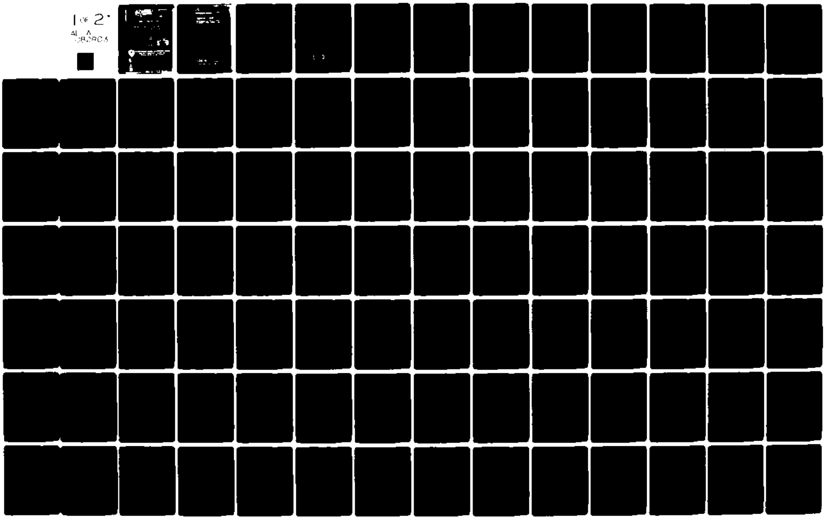
AD-A082 803 ARMY ARMAMENT RESEARCH AND DEVELOPMENT COMMAND ABERD--ETC F/G 21/2
A METHOD FOR COMPUTING THE FLAME SPEED OR A LAMINAR, PREMIXED, --ETC(U)
JAN 80 T P COFFEE, J M HEIMERL
UNCLASSIFIED ARBRL-TR-02212 SBIE -AD-E430 386 NL

1 OF 2
AD A082803



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER TECHNICAL REPORT ARBRL-TR-02212	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A METHOD FOR COMPUTING THE FLAME SPEED OR A LAMINAR, PREMIXED, ONE DIMENSIONAL FLAME		5. TYPE OF REPORT & PERIOD COVERED BRL TECHNICAL REPORT 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) T. P. COFFEE J. M. HEIMERL		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS US Army Armament Research & Development Command US Army Ballistic Research Laboratory ATTN: DRDAR-BLP Aberdeen Proving Ground, MD 21005		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 1L161102AH43
11. CONTROLLING OFFICE NAME AND ADDRESS US Army Armament Research & Development Command US Army Ballistic Research Laboratory ATTN: DRDAR-BL Aberdeen Proving Ground, MD 21005		12. REPORT DATE JANUARY 1980 13. NUMBER OF PAGES 152
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Species Profiles
Temperature Profiles
One Dimensional Flame
Premixed Flame
Non-uniform adaptive grid

20. ABSTRACT (Continue on reverse side if necessary and identify by block number) (clt)

A method of numerically solving the equations for a one-dimensional, premixed, laminar, steady state flame that propagates into an unbounded medium is described. Particular attention is given to the description of an adaptive grid, which concentrates gridpoints within the flame front.

TABLE OF CONTENTS

Page

I. INTRODUCTION

This report describes a method of numerically solving the equations governing a one-dimensional, premixed, laminar, steady state flame that propagates into an unbounded medium. To solve such a system, we adopted a standard package for integrating one-dimensional partial differential equations known as PDECOL. The program is modified to handle flame equations efficiently by concentrating the spatial grid about the flame front itself. How and why this is accomplished constitutes the bulk of this report. As a test case, the program has been implemented for the ozone flame in which there are three chemical species: O, O_2 and O_3.

In passing it should be pointed out that two common simplifying assumptions are not made. These are (1) that the species diffusion velocity is proportional to the gradient of the logarithm of the species mass fraction (Fick's law) and (2) that the Lewis number is a fixed, predetermined value or function.

II. THE FLAME EQUATIONS

The derivation of the conservation equations for multicomponent reacting ideal gas mixtures can be found in the literature[1,2,3]. Here we are interested in those equations that adequately describe a one-dimensional, laminar, premixed flame. The effects of viscosity, thermal diffusion and body forces are ignored. For such flames we recognize the fact that the flame velocity is small compared with the local speed of sound and so take the pressure to be constant[3,4].

The equations pertinent to our present case are:

Overall Continuity:

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u)}{\partial x} = 0 \tag{1}$$

---

[1] *J. O. Hirschfelder, C. F. Curtiss and R. B. Byrd, Molecular Theory of Gases and Liquids, John Wiley and Sons, New York, second printing with notes, March 1964.*

[2] *R. B. Bird, W. E. Stewart and E. N. Lightfoot, Transport Phenomena, John Wiley and Sons, New York, 1960.*

[3] *F. A. Williams, Combustion Theory, Addison-Wesley, Reading Mass., 1965.*

[4] *R. M. Fristrom and A. A. Westenberg, Flame Structure, McGraw-Hill, New York, 1965, p. 319.*

Continuity of Species:

$$\rho \frac{\partial Y_k}{\partial t} + \rho u \frac{\partial Y_k}{\partial x} = R_k M_k - \frac{\partial}{\partial x} (\rho Y_k V_k), \ k = 1,2,\ldots N \tag{2}$$

and

Conservation of Energy:

$$\rho c_p \frac{\partial T}{\partial t} + \rho u c_p \frac{\partial T}{\partial x} = \frac{\partial}{\partial x} \left(\lambda \frac{\partial T}{\partial x}\right) - \rho \sum_{k=1}^{N} c_{pk} Y_k V_k \frac{\partial T}{\partial x} - \sum_{k=1}^{N} R_k M_k h_k, \tag{3}$$

where the variables are defined in the glossary. In equations (2) and (3), V_k is determined from the relation

$$\frac{\partial X_k}{\partial x} = \sum_{j=1}^{N} \frac{X_k X_j}{D_{kj}} (V_j - V_k). \tag{4}$$

For the thermal equation of state we have taken the ideal gas law,

$$p = \rho RT \sum_{k=1}^{N} Y_k / M_k , \tag{5}$$

and for the caloric equation of state we use

$$h_k = h_k^{\circ} + \int_{T_o}^{T} c_{pk} \ dT. \tag{6}$$

From the definition of mass fraction, mole fraction and diffusion velocity we have the identities:

$$\sum_{k=1}^{N} Y_k = 1, \tag{7}$$

$$X_k = (Y_k/M_k)/\sum_{j=1}^{N} (Y_j/M_j), \tag{8}$$

and

$$\sum_{k=1}^{N} Y_k V_k = 0, \tag{9}$$

respectively. Notice that equation (9) can be used to eliminate one of the diffusion velocities in equation (4). (In our coding for the ozone flame we chose to eliminate the diffusion velocity of molecular oxygen.)

In order to avoid solving equation (1) explicity we introduce a Lagrangian coordinate ψ such that

$$\psi(x,t) = \int_0^x \rho(x',t)dx'. \tag{10}$$

Then $\dfrac{\partial\psi}{\partial x} = \rho$ and (11)

$$\frac{\partial\psi}{\partial t} = \int_0^x \frac{\partial\rho(x',t)}{\partial t}\, dx' = -\int_0^x \frac{\partial}{\partial x'}(\rho u)dx' = -\rho u + m_o(t), \tag{12}$$

where $\rho u|_{x=0} = m_o(t)$. With this notation equations (2) and (3) become

$$\frac{\partial Y_K}{\partial t} + m_o \frac{\partial Y_k}{\partial\psi} = R_k M_k/\rho - \frac{\partial}{\partial\psi}(\rho Y_k V_k) \tag{13}$$

and

$$\frac{\partial T}{\partial t} + m_o \frac{\partial T}{\partial\psi} = \frac{1}{c_p}\frac{\partial}{\partial\psi}\left(\rho\lambda\frac{\partial T}{\partial\psi}\right) - \sum_{k=1}^{N} \frac{c_{pk}}{c_p}\rho Y_k V_k \frac{\partial T}{\partial\psi} - \frac{1}{\rho c_p}\sum_{k=1}^{N} R_k M_k h_k, \tag{14}$$

respectively. For initial conditions symmetric about x=0 only the interval $0 \leq x \leq \infty$ need be considered and the boundary conditions are

$$\frac{\partial T}{\partial\psi} = 0 \text{ at } x = 0 \text{ and } x = \infty \text{ for } t \geq 0$$

and

$$\frac{\partial Y_k}{\partial\psi} = 0\ , \quad (k = 1,2,\ldots N) \text{ at } x = 0 \text{ and } x = \infty \text{ for } t \geq 0. \tag{15}$$

In principle, specification of the initial conditions and integration of equations (13) and (14) subject to the boundary conditions of equations (15) would provide steady state profiles of the temperature, T, and the mass fractions, Y_k.

In practice there are a few steps that remain before actual coding of these equations can begin. For numerical convenience we introduce several non-dimensional quantities:

$$t^* = t/t_\infty,$$

$$\psi^* = \psi/\psi_\infty,$$

and

$$T^* = T/T_\infty.$$

where

$$t_\infty = 5 \times 10^{-5} \text{ seconds},$$

$$\psi_\infty = 5 \times 10^{-6} \text{ gm-cm}^{-2}, \text{ and}$$

$$T_\infty = 300\text{K}.$$

These quantities pertain to the particular case of an ozone flame, though they might possibly be convenient divisors for other flames, too. It is also convenient to define $m_o^* = (t_\infty/\psi_\infty)\, m_o$. The dimensionless forms of equations (13) and (14) are

$$\frac{\partial Y_k}{\partial t^*} + m_o^* \frac{\partial Y_k}{\partial \psi^*} = \frac{5 \times 10^{-5}}{\rho} R_k M_k - 10 \frac{\partial}{\partial \psi^*} (\rho Y_k V_k) \tag{16}$$

and

$$\frac{\partial T^*}{\partial t^*} + m_o^* \frac{\partial T^*}{\partial \psi^*} = \frac{2 \times 10^6}{c_p} \frac{\partial}{\partial \psi^*} \left(\rho\lambda \frac{\partial T^*}{\partial \psi^*}\right)$$

$$- 10 \sum_{k=1}^{N} \frac{c_{pk}}{c_p} \rho Y_k V_k \frac{\partial T^*}{\partial \psi^*} - \frac{5 \times 10^{-7}}{3\, \rho c_p} \sum_{k=1}^{N} R_k M_k h_k, \tag{17}$$

respectively.

For the ozone flame

$$\begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \end{bmatrix} = \begin{bmatrix} O \\ O_2 \\ O_3 \end{bmatrix} ,$$

and molecular oxygen concentration is eliminated by using equation (7).

Equations (16) and (17) are actually used to find the steady state mass fraction and temperature profiles.

For the particular case of ozone these profiles have not yet been measured. Only the flame speed as a function of initial mole fraction of ozone has been recorded[5]. The flame speed, u_f, is the propagation velocity of the flame relative to the fluid at rest; i.e., at infinity. To compute u_f first the steady state profiles are required. Then we transform to a coordinate system in which the origin moves through the fluid at the same speed as the propagation velocity of the flame relative to the fluid at the origin. In this coordinate system all variables are independent of time; specifically,

$$\frac{\partial Y_k}{\partial t} = 0 ,$$

$$\frac{\partial \rho}{\partial t} = 0 ,$$

and from equation (1)

$$\frac{\partial (\rho u)}{\partial x} = 0.$$

We now compute the flame speed from the mass fraction profile(s). Take any one of the equations (2) and integrate to obtain

$$\int_0^\infty \rho u \frac{\partial Y_k}{\partial x} dx = \int_0^\infty R_k M_k dx - \int_0^\infty \frac{\partial}{\partial x} (\rho Y_k V_k) dx . \tag{18}$$

---

[5] *A. G. Streng and A. V. Grosse, "The Ozone to Oxygen Flame", Sixth Symposium (International) on Combustion, August 1956, Reinhold Publishing Corporation, New York, 1957, pp. 264-273.*

Performing the integration of equation (18) we find

$$\rho u[Y_k(\infty) - Y_k(0)] = \int_0^\infty R_k M_k dx - \rho Y_k V_k \Big|_0^\infty . \tag{19}$$

To evaluate the upper limit of the last term of equation (19) we know from equations (15) that

$$\frac{\partial Y_k}{\partial x}\Big|_0 = \frac{\partial Y_k}{\partial x}\Big|_\infty = 0.$$

Employing equation (8) we find that the boundary conditions can be written as

$$\frac{\partial X_k}{\partial x}\Big|_0 = \frac{\partial X_k}{\partial x}\Big|_\infty = 0.$$

Then equation (4) evaluated at the boundaries is

$$0 = \sum_{j=1}^{N} \frac{X_k X_j}{D_{kj}} (V_j - V_k),$$

which admits only the unique solution $V_j = 0$ for all j. Thus the last term of equation (19) vanishes and we obtain

$$u(\infty) = \frac{\int_0^\infty R_k M_k dx}{\rho(\infty)[Y_k(\infty)-Y_k(0)]} = \frac{\int_0^\infty \rho^{-1} R_k M_k d\psi}{\rho(\infty)\ Y_k(\infty)-Y_k(0)]} . \tag{20}$$

In this coordinate system, $u_f = - u(\infty)$.

