SHTP-E and are applicable to the ablation of a subliming material subject to aerodynamic heating during reentry into the earth's atmosphere.

The procedures involve two features that are new in SHTP-E and require some explanation. One is EM; the other is approximate factorization (AF) (Ref. 1), an implicit FD method that permits the decomposition of an N-dimensional nodal energy-balance equation (EBE) into N one-dimensional nodal EBE's.

Ref. 1. J. D. Randall, "An Investigation of Finite Difference Recession Computation Techniques Applied to a Nonlinear Recession Problem," APL/JHU ANSP-M-15, Mar 1978.

Ref. 2. J. D. Randall, "Finite Difference Solution of the Inverse Heat Conduction Problem and Ablation," Proc. 1976 Heat Transfer and Fluid Mechanics Institute, Stanford University Press, Stanford, CA, 1976, pp. 257-269.

Ref. 3. R. K. Frazer, "Further Additions and Improvements to the BBE Heat Transfer Program," APL/JHU EM-4274, 16 Jun 1969; Rev. 1, 29 Oct 1969; Rev. 2, 22 Mar 1971; Rev. 3, 29 Mar 1972; Rev. 4, 22 Jan 1973.

Ref. 4. R. K. Frazer, "URLIM-A Unified Radome Limitations Computer Program, Volume 2 — Users Guide," APL/JHU TG 1293B, Apr 1978.

This report briefly describes EM, AF, their combination and associated terminology, theory not covered elsewhere, the general numerical algorithm used, and the use of each of the required PL/I procedures. Familiarity of the user with the SHTP (Refs. 3 and 4) and the PL/I (Ref. 5) language is assumed. Each procedure description is started on a new page so that descriptions of new procedures and revisions of old ones can be inserted from time to time.

Ref. 5. "OS PL/I Checkout and Optimizing Compilers: Language Reference Manual," IBM File No. S360/S370-29, Order No. GC33-0009-4, Oct 1976.

2. DESCRIPTION OF EMBEDDING

Figure 1 illustrates some of the basic terminology needed in discussing EM. EM is used to determine the in-depth thermal response of an ablating material, which involves moving boundaries, by embedding the ablation problem in an inverse problem having no moving boundaries and no material removal. The embedding problem is properly posed by appling a fictitious surface-heating-rate distribution, Q_F, to a stationary embedding surface that covers the receding surface associated with the ablation problem. The embedding model used is of the embedding problem, not the ablation problem that it contains. The fictitious solid between the embedding and ablating surfaces is retained in the analysis. However, the physical boundary conditions of the ablation problem are used to determine the fictitious boundary conditions of the embedding problem. As shown in Section 4, interpolation can be used to find Q_F if the receding surface temperature and location ("s_E", in Fig. 1) are known.

Conductive heat fluxes in the direction of ablation (to be called the A direction) in the ablating material are estimated using new temperature (implicitly), while conductive heat fluxes normal to the A direction (to be called the B and C directions) are estimated either implicitly or by using old temperatures (explicitly). The implicit A-direction internodal heat-flow connections form columns heated on the embedding surface by Q_F, and each column runs normal to the embedding surface at least through the ablating material. These columns will be called embedding columns.

To begin solving a thermal-analysis problem using EM, one needs to know the distance (e_E in Fig. 1) of the embedding surface from a fixed reference surface, and the initial location of the ablating surface; i.e., the initial value of s_E in Fig. 1. Often the reference surface, embedding surface, and initial ablating surface will coincide so that the value of e_E will be zero and the initial value of s_E will be zero. External heating rates are known in terms of fluxes (Ref. 6), but nodal energy balances are established in terms of total heat-transfer rates (Refs. 4 and 5),

---

Ref. 6. R. W. Newman, "A User's Guide for the Continuous Wave Laser Damage Computer Program," APL/JHU TG 1268, Dec 1974.

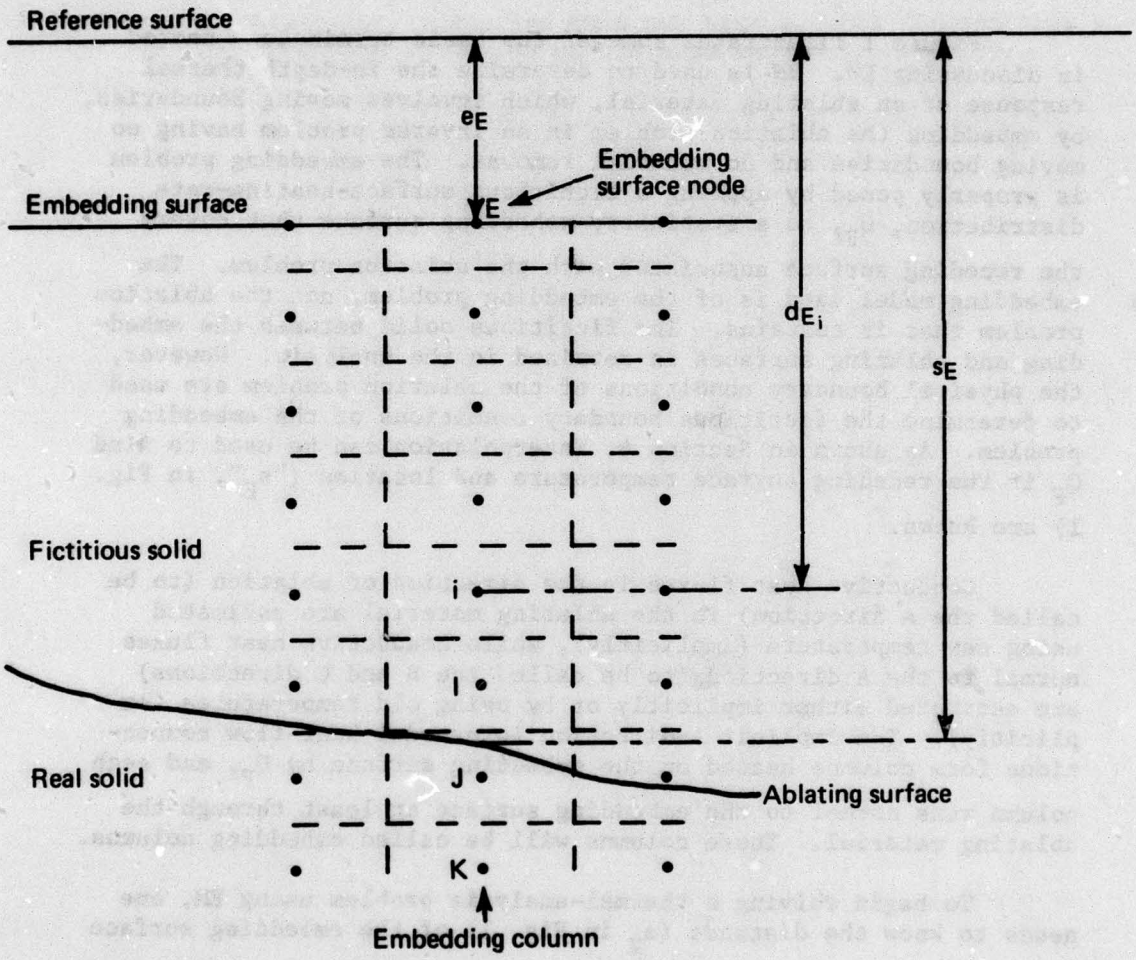


Fig. 1 Embedding Node Structure and Terminology

therefore embedding-surface node areas have to be known to start the calculations if the embedding surface is the initial ablating surface. In order to find Q_{FE}, the current recession distance, s_E, and the distance, d_{E_i}, of each grid point, i, in the embedding column from the reference surface have to be known so that the required interpolation can be carried out.

The temperature of the receding surface is found by solving a surface-energy-balance equation, which requires an accurate determination of the ablating surface temperature gradient in the A direction and the local inclination of the ablating surface with respect to the A direction (Ref. 1). To determine the temperature gradient properly, the position of the ablating surface (s_E) with respect to the neighboring grid points (characterized by the d_{E_i}) in the same embedding column has to be known. For a proper determination of the surface inclination, one needs to know:

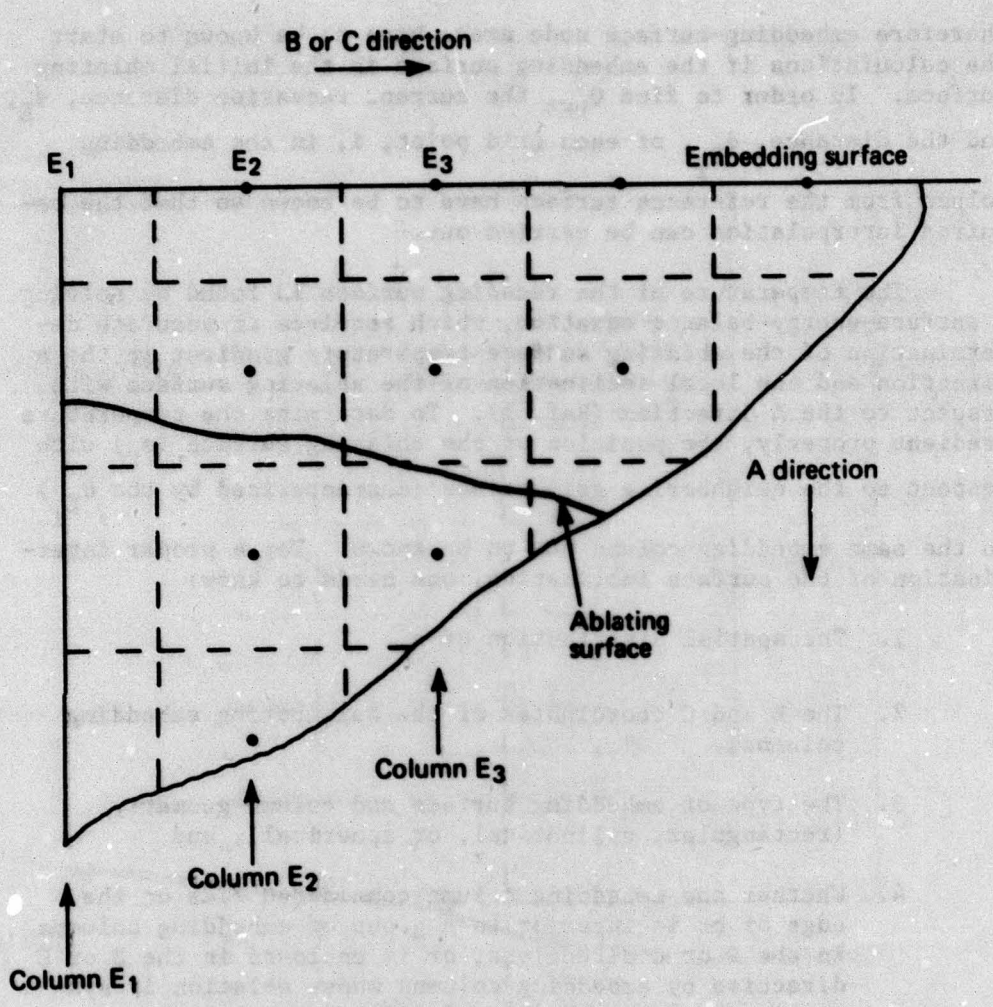
1. The spatial distribution of s_E,

2. The B and C coordinates of the neighboring embedding columns,

3. The type of embedding surface and column geometry (rectangular, cylindrical, or spherical), and

4. Whether the embedding column considered lies on the edge of or is interior to a group of embedding columns in the B or C directions, or is enclosed in the B or C direction by embedding columns whose ablation is symmetric about it. (Figure 2 illustrates the three situations.)

The required initial information described above has to be supplied by the user and a PL/I procedure has been written to facilitate the process. Once the information has been supplied, EM provides a way to perform rapid and accurate ablation computations. The EM procedures have been written in such a way that the initial embedding data need not be revised when a calculation is stopped and restarted. A provision for reading a second file of revised recession data created by a previous job has been incorporated into these procedures.



Column E_1 is an edge column if ablation is not symmetric about it
Column E_1 is a symmetry column if ablation is symmetric about it
Column E_2 is an interior column if $s_{E_1} \neq s_{E_3}$ in general

Fig. 2 Types of Embedding Column

3. APPROXIMATE FACTORIZATION

As stated in Section 2, internodal connections in the A direction, at least through the ablating material, are implicit, while internodal connections in the B and C direction may be either explicit or implicit. For any node that is treated implicitly in more than one coordinate direction, the method of AF (Ref. 1) is used in the procedures described here to solve the nodal-energy-balance equation. AF permits the decomposition of an N-dimensional implicit EBE into N one-dimensional EBE's. For example, if a node has implicit connections in all three coordinate directions, there are three AF equations: one in each of the directions, A, B, and C, respectively.

Applied to conventional heat conduction problems, the SHTP is designed to solve systems of lumped-parameter, nodal-energy-balance equations, each of which has the following very general form:

$$\left[\sum K_{ij}^{\ell} a_{ij}(T_j^{\ell} - T_i^{\ell}) + \sum_j K_{ij}^{\ell}(1 - a_{ij})(T_j^{\ell+1} - T_i^{\ell+1}) + q_i^{\ell+1}\right] \Delta t = C_i^{\ell}(T_i^{\ell+1} - T_i^{\ell}), \quad (1)$$

where T is temperature, i denotes the node of interest, j denotes a neighboring node having contact with node i, q_i^{ℓ} is the distributed internal heat generation in node i, K_{ij}^{ℓ} is a total conductance that can refer to external convection or radiation or internal conduction or radiation, C_i^{ℓ} is the node's thermal capacitance (volumetric specific heat times node volume), ℓ denotes time-level ℓ, t is time, $\Delta t = t_{\ell+1} - t_{\ell}$, and $0 \leq a_{ij} \leq 1$. Internodal links with $a_{ij} = 1$ are explicit, those with $a_{ij} = 0$ are implicit. Hybrid implicit-explicit links are also possible but will not be considered here. For brevity, the superscript ℓ will be dropped and $\ell + 1$ will be replaced by a + in most of the following discussion.

Internally, SHTP-E defines a vector $\underline{Q}$, whose $i^{\underline{th}}$ element is

$$Q_i = \sum_j K_{ij}(T_j - T_i) + q_i^+ \tag{2}$$

so that Eq. 1 becomes

$$\left[Q_i + \sum_j K_{ij}(T_j^+ - T_i^+)\right] \Delta t = C_i(T_i^+ - T_i). \tag{3}$$

The summation in Eq. 2 applies only to explicit links while that in Eq. 3 applies only to implicit links. Letting conductances for implicit links in the A, B, and C directions be given by K_{ij}^A, K_{ij}^B, and K_{ij}^C, Eq. 3 can be rearranged to give

$$\left[1 + \frac{\Delta t}{C_i}\left(\sum_j K_{ij}^A \Delta_{ij} + {}_j\, K_{ij}^B\, \Delta_{ij} + \sum_j K_{ij}^C \Delta_{ij}\right)\right] T_i^+ = T_i + \frac{Q_i \Delta t}{C_i}, \tag{4}$$

where Δ_{ij} is a difference operator such that

$$\Delta_{ij} T_i^+ = T_i^+ - T_j^+ . \tag{5}$$

In order to obtain accurate temperature predictions, Δt should be small, therefore Eq. 4 can be approximated by

$$(1 + \frac{\Delta t}{C_i}\sum_j K_{ij}^C \Delta_{ij})(1 + \frac{\Delta t}{C_i}\sum_j K_{ij}^B \Delta_{ij})\cdot(1 + \frac{\Delta t}{C_i}\sum_j K_{ij}^A \Delta_{ij})T_i^+ = T_i + \frac{Q_i \Delta t}{C_i} \tag{6}$$

Equation 6 is the approximate factorization of Eq. 4. Assuming consistent node definition, Eq. 6 remains consistent with the Fourier heat-conduction equation if the A, B, and C directions are mutually orthogonal.

By defining U_i and V_i as

$$U_i = (1 + \frac{\Delta t}{C_i}\sum_j K^A_{ij}\Delta_{ij})T^+_i \tag{7}$$

$$V_i = (1 + \frac{\Delta t}{C_i}\sum_j K^B_{ij}\Delta_{ij})U_i \ , \tag{8}$$

one can decompose Eq. 6 into the following three one-dimensional nodal EBE's:

$$\sum_j K^C_{ij}(V_j - V_i) + Q_i)\Delta t = C_i(V_i - T_i), \tag{9}$$

$$\sum_j K^B_{ij}(U_j - U_i)\Delta t = C_i(U_i - V_i), \tag{10}$$

and

$$\sum_j K^A_{ij}(T^+_j - T^+_i)\Delta t = C_i(T^+_i - U_i). \tag{11}$$

Equations 7 and 8 show that $\underline{U}$ and $\underline{V}$ give first approximations of $\underline{T}^+$ while Eqs. 9 through 11 show how AF can be used in a lumped-parameter interpretation applicable to SHTP-E. The AF B and C links also form columns, just as the A links form embedding columns. Equation 9 is applied to all nodes in the column for each AF C column. The resulting system of equations can be arranged tridiagonally and solved rapidly (Ref. 7) to give all the V_i values in that column, because all the T_i values are known beforehand. The process is repeated for all the AF C columns until all the elements in $\underline{V}$ are found. Similarly, knowing $\underline{V}$, $\underline{U}$ can be found by applying Eq. 10 to all of the nodes in all of the AF B columns, after which the application of Eq. 11 to all AF A (i.e., embedding) columns will give $\underline{T}^+$. If there are no implicit links in one or more of the three orthogonal directions, one or more of Eqs. 9 through 11 will become trivial, but the remaining equations will still be proper AF nodal EBE's.

Ref. 7. G. D. Smith, Numerical Solution of Partial Differential Equations, Oxford University Press, London, 1965.

The use of implicit internodal links has a beneficial effect on the time step, Δt, allowed by numerical stability considerations. Using Eq. 2, Eqs. 9, 10, and 11 can be arranged to give

$$V_i(C_i + \sum_j K_{ij}^C \Delta t) - \sum_j K_{ij}^C V_j \Delta t =$$

$$T_i(C_i - \sum_j K_{ij}^E \Delta t) + \sum_j K_{ij}^E \Delta T_j \Delta t + q_i^+ \Delta t, \tag{12}$$

$$U_i(C_i + \sum_j K_{ij}^B \Delta t) - \sum_j K_{ij}^B U_j \Delta t = C_i V_i, \tag{13}$$

and

$$T_i^+(C_i + \sum_j K_{ij}^A \Delta t) - \sum_j K_{ij}^A T_j^+ \Delta t = C_i U_i. \tag{14}$$

The superscript E in Eq. 12 signifies an explicit link. A practical stability criterion can be obtained from Eqs. 12 through 14 using matrix theory and the method of row or column norms (Ref. 8). Without going through the details, the matrix analysis indicates that the sums of the absolute values of the temperature coefficients on the left-hand sides of Eqs. 12 through 14, which contain unknown quantities, should be greater than or equal to the sums of the absolute values of the temperature coefficients on the right-hand side of those equations, which contain known quantities.

For brevity, the following definitions will be adopted in performing the stability analysis:

$$\sigma_{A,B,C} = C_i + 2\sum_j K_{ij}^{A,B,C} \Delta t, \tag{15}$$

which are the appropriate sums from the left-hand side of Eqs. 12 through 14. From the right-hand side of Eq. 12, there are two

Ref. 8. P. D. Lax and R. D. Richtmeyer, "Survey of the Stability of Linear Finite Difference Equations," Communications on Pure and Applied Mathematics, Vol. 9, May 1956, pp. 112-113.

possibilities for the sum, σ_E, which is needed:

$$\sigma_E = C_i, \text{ if } C_i - \sum_j K^E_{ij} \Delta t \leq 0 \tag{16}$$

and

$$\sigma_E = 2\sum_j K^E_{ij} \Delta t - C_i, \text{ if } C_i - \sum_j K^E_{ij} \Delta t \leq 0 \; . \tag{17}$$

The ultimate goal in solving Eqs. 12 through 14 is to find $\underline{T}^+$, knowing $\underline{T}$; therefore the stability analysis must apply to the collective stability of solving all three equations. The numbers under the inequality signs given below indicate the equation or inequality from which that particular step originates.

$$\sigma_C \underset{(12)}{\geq} \sigma_E \text{ or } 1 \geq \frac{\sigma_E}{\sigma_c} \; , \tag{18}$$

$$\sigma_B \underset{(13)}{\geq} C_i \underset{(18)}{\geq} C_i \frac{\sigma_E}{\sigma_C} \text{ or } 1 \geq \frac{C_i \sigma_E}{\sigma_C \sigma_B} \; , \tag{19}$$

and

$$\sigma_A \underset{(14)}{\geq} C_i \underset{(19)}{\geq} \frac{C_i^2 \sigma_E}{\sigma_C \sigma_B} \text{ or } 1 \geq \frac{C_i^2 \sigma_E}{\sigma_A \sigma_B \sigma_C} \; . \tag{20}$$

The final stability criterion is best expressed in terms of sums of conductances S_{T_i} and S_{E_i}, where

$$S_{I_i} = \sum_j K^A_{ij} + \sum_j K^B_{ij} = \sum_j K^C_{ij}, \; S_{E_i} = \sum_j K^E_{ij} \; . \tag{21}$$

Using Inequality 20 and ignoring $0(\Delta t^2)$ and $0(\Delta t^3)$ terms, the following conservative numerical stability criterion results:

$$C_i + 2S_{I_i} \Delta t \geq \sigma_E . \tag{22}$$

If σ_E is given by Eq. 16, Inequality 22 shows that

$$C_i + 2S_{I_i}\Delta t \geq C_i \ , \tag{23}$$

which is always true but, according to Eq. 16, is based on having $\Delta t \leq C_i/S_{E_i}$, the purely explicit limit. Equation 17 permits a higher value of Δt when applied to Inequality 22:

$$C_i + (S_{I_i} - S_{E_i})\Delta t \geq 0. \tag{24}$$

Inequality 24 is satisfied unconditionally if $S_{I_i} \geq S_{E_i}$, but for those nodes for which $S_{E_i} > S_{I_i}$

$$\Delta t \leq \min_i \frac{C_i}{S_{E_i} - S_{I_i}} \ . \tag{25}$$

If $S_{I_i} > 0$, the upper limit allowed by Inequality 25 is larger than C_i/S_{E_i}, the limit based only on the explicit links and $C_i/(S_{E_i} + S_{I_i})$, the limit for no implicit links.

4. COMBINING EMBEDDING AND APPROXIMATE FACTORIZATION

EM can be combined with AF by incorporating the embedding-surface boundary condition into Eq. 11, giving

$$\left[\sum_j K^A_{ij}(T^+_j - T^+_i) + \delta_{Ei} Q^+_{FE}\right] \Delta t = C_i(T_i - U_i), \tag{26}$$

where Q_{FE} is the fictitious heating rate applied to embedding node E (see Fig. 1), and

$$\delta_{Ei} = \begin{cases} 0, & E \neq i \\ 1, & E = i. \end{cases} \tag{27}$$

For each embedding column, Eq. 26 is applied to all nodes in the column to form a system of equations having the solution (Refs. 1 and 2)

$$T^+_i = T^+_{AE_i} + Q^+_{FE} T^+_{UE_i}, \tag{28}$$

where $T^+_{AE_i}$ and $T^+_{UE_i}$ are obtained as follows: $T^+_{AE_i}$ is found using $\underline{U}$ as initial data in Eq. 11 and no heating applied to the embedding node E, and it is called the adiabatic thermal response (Ref. 1). $T^+_{UE_i}$ is found by setting $\underline{U} = \underline{0}$ and applying a heat flow of 1 (in appropriate units, Btu/s, for example) to the embedding node E, and it is called the unit incremental thermal response (Ref. 1). Both T_{AE} and T_{UE} are therefore known vectors, but at this point, T^+_i and Q^+_{FE} in Eq. 28 are not known. The following discussion tells how to find Q^+_{FE} so that T^+_i can be found from Eq. 28.

The temperature, $T^+_{s_E}$ of the ablating surface associated with embedding surface node E, is, from Eq. 28

$$T^+_{s_E} = T^+_{AEs} + Q^+_{FE} T^+_{UEs} \quad . \tag{29}$$

Generally the ablating surface does not fall on a grid point and T^+_{AEs} and T^+_{UEs} have to be found by interpolation. For consistency, accuracy of third order or better is required. In the PL/I procedures described here, third-order Lagrange interpolation (Ref. 9) involving three successive grid points, I, J, and K in the embedding column near the ablating surface (see Fig. 1) is used:

$$T^+_{AEs} = \sum_{i=I,J,K} a_i T^+_{AE_i}, \tag{30}$$

where

$$a_i = \prod_{\substack{j=I,J,K \\ j \neq i}} \left(\frac{(s_E - d_{Ej})}{(d_{Ei} - d_{Ej})} \right). \tag{31}$$

At this point, Q_{FE} and T_{s_E} are unknown but related through Eq. 29. As shown below, T_{s_E} is found from the ablating surface energy balance equation (SEBE). Equation 29 will then give Q_{FE} after which Eq. 28 will give all the temperatures in the embedding column.

In general terms, the SEBE is

$$\mu_E q_{radin} + q_{chem} + q_{conv} - q_{radout} + q_{rem} + q_{cond} = 0. \tag{32}$$

Each q term is a heat flux. The terms q_{radin}, q_{chem}, q_{conv}, and q_{radout} are adequately described by Newman (Ref. 6). All of the q's are nonlinear functions of T_{s_E}.

Ref. 9. S. D. Conte, Elementary Numerical Analysis, McGraw-Hill Book Co., New York, 1965.

μ_E is the cosine of the angle between the normal to the ablating surface and the A direction, and in Ref. 1 it is shown that

$$\mu_E = \left[1 + (\nabla_{BC} s_E)^2\right]^{-1/2}, \tag{33}$$

where ∇_{BC} is the two-dimensional gradient operator in the BC plane. The components of $\nabla_{BC} s$ are found by differentiating a Lagrange interpolating polynomial to give second-order accurate estimates. Referring to Fig. 2, the equation for $\partial s_{E_1}/\partial B$, where E_1 is an edge column, is

$$\frac{\partial s_{E_1}}{\partial B} = b(E_1,E_2)s_{E_1} + b(E_1,E_2)s_{E_2} + b(E_1,E_3)s_{E_3}. \tag{34}$$

For $\partial s_{E_2}/\partial B$, where E_2 is an interior column, the proper equation is

$$\frac{\partial s_{E_2}}{\partial B} = b(E_2,E_1)s_{E_1} + b(E_2,E_2)s_{E_2} + b(E_2,E_3)s_{E_3}. \tag{35}$$

Finally, for a symmetry column

$$\frac{\partial s_E}{\partial B} = 0 . \tag{36}$$

Similar equations apply for $\partial s_E/\partial C$. The b coefficients are given by

$$b(E_i,E_j) = \left(2B_{E_i} - \sum_{\substack{k=1,2,3 \\ k\neq j}} B_{E_k}\right) \Bigg/ \prod_{\substack{k=1,2,3 \\ k\neq j}} (B_{E_j} - B_{E_k}) . \tag{37}$$

In Eq. 32, q_{cond} is the heat conducted to the ablating surface by the ablating solid and is given by (Ref. 1)

$$q_{cond} = -\left(\frac{k(T^+_{s_E})}{\mu_E}\right)\left(\frac{\partial T}{\partial A}\right)^+_{s_E}, \tag{38}$$

where $k(T^+_{s_E})$ is the ablating material's thermal conductivity evaluated at $T^+_{s_E}$ and the surface temperature gradient $(\partial T/\partial A)^+_{s_E}$ is evaluated in the A direction, pointing into the ablating material. A second-order accurate estimate of $(\partial T/\partial A)^+_{s_E}$ is

$$\left(\frac{\partial T}{\partial A}\right)^+_{s_E} = \sum_{i=I,J,K} c_i T^+_i , \tag{39}$$

where I, J, and K are defined in Fig. 1 and

$$c_i = (2s_E - \sum_{\substack{j=I,J,K \\ j\neq i}} d_{Ej}) \Bigg/ \prod_{\substack{j=I,J,K \\ j\neq i}} (d_{Ei} - d_{Ej}) . \tag{40}$$

By combining Eq. 28, 29, and 39, one can show that Eq. 38 becomes

$$q_{cond} = -\left(\frac{k(T^+_{s_E})}{\mu_E}\right)\left(\frac{-\nu_{UE}}{T_{UE_s}}\right)\left(T^+_{AEs} - T^+_{UEs}\ \frac{\nu_{AE}}{\nu_{UE}} - T^+_{s_E}\right), \tag{41}$$

where

$$\underset{U}{\nu_{AE}} = \sum_{\ell=I,J,K} c_\ell \underset{U}{T^+_{AE\ell}} . \tag{42}$$

$-T^+_{UEs}/\nu_{UE}$ is positive and has the units of length, $T^+_{AEs} - T^+_{UEs}(\nu_{AE}/\nu_{UE})$ is a temperature, and both quantities can be calculated before the SEBE is solved. Letting L^*_E and T^*_E be given by

$$L^*_E = -\frac{T^+_{UEs}}{\nu_{UE}}$$

and

$$T^*_E = T^+_{AEs} - T^+_{UEs}\left(\frac{\nu_{AE}}{\nu_{UE}}\right), \text{ respectively,} \tag{43}$$

Eq. 40 becomes

$$q_{cond} = \left(\frac{k(T^+_{s_E})}{\mu_E}\right)\left(\frac{1}{L^*_E}\right)(T^*_E - T^+_{s_E}) \quad . \tag{44}$$

The quantity q_{rem} is the energy flux loss by the ablating material, owing to the removal process itself. In terms of mass flux $\dot{m}_E$ (Ref. 6), specific heat, c, and average temperature change, ΔT, of material ablated during time Δt, q_{rem} is

$$q_{rem} = \dot{m}_E c \Delta T \quad . \tag{45}$$

The material loss covers a distance $\dot{m}_E \Delta t/\rho$, where ρ is the ablating material's density and Δt is the time step covered by the calculation. Therefore near the ablating surface ΔT is given approximately by

$$\Delta T = \left(\frac{\dot{m}_E \Delta t}{2\rho}\right)\left(\frac{1}{\mu_E}\right)\left(\frac{\partial T}{\partial A}\right)^+_{s_E} \quad . \tag{46}$$

Combining Eqs. 45 and 46 shows that

$$q_{rem} = \left(\frac{\dot{m}_E^2}{2\rho^2}\right)(\rho c)\left(\frac{(T_E^* - T_{s_E}^+)}{\mu_E L_E^*}\right)\Delta t. \tag{47}$$

The (ρc) has been inserted because SHTP-E uses (ρc) rather than c as a basic thermal property.