III. PDECOL - A GENERAL PDE SOLVER

The package PDECOL, developed by Madsen and Sincovec[6] was used to solve this problem. This is a general package for solving partial differential equations (PDE's) using the method of lines. The spatial

---

[6] *N. K. Madsen and R. F. Sincovec, "PDECOL: General Collocation Software for Partial Differential Equations", Preprint UCRL-78263 (Rev. 1), Lawrence Livermore Laboratory, 1977.*

discretization is accomplished by finite element collocation methods based on B-splines. The time integration is done using a predictor-corrector method, based on the algorithm of Gear.

The basic assumption is that the solution can be written in the form

$$Y_k \cong \sum_{i=1}^{NC} C_k^{(i)}(t^*)\, B_i(\psi^*),\quad k=1\ldots N, \quad \text{and}$$

$$T^* \cong \sum_{i=1}^{NC} C_{N+1}^{(i)}(t^*)\, B_i(\psi^*),$$

where the functions $B_i(\psi^*)$, $i=1 \ldots NC$, span the solution space for any fixed t^* to within a small error tolerance. The time dependent coefficients $C_k^{(i)}$ are determined uniquely by requiring that the expansion above satisfy the given boundary conditions and that they satisfy equations (16) and (17) at (NC-2) interior (collocation) points. If there is a null boundary condition, an extra collocation point is added. The resulting set of coupled nonlinear ordinary differential equations (ODE's) may be numerically integrated by standard means.

The basis functions B_i used are B-splines. A program developed by C. deBoor[7] has been incorporated into PDECOL to handle the generation and evaluation of the B-splines. These basic functions are piecewise polynomials. To define them, the user must supply a set of strictly increasing breakpoints. The piecewise polynomials are joined at these breakpoints (or knots). KORD, the order of the piecewise polynomials and NCC, the number of continuity conditions at the breakpoints must also be specified. Here, the expansion and its first (NCC-1) derivatives will be continuous at the breakpoints. Then the program will generate a set of NC = KORD(NB-1) - NCC(NB-2) basis functions and collocation points, where NB is the number of breakpoints.

By definition, a B-spline will equal zero except over a small interval. At any collocation point at most KORD of the B-splines will be non-zero. Because of this the system of ODE's for the coefficients $C_k^{(i)}$ will not be fully coupled.

This system of ODE's is integrated in time, using a variant of the Gear integrator. The appropriate banded Jacobian is generated internally by the program. Once the integrator has reached a desired output time t^*, the values of Y_k and T^* can be obtained for any ψ^* by substituting into the expansion.

---

[7] *C. de Boor, "Package for Calculating with B-splines", Siam. J. Numer. Anal.* <u>*14*</u>*, 441-472, 1977.*

The user must supply a main program and three subroutines. In the main program the left and right spatial boundaries are specified. The order of the piecewise polynomials is specified, and the continuity and smoothness requirements where the piecewise polynomials are joined. The error criterion for the time integration is chosen. The most important factor is the definition of the breakpoints for the spatial discretization. They must be chosen close enough so that the spatial errors will not be too large. There is at present no way of determining an optimum set of breakpoints for a given problem other than trial and error.

The three subroutines BNDRY, UINIT and F are straightforward. In BNDRY the boundry conditions are specified. UINIT gives original spatial profiles for the unknowns, while F gives the formulas for determining the partial derivative of the unknown variables with respect to time.

IV. PRIOR IMPLEMENTATION

Margolis[8] has attached the ozone flame problem using PDECOL. However, his approach has some serious disadvantages. Margolis solves the problem only for an initial ozone mole fraction of .25, (balance molecular oxygen), using a number of simplifications in the input data. These may affect the validity of his results, but do not change the basic numerical problem. (The details of our input parameters for the ozone test case will be reported elsewhere[9]).

To perform the integration, he makes the standard assumption that the fluid is at rest initially, that is, $m_0 = 0$ in equations (16) and (17). His boundary conditions correspond to zero flux of species and heat at the origin and at infinity. In practice, the boundary condition for $x = \infty$ must be applied at a finite value of x. This must be chosen large enough so that the flame front remains sufficiently remote from the boundary so that it is not affected by the boundary's finite location. Margolis chose the value $\psi = 50$.

His initial conditions[8] are given by

$$Y_1 = \begin{cases} 0.0005 \cos^5 [\frac{\pi}{2} (\frac{\psi^*}{1.2})^7], & 0 \le \psi^* \le 1.2 \\ 0, & \psi^* > 1.2 \end{cases} \tag{21}$$

$$Y_2 = \begin{cases} 2/3 + (\frac{1}{3} - .0005) \cos^5 [\frac{\pi}{2} (\frac{\psi^*}{1.2})^7], & 0 \le \psi^* \le 1.2 \\ 2/3, & \psi^* > 1.2 \end{cases}$$

---

8 *S. B. Margolis, "Time Dependent Solution of a Premixed Laminar Flame", J. Comp. Phys.* <u>*27*</u>*, 410-427, 1978.*

9 *J. M. Heimerl and T. P. Coffee, "The Detailed Modeling of Premixed, Laminar Steady-State Flame to Obtain Validated Reaction Networks I. Ozone". (Manuscript in preparation).*

$$Y_3 = 1 - Y_1 - Y_2, \text{ and}$$

$$T^* = \begin{cases} 1.0 + 3.166667 \cos^5[\frac{\pi}{2} (\frac{\psi^*}{1.2})^7] , & 0 \le \psi^* \le 1.2, \\ 1.0, & \psi^* > 1.2. \end{cases}$$

The conditions at $\psi^* > 1.2$ are those of the unburned medium and the conditions at $\psi^* = 0$ correspond roughly to a hot pocket of nearly burned gas.

The order of the B-splines was chosen to be 6, and the B-splines and their first four derivatives were chosen to be continuous at the breakpoints. Margolis' maximum breakpoint spacing was 0.2 corresponding to 250 breakpoints. His modified equations (13) and (14) were then integrated forward in time until the profiles stabilized. The flame front moves from the origin toward $x = \infty$ and the integration must be stopped before the boundary at infinity (here at $\psi^* = 50$) is reached.

The large number of breakpoints create a need for a large amount of computer "memory". In fact, working on a CDC 7600 in single precision, we could only use about 210 breakpoints because of local small core memory restrictions ($\sim$ 60,000 words). Also, in this coordinate system, it is not obvious when a steady state flame has developed, since the flame, proceeding toward $x = \infty$, is moving relative to a stationary coordinate system.

Moreover, we wanted results for higher initial mole fractions of O_3. But the higher the concentration of O_3, the faster the flame and the narrower the flame-front. As a consequence the finite location of the boundary at infinity would have to be larger to allow the flame sufficient space to develop. Yet, the breakpoints must be closer to adequately model the flame-front. And as a consequence the straightforward implementation of Margolis' method becomes impractical.

V. A MODIFIED FLAME CODE*

We have made substantial improvements by taking advantage of the peculiarities of a flame-front. In the unburned region, practically no changes occur. Most of the changes occur within the relatively narrow flame-front. In the burned regions reactions occur relatively slowly.

**The nomenclature for sections V and VI is defined at the end of section VI.*

What we would like to have is a method of concentrating our breakpoints in the steep flame-front, where greater accuracy is necessary. In principal, a simple way to do this is to transform to a moving coordinate system in which the flame-front is stationary.

First, let us consider the fixed dimensionless coordinate system defined by the standard assumption that no mass flux flows through the origin, i.e., that

$$m_o{}^*(t^*) = (t_\infty/\psi_\infty)\rho u\big|_{x=o} = 0.$$

The origin of this fixed system is defined by O_F in Figure 1. Let $\psi_F{}^*(t^*)$ be a point in the flame front in this system, and

$$S_F{}^* = \frac{\partial\psi_F^*}{\partial t^*}$$

be the speed of the flame in the fixed system. For steady state conditions

$$d\psi_F{}^*/dt^* = 0 \ ,$$

or

$$\partial\psi_F{}^*/\partial t^* \ + \ (\partial\psi_F{}^*/\partial x)\ dx/dt^* = 0.$$

Identifying terms we have

$$S_F{}^* \ + \ (\rho/\psi_\infty)(t_\infty u) = 0,$$

or

$$S_F{}^* = -\ (t_\infty/\psi_\infty)\rho u,$$

where ρu is the constant mass flux through the flame in the x coordinate system.

Next, again referring to Figure 1, we define a moving coordinate system with origin O_M in such a way that a point on the flame profile remains at a fixed distance, ψ_M^* from O_M. This transformation can be simply implemented by letting $m_o{}^*(t) = -S_F{}^*$ in equations (16) and (17). In this case the mass flux through the moving origin, O_M, equals the mass flux through the flame.

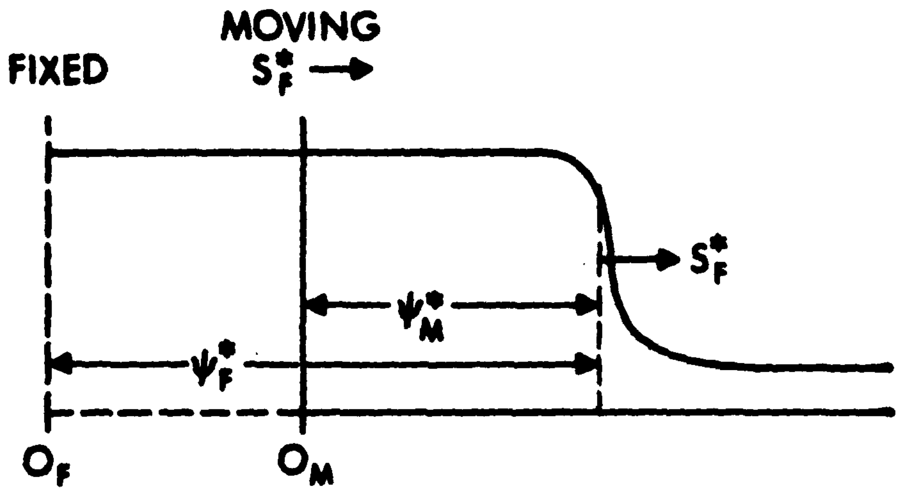


Fig. 1 The curve represents a convenient steady state temperature (or mass fraction) profile that defines a flame. A description of this flame in the fixed, O_F, and in the moving, O_M, coordinate system serves to define the relationship between the two coordinate systems.

In practice, we do not know S_F^* at the beginning of the integration, and the flame will not be at steady state. We will now describe an algorithm to define $m_o^*(t^*)$ such that $m_o^*(t^*)$ approaches $-S_F^*$ as the flame approaches steady state. In the process, we shall require that the flame-front remain about the same distance from the moving origin O_M, in order that the breakpoints can be concentrated in the narrow range of the flame-front.