By defining the first five terms in Eq. 32 as q_{EXT}, that equation becomes

$$q_{EXT}(T_{s_E}^+) + q_{cond}(T_{s_E}^+) = 0. \tag{48}$$

Equation 48 is used in the SHTP-E procedures if the embedding and ablating surfaces do not coincide, which is the general case. Many problems can be started with the ablating surface lying initially on the embedding surface, and so long as the condition persists,

$$Q_{FE} = q_{EXT}(A_E), \tag{49}$$

where A_E is the exposed surface area of embedding surface node E. From Eqs. 28 and 49, the proper surface-temperature equation in this case is

$$T_E^+ = T_{AEE}^+ + q_{EXT}(T_E^+)T_{UEE}^+ . \tag{50}$$

Equation 48 or Eq. 50, whichever is appropriate, is solved by Newton-Raphson iteration for $T_{s_E}^+$ or T_E^+. To begin the process, an error function is defined as

$$\epsilon(T_{s_E}^+) = T_E^+ - T_{AEE}^+ - q_{EXT}(A_E)T_{EUU}^+ \tag{51}$$

if the ablating and embedding surfaces coincide, or

$$\varepsilon(T_{s_E}^{+}) = q_{EXT}\,(T_{s_E}^{+}) + q_{cond}\,(T_{s_E}^{+}) \tag{52}$$

if they do not. In either case, one seeks to find a number T that is a solution of

$$\varepsilon(T) = 0. \tag{53}$$

First, a rough search is made for a solution using

$$T^{n+1} = T^{n} - \Delta T\ \mathrm{sgn}\left(\frac{\varepsilon(T^{n})}{\varepsilon'(T^{n})}\right), \tag{54}$$

where n and n+1 denote successive iterations at the new time level and

$$\mathrm{sgn}\,(x) = \begin{cases} 1 \text{ if } x > 0 \\ 0 \text{ if } x = 0 \\ -1 \text{ if } x < 0 \end{cases} \tag{55}$$

The initial value of ΔT is 1000°R and an initial guess for T is made from initial temperature data or the ablating surface temperature at the previous time level. Whenever the solution point is passed, ΔT is reduced by a factor of 10 until

$$\left|\frac{\varepsilon(T^{n})}{\varepsilon'(T^{n})}\right| < \Delta T, \tag{56}$$

after which Newton-Raphson iteration,

$$T^{n+1} = T^{n} - \frac{\varepsilon(T^{n})}{\varepsilon'(T^{n})}, \tag{57}$$

is invoked. If the solution starts to diverge, Eq. 54 is brought back into play until Inequality 56 is satisfied, after which Eq. 57

s used again. Tolerance limits are set on $|\epsilon(T^n)|$, T^n itself, and the maximum number of iterations. Successful convergence is assumed if $|\epsilon(T^n)|$ and $|T^n - T^{n-1}|$ fall within their tolerance limits in a number of iterations less than, or equal to, the maximum number allowed. If this process leads to $T < 0°R$ or $T > 15\,000°R$, solution failure is assumed and the program execution is stopped.

If the embedding and ablating surfaces coincide, Eqs. 28 and 49 are used to find all the T_i^+ in the embedding column after T_E^+ has been found as described above. If these two surfaces do not coincide, Eqs. 28 and 29 are used to obtain

$$T_i^+ = T_{AEi}^+ + (T_{s_E}^+ - T_{AEs}^+)\left(\frac{T_{UEi}^+}{T_{UEs}^+}\right), \tag{58}$$

from which all the T_i^+ in the embedding column are found after $T_{s_E}^+$ has been found from Eq. 57.

Before computing the surface temperature at time level $\ell+1$, the A-direction recession distance $s_E^{\ell+1}$ is estimated from

$$s_E^{\ell+1} = s_E^{\ell} + \left(\frac{\partial s_E}{\partial t}\right)^{\ell} \Delta t \tag{59}$$

and the SEBE is evaluated and solved for $T_{s_E}^{\ell+1}$ as described above. During this process the mass flux loss $\dot{m}_E^{\ell+1}$ is also determined the A-direction recession rate then can be found from

$$\left(\frac{\partial s_E}{\partial t}\right)^{\ell+1} = \frac{\dot{m}_E^{\ell+1}}{\rho \mu_E^{\ell}} \quad . \tag{60}$$

If desired, the estimate of $s_E^{\ell+1}$ can be refined iteratively by using the relation

$$s_E^{\ell+1} = s_E^{\ell} + \left[G\left(\frac{\partial s_E}{\partial t}\right)^{\ell+1} + (1 - G)\left(\frac{\partial s}{\partial t}\right)^{\ell} \right] \Delta t, \qquad (61)$$

where G is an iteration parameter and $(\partial s_E/\partial t)^{\ell+1}$ is found from Eq. 60 on the previous iteration. As shown in Refs. 1 and 2, this process is fast because it does not require re-evaluation of $\underline{T_{AE}^{+}}$ and $\underline{T_{UE}^{+}}$.

5. PROCESSING ABLATING-SURFACE TEMPERATURE AND PRESSURE-DEPENDENT INFORMATION

As shown in Ref. 6, the individual ablating-surface heating terms in the SEBE (Eq. 32) depend on the ablating-surface temperature and the temperature and pressure of the gas flowing over the ablating surface. The purpose of Section 5 is to show how the ablating-surface quantities, which are influenced by pressure and ablating-surface temperature, are treated by SHTP-E procedures in those cases where this treatment differs from that in previous SHTP ablation procedures (Ref. 6).

The computation of ablating-surface characteristics depends on data read from surface thermochemical tables (Ref. 6). These data are read by one SHTP-E procedure and used later by another. The nature of the data depends on whether or not the Lewis number, Le, of the gas flowing over the ablating surface is unity. In either case, the independent variables for the tables read are gas pressure (p) and the ablation-to-diffusion mass-loss ratio (β). The dependent variables always include the ablating-surface temperature and the gas enthalpy evaluated at this temperature. If $Le \neq 1$, the gas boundary-layer-edge enthalpy and a diffusive chemical-heating term are also read. These data are modified so that π and b, where

$$\pi = \ln (p), \text{ and } b = \ln (\beta) \tag{62}$$

become the independent variables, and a modified chemical heating term (Ref. 6) is added to the tables. Letting x represent any one of the dependent variables, π_i denote a logarithmic pressure table entry, and b_j a logarithmic mass-loss-ratio table entry,

$$x_{ij} = x(\pi_i, b_j). \tag{63}$$

Previous SHTP ablation procedures (Ref. 6) reprocess the data read so that π and ablating-surface temperature become the independent variables, since it is desired to solve the SEBE by iterating with the ablating-surface temperature. During the reprocessing a uniform ablating-surface temperature increment is used in the new tables, which provides poor resolution for the sublimation

ablation regime and unnecessarily fine resolution in the diffusion-limited ablation regime. Fortunately, the raw surface thermochemical data are properly resolved, and the following discussion describes how it can be taken advantage of.

For each embedding column at a particular time level, the SEBE is solved for a constant surrounding gas pressure that is known beforehand. By interpolating on π, a one-dimensional table for the values of $x(\pi,b_j)$ for all values of j can be set up by letting

$$x(\pi,b_j) = x_{ij} + \frac{(\pi - \pi_i)}{(\pi_{i+1} - \pi_i)} (x_{i+1,j} - x_{i,j}), \tag{64}$$

where

$$\pi_i \leq \pi < \pi_{i+1}. \tag{65}$$

Applying Eq. 64 to all of the dependent variables gives a table that coordinates the ablating-surface thermochemical data for the current pressure. The choice of independent variable in this table is arbitrary, but, as shown above, the most convenient choice for present purposes is the ablating-surface temperature.

The constant-pressure-surface thermochemical table just described contains data that are not valid in the diffusion-kinetic ablation regime (Ref. 10). For Le = 1, Ref. 10 describes a method of modification that can be used to insert valid diffusion kinetic values of b_j, ablating-surface gas enthalpy, and the modified chemical-heating term into the table for those tabular ablating-surface temperatures that fall in the diffusion kinetic regime. This modification has been incorporated into the SHTP-E ablation procedure. To date, no modification is available for Le $\neq$ 1.

At this point, one has a well-resolved constant-pressure surface thermochemical table (STT) that is useful in evaluating terms in the SEBE so that the ablating-surface temperature, T_{s_E}, may be found. Section 4 describes how the evaluation is done using Newton-Raphson iteration, which requires the evaluation of first-

Ref. 10. L. L. Perini, "Graphite Oxidation Modelling in Subsonic Flow," APL/JHU ANSP-111, 9 Aug 1977.

derivative of all terms in the SEBE with respect to ablating-surface temperature. While previous versions of the SHTP's ablation procedure properly differentiate most of the SEBE's terms with respect to ablating-surface temperature, those terms involving the blowing corrections to the heat- and mass-transfer Stanton numbers (Ref. 6) have been assumed to be constant in the iteration process. It is valid to do this for low-temperature diffusion-limited ablation, but in the high-temperature sublimation regime, the blowing-corrected Stanton-number variation should be taken into account.

The heat- and mass-transfer Stanton numbers are denoted here by C_H and C_m and their corrections for blowing are C_{Hb} and C_{mb}. The C_{Hb} and C_{mb} terms depend on a modified mass-loss ratio β_o, defined as:

$$\beta_o = \left(\frac{C_{mb}}{C_H}\right)\beta \tag{66}$$

The funtional relationships describing $C_{Hb}(\beta_o)$ and $C_{mb}(\beta_o)$ have been the subject of many experimental and theoretical investigations. Two of the better correlations are based on work by Mickley et al. (Ref. 11) (called the Mickley-Spalding correlation) and Putz and Bartlett (Ref. 12).

The Mickley-Spalding correlation equations are

$$C_{Hb} = \frac{C_H\, 2\lambda\beta_o}{\exp(2\lambda\beta_o)-1} \tag{67}$$

and

$$C_{mb} = \left(\frac{C_m}{C_H}\right) C_{Hb}. \tag{68}$$

Ref. 11. H. S. Mickley, R. C. Ross, A. L. Squyers, and W. E. Stewart, Heat, Mass, and Momentum Transfer for Flow Over a Flat Plate with Blowing or Suction, NACA TN 3208, Jul 1954.

Ref. 12. K. E. Putz and E. P. Bartlett, "Heat Transfer and Ablation Rate Correlations for Reentry Heat Shield and Nose Tip Applications," AIAA Preprint No. 72-91, Jan 1972.

The λ and the Stanton-number ratio (C_m/C_H) are assumed to be constants. Reference 13 recommends $\lambda = 0.4$ for turbulent flow and $\lambda = 0.5$ for laminar flow over the ablating body. C_H and C_m do not depend on the ablating-surface temperature, but in light of Eq. 66, which contains β, C_{Hb}, and C_{mb}, they are functions of ablating-surface temperature and pressure. By combining Eqs. 66, 67, and 68, one can show that

$$\beta_o = \frac{1}{2\lambda}\left[\ln\left(1 + 2\lambda\beta\frac{C_m}{C_H}\right)\right] . \tag{69}$$

Equation 69 can be used to find $d\beta_o/d\beta$, and the STT constructed from Eq. 64 can be used to find $d\beta/dT_s$, where T_s is the STT's ablating-surface-temperature variable. The $dC_{Hb}/d\beta_o$ and $dC_{mb}/d\beta_o$ can be found easily from Eqs. 67 and 68, which, after using the chain rule, lead to

$$\frac{dC_{Hb}}{dT_s} = \left(\frac{dC_{Hb}}{d\beta_o}\right)\left(\frac{d\beta_o}{d\beta}\right)\left(\frac{d\beta}{dT_s}\right) \tag{68}$$

and

$$\frac{dC_{mb}}{dT_s} = \left(\frac{dC_{mb}}{d\beta_o}\right)\left(\frac{d\beta_o}{d\beta}\right)\left(\frac{d\beta}{dT_s}\right) . \tag{69}$$

The Putz and Bartlett correlation equations have the forms

$$C_{Hb} = C_H F(\beta_o) \tag{70}$$

and

$$C_{mb} = C_H \frac{2\lambda\beta_o}{\exp(2\lambda\beta_o)-1} , \tag{71}$$

Ref. 13. "User's Manual Aerotherm Charring Material Thermal Response and Ablation Program," Version 3, Aerotherm Corporation, Mountain View, CA, Report UM-70-14, Apr 1970.

where

$$\lambda = \lambda(\beta_o) \quad . \tag{72}$$

The functional relationships for $F(\beta_o)$ and $\lambda(\beta_o)$ are polynomials and are given in Ref. 6. The combination of Eqs. 66 and 71 yields the relation

$$\beta = \frac{\exp(2\lambda\beta_o)-1}{2\lambda} \quad . \tag{73}$$

Equation 73 gives a simple relationship for finding β given β_o, when in fact, one desires β_o, given β. The latter relationship is obtainable by assuming a set of several values of β_o and setting up a table of β and β_o values, using Eqs. 72 and 73. In the SHTP-E ablation procedure, a table coordinating β, β_o, and λ is set up before any ablation calculations are performed. With a value of $T_{s_E}^+$ obtained during the iteration process, one can find β and $d\beta/dT_s$ from the STT, and β_o, λ, $d\beta_o/d\beta$, and $d\lambda/d\beta_o$ from the β-β_o-λ table. The $dC_{Hb}/d\beta_o$ and $dC_{mb}/d\beta_o$ can then be found from Eqs. 70 and 71, and finally Eqs. 68 and 69 will give dC_{Hb}/dT_s and dC_{mb}/dT_s at $T_s = T_{s_E}^+$.