VI. ALGORITHM TO DEFINE $m_o^*(t^*)$

We chose the same conditions on the B-splines and boundary conditions as before. (The sensitivity of the computed flame speed on the order of B-spline is discussed in Appendix A). The boundary at infinity is placed at $\psi^* = 25$. For the case of .25 mole fraction we choose the initial conditions:

$$Y_2 = \begin{cases} 0.9995 \,, & \psi^* \le 9.9575 \\ 2/3 + \left(\frac{1}{3} - .0005\right) \cos^5\left[\frac{\pi}{2}\left(\frac{\psi^* - 9.9575}{1.2}\right)^7\right], & 9.9575 \le \psi^* \le 11.1575 \\ 2/3 \,, & \psi^* \ge 11.1575 \end{cases} \tag{22}$$

$$Y_3 = \begin{cases} 10^{-7} \,, & \psi^* \le 9.9575 \\ \frac{1}{3} - \frac{1}{3}\cos^5\left[\frac{\pi}{2}\left(\frac{\psi^* - 9.9575}{1.2}\right)^7\right] \,, & 9.9575 \le \psi^* \le 11.1575 \\ \frac{1}{3} \,, & \psi^* \ge 11.1575 \end{cases}$$

$$Y_1 = 1 - Y_2 - Y_3$$

$$T^* = \begin{cases} 4.166667 \,, & \psi^* \le 9.9575 \\ 1.0 + 3.166667 \cos^5\left[\frac{\Pi}{2}\left(\frac{\psi^* - 9.9575}{1.2}\right)^7\right] \,, & 9.9575 \le \psi^* \le 11.1575 \\ 1.0 \,, & \psi^* \ge 11.1575 \end{cases}$$

These profiles correspond roughly to a flame-front where $Y_2 = 0.8$ at $\psi_F^* = 11.0$. Arbitrarily, we try to keep the flame centered at this point, that is $Y_2(\psi_M^*, t^*) = 0.8$ at $\psi_M^* = 11$. (We define $\psi_C^*(t^*)$ as the value of ψ^* for which $Y_2(\psi^*, t^*) = 0.8$).

A total of 58 breakpoints are chosen. Across the flame-front, $10 \leq \psi^* \leq 13$, the breakpoint spacing is 0.15, gradually increasing to 0.5 at $\psi_M^* = 0$ and 1.0 at $\psi_M^* = 25$.

For numerical reasons, we would like the function $m_o^*(t^*)$ to be continuous. For simplicity, we let it be a piecewise straight line.

We define a sequence of output times t_o^*, t_1^*, t_2^*, ..., t_k^* where m_o^* will be redefined. This must be done often enough to keep up with changes in the propagation velocity as the flame approaches steady state. The function m_o^* is then defined by the values at these ouput times, that is $m_o^*(t_o^*) = - S_o^*$, $m_o^*(t_1^*) = - S_1^*$ and so on. The problem lies in determining an appropriate sequence S_i^* such that $S_i^* \to S_F^*$.

To begin the iteration, we make a guess as to the initial, unrelaxed flame speed, S_{Fo}^*. We let $S_{Fo}^* = S_o^* = S_1^*$, so $m_o^*(t_o^*) = - S_o^*$ in the interval $t_o^* \leq t^* \leq t_1^*$, as depicted in Figure 2. By interpolation, we can find $\psi_{FC}^*\ (t_o^*) = \psi_{MC}^*\ (t_o^*)$, the preselected centering point of the flame front at t_o^*. At this starting time, the fixed coordinate system and the moving coordinate system have a common origin.

We then integrate to time t_1^* and find $\psi_{MC}^*\ (t_1^*)$. The relationship between fixed and moving coordinate system at time t_1^* is shown in Figure 3 and is given by

$$\psi_{FC}^*\ (t_1^*) = \psi_{MC}^*\ (t_1^*) + S_o^*\ (t_1^* - t_o^*).$$

The average speed of the flame between t_o and t_1, denoted S_{F1}^*, is defined by

$$S_{F1}^* = (\psi_{FC}^*\ (t_1^*) - \psi_{FC}^*\ (t_o^*))/(t_1^* - t_o^*). \tag{23}$$

We need to determine the next value, $m_o^*(t_2^*) = S_2^*$. One simple algorithm is to choose $S_2^* = S_{F1}^*$. Then at $t^* = t_2^*$, the origin O_M will be moving at the average speed of the flame-front between t_o^* and t_1^*. As the flame approaches steady state, S_{Fi}^* will approach S_F^* and the origin will eventually match the flame speed.

The disadvantage of this procedure is the time lag in adjusting O_M (See Figure 4). During this lag the flame can drift away from its initial position at $\psi_M^* = 11$. If we were to adopt this algorithm the region of dense breakpoints would have to be much larger to allow for drift.

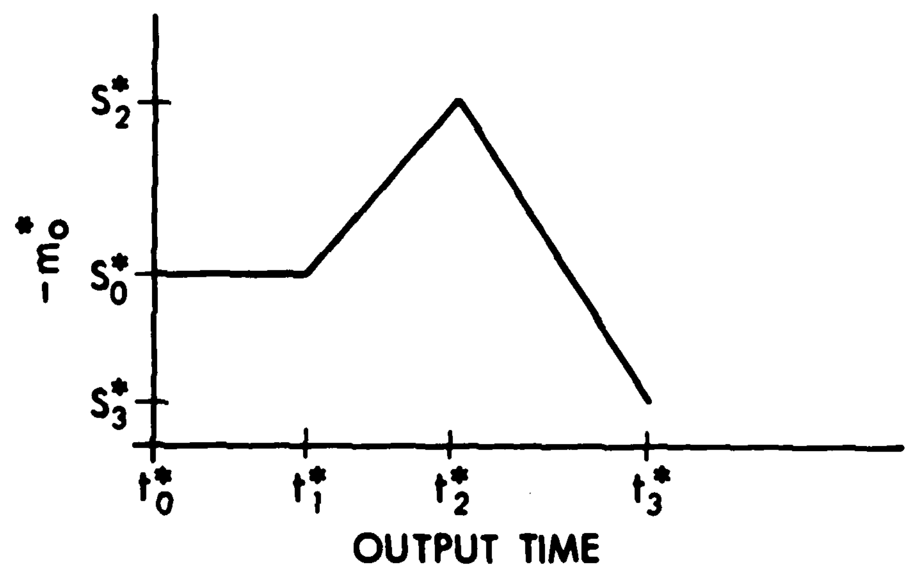


Fig. 2. A Sketch of m_o* as a Function of Non-dimensionalized Output Time. (S_1* = S_o*, see text).

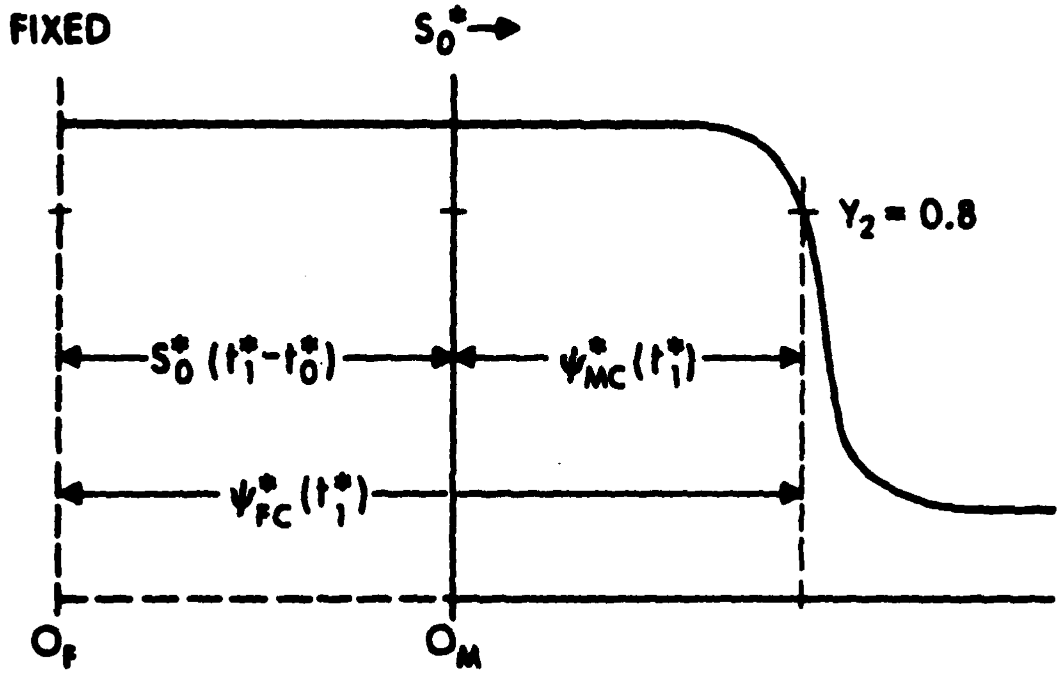


Fig. 3. Oxygen profile, Y_2, is used to illustrate the relationship between the fixed and moving coordinate system at time t_1^*.

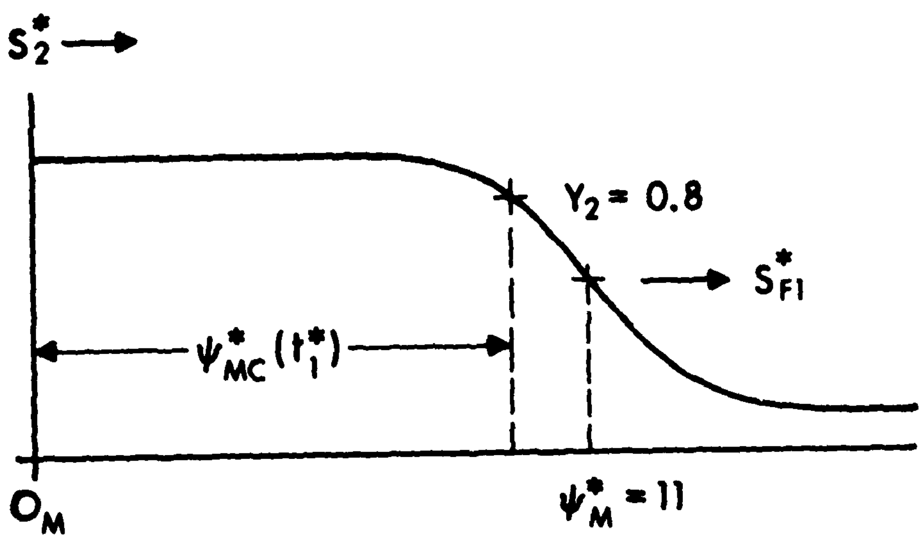


Fig. 4. The molecular oxygen mass fraction profile is used to characterize the flame front (exaggerated for clarity) at $t^* = t_1^*$. The case sketched here, $11.0 > \psi_{MC}^*$ (t_1^*) indicates the speed of the coordinate systems, S_2^*, should be increased to move ψ_{MC}^* closer to 11.0.

An alternative method to determine S_2^* is to try to recenter the flame by the time t_2^*. To do this, we assume that $S_{F2}^* = S_{F1}^*$. As the flame approaches steady state, this will become very nearly true. Consider the average speed of the moving coordinate system between t_1^* and t_2^*; i.e., $(S_1^* + S_2^*)/2$. If we let this average speed equal

$$S_{F1} + \frac{11 - \psi_{MC}^* (t_1^*)}{t_2^* - t_1^*}$$

the average speed will be modified just enough so that $\psi_{MC}^* (t_2^*) = 11$. This centering constraint is sufficient to determine S_2^*. Specifically:

$$S_2^* = 2\, S_{F1}^* - S_1^* + 2\, (11 - \psi_{MC}^*(t_1^*))/(t_2^*-t_1^*). \qquad (24)$$

However, this algorithm for S_2^* is observed to have numerical difficulties. In order to center the flame, the speed of the origin must substantially over-shoot the flame speed and the resulting oscillations are hard to follow numerically, as illustrated in Figure 2.

An effective empirical rule was found. We simply took the average of the above two values for S_2^*. That is

$$S_2^* = 3/2\, S_{F1*} - 1/2\, S_1^* + (11 - \psi_{MC}^* (t_1^*))/(t_2^*-t_1^*). \qquad (25)$$

This helps cut down oscillations in m_o^* while keeping the flame reasonably centered.

The generalization is straightforward at time t_i^*,

$$S_{Fi}^* = (\psi_{FC}^* (t_i^*) - \psi_{FC}^* (t^*_{i-1}))/(t_i^*-t_{i-1}^*) \qquad (26)$$

and

$$S_{i+1}^* = 3/2\, S_{Fi}^* - 1/2\, S_i^* + (11 - \psi_{MC}^* (t_i^*)/(t_{i+1}-t_i^*). \qquad (27)$$

Once the flame reaches steady state, $S_{Fi}^* = S_F^*$. The quantity $1 - \psi_{MC}^* (t_i^*)$ quickly goes to zero, and equation (27) rapidly approaches the desired steady state solution $S_i^* = S_{i+1}^* = S_{Fi}^* = S_F^*$.

A slight modification should be made in evaluating the flame speed. Since the coordinate system is moving, the gradients of the profiles are not necessarily zero at $\psi_M^* = 0$. So in running the code, no boundary conditions are imposed at $\psi_M^* = 0$ and V_k at $\psi_M^* = 0$ is not necessarily zero.

This term can be evaluated, and equation (20) becomes

$$u\ (\infty) = \frac{\int_0^\infty \rho^{-1} R_k M_k d\psi}{\rho(\infty)[Y_k(\infty) - Y_k(0)]} - \frac{\rho(0)\ Y_k(0)V_k(0)}{\rho(\infty)[Y_k(\infty) - Y_k(0)]}. \tag{28}$$

This additional term is found to be extremely small.

VII. SENSITIVITY OF THE COMPUTED FLAME SPEED UPON THE PARAMETERS USED IN THE NUMERICAL METHOD

A question that might be brought up is whether the computed values of the flame speed depend on the particulars of the numerical method. That is, is the flame speed sensitive to the number of intervals (NINT) chosen? (The number of breakpoints equals one plus the number of intervals). Is it sensitive to the order of the B-spline (KORD) or the number of continuity conditions imposed (NCC)? To answer these questions we executed the code starting from the fixed standard profile for each initial mole fraction of ozone to a fixed value for t*. (For an example of the 0.25 initial mole fraction standard profile see equation (22)). A value for t* was chosen to be more than adequate to insure that the steady state had been achieved i.e. t* (.25 ozone) = 40 and t* (1.00 ozone) = 10. [This procedure is contrasted to the more normal and less time consuming method of iterating from previously saved profiles]. The advantage gained in starting from standard profiles is that the run time is then a measure of the relative efficiency of the change.

Table I shows the effect of doubling the number of breakpoints (grid points) on the computed flame speed. The standard we had selected for the 0.25 initial mole fraction ozone case was 57 breakpoints and sixth order spline with five continuity conditions. The flame speed was computed for each species (see equation 20) and shows little difference when the breakpoints are doubled. One also sees that the value of the flame speed is independent of the species selected for use in equation (20). As expected the run time, execution time less initialization time, increases

TABLE I. SENSITIVITY OF COMPUTED FLAME SPEED TO CHANGES IN NUMBER OF BREAKPOINTS FOR INITIAL OZONE MOLE FRACTIONS OF 0.25 AND 1.00

INITIAL OZONE MOLE FRACTION	NINT	KORD	NCC	FLAME SPEED (cm/sec)			RUN TIME (SEC)	# INTEGRATION STEPS
				O	O_2	O_3		
0.25	57	6	5	62.5	62.5	62.5	31.4	236
	114	6	5	62.3	62.5	62.5	64.0	248
1.00	70	6	5	493	495	495	57.5	358
	140	6	5	493	495	495	88.5	289

with increasing number of breakpoints. The number of integration steps is also indicated and are a measure of the relative effort to attain the steady state condition. Rows 3 and 4 of this table show the same results for the 1.00 initial mole fraction ozone case. Here we see that the flame speed is essentially unchanged whether NINT in doubled or whether O, O_2, or O_3 is chosen to evaluate equation (20).

Tables II and III show the sensitivity of the computed flame speed to changes in the order of the B-spline selected and the number of continuity conditions imposed for initial mole fraction of O_3 of 0.25 and 1.00, respectively. The flame speed values are essentially all the same. By comparing run time and number of steps it can be seen that some improvement in run time could have been made over our standard case where KORD = 6 and NCC = 5.

In every case the previous t* printed out [t* (0.25 ozone) = 35 and t* (1.00 ozone) = 9] also showed steady state had been attained. There were no differences to three significant figures for the 0.25 initial ozone mole fraction. With the exception of the KORD = 3 and NCC = 2 case, again no differences to three significant figures were observed for the 1.00 initial ozone mole fraction. The KORD = 3 and NCC = 2 case showed differences of six parts in 500 for O, two parts in 500 for O_2 and three parts in 500 for O_3, still an adequately close approximation to steady state.

In conclusion, we find that the observed insensitivity indicates that our standard cases are sufficient to supply a numerically reliable solution.

VIII. SUMMARY

We have reported a method to efficiently solve the equations that characterize a one-dimensional, premixed, laminar steady-state flame propagating into an unbounded medium. The general package PDECOL that solves partial differential equations using the method of lines has been employed. Spacial discretization is accomplished by finite element collocation methods based on B-splines, while the temporal integration is accomplished by a Gear-type predictor-corrector method. We have implemented PDECOL in conjuction with a method of concentrating the breakpoints in the steep flame-front where great accuracy is required. A description of this method comprises the bulk of this report. The method enables one to solve efficiently the governing equations. Sensitivity tests concerning the evaluation of the flame speed for different numbers of breakpoints, different numbers of continuity conditions show that we in fact have obtained a numerically reliable solution.

TABLE II. SENSITIVITY OF COMPUTED FLAME SPEED TO CHANGES IN ORDER OF B-SPLINE AND NUMBER OF CONTINUITY CONDITIONS IMPOSED FOR THE INITIAL OZONE MOLE FRACTION OF 0.25

KORD	NCC	FLAME SPEED (cm/sec) O	O_2	O_3	RUN TIME (SEC)	# INTEGRATION STEPS
6	5	62.5	62.5	62.5	31.4	236
5	4	62.2	62.5	62.5	30.7	254
4	3	62.7	62.6	62.6	26.9	250
3	2	63.5	62.7	62.7	23.7	218
6	4	62.5	62.5	62.5	65.3	248
4	2	62.4	62.5	62.5	58.7	266

TABLE III. SENSITIVITY OF COMPUTED FLAME SPEED TO CHANGES IN ORDER OF B-SPLINE AND NUMBER OF CONTINUITY CONDITIONS IMPOSED FOR THE INITIAL OZONE MOLE FRACTION OF 1.00

		FLAME SPEED (cm/sec)				
KORD	NCC	O	O_2	O_3	RUN TIME (SEC)	# INTEGRATION STEPS
6	5	493	495	495	57.5	358
8	7	493	494	495	65.5	360
5	4	489	494	493	46.8	342
4	3	480	493	493	63.0	461
3	2	497	500	500	61.8	460
4	2	492	494	494	73.2	303
6	4	494	495	495	95.7	314

Nomenclature for Sections V and VI.
(All Quantities are Non-dimensional)

O_F The origin of the fixed coordinate system (that is, there is no mass flow through O_F).

O_M The origin of the moving coordinate system that will track the flame (at $t^* = 0$, $O_M = O_F$).

$S_F{}^*$ The speed of the steady state flame with respect to O_F (this is a constant).

$m_o{}^*$ The non-dimensional mass flow through O_M.

$\psi_F{}^*$ "Distance" in the fixed system.

$\psi_M{}^*$ "Distance" in the moving system.

ψ^*_{FC} The "distance" from O_F to some specified point in the flame-front.

ψ^*_{MC} The "distance" from O_M to some specified point in the flame-front.

$S_i{}^*$ The speed of O_M with respect to O_F at $t_i{}^*$ (equals the negative of the non-dimensional mass flow through O_M at $t_i{}^*$).

S^*_{Fi} The average speed of the flame with respect to O_F between t_{i-1}^* and $t_1{}^*$.

REFERENCES

1. J. O. Hirschfelder, C. F. Curtiss and R. B. Byrd, Molecular Theory of Gases and Liquids, John Wiley and Sons, New York, second printing with notes, March 1964.

2. R. B. Bird, W. E. Stewart and E. N. Lightfoot, Transport Phenomena John Wiley and Sons, New York, 1960.

3. F. A. Williams, Combustion Theory. Addison-Wesley, Reading Mass., 1965.

4. R. M. Fristrom and A. A. Westenberg, Flame Structure, McGraw-Hill, New York, 1965, p. 319.

5. A. G. Streng and A. V. Grosse, "The Ozone to Oxygen Flame", Sixth Symposium (International) on Combustion, August 1956, Reinhold Publishing Corporation, New York, 1957, pp. 264-273.

6. N. K. Madsen and R. F. Sincovec, "PDECOL: General Collocation Software for Partial Differential Equations", Preprint UCRL-78263 (Rev. 1), Lawrence Livermore Laboratory, 1977.

7. C. de Boor, "Package for Calculating with B-splines", Siam. J. Numer. Anal. 14, 441-472, 1977.

8. S. B. Margolis, "Time Dependent Solution of a Premixed Laminar Flame", J. Comp. Phys. 27, 410-427, 1978.

9. J. M. Heimerl and T. P. Coffee, "The Detailed Modeling of Premixed, Laminar, Steady-State Flames to Obtain Validated Reaction Networks I. Ozone", (manuscript in preparation).

APPENDIX A.
Recent Code Changes

A number of changes have recently been made in the code. The purpose is to make the code easier to modify, so that a minor change in a problem does not involve major changes in the code. Also, the form was chosen to be as easy as possible to extend to other flame simulations.

To begin the execution of the code, a number of input values are required. We must specify the right and left hand boundaries, ψ_R^* and ψ_L^* respectively. The corresponding starting temperatures, T_R^* and T_L^*, and mass fraction, Y_{kR} and Y_{kL}, must also be given. In addition, we need to specify t_f^*, the final integration time. Finally, we specify NINT, the numbers of intervals chosen, KORD, the degree of the piecewise polynomial functions, and NCC, the number of continuity conditions.

The subroutine UINIT will now automatically generate the initial guessed profile. Since the exact form of the profile is not important, so we use the simple formula

$$T = \begin{cases} T_L \quad , & \psi_L^* \le \psi^* \le \psi_L^* + 0.4\ (\psi_R^* - \psi_L^*) \\ T_L + \dfrac{(T_R - T_L)\ [\psi^* - \psi_L^* - 0.4\ (\psi_R^* - \psi_L^*)]}{0.2\ (\psi_R^* - \psi_L^*)}\ , & \\ \psi_L^* + 0.4\ (\psi_R^* - \psi_L^*) \le \psi^* \le \psi_L^* + 0.6\ (\psi_R^* - \psi_L^*) & \\ T_R \quad , & \psi_L^* + 0.6\ (\psi_R - \psi_L^*) \le \psi^* \le \psi_R^* \end{cases}$$

with corresponding formulas for the Y_k.

The breakpoint sequence is also chosen automatically, with the breakpoints concentrated in the center of the interval (ψ_L^*, ψ_R^*).

The program will produce the original guess S_o^* for the speed of the origin. It does this by numerically integrating equation (20) for one of the given Y_k profiles. This can be converted to a corresponding mass flow rate m_o^*.

The sequence of output times is also selected automatically. As $|\ S_{Fi}^* - S_{Fi-i}^*|$ decreases, $|\ t_{i+1}^* - t_i^*\ |$ is increased.

When the end time t_f^* is reached, a data file is created. This contains the collocation points with the corresponding temperature and mass fractions. It also contains the last value for the speed of the origin, S_i^*, and the left and right boundaries of the flame front. For convenience, the left boundary is defined as the value of ψ^* for which

$$T^* = T_L^* + 0.1\ (T_R^* - T_L^*)\ ,$$

and the right boundary as the value of ψ^* for which

$$T^* = T_R^* - 0.1\ (T_R^* - T_L^*).$$

The integration can be restarted from this data file by setting NSTART = 2 within the program. If desired, any of the parameters ψ_R^*, ψ_L^*, NINT, KORD, or NCC may be changed.

The breakpoint sequence is again generated. For this case, the breakpoints are adjusted so the greatest concentration is in the given flame front.

The old collocation points and function values are read in the subroutine UINIT. If ψ_L^* or ψ_R^* have been changed, these values are adjusted to recenter the flame. If NINT, KORD, or NCC have been changed, the set of collocation points generated by PDECOL will be different. In this case, the required values at the new collocation points are found by interpolation. The original guess for S_0^* is the value for the old speed of the origin.