To find the heat loss due to material removal (q_{rem}) in Eq. 32, one needs to know the local mass-flux-loss, $\dot{m}_E$, for the ablating surface in the embedding column with embedding-surface node E. The $\dot{m}_E$ has a thermochemical component, $\dot{m}_{ETC}$, and a mechanical erosion component $\dot{m}_{EMC}$, so that

$$\dot{m}_E = \dot{m}_{ETC} + \dot{m}_{EMC} \tag{74}$$

Reference 14 gives the ablating-surface temperature and surrounding gas-pressure ranges in which mechanical erosion is significant and shows that for graphite

$$\dot{m}_{EMC} = \dot{m}_{ETC} H(T_{s_E}) \quad . \tag{75}$$

For the Mickley-Spalding correlations, it is assumed that $H(T_{s_E}) = 0$, but for the Putz and Bartlett correlation, the functional form of $H(T_{s_E})$ given by Ref. 14 is used. $\dot{m}_{ETC}$ is expressible in terms of the local surrounding gas boundary-layer edge-mass flux $\dot{m}_{GE}$, taken parallel to the ablating surface, as

$$\dot{m}_{ETC} = \beta \dot{m}_{GE} C_{mb} \quad . \tag{76}$$

Therefore

$$\dot{m}_E = \beta \dot{m}_{GE} C_{mb} \left[1 + H(T_{s_E}) \right] \quad , \tag{77}$$

and

$$d\dot{m}_E / dT_s \text{ at } T_S = T_{s_E}$$

can be found directly from Eq. 77 using the STT Eqs. 71 and 69, and, when appropriate, the relationship for $H(T_s)$ given by Ref. 14.

Ref. 14. L. L. Perini, "Review of Graphite Ablation Theory and Experimental Data," APL/JHU ANSP-M-1, Dec 1971.

6. GENERAL SOLUTION PROCEDURE SUMMARY

The general-solution algorithm is summarized as follows.

For each node (e.g., node i):

1. Follow the consistent node definition rules given in Ref. 1. Nodal control surfaces should be located halfway between, and normal to, the line connecting two adjacent nodal grid points, regardless of the geometry used.

2. Specify an initial temperature.

3. Determine the nodal capacitance C_i, which is the volumetric specific heat $(\rho c)_i$ multiplied by the volume of node i.

4. Determine internodal conductances K_{ij} (j = neighboring node), basing thermal properties on old temperatures (those at the beginning of the time step). A record is kept of whether K_{ij} pertains to an implicit or explicit connection. If the connection is implicit, a record of the directionality of the connection is also kept.

5. Accumulate the sum of explicit conductances, S_{E_i} and the sum of implicit conductances S_{I_i}.

6. Determine the explicit heating rate contribution Q_i. This includes internal heat sources, external heat flows treated explicity, and explicit internal heat flow based on explicit internodal conductances.

For all the nodes:

1. Find the minimum stability time step, Δt. This is done by comparing S_{E_i} and S_{I_i} for each node. From Inequalities 24 and 25, if $S_{I_i} \geq S_{E_i}$ there is no time step

restriction, but for nodes that have $S_{E_i} > S_{I_i}$

$$\Delta t = 0.9 \min_i \frac{C_i}{S_{E_i} - S_{I_i}}$$

or a preset maximum value (whichever is smaller) is used.

2. Determine the new temperatures of nodes with no implicit connections:

$$T_i^+ = T_i + \frac{Q_i \Delta t}{C_i}$$

3. Estimate the temperatures of nodes with B- and C-directed AF connections.

For each embedding column:

1. Estimate the A-direction recession distance, using the recession rate of the previous time level.
2. Determine the ablating-surface temperature.
3. Compute the fictitious heating rate and find all the temperatures in the embedding column.
4. Find the new recession rate in the A direction.
5. (Optional) Refine the estimate of the recession distance and repeat steps 2 through 4.

7. PL/I PROCEDURE DESCRIPTION

As usual with the SHTP (Refs. 4 and 5), the PL/I procedures described here fall into two general categories, initialization and repetitive thermal-analysis computation. The user writes a problem-oriented main program that calls the procedures described here, as they are needed. The initialization procedures have to precede the thermal-analysis procedures. Within each group, some procedures have to precede others and the calling order will be indicated as the procedures are described.

The initialization procedures are designed to aid the user in supplying the initial data needed to perform a satisfactory thermal analysis. The procedures perform the following general tasks:

1. Allocating storage for external identifiers;
2. Specifying starting and stopping times;
3. Specifying initial temperature data;
4. Reading node geometry information, including quantitive geometric characteristics as well as the relationship of each node to other nodes with which it has direct links;
5. Reading embedding-surface and column-geometry information;
6. Reading trajectory data;
7. Reading thermal-property data; and
8. Reading surface thermochemical data.

The procedures for thermal-analysis computation form a loop and must follow all of the initialization procedures. The procedures are designed to help the user perform the actual repetitive calculations necessary to determine a thermal-response history of an ablating material by carrying out the following tasks.

1. Finding node capacitances and explicit and implicit internodal conductances;

2. Finding the time step to be used;

3. Determining temperatures of nodes with no implicit connections;

4. Making approximate factorization estimates of node temperatures for nodes with implicit B- or C-direction connections;

5. Applying the embedding technique to each embedding column;

6. Printing temperature distributions and ablating-surface information at selected time levels; and

7. Storing data in data sets for future use such as plotting, restarting the calculation from its latest time level, and saving data for thermal-stress calculations.

The individual procedures are described on the following pages. Each description begins with the appropriate CALL statement, followed by a list of the arguments of each procedure, the attribute of each argument, and a brief description of its purpose. The argument list is followed by a clarifying discussion.

INITIALIZATION PROCEDURES

The initialization procedures follow, each procedure being started on a page by itself to ensure clarity. These procedures are STØRE, SET, READCPE, APFAC, READSU, READTM, READPR, and READABN.

STØRE

CALL STØRE (#CAPS, XCAPLIM, CØDE);

#CAPS	FIXED BIN.	The number of nodes in the FD model.
XCAPLIM	FIXED BIN.	The number of nodes plus the number of indirect addresses to nodes (see discussion of READCPE below), or a number at least as large as the largest indirect address.
CØDE	FIXED BIN.	A code indicating the type of analysis being performed. For EM, CØDE = 5.

STØRE dynamically allocates and initializes to zero-controlled external identifiers used by several of the other procedures. STØRE must precede SET, READCPE, and READSU.

SET

CALL SET (TSTART, TSTØP, TINIT, ØTHER, IØTHER, TØTHER);

TSTART	FLØAT BIN.	The time, in seconds, for thermal-analysis computations to begin.
TSTØP	FLØAT BIN.	The time, in seconds, for thermal-analysis computations to end.
TINIT	FLØAT BIN.	The initial temperature, in °R, of all nodes in the FD system.
ØTHER	FIXED BIN.	The number of nodes whose initial temperatures are not to be TINIT.
IØTHER	(*) FIXED BIN.	An array of ØTHER elements specifying the nodes whose initial temperatures are not to be TINIT.
TØTHER	(*) FLØAT BIN.	An array of ØTHER elements specifying the temperatures, in °R, of the nodes whose initial temperatures are not to be TINIT.

SET, which must follow STØRE, initializes all node temperatures in °R, and specifies the starting and ending times, in seconds, for the thermal-analysis computations. It is identical to the SET procedure described in Refs. 4 and 6.

Using T as the temperature array, SET first sets all temperatures equal to TINIT using T = TINIT:. It then overides this specification for some of the nodes using T(IØTHER(I));.

READCPE

CALL READCPE (GEØFILE);

GEØFILE	FILE.	The name of the file containing geometric data about the nodes. READCPE does not open or close GEØFILE.

READCPE is similar to READCP (Refs. 4 and 6), but contains features relating to EM and AF. Its purpose is to read and store the contents of GEØFILE, which contains data characterizing each node's geometry and its relationship to other nodes.

The first card in GEØFILE should contain a descriptive title of the file's contents; the second card should have title headings for the remaining cards, which require the following information.

COLUMNS	DATA DESCRIPTION
1-4	Fixed format (no decimals). A node number (direct address) or an indirect address (greater than the largest node number). For further discussion, see "Direct and Indirect Addressing."
10-19	E format. The node volume (ft^3) for directly addressed information or 0.E0 for indirectly addressed information.
20-29	F format (decimal required). Material code. This can vary from 1. to 999999. For further discussion see "Material Code Conventions."
30-39	F format (decimal required). Number of neighboring nodes with directly addressed connections, if equal to 0., 1., or 2.; reference to indirect address if greater than two.
40-49	F format (decimal required). First neighboring node. If positive, this signifies a conductive link or a contact link with another material. If negative, it signifies a radiation link. If an A, B, or C appears in this field to the left of the numerical entry, a type A, B, or C implicit link is signified. If only

the numerical entry appears, the link is explicit. For more discussion of the use of the A, B, and C codes, see "Direct and Indirect Addressing."

50-59 — E format. The geometric part of the conductance to the first neighboring node. For a conductive link, this field gives the internodal contact area in square feet divided by the distance in feet between the grid points of the node of interest and its first neighbor. For a contact or radiation link, it gives the appropriate area in square feet.

60-69, 70-79 — These entries have the same significance for the second neighboring node that the entries in columns 40-59 have for the first neighboring node. They are used if the entry in columns 20-29 is greater than 99, or if the entry in columns 30-39 is greater than or equal to two. Zero or blank entries are permissible.

The last card in GEØFILE should be a blank card. READCPE must precede APFAC and follow STØRE. The numerical entries in the seven fields in columns 10-79 are stored in an external identifier called XCAP. For node I, XCAP (I,1) is the node volume, etc. XCAP's declaration is

```
DCL XCAP (*,7) FLØAT BIN CTL EXT;
```

Users may find this information useful when defining distributed heat-source terms that require node volume.

Direct and Indirect Addressing

If N is the number of nodes in the FD model, the entry in columns 1-4 is I, and I ≤ N, then I is a node number and the XCAP information on the card pertains directly to node I. This is called direct addressing. If the entry in columns 30-39 is M and M > N, then further XCAP information is expected on a card containing M in columns 1-4. XCAP (M,1) should be zero, but the remaining entries, XCAP (M,2) - XCAP (M,7) will contain more information about node I. This is called indirect addressing. If necessary, indirect addresses can be used to access more information on other indirectly addressed cards. Indirect addressing is convenient for connecting more than two neighbors to a node and for specifying parallel heat-transfer links, such as gas conduction and radiation across a gap.

Type A (i.e., the A direction in Fig. 1) implicit internodal connections must be initiated by direct addressing but indirect addressing can be used to construct parallel connections. Type B or C implicit connections can be initiated by either direct or indirect addressing. Regardless of direct or indirect addressing, implicit links must always have an A, B, or C code, even if there are parallel connections. For each node that has implicit links, A, B, and C neighbors can be addressed simultaneously, but not more than one neighbor of each type can be addressed directly or indirectly. Parallel implicit connections have to be made on separate cards.

As is the case with READCP (Refs. 4 and 6), nodal connections should be specified in only one direction in GEØFILE. If node 69 has been designated as node 1's neighbor in the table, the latter is understood when node 69 is identified as node 1's neighbor.

Material Code Conventions

Table 1 describes the material code conventions used in the SHTP-E procedures. I is the node of interest, N1 and N2 are its directly or indirectly addressed first and second neighbors in columns 40-49 and 60-69, and a, b, c, d, e, and f are any of the single digits (0-9). The table describes the general convention and gives some examples.

As indicated in the discussion under READPR, each individual material has a single material code that can vary from 1 to 99. Gaps and contacts, as well as materials with finite volume, can be treated as individual materials. For materials with finite volume and directional conductivities, separate codes can be defined for each direction. In contrast to READCP (Refs. 4 and 6), the limit on the number of materials is 99 instead of 9 and no special material code convention is adopted for contacts. The potentially harmful effect of high contact conductance on the allowable time step can be eliminated by making the contact link implicit.

Table 1

Material Code Conventions Used in SHTP-E

Material Code	I's Material	Material Connecting I to:	
		N1	N2
ab.	ab	ab	ab
abcd.	ab	ab	cd
abcdef.	ab	cd	ef
2.	2	2	2
10.	10	10	10
105.	1	1	5
1105.	11	11	5
1125.	11	11	25
10203.	1	2	3
10201.	1	2	1
11213.	1	12	13
111213.	11	12	13

APFAC

CALL APFAC (CØDE, APFILE);

CØDE	CHAR(*).	CØDE must be 'B' or 'C'. It is used to indicate whether the contents of APFILE refer to implicit internodal links in the B or C direction.
APFILE	FILE.	The contents of APFILE contain information about the starting nodes in type B or C AF columns. APFAC opens and closes APFILE.

APFAC defines each AF column (see discussion following Eq. 11 in Section 3) in a way that can be understood by the appropriate procedures in the thermal-analysis computation loop. READCPE only reads information that may result in implicit B or C AF internodal links; APFAC finalizes those links.

The first card in APFILE is a title card; the remaining cards contain a list of the node numbers of the starting nodes in the AF columns desired. These entries are LIST controlled, i.e., format free, and the numbers read should be first nodes in the AF columns read by READCPE. If APFAC does not define an AF column that has been read by READCPE, the corresponding internodal links are assumed to be explicit, even though they have been tagged as implicit by a B or C in READCPE. This is accomplished by not entering the starting node number in APFILE. If APFILE is undefined (no DD card), empty, or contains no starting node numbers, there are no AF links in the CØDE direction.

There are also various checks built into APFAC. Program execution will be terminated if CØDE is not 'B' or 'C' or if an attempt is made to cross an AF column with itself or another AF column of the same type. However, cyclic AF columns, in which the last node in the column is implicitly connected to the first node, are permitted. These could occur in the angular direction around a sphere or cylinder. APFAC also checks the AF column structure read by READCPE to make sure that no more than one implicit B or C neighbor is addressed directly or indirectly by each node that has AF connections. The AF columns generated by APFAC are printed out so that the user can check them to be sure they are constructed as desired.

APFAC must follow READCPE.

READSU

CALL READSU (NBØDY, NNPERS, SRFDATA);

NBØDY	FIXED BIN.	The total number of embedding surface body stations. A body station is a collection of embedding surface nodes whose associated ablating surfaces are exposed to the same external environment.
NNPERS	(*) FIXED BIN.	An array of NBØDY numbers giving the number of embedding surface nodes in each body station. NNPERS(I) = J means that there are J embedding surface nodes in the I^{th} body station.
SRFDATA	FILE.	The file containing embedding and initial receding surface geometry information. This file should <u>not</u> be called SURFACE. READSU does not open or close SRFDATA.

READSU is used to read embedding-surface data, column-geometrical data, and related initial ablating-surface data. The file SRFDATA is composed of cards that have to be arranged as described below.

Cards 1-6: Text describing information on the following cards, e.g., comments, titles, and column headings.

Card 7: An asterisk in column 1.

Card 8: The name of a file (starting in column 1) containing the distances in inches of the grid points in the embedding column in the ablating material from the fixed reference surface in the A direction, i.e., the d's in Fig. 1. This file should contain nothing but the d's in ascending order. The input data are format free (LIST controlled). The d's given should not go beyond the back face of the ablating material, even if the embedding column does.

Card 9: A code indicating the type of geometry through which the embedding columns pass. This code should start in column 1. The following four codes are acceptable:

RECT: Rectangular, or Cartesian, geometry. The embedding columns can run in any one of the three coordinate directions.