The boundary conditions specified in BNDRY have also been modified. We now use the conditions

$$\left.\begin{aligned} T^* &= T_L^* \\ Y_k &= Y_{kL} \end{aligned}\right\} \qquad \psi^* = \psi_L^*$$

$$\left.\begin{aligned} \frac{\partial T^*}{\partial \psi^*} &= 0 \\ \frac{\partial Y_k}{\partial \psi^*} &= 0 \end{aligned}\right\} \qquad \psi^* = \psi_R^*\ .$$

Then even if ψ_L is accidentally chosen too close to the flame front, we will still have the proper values for T and Y_k.

The subroutine F and RT give the information necessary to compute the time derivatives of T and Y_k. These are complicated subroutines that depend on the specific kinetics and transport properties used.

We have developed a loader program that will write these subroutines, given the appropriate data. So changes in kinetics and transport, or changes to another flame, can be made easily.

APPENDIX B.

Comment on the Method of Spalding and Patankar.

We have recently become aware of another program for solving steady state flame problems. Like the method documented here, it concentrates the spacial grid in the flame front.

This procedure is based on a computer code developed by Spalding and Patankar[B1]. It is applicable to systems of parabolic equations of the form

$$\frac{\partial \phi}{\partial x} + (a+b\omega)\frac{\partial \phi}{\partial \omega} = \frac{\partial}{\partial \omega}\left(c\,\frac{\partial \phi}{\partial \omega}\right) + d, \tag{B1}$$

where a and b are functions of x, and c and d are arbitrary functions. The code was originally used to predict boundary-layer phenomena. The method was modified by Spalding, Stephenson, and Taylor[B2] to solve the equations of one-dimensional laminar flame propagation.

We can start with our set of equations (1), (2), and (3). The standard transformation

$$\frac{\partial \psi}{\partial x} = \rho \quad \frac{\partial \psi}{\partial t} = -\rho u \tag{B2}$$

is applied. Note that there is no mass flow through the left boundary. Equation (2) becomes

$$\frac{\partial Y_k}{\partial t} = -\frac{\partial}{\partial \psi}(\rho Y_k V_k) + R_k M_k/\rho \tag{B3}$$

and the temperature equation is similar.

To put this in a form similar to equation (B1), we need to apply some form equivalent to Ficks law. The most common approximation is

---

B1. D. B. Spalding and S. V. Patankar, <u>*Heat and Mass Transer in Boundary Layers*</u>*, Morgan-Grampian, London, 1967.*

B2. D. B. Spalding, P. L. Stephenson, and R. G. Taylor, "A Calculation Procedure for the Prediction of Laminar Flame Speeds", Combust. Flame <u>*17*</u>*, 55-64, 1971.*

$$\rho \, Y_k \, V_k = - \rho^2 \, D_k \, \frac{\partial Y_k}{\partial \psi} \, , \tag{B4}$$

where D_k is some diffusion coefficient. (More complicated relationships between V_k and $\partial Y_k / \partial \psi$ can be put into this form). Then equation (B3) becomes

$$\frac{\partial Y_k}{\partial t} = \frac{\partial}{\partial \psi} \left(\rho^2 \, D_k \, \frac{\partial Y_k}{\partial \psi} \right) + R_k M_k / \rho . \tag{B5}$$

To concentrate the computational effort within the flame-front, another transformation is made. We define $\omega = (\psi - \psi_u) \, / \, (\psi_b - \psi_u)$, where ψ_u is a point near the unburned boundary of the flame-front and ψ_b is a point near the burned boundary of the flame. Then $\partial \psi_u / \partial t = - m_u$ and $\partial \psi_b / \partial t = - m_b$, where m is the mass flow through the points. At steady state, $m_u = m_b = \rho u$ is constant. Equation (B5) now is written

$$\frac{\partial Y_k}{\partial t} + \frac{m_u + \omega \, (m_b - m_u)}{\psi_b - \psi_u} \, \frac{\partial Y_k}{\partial \omega} = \frac{\partial}{\partial \omega} \left(\rho^2 \, D_k \, \frac{\partial Y_k}{\partial \psi} \right) / (\psi_b - \psi_u)^2 + \frac{R_k M_k}{\rho} \, . \tag{B6}$$

The critical part of the code is the selection of m_u and m_b, so that they approach the steady state mass flow without causing instability in the code. In another paper, Spalding and Stephenson[B3] give suggestions for such "entrainment laws", or formulas for determining mass flow rates.

The two codes have the same basic aim, to concentrate attention on the flame-front. Both codes accomplish this by properly choosing mass flow rates, either through one or two points.

The Spalding and Patankar code is restricted to equations of the form (B1). Also, the code uses finite differences, instead of the finite element method of PDECOL.

B3. *D. B. Spalding and P. L. Stephenson, "Laminar Flame Propagation in Hydrogen and Bromine Mixtures", Proc. R. Soc. London* A.324, *315-337, 1971.*

APPENDIX C.
Program listing

A listing of the computer code follows. The subroutines MAIN,F, UINIT, FLSP, RT, BNDRY, DECOMP, and SOLVE are written for laminar flame problems. The code is set up to compute an ozone flame with an initial ozone mole fraction of 0.25.

The rest of the subroutines are from PDECOL. The only change has been made in the driver routine. The error criterion has been changed so that any quantity smaller than a user supplied parameter SREC is controlled by an absolute error criterion instead of a relative error criterion. This prevents the program from overworking when concentrations become negligible. Also, the collocation points are placed in the vector XCPTS, to permit their use in MAIN.

We have also included a listing of the loader programs. The first routine, LOADF, writes the part of the code concerned with computing the transport coefficients. LOAD writes the part of code concerned with the kinetics. LOAD can also write the appropriate subroutine for EPISODE, an ordinary differential equation solver. Both codes write the subroutine on a file called TAPE3. This can be cataloged and then attached to the main program. The routines are also written on the output file (TAPE6) along with additional information on the input values.

```
PROGRAM MAIN        76/76    OPT=1 ROUND=+-*/ TRACE          FTN 4.6+452          03/19/78  14.26.17        PAGE     1

          PROGRAM MAIN(OUTPUT,TAPE3=OUTPUT,TAPE1,TAPE2,TAPE4,                    MAIN       2
         * TAPE9,TAPE10)                                                         MAIN       3
     C  OZONE FLAME.                                                             MAIN       4
          COMMON/TABSP/LB(4)                                                     MAIN       5
          COMMON/ENDPT/PH0,PH5                                                   MAIN       6
          COMMON/TABP/PRESS,PSR                                                  MAIN       7
          COMMON/GEAR0/DTUSED,NQ,NSTEPS,NF,NJ                                    MAIN       8
          COMMON/TABF/SREC,PHCPTS(100)                                           MAIN       9
          COMMON/START/NSTART                                                    MAIN      10
          COMMON/TABAB/ASP,BSP,TPN,PHN,TMN                                       MAIN      11
          COMMON/MAIN/SCTCH(50),WORK(20000),IWORK(1200)                          MAIN      12
          DIMENSION PHBKPT(31)                                                   MAIN      13
          COMMON/TABIN/PHFFL,PHFFR,PHCTO,PHCT,UC(3),UB(3)                        MAIN      14
          DIMENSION UT(3)                                                        MAIN      15
          COMMON/TABW/W(3)                                                       MAIN      16
          COMMON/OUT/PHVAL(241)                                                  MAIN      17
          COMMON/OUTP/U1(241),U(3,241)                                           MAIN      18
     C  ************************                                                 MAIN      19
     C  ************************                                                 MAIN      20
     C  THE FOLLOWING INPUT DATA MUST BE SPECIFIED.                              MAIN      21
     C  DIMENSIONING IN MAIN, F, UINIT, FLSP, RT, DECOMP, AND SOLVE              MAIN      22
     C  MUST BE CORRECT.                                                         MAIN      23
          LB(1)=10HTEMP/1000                                                     MAIN      24
          LB(2)=10HO                                                             MAIN      25
          LB(3)=10HO3                                                            MAIN      26
          LB(4)=10HO2                                                            MAIN      27
          PRESS=1.0                                                              MAIN      28
          TPN=1000.                                                              MAIN      29
          PHN=5.0E-6                                                             MAIN      30
          TMN=5.0E-5                                                             MAIN      31
          W(1)=32.                                                               MAIN      32
          W(2)=16.                                                               MAIN      33
          W(3)=48.                                                               MAIN      34
          UB(1)=1256./TPN                                                        MAIN      35
          UC(1)=300./TPN                                                         MAIN      36
          UB(2)=1.0E-5                                                           MAIN      37
          UC(2)=1.0E-10                                                          MAIN      38
          UB(3)=1.0E-5                                                           MAIN      39
          UC(3)=1./3.                                                            MAIN      40
          NPDE=3                                                                 MAIN      41
          TFINAL=30.0                                                            MAIN      42
     C  LEFT BOUNDARY.                                                           MAIN      43
          PH0=0.0                                                                MAIN      44
     C  RIGHT BOUNDARY.                                                          MAIN      45
          PH5=15.0                                                               MAIN      46
     C  ERROR IS MEASURED BY (YC-YP)/MAX(YC,SREC)                                 MAIN      47
          SREC=1.0E-5                                                            MAIN      48
     C  IF NSTART = 1, BEGIN FROM A GUESSED PROFILE.                             MAIN      49
     C  IF NSTART = 2, BEGIN FROM A PROFILE ON TAPE1.                            MAIN      50
          NSTART=2                                                               MAIN      51
     C  NINT MUST BE A MULTIPLE OF TEN.                                          MAIN      52
          NINT=30                                                                MAIN      53
          KORD=4                                                                 MAIN      54
          NCC=2                                                                  MAIN      55
          T0=0.0                                                                 MAIN      56
          EPS=1.0E-4                                                             MAIN      57
          MF=22                                                                  MAIN      58
```

```
      IWORK(1)=20000                                              MAIN
      IWORK(2)=1200                                               MAIN
C  IF KFILE = 1, CREATE A RESTART FILE.                          MAIN
      KFILE=1.                                                    MAIN
C  ************************                                      MAIN
C  ************************                                      MAIN
      WRITE(3,7)PRESS                                             MAIN
7     FORMAT(/5X,10HPRESSURE =,1PE14.4,2X,4HATM,/)               MAIN
      PSR=PRESS/82.05                                             MAIN
      YSSM=0.0                                                    MAIN
      Y1=1.0                                                      MAIN
      DO 5 K=2,NPDE                                               MAIN
      Y1=Y1-UC(K)                                                 MAIN
5     YSSM=YSSM+UC(K)/W(K)                                        MAIN
      YSSM=YSSM+Y1/W(1)                                           MAIN
      T=UC(1)*TPN                                                 MAIN
      RHO=PSR/(T*YSSM)                                            MAIN
      WRITE(3,8)RHO                                               MAIN
8     FORMAT(/5X,17HINITIAL DENSITY =,1PE12.4/)                  MAIN
      NPTS=NINT+1                                                 MAIN
      DT=TFINAL*1.0E-10                                           MAIN
      TOUT=TFINAL*1.0E-8                                          MAIN
      TPRINT=TFINAL*0.78                                          MAIN
      INDEX=1                                                     MAIN
C  DEFINE THE BREAKPOINTS FOR THE INTEGRATOR.                    MAIN
      NINT10=NINT/10                                              MAIN
      IF(NSTART.EQ.2)GO TO 505                                    MAIN
      PHFFL=PH0+0.4*(PH5-PH0)                                     MAIN
      PHFFR=PH0+0.6*(PH5-PH0)                                     MAIN
      PH2=PHFFL                                                   MAIN
      PH3=PHFFR                                                   MAIN
      PHCT=0.5*(PH0+PH5)                                          MAIN
      GO TO 510                                                   MAIN
505   READ(1,22)PHFFL,PHFFR,PHCTO                                 MAIN
      WRITE(3,22)PHFFL,PHFFR,PHCTO                                MAIN
22    FORMAT(1P3E12.4)                                            MAIN
      PHCT=0.5*(PH0+PH5)                                          MAIN
      DPH=1.5*(PHFFR-PHFFL)                                       MAIN
      IF(DPH.GT.(PH5-PH0)/3.0)DPH=(PH5-PH0)/3.0                  MAIN
      PH2=PHCT-DPH/2.0                                            MAIN
      PH3=PHCT+DPH/2.0                                            MAIN
      IF(PH2.LE.PH0)PH2=PH0+0.1*(PH3-PH2)                         MAIN
510   PH1=0.5*(PH0+PH2)                                           MAIN
      PH4=0.5*(PH3+PH5)                                           MAIN
      WRITE(3,220)PH0,PH1,PH2,PH3,PH4,PH5                         MAIN
      D10=FLOAT(NINT10)                                           MAIN
      NU=NINT10+1                                                 MAIN
      DP1=(PH1-PH0)/D10                                           MAIN
      DP2=0.5*(PH2-PH1)/D10                                       MAIN
      DP3=0.25*(PH3-PH2)/D10                                      MAIN
      DP4=0.5*(PH4-PH3)/D10                                       MAIN
      DP5=(PH5-PH4)/D10                                           MAIN
      DO 10 K=1,NU                                                MAIN
10    PHBKPT(K)=FLOAT(K-1)*DP1                                    MAIN
      NL=NU+1                                                     MAIN
      NU=NU+2*NINT10                                              MAIN
      DO 11 K=NL,NU                                               MAIN
```

```
11    PHBKPT(K)=PH1+FLOAT(K+1-NL)*DP2
      NL=NU+1
      NU=NU+4*NINT10
      DO 12 K=NL,NU
12    PHBKPT(K)=PH2+FLOAT(K+1-NL)*DP3
      NL=NU+1
      NU=NU+2*NINT10
      DO 13 K=NL,NU
13    PHBKPT(K)=PH3+FLOAT(K+1-NL)*DP4
      NL=NU+1
      NU=NU+NINT10
      DO 14 K=NL,NU
14    PHBKPT(K)=PH4+FLOAT(K+1-NL)*DP5
      WRITE(3,39)PHBKPT
C  DEFINE THE EVALUATION POINTS.
      NSP=4*(KORD-NCC)
      NVPTS=NSP*NINT+1
      DO 17 K=1,NPTS
      KV=1+(K-1)*NSP
17    PHVAL(KV)=PHBKPT(K)
      NSPP=NSP+1
      DO 19 K=NSPP,NVPTS,NSP
      KM=K-NSP
      DP=(PHVAL(K)-PHVAL(KM))/FLOAT(NSP)
      DO 19 J=2,NSP
      KT=KM+J-1
      PHVAL(KT)=PHVAL(KM)+DP*FLOAT(J-1)
19    CONTINUE
      NSKIP=2
      WRITE(3,39)(PHVAL(I),I=1,NVPTS,NSKIP)
C  DEFINE THE CENTER OF THE FLAME.
      KCEN=NPDE
      VCEN=0.5*(UB(KCEN)+UC(KCEN))
C  INITIAL VALUE FOR THE SPEED OF THE ORIGIN (MASS FLOW).
      IF(NSTART.EQ.2)GO TO 350
      DO 310 K=1,NVPTS
      PH=PHVAL(K)
      CALL UINIT(PH,UT,NPDE)
      DO 305 J=1,NPDE
305   U(J,K)=UT(J)
      U1(K)=1.0
      DO 308 J=2,NPDE
308   U1(K)=U1(K)-U(J,K)
310   CONTINUE
      CALL FLSP(PH0,PH5,NPDE,TOUT,TPRINT,TFINAL,NVPTS,RHO,KCEN,FSP,1)
      FM=RHO*FSP
      SPEED0=-FM*TMN/PHN
      WRITE(3,47)FM,FSP
      DS=0.0
      WRITE(3,43)SPEED0,DS
350   CONTINUE
      IF(NSTART.EQ.2)READ(1,255)SPEED0
255   FORMAT(1PE12.4)
      FT=SECOND(CP)
C  CALL INTEGRATOR AND WRITE OUTPUT.
20    CALL PDECOL(T0,TOUT,DT,PHBKPT,EPS,NINT,KORD,NCC,NPDE,MF,
     * INDEX,WORK,IWORK)
```

```
      IF(INDEX.NE.0)GO TO 70
      WRITE(3,30)TOUT,DTUSED,NSTEPS
30    FORMAT(//10X,3HT =,1PE12.4,4X,4HDT =,1PE12.4,4X,
     * 13HTOTAL STEPS =,I8/)
      GT=SECOND(CP)
      HT=GT-FT
      WRITE(3,45)HT
45    FORMAT(/10X,10HRUN TIME =,1PE12.4/)