CYLZ: Cylindrical geometry. Embedding column runs in the axial direction.

CYLR: Cylindrical geometry. Embedding column runs in the negative radial direction.

SPH: Spherical geometry. Embedding column runs in the negative radial direction.

The next several cards contain information about embedding surface nodes. The first two columns of each card starting a new data group contain information about the nature of the embedding surface node as viewed in the B and C directions, the first pertaining to the B direction and the second to the C direction. Each data group describes a single embedding surface node. An S in column 1 indicates that the embedding surface node's column is a symmetry column in the B direction, as it is defined in Fig. 2. An E in column 1 means that the embedding surface node's column is an edge column when viewed in the B direction. A blank or any character other than S or E in column 1 means that the embedding surface node's column is an interior column when viewed in the B direction. Similar remarks apply about C direction codes in column 2. The remaining eleven input items are LIST directed and may appear on the first card in the data group or following cards in the same data group. These items have the roles described below and must appear in the order indicated.

1. Embedding surface node number. If this number is positive, printing of the temperatures of the nodes in the fictitious ablating material (see Fig. 1) is suppressed; if it is negative, temperatures for these nodes are printed.

2. Body station number.

3. Node number within the body station.

4. Embedding surface area (ft^2).

5. A coordinate of the embedding surface node.

6. B coordinate of the embedding surface node.

7. C coordinate of the embedding surface node.

8. Node number of the adjacent embedding surface node in the B direction.

9. Node number of the adjacent embedding surface node in the C direction.

10. The initial distance, in inches, of the ablating surface in the embedding column from the reference surface in the A direction, i.e., the initial value of s_E in Fig. 1.

11. The distance, in inches, of the embedding surface node's grid point from the reference surface in the A direction, i.e., e in Fig. 1.

If embedding surface node 69 is the third node in the eighth body station, the first three items described above should be 69 8 3.

If there are no B- or no C-direction items, zeroes should be entered in the appropriate places.

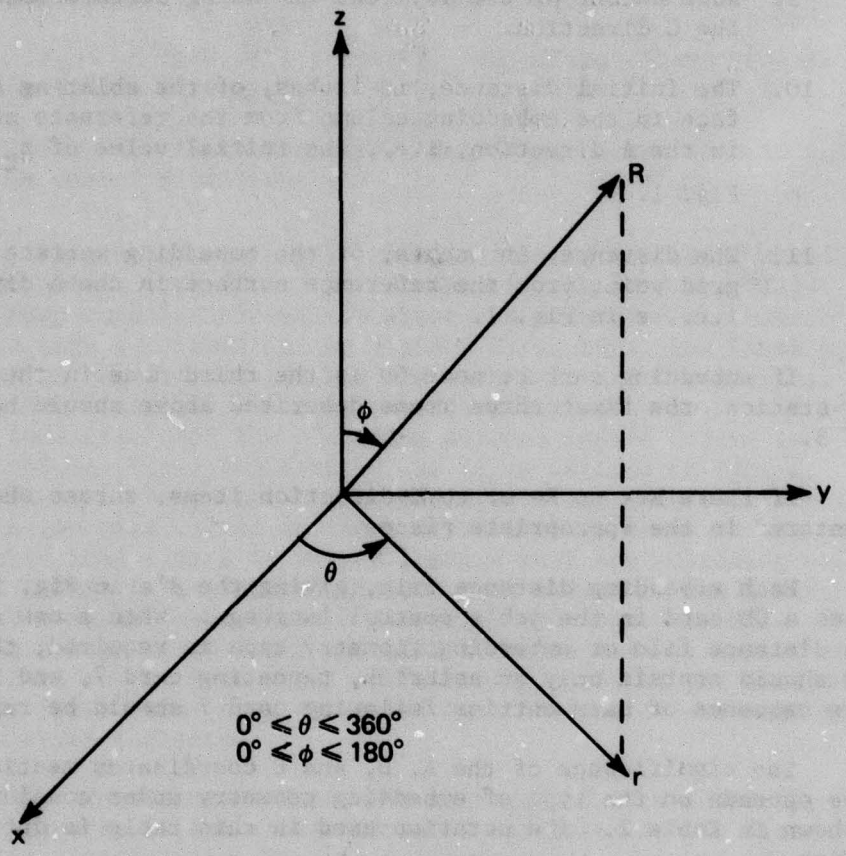
Each embedding distance file, giving the d's in Fig. 1, requires a DD card in the job's control language. When a new embedding distance file or embedding geometry type is required, the next card should contain only an asterisk, repeating card 7, and the above sequence of data entries following card 7 should be repeated.

The significance of the A, B, and C coordinates mentioned above depends on the type of embedding geometry under consideration, as shown in Table 2. The notation used in this table is defined in Fig. 3.

Table 2

Definitions of A, B, and C

Embedding Geometry	A	B	C
RECT	x (in.)	y (in.)	z (in.)
CYLZ	z (in.)	r (in.)	θ (deg.)
CYLR	r (in.)	z (in.)	θ (deg.)
SPH	R (in.)	ϕ (deg.)	θ (deg.)



Rectangular: x, y, z $x = r\cos\theta = R\sin\phi\cos\theta$
Cylindrical: r, θ, z $y = r\sin\theta = R\sin\phi\sin\theta$
Spherical: R, ϕ, θ $z = R\cos\phi$

Fig. 3 Coordinate Definitions

The data supplied in the file SRFDATA pertain to a situation before any ablation has taken place, when all recession rates are zero. It is not necessary to revise SRFDATA to restart a calculation after some ablation has occurred and is proceeding at a finite rate. The appropriate data revisions are performed by ABAERØE, described later.

READSU should follow STØRE.

READTM

CALL READTM (I, IND, D1, D2, D3, D4, D5, NR, INFILE);

I	FIXED BIN.	An index that should be different each time READTM is called.
IND	(*) FLØAT BIN.	An array of independent variables with N + 1 elements.
D1, D2, D3, D4, D5	(*) FLØAT BIN.	Arrays of dependent variables each having N elements.
NR	FIXED BIN.	The number of items read from the arrays described above, usually 0 or N.
INFILE	FILE.	The file containing the list to be read.

READTM (Refs. 4 and 5) is a general table reader. IND(1) = N, the number of table values. D1(J), D2(J), etc. correspond to IND (J+1).

When used in conjunction with SHTP-E, READTM's primary purpose is to store the addresses of time-dependent trajectory data for each body station. I represents the body station; IND is a list of flight times (seconds); and D1, D2, D3, D4, and D5 are lists of recovery enthalpy (Btu/lbm), incident radiant heat flux (Btu/s-ft^2), enthalpy-based cold-wall heat transfer coefficient (lbm/s-ft^2), local total pressure (atmospheres), and a dummy variable, respectively. These data are read or calculated from raw data produced by the three-degree-of-freedom trajectory code (Ref. 15) prior to calling READTM. Therefore, NR = 0 and INFILE = SYSIN; i.e., READTM does not read the data but merely stores the addresses of the data.

READTM can also be used in conjunction with other tables, but care should be taken that I is not one of the body-station numbers. If data are being read by READTM, the first two cards in INFILE are comment cards and the remaining NR cards are arranged as described below, where $0 < J \leq NR$.

Ref. 15. L. L. Perini, "User's Manual for the 3DOF Trajectory Computer Program," APL/JHU ANSP-M-6, Sep 1973.

COLUMNS	DATA DESCRIPTION
1-9	IND (J+1)
10K-10K+9	DK(J), K=1, 2, 3, 4, or 5 (F-format, decimals required)

READTM can be used anywhere in the main program ahead of the thermal-analysis computation loop.

READPR

CALL READPR (MATFILE, ABMAT, GAPM);

MATFILE	FILE.	A file containing the names of the thermal-property files. READPR opens and closes this file.
ABMAT	FIXED BIN.	The material code for the ablating material.
GAPM	(*) FIXED BIN.	A list of gap material codes.

READPR is used to read thermal property tables. One call to READPR replaces several calls to READRK (Refs. 4 and 6).

The first card in MATFILE is a comment card. The remaining cards contain a LIST-directed list of thermal-property file names (quotes are required around each name). The first file listed pertains to material 1, the second to material 2, and so on. A DD card is required in the job's JCL for each file listed. As many as 99 file names are allowed.

READPR counts the number of material files in MATFILE and then allocates storage to the thermal-property pointer to cover that many materials. It then opens each property file and counts the number of cards in the file, after which it closes the file, allocates storage for the thermal properties, reopens the property file, and calls READRKD, a version of READRK (Refs. 4 and 6) with dynamically allocated property pointers. READRKD reads the property data and stores their addresses. When control is returned to READPR, the property file is closed.

Each thermal-property file begins with two comment cards, followed by several cards giving property data as described below.

COLUMNS	DATA DESCRIPTION (All F format, decimals required, blank fields are zeroes)
1-14	Temperature (°R)
15-29	Volumetric specific heat (Btu/ft^3-°R)
30-44	Thermal conductivity (Btu/h-ft-°F) or Contact conductance (Btu/h-ft^2-°F)

45-59	Emissivity, for use in computing gap radiation.
60-69	Exposed surface absorptivity.

GAPM is a list of material property codes that are intended to refer to gaps. The property files for these materials are expected to list gas conductivities and radiant interchange factors (in the emissivity column) in the gaps. If GAPM (I) = +J > 0, then the property data for material J is unaltered after it is read. If GAPM (I) = -J, then the thermal conductivity data items for material J are all set equal to zero and the gap medium is treated as a vacuum.

READPR should precede READABN (described next).

READABN

CALL READABN (#PL, #BETA, LEW#, RHØ, CØNV, AIRPRNT, THRMØCH);

#PL	FIXED BIN.	The number of pressure entries in the thermochemical tables (Ref. 6).
#BETA	FIXED BIN.	The number of ablation-to-diffusion mass-loss ratios in the thermochemical tables.
LEW#	FLØAT BIN.	The Lewis number of the fluid surrounding the ablating surface, usually assumed to be 1.
RHØ	FLØAT BIN.	The density of the ablating material in lbm/ft^3.
CØNV	FLØAT BIN.	A conversion factor for converting the units in thermochemical table entries to English units. The tables are usually given in terms of calories, grams, and °K, so that CØNV is usually 1.8.
AIRPRNT	FIXED BIN.	If AIRPRNT = 1, READABN prints a listing of the surface thermochemical tables. If AIRPRNT ≠ 1, the surface thermochemical tables are not printed.
THRMØCH	FILE.	The name of the file containing the surface thermochemical tables. READABN opens and closes THRMØCH.

READABN is used to read the surface thermochemical tables, which are described in detail by Newman (Ref. 6) and to construct a table of ablating material enthalpy as a function of temperature, also described by Newman. READABN is similar to Newman's READAB, but has five fewer arguments because:

1. The list of the ablating material's temperatures and volumetric specific heats needed by READABN is passed to it internally by READPR.

2. Uniform-temperature-increment tables are not used in EM's ablation procedure, ABAERØE. Tables interpolated by pressure are constructed from the original nonuniform temperature increment tables by ABAERØE, as described in Section 5.

3. AIRPRNT is now an argument rather than an external identifier.

READABN must follow READPR.

THERMAL-ANALYSIS COMPUTATION LOOP

The remaining procedures perform the repetitive thermal-analysis computations, beginning with SET's time TSTART and ending with TSTØP. During the first pass through this loop, no computations are made, but several initializations and storage allocations are carried out and the final structuring of the embedding columns is performed.

The first procedure called is CAPCØNE. It may be desirable to precede CAPCØNE with a definition of distributed internal-heat-source terms. For example, suppose that nodes I_1 through I_2 have a heat generation rate of HINT Btu/h-ft^3. To include the effect of HINT in his model, the user should first write the declare statement

```
DCL(Q(*), XCAP(*,7)) FLOAT BIN CTL EXT;
```

in his main program. Then the computation loop would have the following general appearance.

```
SØURCE:  DØ I = I1 TØ I2;
         Q(I) = XCAP(I,1) * HINT;
         END SØURCE;
CALL CAPCØNE;
     :
     other call statements
     :
GØ TØ SØURCE;
```

This looks like an infinite loop, but the called procedures keep a record of the current time variable and CAPCØNE stops the execution of the program when the time passes TSTØP.

Unless otherwise indicated, the thermal analysis computation loop (TACL) procedures must be called in the order in which they are described on the following pages.

CAPCØNE

CALL CAPCØNE;

CAPCØNE is one of the most important of all the SHTP-E procedures. CAPCØNE is a version of CAPCØN (Refs. 4 and 5) adapted for implicit AF internodal connections and the new material code conventions discussed under READCPE.

On the first pass through the TACL, CAPCØNE finalizes the embedding column structure read by READCPE. The embedding column nodes are printed out so that the user may check to see if the desired columns have indeed been formed. On subsequent TACL passes, CAPCØNE computes the thermal capacitances of each node, internodal conductances, sums of implicit and explicit conductances for each node, and explicit heat flows between neighboring nodes, all of which it does while it keeps track of the proper material codes and the directionality of implicit internodal connections. When TIMS, the SHTP-E time variable, passes TSTØP, CAPCØNE stops program execution.

Unlike CAPCØN (Refs. 4 and 6), CAPCØNE does not call a procedure called CØMCØN to compute the effects of contact conductances. Contact conductances are treated as described under READCPE and READPR.

CAPCØNE must precede STEPE.

STEPE

CALL STEPE (TSWITCH, MINSTP, MAXSTP);

TSWITCH	(*) FLØAT BIN.	A list of #TSW times (seconds) at which the minimum and maximum allowable time steps should be changed.
MINSTP	(*) FLØAT BIN.	A list of #TSW minimum allowable time steps (seconds).
MAXSTP	(*) FLØAT BIN.	A list of #TSW maximum allowable time steps (seconds).

During each pass (except the first one) in the TACL, STEPE computes the stability time step, advances TIMS by one step, computes the temperatures of nodes with no implicit internodal links, and calls a procedure called BAPSUB, which makes B and C direction AF temperature estimations for those nodes that have B and C AF connections.

The three arguments passed to STEPE should have the same array length, say #TSW. TSWITCH should be given in ascending order; i.e., TSWITCH(1) < TSWITCH(2) < TSWITCH(3), etc. The minimum and maximum allowable time steps used by STEPE are selected from the MINSTP and MAXSTP lists, depending on the current value of TIMS, whose units are also seconds. The table below shows how the selection is made.

TIMS (s)	Minimum Time Step (s)	Maximum Time Step (s)
< = TSWITCH(1)	MINSTP(1)	MAXSTP(1)
> TSWITCH(I - 1) & < = TSWITCH(I) (I > 1)	MINSTP(I)	MAXSTP(I)
> TSWITCH(#TSW)	MINSTP(#TSW)	MAXSTP(#TSW)

If there is no limit on the stability time step, or if the stability time step is greater than the maximum allowable time step, or if the maximum allowable time step is given as a negative number, the time step used is the absolute value of the maximum time step. If the stability time step is less than the minimum allowable time step, the program is stopped. Under all other circumstances, the time step used is the stability time step.

STEPE must follow CAPCØNE and precede ABAERØE.

ABAERØE

CALL ABAERØE (ID, CMH, LEW#, VIEW, TSPACE, LAMBDA, RHØ, TLIM, QLIM, ITLIM, PRNTM, PLTCØDE, ADIABATIC, SITLIM, G);

ID	FIXED BIN.	Body station number.
CMH	FLØAT BIN.	Stanton number ratio (C_m/C_H in Section 5).
LEW#	FLØAT BIN.	Lewis number.
VIEW	FLØAT BIN.	View factor for radiation incident on the ablating surface.
TSPACE	FLØAT BIN.	Space temperature (°R).
LAMBDA	FLØAT BIN.	Blowing correction factor.
RHØ	FLØAT BIN.	Ablating material density (lbm/ft^3).
TLIM	FLØAT BIN.	Convergence tolerance limit on surface temperatures (°R).
QLIM	FLØAT BIN.	Convergence tolerance limit on surface heat flux imbalance (Btu/ft^2-s).
ITLIM	FIXED BIN.	Maximum number of iterations allowed in finding the ablating surface temperature.
PRNTM	(*) FLØAT BIN.	List of times (seconds) at which detailed information about ablation calculations is to be printed.
PLTCØDE	FIXED BIN.	Plotter code. If PLTCØDE ≠ 1, nothing is saved for plotting. If PLTCØDE = 1, data are saved for the Rubinstein-Klein plotter code (Ref. 16).

Ref. 16. N. Rubinstein and L. E. Klein, "GENPLOT-A User Oriented Executable Program for Linear, Semi-Log and Log-Log CALCOMP Plots — User's Guide," APL/JHU F1C(2)-75-U-035, 19 Dec 1975.

ADIABATIC	FIXED BIN.	If ADIABATIC = 1, the receding surface is insulated, otherwise it is not.
SITLIM	FIXED BIN.	The number of iterations desired to improve the estimation of the current recession distance. For no iterations, SITLIM = 0.
G	FLØAT BIN.	Ablating surface location iteration parameter. See Eq. 61 and the following discussion.

ABAERØE solves the ablating surface energy balance equation (SEBE) for the ablating surface temperature, and calculates the local recession rate and distance in the A direction for sublimation ablation. It also calculates temperatures for the nodes in each embedding column for both explicit and internodal links in the AF B and C direction. ABAERØE has to be called for each body station. It analyzes all the embedding columns' ablating surfaces for each body station. Embedding surface node numbers and areas are obtained internally from data stored by READSU.

ABAERØE computes ablation using the Mickley-Spalding correlation (Refs. 1 and 6) if CMH > 0 or the Putz and Bartlett correlation (Refs. 6 and 13) plus a mechanical erosion modification (Refs. 6 and 14), if CMH ≤ 0.

In many applications, the use of a Lewis number of 1 is adequate, but one can also use nonunity Lewis numbers in ABAERØE. Care should be taken that the proper data have been read by READABN.

The incident-radiation view factor is usually 1, and for reentry work the space temperature is generally assumed to be 0°R.

The numerical value of LAMBDA, the blowing correction factor, passed to ABAERØE is used only if CMH > 0. Recommended values are 0.5 for laminar flow over the ablating surface and 0.4 for turbulent flow. When CMH ≤ 0, ABAERØE computes LAMBDA internally. See Section 5 for further discussion.

The ablating material's density is assumed to be constant, and all accounting for thermochemical phenomena is confined to the ablating surface.

The SEBE solving process in ABAERØE tries to force the net heat flux imbalance on the ablating surface to be zero, taking QLIM

as an acceptably small imbalance and maintaining a small error in the ablating-surface temperature, less than or equal to TLIM, within ITLIM iterations.

Numerical values in the PRNTM should be in ascending order (i.e., PRNTM(1) < PRNTM(2) < PRNTM(3), etc.). If PRNTM(1) < 0, ABAERØE prints information at every time level. If all values of PRNTM are positive, ABAERØE prints information only for each of them until TIMS, the current time, is greater than PRNTM's maximum value, after which ABAERØE prints information for all subsequent time levels.

If PLTCØDE = 1, ABAERØE stores current values of 19 ablating-surface characteristics for each embedding column in a file called ABPLØT. The file ABPLØT should be allocated to a sequential card data set. The Rubinstein-Klein plotter program (Ref. 16) can be used to plot the data saved in ABPLØT's data set as a function of time. The 19 data items, in the order of their listing in the file, are:

1. Current ablating-surface temperature (°R);
2. Total mass loss per unit area (lbm/ft^2) in the embedding column;
3. Current mass loss rate ($lbm/s\text{-}ft^2$);
4. Integrated ablating surface energy imbalance (Btu/ft^2);
5. Current ablating surface energy imbalance ($Btu/s\text{-}ft^2$);
6. Current recession rate (in./s) in the A direction;
7. Total recession distance (in.) in the A direction;
8. Convective heating rate ($Btu/s\text{-}ft^2$) (for further explanation see T(IPLØT+7), page 55 of Ref. 6);
9. Integrated convective heating (Btu/ft^2);
10. Heating due to ablating surface chemical reaction ($Btu/s\text{-}ft^2$) (for further explanation see T(IPLØT+9), page 56 of Ref. 6);
11. Integrated chemical heating (Btu/ft^2);
12. Radiative heat flux absorbed by ablating surface ($Btu/s\text{-}ft^2$);

13. Integrated radiative heating absorbed by ablating surface (Btu/ft^2);

14. Radiation relief to space from ablating surface ($Btu/s\text{-}ft^2$);

15. Integrated radiation relief to space from ablating surface (Btu/ft^2);

16. Conductive heat flux from ablating material to ablating surface ($Btu/s\text{-}ft^2$);

17. Integrated conductive heat flux from ablating material to ablating surface (Btu/ft^2);

18. Net external heating loss given up to material removal ($Btu/s\text{-}ft^2$); and

19. Integrated material removal heat loss (Btu/ft^2).

The times at which these data are to be plotted are read from a file called ABPLTM, a card file containing a title card followed by cards containing a LIST-directed list of times (seconds). After TIMS passes the last time in this file, no further information is passed to the file ABPLØT.

ADIABATIC = 1 imposes an adiabatic boundary condition on the ablating surface. If this happens when the embedding and ablating surfaces coincide, q_{EXT} is set equal to zero in Eq. 50. If the two surfaces do not coincide, $T^+_{s_E}$ is set equal to T^*_E in Eq. 44.

Using Eq. 61, the estimate of the position of the ablating surface in the embedding column can be refined by going through SITLIM iterations using G as an iteration parameter. Equation 59 is used to obtain the first estimate of the ablating surface position. If SITLIM = 0, Eq. 61 is not used at all. For fast ablation G = 0.5 gives good results, while G = 1.5 works better for slow ablation (Refs. 1 and 2).

On the first pass through the TACL, ABAERØE allocates storage for several identifiers and, if CMH $\leq$ = 0, sets up the β, β_o, and $\lambda(\beta_o)$ table discussed in Section 5. Subsequently, for each

body station at a given time level, all ablating-surface energy-balance calculations are performed at a constant surrounding gas pressure, which ABAERØE finds from the trajectory tables whose contents can be found from the addresses saved by READTM. For the fixed pressure, ABAERØE sets up the constant-pressure ablating surface thermochemical tables described in Section 5. For Le = 1, ABAERØE calls DIKI*, a procedure that corrects the table in the diffusion kinetic ablation regime. There is no correction procedure for nonunity Lewis number.

DIKI performs the required table corrections using several default parameters that can be overridden using data introduced in a file called LØWABF. If there is no DD card for LØWABF, the default table corrections are made. If LØWABF has a DD card and the file contains only the entry

ITB = 0;

the default table corrections are made. On the other hand if

ITB = (any negative number);

appears in LØWABF, no table corrections are made. For values of ITB that are neither negative nor zero, the following additional data are expected by DIKI in the LØWABF file. The first group of data is DATA directed.

KØ: Reaction rate constant ($lbm/ft^{3/2}s\ atm^{1/2}$).

E: Activation energy (cal/mole).

ZD: Diffusion limited CO/CO_2 ratio.

TLØW: Lowest temperature (°R) at which table corrections are to be made.

THIGH: Highest temperature (°R) at which table corrections are to be made.

The semicolon should follow the THIGH assignment. Following these data is a LIST-directed list of ITB pairs of wall temperatures (°R) and CO/CO_2 ratios in ascending order of wall temperature (Table 3). The first wall temperature should be TLØW and the ITB^{th} one should be THIGH. Reference 10 shows how to select the parameters to override DIKI defaults, listed in Table 4.

* Private communication with L. L. Perini, APL/JHU, 1977.

Table 3

Wall Temperature and CO/CO_2 Ratio

Wall Temperature (°R)	CO/CO_2 Ratio
1800	0.001
1980	0.245
2070	0.945
2160	1.910
2250	4.560
2430	12.520
2610	16.580
2880	117.000
3600	200.000

Table 4

Default Parameters for DIKI (Graphite Values)

KØ	$= 9.65 \times 10^{-5}$ lbm/ft$^{3/2}$s atm$^{1/2}$
E	$= 4.4 \times 10^{4}$ cal/mole
ZD	= 17
TLØW	= 1800°R
THIGH	= 3600°R

ABAERØE sets up a thermochemical table as described above and treats wall or gas enthalpy, mass-loss ratio, and chemical heating as functions of wall temperature.

In the process of preparing to solve and solving the SEBE, ABAERØE has to search three different types of tables: those having surrounding gas pressure, ablating surface temperature, and (for CMH $\leq$ 0) ablation-to-diffusion mass-loss ratio as independent variables. In any of these cases, the independent variable may fall outside the range of table values. If this happens, a message is printed but the calculations continue. If the independent variable is less than the lowest table value, the values of the dependent variable corresponding to the lowest independent variable are used. On the other hand, for independent variables higher than the highest table value, a linear extrapolation from the end of the table is used to find the dependent variable.

Once the constant-pressure surface-thermochemical table is set up, ABAERØE solves the SEBE for the ablating-surface temperature for each embedding column passing through the body station. In each embedding column, ABAERØE calls BACKSUB, a procedure that computes the adiabatic and unit incremental thermal-response vectors for that column. After this is done, ABAERØE calls SURFACE, a procedure that computes the local surface inclination with respect to the A direction, L_E^* and T_E^* for use in Eq. 44, and the adiabatic and unit incremental temperatures on the ablating surface. These tasks are performed using geometry information obtained from READSU. Now the Newton-Raphson solution of either Eq. 51 or 52 is carried out to find the ablating-surface temperature. Once it is found, ABAERØE computes the fictitious heating rate on the embedding surface node and uses Eq. 29 to find the temperatures of all the nodes in the embedding column.

If one desires to stop the job or job step using the SHTP-E procedures and to restart the ablation calculation on a subsequent job or job step, ABAERØE can save ablating-surface heat-balance data needed to do this at the end of one job or job step and read it at the beginning of a subsequent job or job step. Output data are saved in a file called RECRSTØ. If there is no DD card for this file, the file is ignored. RECRSTØ should be allocated to a sequential data set. Input data are read from a file called RECRSTI, consisting of data saved by a previous job's or job step's RECRSTØ file. As with RECRSTI, if there is no DD card for RECRSTØ, the file is ignored. The contents of either file are cards or card images (logical record length of 80). The first card in the file gives the time at which the job producing the cards stopped, and

the remaining cards contain ablating-surface heat-balance data in groups of four cards for each embedding column. This information is useful for setting aside space for a data set on the RECRSTØ DD card.

File Summary

There are five files that can be used with ABAERØE, although it is not necessary to use any of them. If PLTCØDE = 1, the user has to supply DD cards for the files ABPLTM (input) and ABPLØT (output). The default diffusion kinetic modification to the ablating surface thermochemical tables can be overridden using the file LØWABF. If there is no DD card for LØWABF, DIKI is executed using default graphite parameters. Data can be saved for restarting the thermal analysis simply by supplying a DD card for the file RECRSTØ, an output file. If there is no DD card for this file, no attempt is made to save any ablating surface data for future use. Data from a previous job's or job step's RECRSTØ file can be read by ABAERØE through the file RECRSTI. No attempt is made to read such data if there is no DD card for RECRSTI. The output files ABPLØT and RECRSTØ should be allocated to sequential, not partitioned, data sets. All of the files are card files; i.e., their logical record length is 80.

ABAERØE must follow STEPE and precede WRITEE and PLTSTRS.

WRITEE

CALL WRITEE (TIMEP);

TIMEP	(*) FLØAT BIN.	A list of times (seconds) at which temperature distributions are to be printed. This list should be in ascending order; i.e., TIMEP(1) < TIMEP(2) < TIMEP(3) etc.

WRITEE serves the same function as WRITE (Refs. 4 and 6), printing out temperature distributions (°F) at selected times. WRITEE suppresses the printing of fictitious node temperatures if the embedding-surface node numbers read by READSU are positive but prints those temperatures if the embedding-surface node numbers read by READSU are negative. After printing the temperature distibution of the body being analyzed, WRITEE prints the ablating-surface temperatures (°F) and recession distances (in.) and rates (in./s) in the A direction for each embedding column, distinguished in the output by its embedding-surface node number.

Each temperature distribution listing has the following general appearance:

25 = 1025.3 26 = 1040.9

This means that the temperature at node 25 is 1025.3°F and the temperature at node 26 is 1040.9°F. The ablating surface listing has the appearance

1 = 7055.2 4.562E-04 3.411E-02

for each embedding column. The numbers signify that the ablating surface in the embedding column with embedding surface node 1 has a temperature of 7055.2°F, is located 4.562×10^{-4} in. from the inertial reference surface (see Fig. 1) in the A direction, and is moving at a rate of 3.411×10^{-2} in./s in the A direction away from the inertial reference surface. There is no node number associated with the ablating surface, each point of intersection between the ablating surface and an embedding column is identified by both WRITEE and ABAERØE by the associated embedding-surface node number.

When WRITEE prints temperature and ablating-surface information, it prints the time at which these data apply before printing anything else. After everything is printed, WRITEE causes STEPE to print the time step used and the maximum allowable time step, based on the temperatures printed. STEPE also prints a line of asterisks to separate the current temperature distribution from the next one to be printed.

When TIMS is greater than the largest value of TIMEP, and there is more than one value of TIMEP, WRITEE prints temperature and ablating surface information at every time level. If TIMEP(1) < 0, WRITEE prints information at every time step. If TIMEP(1) = ±999, additional information useful for debugging is printed. If there is only one value of TIMEP, TIMEP(1) is treated as a print-time interval if it is positive, and WRITEE prints information every TIMEP(1) seconds. If there is more than one value of TIMEP, and TSTART (saved by SET) is greater than the first several values of TIMEP, no printing is done by WRITEE for these times.

WRITEE must follow ABAERØE but may either precede or follow PLTSTRS (described next).