      CALL VALUES(PHVAL,U,SCTCH,NPDE,NVPTS,NVPTS,0,WORK)
      DO 25 K=1,NVPTS
      U1(K)=1.0
      DO 25 J=2,NPDE
25    U1(K)=U1(K)-U(J,K)
      IF(TOUT.GT.TFINAL*1.1E-08.AND.TOUT.LT.TPRINT)GO TO 65
      DO 35 K=1,NPDE
      WRITE(3,37)LB(K)
37    FORMAT(/10X,A10/)
      WRITE(3,39)(U(K,I),I=1,NVPTS,NSKIP)
39    FORMAT(1P10E12.4)
35    CONTINUE
      NPDEP1=NPDE+1
      WRITE(3,37)LB(NPDEP1)
      WRITE(3,39)(U1(I),I=1,NVPTS,NSKIP)
65    CONTINUE
      CALL FLSP(PH0,PH5,NPDE,TOUT,TPRINT,TFINAL,NVPTS,RHO,KCEN,FSP,2)
C  FIND THE FORMULA FOR ADJUSTING THE ORIGIN SPEED AND CENTERING
C  THE FLAME.
100   DO 110 K=1,NVPTS
      KP=K+1
      UMAX=AMAX1(U(KCEN,K),U(KCEN,KP))
      UMIN=AMIN1(U(KCEN,K),U(KCEN,KP))
      IF(UMAX.GT.VCEN.AND.UMIN.LT.VCEN)KS=K
110   IF(UMAX.GT.VCEN.AND.UMIN.LT.VCEN)GO TO 115
115   KP=KS+1
      PC=(VCEN-U(KCEN,KS))/(U(KCEN,KP)-U(KCEN,KS))
      PHNEW=PHVAL(KS)+PC*(PHVAL(KP)-PHVAL(KS))
      IF(TOUT.GT.TFINAL*1.1E-8)GO TO 150
      TINC=TFINAL/50.
      TOLD=TOUT
      TOUT=TINC
      DPH=PHNEW-PHCT
      DT=TOUT-TOLD
      PHOLD=PHNEW-SPEEDO*DT
      BSP=0.0
      ASP=SPEEDO
      SPEEDN=ASP+BSP*TOUT
      GO TO 20
150   SPEEDO=SPEEDN
      SPN1=(PHNEW-PHOLD)/(TOUT-TOLD)
      FM=SPN1*PHN/TMN
      FSP=FM/RHO
      WRITE(3,47)FM,FSP
47    FORMAT(/5X,4HM0 =,1PE12.4,6X,6HFLSP =,1PE12.4/)
      DS=SPN1-SPEEDO
      TOLD=TOUT
      PC=ABS(DS/SPEEDO)
      IF(PC.LT.0.10)TINC=2.0*TINC
```

```
      IF(PC.LT.0.01)TINC=TINC*2.0
      IF(TINC.GT.TFINAL*0.2)TINC=TFINAL*0.2
      TOUT=TOUT+TINC
      IF(TOUT.GT.TFINAL)TOUT=TFINAL
      IF(TOLD.EQ.TFINAL)GO TO 201
      DT=TOUT-TOLD
      DPH=PHNEW-PHCT
      DS=SPN1-SPEEDO
      BSP=2.0*(DS+DPH/DT)/DT
      ASP=SPEEDO-BSP*TOLD
      SPN2=ASP+BSP*TOUT
      SPEEDN=0.5*(SPN1+SPN2)
      BSP=(SPEEDN-SPEEDO)/(TOUT-TOLD)
      ASP=SPEEDO-TOLD*BSP
      DS=SPEEDN-SPEEDO
      WRITE(3,43)SPEEDN,DS
   43 FORMAT(/5X,14HORIGIN SPEED =,1PE12.4,6X,8HCHANGE =,1PE12.4/)
      PHOLD=PHNEW-DT*(ASP+BSP*(TOUT+TOLD)*0.5)
      IF(TOUT.LE.TFINAL)GO TO 20
  201 CONTINUE
      IF(KFILE.EQ.0)GO TO 500
C  CREATE DATA FILE FOR RESTART
      WRITE(3,203)PHFFL,PHFFR,PHNEW
      WRITE(2,203)PHFFL,PHFFR,PHNEW
  203 FORMAT(1P3E12.4)
      WRITE(2,205)SPEEDN
      WRITE(3,205)SPEEDN
  205 FORMAT(1PE12.4)
      WRITE(3,200)
  200 FORMAT(/10X,18HCOLLOCATION POINTS/)
      NCPTS=(KORD-NCC)*NINT+NCC
      CALL VALUES(PHCPTS,U,SCTCH,NPDE,NCPTS,NCPTS,0,WORK)
      WRITE(2,210)NCPTS
      WRITE(3,210)NCPTS
  210 FORMAT(I8)
      DO 250 K=1,NCPTS
      WRITE(2,220)PHCPTS(K),(U(L,K),L=1,NPDE)
      WRITE(3,230)K,PHCPTS(K),(U(L,K),L=1,NPDE)
  250 CONTINUE
  220 FORMAT(1P10E12.4)
  230 FORMAT(I8,1P4E12.4)
  500 CONTINUE
   70 WRITE(3,80)INDEX
   80 FORMAT(/10X,7HINDEX =,I3)
      STOP
      END
```

```
      SUBROUTINE F(TIME,PH,U,UPH,UPH2,FVAL,NPDE)
      COMMON/TABAB/ASP,BSP,TPN,PHN,TMN
      COMMON/TABP/PRESS,PSR
      COMMON/TABRY/T,Y1,Y2,Y3,Y4
      COMMON/TABUV/RH,UV(3)
      COMMON/TAB6/Z(3,3)
      DIMENSION W(3),DUV(3)
      DIMENSION R(3),H(3)
      DIMENSION U(NPDE),UPH(NPDE),UPH2(NPDE),FVAL(NPDE)
      Y1=1.0
      DO 10 K=2,NPDE
   10 Y1=Y1-U(K)
      CALL RT(U,R)
C  ********* **************
C  THE REST OF THIS SUBROUTINE IS WRITTEN BY THE PROGRAM LOADF.
C  ************************
C  OZONE FLAME.
C  U=MASS FRACTION.  Y=MASS FRACTION/MOLECULAR WEIGHT.
C  NTIS ENTHALPIES AND HEAT CAPACITIES.
      IF(T.GT.1000.)GO TO 2000
      C 1=1.9872*(  3.62559850E+00+T*( -1.87821840E-03
     * +T*(  7.05545440E-06+T*( -6.76351370E-09+T*(  2.15559930E-12)))))
      H 1=1.9872*( -1.04752260E+03+T*(  3.62559850E+00
     * +T*( -1.87821840E-03/2.0+T*(  7.05545440E-06/3.0
     * +T*( -6.76351370E-09/4.0+T*(  2.15559930E-12/5.0))))))
      C 2=1.9872*(  2.95586620E+00+T*( -1.70615360E-03
     * +T*(  2.59251540E-06+T*( -1.78379800E-09+T*(  4.57090120E-13)))))
      H 2=1.9872*(  2.91436540E+04+T*(  2.95586620E+00
     * +T*( -1.70615360E-03/2.0+T*(  2.59251540E-06/3.0
     * +T*( -1.78379800E-09/4.0+T*(  4.57090120E-13/5.0))))))
      C 3=1.9872*(  2.46606170E+00+T*(  9.17032090E-03
     * +T*( -4.96984800E-06+T*( -2.06342300E-09+T*(  2.00155950E-12)))))
      H 3=1.9872*(  1.60595560E+04+T*(  2.46606170E+00
     * +T*(  9.17032090E-03/2.0+T*( -4.96984800E-06/3.0
     * +T*( -2.06342300E-09/4.0+T*(  2.00155950E-12/5.0))))))
      GO TO 3000
 2000 CONTINUE
      C 1=1.9872*(  3.62195350E+00+T*(  7.36182640E-04
     * +T*( -1.96522280E-07+T*(  3.62015580E-11+T*( -2.89456270E-15)))))
      H 1=1.9872*( -1.20198250E+03+T*(  3.62195350E+00
     * +T*(  7.36182640E-04/2.0+T*( -1.96522280E-07/3.0
     * +T*(  3.62015580E-11/4.0+T*( -2.89456270E-15/5.0))))))
      C 2=1.9872*(  2.53526380E+00+T*( -1.43718980E-05
     * +T*( -1.13601390E-08+T*(  6.60051310E-12+T*( -6.11816260E-16)))))
      H 2=1.9872*(  2.92302650E+04+T*(  2.53526380E+00
     * +T*( -1.43718980E-05/2.0+T*( -1.13601390E-08/3.0
     * +T*(  6.60051310E-12/4.0+T*( -6.11816260E-16/5.0))))))
      C 3=1.9872*(  5.46652390E+00+T*(  1.73260310E-03
     * +T*( -7.22048890E-07+T*(  1.37216600E-10+T*( -9.62338280E-15)))))
      H 3=1.9872*(  1.52140960E+04+T*(  5.46652390E+00
     * +T*(  1.73260310E-03/2.0+T*( -7.22048890E-07/3.0
     * +T*(  1.37216600F-10/4.0+T*( -9.62338280E-15/5.0))))))
 3000 CONTINUE
C  THERMAL CONDUCTIVITY.
C  LO  DALGARNO AND SMITH FORMULA.  T=100,2000.
C  LO2 LEAST SQUARES FIT OF HANLEY AND ELY DATA.  T=300,2000.
C  LO3 BROMLEY CORRELATION.  T=300,2000.
```

```
      C  O3  LENNARD-JONES PARAMETERS FROM A FIT OF TRETTER VISCOSITY DATA.
            RL 1=  5.743800E-07*(T**  6.267700E-01)
            RL 2=  1.603700E-06*(T**  7.100000E-01)
            RL 3=  3.899200E-07*(T**  6.624200E-01)
      C  BINARY DIFFUSION COEFFICIENTS.
      C  O-O2  FIT BY MARRERO AND MASON.
      C  O2-O3 AND O-O3  LENNARD JONES POTENTIAL.
      C  O   LENNARD JONES PARAMETERS FROM DALGARNO AND SMITH VISCOSITY DATA.
      C  O2  LENNARD JONES PARAMETERS FROM HANLEY AND ELY DVISCOSITY DATA.
      C  O3  LENNARD JONES PARAMETERS FROM A FIT OF TRETTER VISCOSITY DATA.
      C  A LEAST SQUARES FIT OF THE RESULTING DIFFUSIONS COEFFICIENTS IS
      C  MADE FOR T=300,2000.
      C  PRESS =  1.00
            D 1 2=  1.320000E-05*(T**  1.774000E+00)
            D 1 3=  1.175200E-05*(T**  1.665400E+00)
            D 2 3=  1.659600E-05*(T**  1.665400E+00)
      C  SPECIFIC HEAT OF THE MIXTURE.  NTIS FORMULA.
            CM=+C 1*Y 1+C 2*Y 2+C 3*Y 3
      C  SPECIFIC HEATS AND SPECIFIC ENTHALPIES.
            C 1=C 1/ 32.00
            H 1=H 1/ 32.00
            C 2=C 2/ 16.00
            H 2=H 2/ 16.00
            C 3=C 3/ 48.00
            H 3=H 3/ 48.00
      C  MOLE FRACTIONS AND VARIOUS SPACE DERIVATIVES.
            YS=Y 4
            X 1=Y 1/YS
            X 2=Y 2/YS
            X 3=Y 3/YS
            U 2=U( 2)
            DU 2=UPH( 2)
            DDU 2=UPH2( 2)
            U 3=U( 3)
            DU 3=UPH( 3)
            DDU 3=UPH2( 3)
            U 1=1.0-U 2-U 3
            DU 1=-DU 2-DU 3
            DDU 1=-DDU 2-DDU 3
            DYS=+DU 1/ 32.00+DU 2/ 16.00+DU 3/ 48.00
            DDYS=+DDU 1/ 32.00+DDU 2/ 16.00+DDU 3/ 48.00
      C  TPN =  1.0000E+03       PHN =  5.0000E-06      TMN =  5.0000E-05
            DT=UPH(1)*  1.0000E+03
            DRH=-  1.2187690433E-02*(DT/T+DYS/YS)/(T*YS)
            DL 1=  4.748802E-07*DT*(T**( -1.732300E-01))
            DL 2=  1.138627E-06*DT*(T**( -2.900000E-01))
            DL 3=  3.284764E-07*DT*(T**( -1.575800E-01))
            DX 1=DU 1/( 32.00*YS)-Y 1*DYS/(YS*YS)
            DX 2=DU 2/( 16.00*YS)-Y 2*DYS/(YS*YS)
            DX 3=DU 3/( 48.00*YS)-Y 3*DYS/(YS*YS)
            DD 1 2=  2.341660E-05*(T**  7.740000E-01)*DT
            DD 1 3=  1.957178E-05*(T**  6.654000E-01)*DT
            DD 2 3=  2.763898E-05*(T**  6.654000E-01)*DT
      C  THERMAL CONDUCTIVITY OF THE MIXTURE.
      C  BROKAW METHOD.  P. 486. REID AND SHERWOOD.
            RLM = +X 1*RL 1 +X 2*RL 2 +X 3*RL 3
            RLMV= +X 1/RL 1 +X 2/RL 2 +X 3/RL 3
```

```
      RLMV=1.0/RLMV
      RLM=0.5*(RLM+RLMV)
C  SPACE DERIVATIVE OF THE THERMAL CONDUCTIVITY.
      DRLM=+RL 1*DX 1+DL 1 *X 1+RL 2*DX 2+DL 2 *X 2+RL 3*DX 3+DL 3 *X 3
      DRLMV=-RLMV*RLMV*(
     *  +DX 1 /RL 1  -X 1 *DL 1/(RL 1 *RL 1)
     *  +DX 2 /RL 2  -X 2 *DL 2/(RL 2 *RL 2)
     *  +DX 3 /RL 3  -X 3 *DL 3/(RL 3 *RL 3))
      DRLM=0.5*(DRLM+DRLMV)
C  SOLVE FOR UV.
      Z( 1, 1)=1.0
      Z( 1, 2)=1.0
      Z( 1, 3)=1.0
      W(1)=0.0
      Z( 2, 1)=U 2/( 32.00*D 1 2)
      Z( 2, 3)=U 2/( 48.00*D 2 3)
      Z( 3, 1)=U 3/( 32.00*D 1 3)
      Z( 3, 2)=U 3/( 16.00*D 2 3)
      Z( 2, 2)=-U 1/( 32.00*D 1 2)
     * -U 3/( 48.00*D 2 3)
      Z( 3, 3)=-U 1/( 32.00*D 1 3)
     * -U 2/( 16.00*D 2 3)
      W( 2)=RH*(DU 2*YS-U 2*DYS)/PHN
      W( 3)=RH*(DU 3*YS-U 3*DYS)/PHN
      CALL DECOMP(NPDE)
      CALL SOLVE(NPDE,W,UV)
C  SOLVE FOR THE PARTIAL OF UV.
      W(1)=0.0
      W( 2)=RH*(DDU 2*YS-U 2*DDYS)/PHN
     * +DRH*(DU 2*YS-U 2*DYS)/PHN
      W( 2)=W( 2)+(-DU 2*UV( 1)+UV( 2)*DU 1)/( 32.00*D 1 2)
     * +(U 2*UV( 1)-UV( 2)*U 1)*DD 1 2/( 32.00*D 1 2*D 1 2)
      W( 2)=W( 2)+(-DU 2*UV( 3)+UV( 2)*DU 3)/( 48.00*D 2 3)
     * +(U 2*UV( 3)-UV( 2)*U 3)*DD 2 3/( 48.00*D 2 3*D 2 3)
      W( 3)=RH*(DDU 3*YS-U 3*DDYS)/PHN
     * +DRH*(DU 3*YS-U 3*DYS)/PHN
      W( 3)=W( 3)+(-DU 3*UV( 1)+UV( 3)*DU 1)/( 32.00*D 1 3)
     * +(U 3*UV( 1)-UV( 3)*U 1)*DD 1 3/( 32.00*D 1 3*D 1 3)
      W( 3)=W( 3)+(-DU 3*UV( 2)+UV( 3)*DU 2)/( 16.00*D 2 3)
     * +(U 3*UV( 2)-UV( 3)*U 2)*DD 2 3/( 16.00*D 2 3*D 2 3)
      CALL SOLVE(NPDE,W,DUV)
C  FIND THE TIME DERIVATIVES.
      SP=ASP+BSP*TIME
      TL=  2.0000000000E+06*(RH*RLM*UPH2(1)+UPH(1)*(DRH*RLM+RH*DRLM))
      TR=-  5.0000000000E-08*(+R( 1)*H 1
     * +R( 2)*H 2 +R( 3)*H 3)
      TD=-  1.0000000000E+01*UPH(1)*RH*(+UV( 1)*C 1
     * +UV( 2)*C 2 +UV( 3)*C 3)
      FVAL(1)=SP*UPH(1)+(TL+TR+TD)/CM
      DY 2=-  1.0000000000E+01*(DRH*UV( 2)+RH*DUV( 2))
      RY 2=  5.0000000000E-05*R( 2)
      FVAL( 2)=SP*UPH( 2)+DY 2+RY 2
      DY 3=-  1.0000000000E+01*(DRH*UV( 3)+RH*DUV( 3))
      RY 3=  5.0000000000E-05*R( 3)
      FVAL( 3)=SP*UPH( 3)+DY 3+RY 3
      RETURN
      END
```

```
      SUBROUTINE UINIT(PH,U,NPDE)
      DIMENSION U(1)
      COMMON/TABIN/PH2,PH3,PHCTO,PHCT,UC(3),UB(3)
      COMMON/OUTP/PHS(241),UP(3,241)
      COMMON/TABAB/ASP,BSP,TPN,PHN,TMN
      COMMON/TABW/W(3)
      COMMON/START/NS
      COMMON/TABUV/RH,UV(3)
C  DETERMINE THE INITIAL STARTING PROFILE.
C  X WILL BE A COLLOCATION POINT.
C  U STORES THE CORRESPONDING FUNCTION VALUES.
C  *************************
      IF(NS.GT.1)GO TO 50
C  DETERMINE AN INITIAL GUESSED PROFILE FROM VALUES AT THE
C  BURNED AND UNBURNED ENDS.
      IF(PH.GT.PH2)GO TO 5
      DO 3 K=1,NPDE
    3 U(K)=UC(K)
      RETURN
    5 IF(PH.GT.PH3)GO TO 10
      PHD=(PH-PH2)/(PH3-PH2)
      DO 8 K=1,NPDE
    8 U(K)=UC(K)+(UB(K)-UC(K))*PHD
      RETURN
   10 DO 13 K=1,NPDE
   13 U(K)=UB(K)
      RETURN
   50 IF(NS.GT.2)GO TO 100
C  READ THE COLLOCATION VALUES AND THE CORRESPONDING FUNCTION
C  VALUES FROM A PREVIOUS RUN.
      READ(1,55)NPS
   55 FORMAT(I8)
      DO 60 K=1,NPS
      READ(1,65)PHS(K),(UP(J,K),J=1,NPDE)
      SUM=0.0
      DO 58 J=2,NPDE
      IF(UP(J,K).LT.0.0)UP(J,K)=0.0
      IF(UP(J,K).GT.1.0)UP(J,K)=1.0
      SUM=SUM+UP(J,K)
   58 CONTINUE
      IF(SUM.LE.1.0)GO TO 66
      UMAX=0.0
      DO 62 J=2,NPDE
      IF(UP(J,K).GT.UMAX)JMAX=J
      IF(UP(J,K).GT.UMAX)UMAX=UP(J,K)
   62 CONTINUE
      UP(JMAX,K)=UP(JMAX,K)-(SUM-1.0)
   66 CONTINUE
   60 CONTINUE
   65 FORMAT(1P10E12.4)
C  RECENTER THE FLAME.
      DO 70 K=1,NPS
      PHS(K)=PHS(K)+PHCT-PHCTO
   70 CONTINUE
      NS=3
      DO 75 J=1,NPUF
   75 U(J)=UC(J)
```

```
      RETURN
C  DETERMINE THE NEW VALUES BY INTERPOLATION.
100   K=2
      IF(PH.GT.PHS(1))GO TO 104
      DO 102 J=1,NPDE
102   U(J)=UC(J)
      RETURN
104   IF(PH.LT.PHS(NPS))GO TO 108
      DO 106 J=1,NPDE
106   U(J)=UP(J,NPS)
      RETURN