PLTSTRS

CALL PLTSTRS (PLTM, #CAPS, A, B);

PLTM	(*) FLØAT BIN. A list of times (seconds) at which PLTSTRS is to save data for plotting or thermal-stress purposes.
#CAPS	FIXED BIN. The number of nodes in the finite-difference model.
A	(*) FLØAT BIN. A list of #CAPS A coordinates. A(I) = node I's A coordinate. The B or C coordinates can also be used here.
B	(*) FLØAT BIN. A list of #CAPS B coordinates. B(I) = node I's B coordinate. The A or C coordinates can also be used here.

PLTSTRS is an optional procedure that can be used for saving data for the Rubinstein-Klein (Ref. 16) plotter code, thermal stress calculations, and restarting the thermal analysis computations. PLTSTRS has six files (two input and four output) available for these purposes. All of the files are card files with a logical record length of 80. The temperature field should depend on only one or two spatial variables.

JIMRØG (named after APL's J. T. Stadter and R. O. Weiss, whose need for temperature data in a special format for thermal stress computations first motivated the writing of PLTSTRS) is an input file used to supply comments for use in the four output files. Up to twelve lines of comments, each line being no more than forty characters long, can be read from JIMRØG. Only the first forty columns of each card should be used. There has to be a DD card for JIMRØG.

NRPLØTS is an output file. For each value of PLTM and all times larger than the maximum PLTM, PLTSTRS stores the time (seconds), the A and B coordinates, temperature (°F), and index of each real node and uses the entire contents of JIMRØG. Fictitious node data are also stored if the embedding surface node numbers read by READSU are negative. In addition to node data, PLTSTRS saves the A and B coordinates and temperatures (°F) of the ablating surface's intersection points with each embedding column and identifies these points by minus the embedding surface node numbers in

NRPLØTS. There is one card image generated for each node in a format suitable for both thermal stress calculations and use with the Rubinstein-Klein plotter. If there is no DD card for NRPLØTS, the file is ignored.

STRESS is an output file very similar to NRPLØTS, differing only in that it is always used to store both real and fictitious node data and it does not store any ablating-surface temperatures. If there is no DD card for STRESS, the file is ignored.

RECPLØT is an output file used to store times (seconds), A-direction recession distances (in.) and rates (in./s), and embedding-surface node numbers for each embedding column at times corresponding to each value of PLTM and greater than the maximum value of PLTM. These data can be used with the Rubinstein-Klein plotter. RECPLØT uses the first line in the JIMRØG file as a descriptive title. If there is no DD card for RECPLØT, the file is ignored.

NEXTRUN is an output file that saves the final temperature distribution and time for use in a subsequent job or job step. It uses the entire contents of JIMRØG, and its formatting is the same as that used for NRPLØTS and STRESS. If there is no DD card for NEXTRUN, the file is ignored.

PREVRUN is an input file containing the contents of the NEXTRUN file of a previous job or job step. When there is a DD card for PREVRUN, PLTSTRS overrides SET and initializes the time and temperatures as they are read from the PREVRUN file. The final time in this case is taken to be the maximum value of PLTM. All this is done on the first pass through the TACL. If there is no DD card for PREVRUN, the file and the actions associated with it are ignored.

If PLTM(1) is less than the starting time, PLTSTRS will stop program execution. All of the output files used by PLTSTRS should be allocated to sequential data sets if they are used at all.

8. DEBUGGING AIDS AND USEFUL EXTERNAL IDENTIFIERS

As it is with other SHTP procedures, the SHTP-E procedures have several external identifiers that the user need not know. However, there are some external identifiers that a user may desire to use in his main program; the purpose of this Section is to describe them. Each identifier will be introduced by its DCL statement, which will be followed by a description.

```
DCL (DBGEB INIT ('0'B))  BIT(1)  EXT;
```

DBGEB is a bit flag used to print debugging messages throughout the SHTP-E procedures. These messages are printed when DBGEB = '1'B. A job using DBGEB = '1'B should be run for only very few time steps because of the copious output produced.

```
DCL (DBPTAB INIT ('0'B))  BIT(1)  EXT;
```

When DBPTAB = '1'B, ABAERØE prints the constant-pressure thermochemical tables that it generates from the raw data obtained from READABN. This is done only when ABAERØE prints ablating-surface information as dictated by its print-time argument.

```
DCL XCAP(*,7) FLØAT BIN CTL EXT;
```

XCAP is the array of geometric parameters read by READCPE. XCAP(I,1) through XCAP(I,7) are the numerical data entries read in columns 10 through 79 on the cards in READCPE's geometry file.

```
DCL T(*) FLØAT BIN CTL EXT;
```

T(I) is the temperature of node I in Rankine degrees.

```
DCL Q(*) FLØAT BIN CTL EXT;
```

Q(I) is the sum of internal source heating and explicit internodal heating (in Btu/h) for node I. For each time level, Q should not be altered after calling CAPCØNE.

DCL TIMS FLØAT BIN EXT;

TIMS is the current thermal model time, in seconds.

DCL PP(*) PØINTER CTL EXT;

PP is an array of addresses for thermal properties. Its dimension is five times the number of material files read by READPR, when PP is used to access a particular property, a based variable with the declaration, DCL PRDUM(100) BASED(PR) FLØAT BIN; is needed.

For the J^{th} data listing of material, I, PP, and PRDUM can be used to retrieve the properties listed below.

PP(5*I-4) - > PRDUM(1)	Number of entries in property list, maximum value of J.
PP(5*I-4) - > PRDUM(J+1)	Temperature (°R).
PP(5*I-3) - > PRDUM(J)	Volumetric specific heat (Btu/ft^3-°R).
PP(5*I-2) - > PRDUM(J)	Thermal conductivity (Btu/h-ft-°R).
PP(5*I-1) - > PRDUM(J)	Emissivity or gap radiation factor.
PP(5*I) - > PRDUM(J)	Surface absorptivity.

9. SHTP-E COMPUTER LIBRARIES

Computer libraries for PL/I source code, PL/I optimizer-compiler cross-referenced listings, and compiled and link-edited load modules for the SHTP-E procedures have been created under the index BBE.RANDALL.

BBE.RANDALL.EMBEDPLI contains the PL/I source code for all the procedures needed to support SHTP-E. Included are all of the procedures described in Section 7. It also contains the source code for DIKI, BACKSUB, and SURFACE which are called by ABAERØE; BAPSUB which is called by STEPE; READRKD which is called by READPR; and DECIDE (Ref. 3) and PIF1D (Ref. 3) which are two general-purpose interpolation subroutines. The library member names for the source code are the procedure names; e.g., ABAERØE's source code is contained in BBE.RANDALL.EMBEDPLI(ABAERØE).

In addition to the procedure source codes, BBE.RANDALL.EMBEDPLI has three other members: ENTRIES, CØMMENTS, and LINKS.

ENTRIES contains the DCL statements for all of the procedures described in Section 7. CØMMENTS contains a main program heading that can be used with ENTRIES to avoid blocksize difficulties when one attempts to compile a main program from card input. Compiling a main program from card input is done by arranging the P.SYSIN cards in the following way:

```
//P.SYSIN DD DSN=BBE.RANDALL.EMBEDPLI(CØMMENTS),DISP=SHR
// DD  *
  User supplied source code.
// DD DSN=BBE.RANDALL.EMBEDPLI(ENTRIES),DISP=SHR
// DD  *
  More user supplied source code.
```

Use of CØMMENTS AND ENTRIES in this way makes it unnecessary for the user to write DCL statements for the embedding procedures he has to call. If the user's source code is in a data set (e.g., BBE.USER.SOURCE) with a blocksize of 6160 or larger, the following

arrangement can be used.

```
//P.SYSLIB  DD  DSN=BBE.RANDALL.EMBEDPLI,DISP=SHR

//P.SYSIN   DD  DSN=BBE.USER.SØURCE,DISP=SHR
```

The data set BBE.USER.SOURCE should contain the statement

```
%INCLUDE SYSLIB(ENTRIES);
```

in the calling program.

LINKS is a set of linkage editor input cards that can be used to take advantage of the linkage editor overlay feature when an executable load module is created. These cards can be accessed by using the JCL card

```
//L.SYSIN   DD  DSN=BBE.RANDALL.EMBEDPLI(LINKS),DISP=SHR
```

after the //L.SYSLMØD statement in the job used to create the executable load module.

A listing of the contents of any or all of the members of BBE.RANDALL.EMBEDPLI can be obtained by using the LISTSRC or LISTPCH utilities described in Ref. 17.

Listings of the PL/I procedures cross referenced by the PL/I optimizer in BBE.RANDALL.EMBEDPLI are kept in members of a partitioned data set designated by

```
BBE.RANDALL.EBXREF(Procname)
```

where Procname is the name of one of the procedures. The CSDS utility program (Ref. 18) can be used to obtain these listings. For example, to obtain a hard copy of ABAERØE's PL/I cross referenced listing, one can use the following JCL cards.

```
//LIST EXEC PGM=CSDS,TIME=(0,10)

//SYSPRINT DD DUMMY
```

Ref. 17. "System 360 Model 91 User's Guide," APL/JHU BCS-1:40, Nov 1973.

Ref. 18. G. L. Bennett, J. C. Hagan, and D. C. Tantino, "Evaluating Aeroshell Materials for the MJS/Multihundred Watt Heat Source," APL/JHU ANSP-M-14, Jul 1976.

```
//INPUT DD DSN=BBE.RANDALL.EBXREF(ABAERØE),DISP=SHR

//ØUTPUT DD SYSØUT=A,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=1922)
```

A microfiche copy is obtainable by replacing SYSØUT=A with SYSØUT=Q on the //ØUTPUT card.

Compiled and link edited SHTP-E procedures have been stored in the library BBE.RANDALL.EMBEDLIB whose member names are: ABAERØE, APFAC. BACKSUB, BAPSUB, CAPCØNE, DECIDE*, DIKI, PIF1D*, PLTSTRS, READABN, READCPE, READPR, READRKD, READSU, READTM*, SET*, STEPE, STØRE, SURFACE, and WRITEE. The members with asterisks were copied directly from BBE.FRAZER.BASICØ2. BBE.RANDALL.EMBEDLIB is needed in the ØL, ØPL, or ØPLG statement when an executable load module is created, and is accessed as follows:

```
//STEP EXEC ØPL,LIB='BBE.RANDALL.EMBEDLIB',...
```

10. EXAMPLE PROBLEM

The EM procedures have been used successfully in reentry thermal analyses of a General Electric Multihundred Watt Generator Heat Source Assembly (MHW/HSA) full-sphere assembly (FSA) (Ref. 18) and a Mound Laboratory radioisotope heater unit (Ref. 19). A specific FSA reentry analysis will be described here to show how the EM procedures are used.

The FSA consists of a PuO_2 fuel sphere surrounded by an iridium post-impact containment shell (PICS), that, in turn, is surrounded by a T-50 wound graphite impact shell (GIS). The fuel sphere is 1.498 in. in diameter. The inside diameter of the PICS is 1.578 in. and it has a thickness of 0.020 in. The inside diameter of the GIS is 1.646 in. and its thickness is 0.460 in. In the example given here, the MHW/HSA breaks up at 11.06 s into a reentry flight and bore FSA's fall from this point to impact at 191.66 s. Breakup occurs at an altitude of 117 443 ft at a flight path angle of -47.14° at a speed of 29 830 ft/s. The GIS's exterior surface temperature is initially 2599°R and the FSA's temperature distribution is steady at breakup time. The fuel-PICS gap contains helium and the PICS-GIS gap is evacuated.

A cross section of the FSA thermal model used is shown in Fig. 4. The true model can be visualized by rotating the cross section about the external flow field's axis of symmetry. It is assumed that the temperature distribution is axisymmetric so that it depends on time, radius from the fuel center, and angle from the forward stagnation point.

The FSA analysis was carried out using thermal properties, thermochemical data, and heating-rate and pressure-distribution data obtained from various sources. PL/I procedures, JCL, and data needed to perform the analysis are stored in the partitioned data set BBE.RANDALL.FSAEX, a listing of which is obtainable using the LISTSRC utility (Ref. 16). Table 5 lists and describes the members of BBE.RANDALL.FSAEX.

Ref. 19. "Safety Analysis Report for the MJS-Radioisotope Heater Unit," Report MLM-ML-76-48-0003, Mound Laboratory, Miamisburg, OH, Aug 1976.

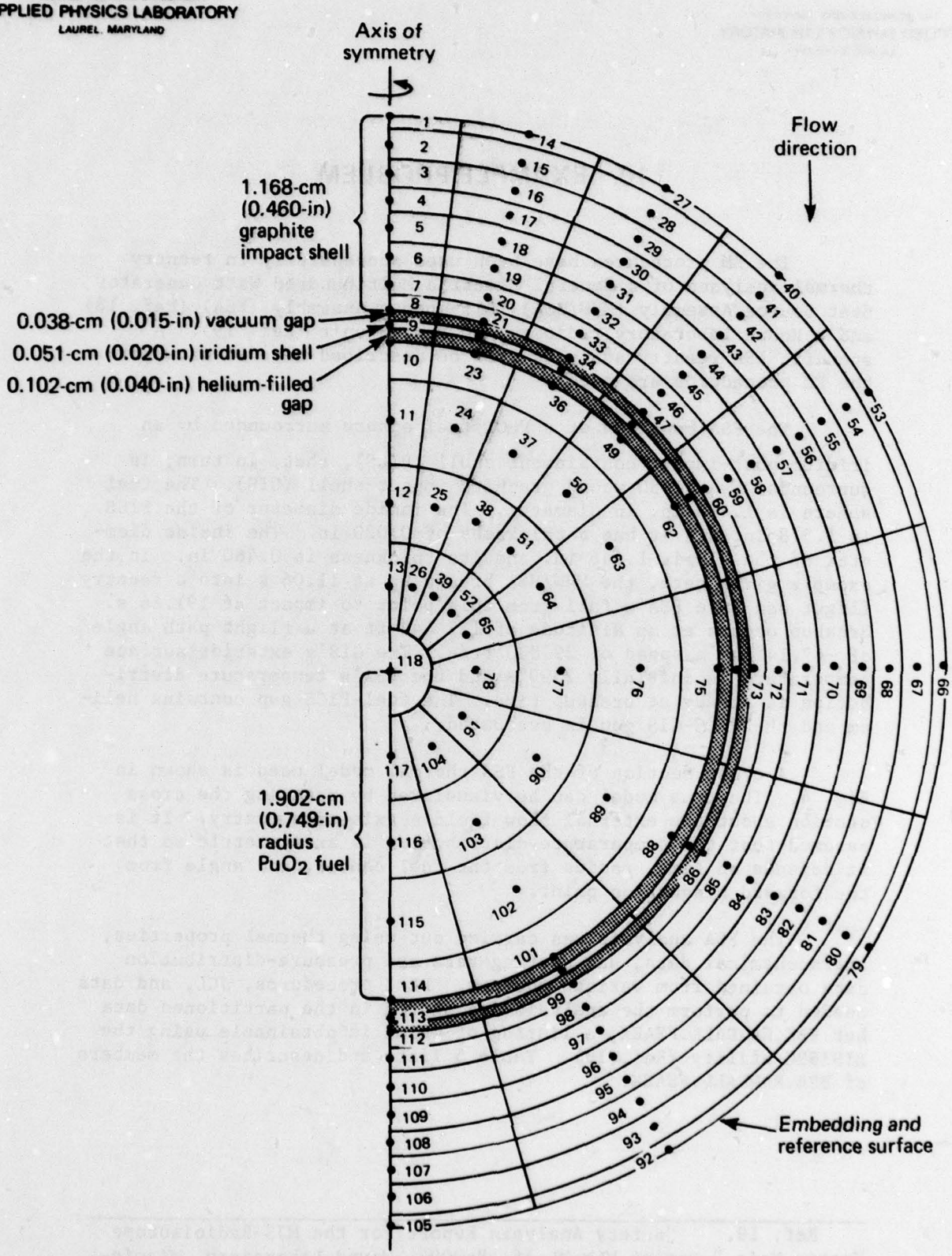


Fig. 4 Cross-Section of FSA Thermal Model

Table 5

BBE.RANDALL.FSAEX

Member	Description
GEØPLI	PL/I source code for generating geometric data for the contents of the members GEØDATA, SURFACE, ABCØØRDS, RSNPHIS, and ØNEDS.
GEØDATA	Node geometry data read by READCPE for the thermal model sketched in Fig. 4.
SURFACE	Embedding-surface geometry data read by READSU.
ABCØØRDS	Node distance from reference-surface data read by READSU.
RSNPHIS	List of node numbers and radial and angular spherical coordinates of nodes needed by PLTSTRS.
ØNEDS	Node geometry data for a purely radially dependent FSA temperature distribution, used to find initial steady state.
FUELTH	Thermal property list for PuO_2 fuel.
GFPTH	Thermal property list for PuO_2-PICS gap.
GPSTH	Thermal property list for PICS-GIS gap.
T5OTH	Thermal property list for GIS.
IRIDIUM	Thermal property list for PICS.
ICPLI	PL/I source code for finding initial radially dependent FSA using steady-state procedure described in Ref. 20.
STJØB	JCL necessary to execute PL/I procedure in member ICPLI.
IC2599	Steady-state temperature distribution generated by steady-state program.

Ref. 20. R. K. Frazer, "An Alternate Solution Technique for Steady State Temperature Analysis," APL/JHU EM-4505, Rev. 1, 26 Oct 1973.

Table 5 (cont'd)

Member	Description
FCØNV	List of convective heating rate factors used to define convective heating distribution on the FSA's outer surface.
TRAJ	Trajectory data generated by program described in Ref. 15.
ANPLI	PL/I source code for the main program using the SHTP-E procedures to solve the FSA reentry problem.
LIBCR	JCL for creating a compiled and link-edited library using the contents of member ANPLI.
LIBREP	JCL for replacing the libary created by the job in member LIBCR with a revised library.
CHANGE	JCL for creating an executable load module using the contents of BBE.RANDALL.EMBEDLIB and the library created by the job in member LIBCR or revised by the job in member LIBREP.
JØB1	JCL for a job using the executable load module created by the job in member CHANGE in the FSA thermal analysis beginning with the initial FSA steady-state data in member IC2599. The JCL shows how to create output data sets from the contents of the output files used by the procedures ABAERØE and PLTSTRS.
JØBM1	JCL for a job that begins with the data created at member JØB1's last time level. The JCL shows how to revise the data sets created by the job in member JØB1.

The aerothermochemical data were taken from

BBE.FRAZER.MHWDATA(TCH14X39).

A PL/I optimizer-compiler cross-referenced listing of the contents of BBE.RANDALL.FSAEX(ANPLI) is kept in the sequential data set BBE.RANDALL.FSAXREF. The printed outputs from the jobs listed in BBE.RANDALL.FSAEX(JØB1) and BBE.RANDALL.FSAEX(JØBM1) are kept in the sequential data sets BBE.RANDALL.FSAPR1 and BBE.RANDALL.FSAPRM1, respectively. Listings of the contents of these three sequential

data sets can be obtained using the JCL given in Section 9 for listing the contents of BBE.RANDALL.EBXREF(ABAERØE).

A microfiche or hard-copy listing that includes PL/I cross-referenced listings of all the embedding procedures, the members of BBE.RANDALL.FSAEX discussed above, the aerothermochemical data, a PL/I cross-referenced listing of the main FSA thermal-analysis procedure, and the two printed outputs kept in BBE.RANDALL.FSAPR1 and BBE.RANDALL.FSAPRM1 can be obtained using the following JCL and data cards.

```
//Job statement

//LIST EXEC PGM=CSDS,TIME=(0,10)

//SYSPRINT DD DUMMY

//INPUT DD DATA

//JØBNAME JØB (ACCT#,C,U,N), "Title"

// 'USER ID',RD=R

//*RØUTE Destination

/*

// DD DSN=BBE.RANDALL.FSAEX(Member),DISP=SHR

//ØUTPUT DD SYSØUT=J,

// DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)

/*
```

The above JCL contains a job that submits another job. The file INPUT is a concatenation of a user-supplied job statement and destination card with JCL necessary to produce the desired output. Using SYSØUT = J, the file ØUTPUT submits the job assembled in INPUT. The JCL of this job is printed at remote terminal destination (e.g. RM001), but the printed (Member = PRXREF) or microfiche (Member = FICHXREF) output is produced at the APL computing center (Building 3).

ACKNOWLEDGMENT

This work was conducted as part of the Applied Physics Laboratory's Aerospace Nuclear Safety Program, supported by the Safety Branch, Advanced Materials and Production Division, U.S. *Department of Energy.*

The author wishes to thank his coworkers at APL for their advice and patience while he was performing this work. Useful discussions with R. K. Frazer, R. W. Newman, and L. L. Perini were especially helpful.

REFERENCES

1. J. D. Randall, "An Investigation of Finite Difference Recession Computation Techniques Applied to a Nonlinear Recession Problem," APL/JHU ANSP-M-15, Mar 1978.

2. J. D. Randall, "Finite Difference Solution of the Inverse Heat Conduction Problem and Ablation," Proc. 1976 Heat Transfer and Fluid Mechanics Institute, Stanford University Press, Stanford, CA, 1976, pp. 257-269.

3. R. K. Frazer, "Further Additions and Improvements to the BBE Heat Transfer Program," APL/JHU EM-4274, 16 Jun 1969; Rev. 1, 29 Oct 1969; Rev. 2, 22 Mar 1971; Rev. 3, 29 Mar 1972; Rev. 4, 22 Jan 1973.

4. R. K. Frazer, "URLIM-A Unified Radome Limitations Computer Program, Volume 2 - Users Guide," APL/JHU TG 1293B, Apr 1978.

5. "OS PL/I Checkout and Optimizing Compilers: Language Reference Manual," IBM File No. S360/S370-29, Order No. GC33-0009-4, Oct 1976.

6. R. W. Newman, "A User's Guide for the Continuous Wave Laser Damage Computer Program," APL/JHU TG 1268, Dec 1974.

7. G. D. Smith, Numerical Solution of Partial Differential Equations, Oxford University Press, London, 1965.

8. P. D. Lax and R. D. Richtmeyer, "Survey of the Stability of Linear Finite Difference Equations," Communications on Pure and Applied Mathematics, Vol. 9, May 1956, pp. 112-113.

9. S. D. Conte, Elementary Numerical Analysis, McGraw-Hill Book Co., New York, 1965.

10. L. L. Perini, "Graphite Oxidation Modelling in Subsonic Flow," APL/JHU ANSP-111, 9 Aug 1977.

11. H. S. Mickley, R. C. Ross, A. L. Squyers, and W. E. Stewart, Heat, Mass, and Momentum Transfer for Flow Over a Flat Plate with Blowing or Suction, NACA TN 3208, Jul 1954.

12. K. E. Putz and E. P. Bartlett, "Heat Transfer and Ablation Rate Correlations for Reentry Heat Shield and Nose Tip Applications," AIAA Preprint No. 72-91, Jan 1972.

13. "User's Manual Aerotherm Charring Material Thermal Response and Ablation Program," Version 3, Aerotherm Corporation, Mountain View, CA, Report UM-70-14, Apr 1970.

14. L. L. Perini, "Review of Graphite Ablation Theory and Experimental Data," APL/JHU ANSP-M-1, Dec 1971.

15. L. L. Perini, "User's Manual for the 3DOF Trajectory Computer Program," APL/JHU ANSP-M-6, Sep 1973.

16. N. Rubinstein and L. E. Klein, "GENPLOT-A User Oriented Executable Program for Linear, Semi-Log and Log-Log CALCOMP Plots — User's Guide," APL/JHU F1C(2)-75-U-035, 19 Dec 1975.

17. "System 360 Model 91 User's Guide," APL/JHU BCS-1:40, Nov 1973.

18. G. L. Bennett, J. C. Hagan, and D. C. Tantino, "Evaluating Aeroshell Materials for the MJS/Multihundred Watt Heat Source," APL/JHU ANSP-M-14, Jul 1976.

19. "Safety Analysis Report for the MJS-Radioisotope Heater Unit," Report MLM-ML-76-48-0003, Mound Laboratory, Miamisburg, OH, Aug 1976.

20. R. K. Frazer, "An Alternate Solution Technique for Steady State Temperature Analysis," APL/JHU EM-4505, Rev. 1, 26 Oct 1973.

NOMENCLATURE

English

A, B, C	Orthogonal coordinates, m or in.
A_E	Embedding surface node area, m^2 or ft^2
a_i	Lagrange interpolation coefficient for temperatures
a_{ij}	Heat flux weighting factor
$b(E_i,E_j)$	Lagrange interpolation coefficient for finding ablating surface gradient, m^{-1} or ft^{-1}
b	Logorithmic ablation-to-diffusion mass-loss ratio
C_i	Thermal capacitance, J/K or Btu/°R
C_H	Heat transfer Stanton number
C_m	Mass transfer Stanton number
c_i	Lagrange interpolation coefficient for finding ablating surface temperature gradient, m^{-1} or ft^{-1}
d_i	Distance of node i in embedding column from inertial reference surface, m or in.
e	Distance of embedding surface from fixed reference surface, m or in.
G	Recession distance iteration parameter
$H(T_s)$	Functional relationship describing ablating surface temperature dependence of the ratio of mechanical erosion mass-loss rate to thermochemical mass-loss rate
K_{ij}	Thermal conductance for heat flow between nodes i and j, W/K or Btu/h-°R

k	Thermal conductivity of ablating material, W/m-K or Btu/h-ft-°R
L_E^*	Equivalent conduction path, m or ft
Le	Lewis number
$\dot{m}$	Mass loss flux due to ablation, kg/m^2-s or lbm/ft^2-s
p	Tabular pressure in thermochemical table; local pressure, atm
Q_F	Fictitious heating rate applied to embedding surface, W or Btu/s
Q	Sum of distributed heat source and explicit internodal heating rates, W or Btu/h
q_{chem}	Ablating surface heat flux due to chemical reaction, W/m^2 or Btu/ft^2-s
q_{cond}	Heat flux conducted from ablating material to ablating surface, W/m^2 or Btu/ft^2-s
q_{conv}	External convective heat flux to ablating surface, W/m^2 or Btu/ft^2-s
q_{EXT}	Total external heat flux to ablating surface, W/m^2 or Btu/ft^2-s
q_i	Internal heat generation rate for node i, W or Btu/h
q_{radin}	Incident radiative heat flux on ablating surface, W/m^2 or Btu/ft^2-s
q_{radout}	Radiation relief from ablating surface to space, W/m^2 or Btu/ft^2-s
q_{rem}	Energy flux required to remove ablating material, W/m^2 or Btu/ft^2-s

S_{E_i}	Sum of explicit conductances for node i, W/K or Btu/h-°R
S_{I_i}	Sum of implicit conductances for node i, W/K or Btu/h-°R
s	A-direction recession distance, m or in.
T	Temperature, K or °R
T_A	Adiabatic thermal response, K or °R
T_U	Unit incremental thermal response, K/W or °R-s/Btu
T_E^*	Temperature used to estimate ablating surface temperature gradient, K or °R
t	Time, s or h
U_i	B-direction temperature approximation, K or °R
V_i	C-direction temperature approximation, K or °R
x	Surface thermochemical table dependent variable

Greek

β	Ratio of actual ablation mass-loss rate to diffusion limited mass-loss rate, uncorrected for blowing
β_o	Blowing-corrected ratio of actual ablation mass-loss rate to diffusion-limited mass-loss rate
δ_{iE}	Kronecker delta: $\delta_{iE} = 1$, if $i = E$; $\delta_{iE} = 0$, if $i \neq E$
ΔT	Ablating surface temperature change during material removal; temperature change during rough search for ablating surface temperature, K or °R
Δt	Time step, $t_{\ell+1} - t_\ell$, s or h

ε	Newton Raphson error function, W/m^2 or Btu/ft^2-s for embedded ablating surface, K or °R when ablating and embedding surfaces coincide
$\varepsilon'(T)$	$d\varepsilon/dT$
λ	Blowing parameter
μ	Cosine of angle between A direction and ablating surface normal
ν_A	Lagrange interpolation polynomial approximation of $\partial T_A/\partial A$, K/m or °R/ft
ν_U	Lagrange interpolation polynomial approximation of $\partial T_U/\partial A$, K/W-m or s-°R/Btu-ft
π	Logarithmic tabular pressure in thermochemical tables; local logarithmic pressure
ρ	Density of ablating material, kg/m^3 or lbm/ft^3
σ	Sum of temperature coefficients, J/K or Btu/°R

Subscripts

b	Blowing corrected value
E	Embedding surface node
G	Gas flowing over ablating surface
I, J, K	Three successive nodes in embedding column
i	Node being analyzed; tabular pressure entry in thermochemical tables
j	Neighboring node; tabular mass-loss ratio entry in thermochemical tables
MC	Mechanical erosion
s	Ablating surface
s_E	Ablating surface intersection with embedding column having embedding surface node E

TC	Thermochemical ablation

Superscripts

A, B, C	A-, B-, C-direction implicit internodal links
E	Explicit internodal link
ℓ	Time level index, $\ell^{\underline{th}}$ time step
+	Time level $\ell + 1$. No superscript: time level ℓ
n	Iteration number

Operators

Δ_{ij}	Difference operator, $\Delta_{ij}T_i = T_i - T_j$
∇_{BC}	Gradient in B-C plane, m^{-1} or ft^{-1}
Π	Product
Σ	Summation

Miscellaneous

$\underline{F}$	Vector of all values of F for all nodes

Acronyms

AF	Approximate factorization
EBE	Energy-balance equation
EM	Embedding method
FD	Finite difference
SEBE	Ablating-surface energy-balance equation
SHTP	Standard Heat Transfer Program
SHTP-E	Embedding Standard Heat Transfer Program

STT	**Surface thermochemical table**
TACL	**Thermal-analysis computation loop**