108   CONTINUE
110   IF(PH.LE.PHS(K))GO TO 120
      K=K+1
      GO TO 110
120   KM=K-1
      P=(PHS(K)-PH)/(PHS(K)-PHS(KM))
      DO 130 J=1,NPDE
130   U(J)=UP(J,K)-P*(UP(J,K)-UP(J,KM))
      RETURN
      END
```

```
      SUBROUTINE FLSP(PHL,PHR,NPDE,TOUT,TPRINT,TFINAL,NVPTS,RHO,          FLSP    2
     * KCEN,FSP,KINIT)                                                    FLSP    3
C  DETERMINE THE FLAME SPEED BY NUMERICALLY INTEGRATING THE SPECIES      FLSP    4
C  EQUATIONS,USING THE TRAPEZOIDAL RULE.                                 FLSP    5
      COMMON/OUT/PHVAL(241)                                               FLSP    6
      COMMON/TABIN/PHFFL,PHFFR,PHCTO,PHCT,UC(3),UB(3)                     FLSP    7
      COMMON/ENDPT/PHO,PH5                                                FLSP    8
      COMMON/TABUV/RH,UV(3)                                               FLSP    9
      COMMON/TABP/PRESS,PSR                                               FLSP   10
      COMMON/OUTP/U1(241),U(3,241)                                        FLSP   11
      COMMON/MAIN/SCTCH(50),WORK(20000),IWORK(1200)                       FLSP   12
      COMMON/TABAH/ASP,BSP,TPN,PHN,TMN                                    FLSP   13
      COMMON/TABW/W(3)                                                    FLSP   14
      COMMON/TABRY/T,Y1,Y2,Y3,Y4                                          FLSP   15
      DIMENSION RINT(3),RINT1(3),RI(3)                                    FLSP   16
      DIMENSION UL(3),ULPH(3),ULPH2(3),FVAL(3),ULS(3,2,3),PHB(2)          FLSP   17
      DIMENSION X(241),R(3),D(3)                                          FLSP   18
      DO 5 J=1,NPDE                                                       FLSP   19
    5 RI(J)=0.0                                                           FLSP   20
      DO 100 K=1,NVPTS                                                    FLSP   21
      DO 10 J=1,NPDE                                                      FLSP   22
   10 UL(J)=U(J,K)                                                        FLSP   23
      Y1=U1(K)                                                            FLSP   24
      CALL RT(UL,R)                                                       FLSP   25
      DO 20 J=1,NPDE                                                      FLSP   26
      RINT(J)=R(J)                                                        FLSP   27
   20 CONTINUE                                                            FLSP   28
      IF(K.EQ.1)GO TO 50                                                  FLSP   29
      KM=K-1                                                              FLSP   30
      DPH=PHVAL(K)-PHVAL(KM)                                              FLSP   31
      DO 30 J=1,NPDE                                                      FLSP   32
   30 RI(J)=RI(J)+0.5*(RINT(J)+RINT1(J))*DPH                              FLSP   33
   50 DO 60 J=1,NPDE                                                      FLSP   34
   60 RINT1(J)=RINT(J)                                                    FLSP   35
  100 CONTINUE                                                            FLSP   36
      DO 110 J=1,NPDE                                                     FLSP   37
  110 RI(J)=PHN*RI(J)                                                     FLSP   38
      WRITE(3,112)(RI(L),L=1,NPDE)                                        FLSP   39
  112 FORMAT(/2X,11HINTEGRALS =,1P10E12.4/)                               FLSP   40
      IF(KINIT.EQ.1)GO TO 135                                             FLSP   41
C  FIND APPROPRIATE VALUES AT PHL AND PHR.                                FLSP   42
      PHB(1)=PHR                                                          FLSP   43
      PHB(2)=PHL                                                          FLSP   44
      CALL VALUES(PHB,ULS,SCTCH,NPDE,2,2,2,WORK)                          FLSP   45
      DO 115 J=1,NPDE                                                     FLSP   46
      UL(J)=ULS(J,1,1)                                                    FLSP   47
      ULPH(J)=ULS(J,1,2)                                                  FLSP   48
      ULPH2(J)=ULS(J,1,3)                                                 FLSP   49
  115 CONTINUE                                                            FLSP   50
      CALL F(TOUT,PHO,UL,ULPH,ULPH2,FVAL,NPDE)                            FLSP   51
      DO 118 J=1,NPDE                                                     FLSP   52
  118 D(J)=-RH*UV(J)                                                      FLSP   53
      DO 120 J=1,NPDE                                                     FLSP   54
      UL(J)=ULS(J,2,1)                                                    FLSP   55
      ULPH(J)=ULS(J,2,2)                                                  FLSP   56
      ULPH2(J)=ULS(J,2,3)                                                 FLSP   57
  120 CONTINUE                                                            FLSP   58
```

```
      CALL F(TOUT,PHS,UL,ULPH,ULPH2,FVAL,NPDE)
      DO 121 J=1,NPDE
  121 D(J)=D(J)+RH*UV(J)
      WRITE(3,122)(D(L),L=1,NPDE)
  122 FORMAT(/2X,12HDIFF TERMS =,1P10E12.4/)
      DO 130 J=1,NPDE
  130 RI(J)=RI(J)+D(J)
  135 CONTINUE
      RI(1)=RI(1)/(RHO*(U1(NVPTS)-U1(1)))
      DO 150 J=2,NPDE
  150 RI(J)=RI(J)/(RHO*(U(J,NVPTS)-U(J,1)))
      WRITE(3,152)RI
  152 FORMAT(/2X,13HFLAME SPEED =,1P10E12.4///)
      FSP=RI(KCEN)
C  FIND THE X VALUES USING THE TRAPEZOIDAL RULE.
      RI(1)=0.0
      X(1)=0.0
      DO 200 K=1,NVPTS
      T=U(1,K)*TPN
      YSSM=U1(K)/W(1)
      DO 160 J=2,NPDE
  160 YSSM=YSSM+U(J,K)/W(J)
      RH=PSR/(T*YSSM)
      RINT(1)=1.0/RH
      IF(K.EQ.1)GO TO 180
      KM=K-1
      DPH=PHVAL(K)-PHVAL(KM)
      RI(1)=RI(1)+0.5*(RINT(1)+RINT1(1))*DPH
      X(K)=PHN*RI(1)
  180 RINT1(1)=RINT(1)
  200 CONTINUE
      IF(TOUT.LT.TPRINT)GO TO 250
      NSKIP=2
      WRITE(3,201)
  201 FORMAT(/10X,7HX IN CM/)
      WRITE(3,202)(X(L),L=1,NVPTS,NSKIP)
  202 FORMAT(1P10E12.4)
  250 CONTINUE
C  COMPUTE THE FLAME THICKNESS.
      TMAX=AMAX1(U(1,NVPTS),U(1,1))
      TMIN=AMIN1(U(1,NVPTS),U(1,1))
      DT=0.1*(TMAX-TMIN)
      TH=TMAX-DT
      TL=TMIN+DT
      DO 310 K=1,NVPTS
      KP=K+1
      IF(TL.GT.U(1,K).AND.TL.LT.U(1,KP))KL=K
      IF(TH.GT.U(1,K).AND.TH.LT.U(1,KP))KH=K
  310 CONTINUE
      KLP=KL+1
      KHP=KH+1
      PL=(U(1,KL)-TL)/(U(1,KL)-U(1,KLP))
      PH=(U(1,KH)-TH)/(U(1,KH)-U(1,KHP))
      PHIL=PHVAL(KL)+PL*(PHVAL(KLP)-PHVAL(KL))
      PHIH=PHVAL(KH)+PH*(PHVAL(KHP)-PHVAL(KH))
      XL=X(KL)+PL*(X(KLP)-X(KL))
      XH=X(KH)+PH*(X(KHP)-X(KH))
```

```
SUBROUTINE FLSP        76/76    OPT=1 ROUND=+-*/ TRACE          FTN 4.6+452          03/19/78  14.26.17          PAGE     13

               FTH=ABS(XLW-XH)                                                  FLSP       116
               WRITE(3,314)PHIL,PHIH                                            FLSP       117
               PHCEN=0.5*(PHL+PHH)                                              FLSP       118
               PHFFL=AMIN1(PHIL,PHIH)                                           FLSP       119
               PHFFR=AMAX1(PHIL,PHIH)                                           FLSP       120
         314   FORMAT(/10X,23HFLAME FRONT FROM PHI* =,1PE12.4,2X,               FLSP       121
              * 9HTO PHI* =,1PE12.4/)                                           FLSP       122
               WRITE(3,318)XLW,XH                                               FLSP       123
         318   FORMAT(/10X,20HFLAME FRONT FROM X =,1PE12.4,2X,                  FLSP       124
              * 6HTO X =,1PE12.4/)                                              FLSP       125
               WRITE(3,330)FTH                                                  FLSP       126
         330   FORMAT(/10X,17HFLAME THICKNESS =,1PE12.4,2X,2HCM/)               FLSP       127
               RETURN                                                           FLSP       128
               END                                                              FLSP       129
```

```
    1           SUBROUTINE WT(U,R)                                              RT        2
                DIMENSION U(1),R(1)                                             RT        3
                COMMON/TABRY/T,Y1,Y2,Y3,Y4                                      RT        4
                COMMON/TABUV/RH,UV(3)                                           RT        5
    5           COMMON/TABP/PRESS,PSR                                           RT        6
                COMMON/TABAB/ASP,BSP,TPN,PHN,THN                                RT        7
          C  *************************                                          RT        8
          C  THE REST OF THIS SUBROUTINE IS WRITTEN BY THE PROGRAM LOADER.     RT        9
          C  *************************                                          RT       10
   10           T=U(1)*TPN                                                      RT       11
                Y 1=Y 1/ 32.00                                                  RT       12
                Y 2=U( 2)/ 16.00                                                RT       13
                Y 3=U( 3)/ 48.00                                                RT       14
                Y4=Y1+Y2+Y3                                                     RT       15
   15           Y5=0.44*(Y1+Y2)+Y3                                              RT       16
                Y6=Y1+3.6*Y2+Y3                                                 RT       17
                RH=PSR/(T*Y 4)                                                  RT       18
                RH2=RH*RH                                                       RT       19
                RK 1=  4.3100E+14*EXP( -11161.00/T)                             RT       20
   20           RK 2=  1.2000E+13*EXP(     976.00/T)                            RT       21
                RK 3=  1.1400E+13*EXP(  -2300.00/T)                             RT       22
                RK 4=  1.1900E+13*EXP( -50600.00/T)                             RT       23
                RK 5=  1.3800E+18*(T**( -1.00))*EXP(    -171.00/T)              RT       24
                RK 6=  2.7500E+19*(T**( -1.00))*EXP( -59732.00/T)               RT       25
   25           R 1=RH*RK 1*Y 3*Y 5                                             RT       26
                R 2=RH2*RK 2*Y 2*Y 1*Y 5                                        RT       27
                R 3=RH*RK 3*Y 3*Y 2                                             RT       28
                R 4=RH*RK 4*Y 1*Y 1                                             RT       29
                R 5=RH2*RK 5*Y 2*Y 2*Y 6                                        RT       30
   30           R 6=RH*RK 6*Y 1*Y 6                                             RT       31
                R( 1)= 32.00*                                                   RT       32
               *(+R 1-R 2+2.*R 3-2.*R 4+R 5-R 6)                                RT       33
                R( 2)= 16.00*                                                   RT       34
               *(+R 1-R 2-R 3+R 4-2.*R 5+2.*R 6)                                RT       35
   35           R( 3)= 48.00*                                                   RT       36
               *(-R 1+R 2-R 3+R 4)                                              RT       37
                RETURN                                                          RT       38
                END                                                             RT       39
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
12	I	U	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
13	I	U	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
33	I	R	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
35	I	R	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.

```
SUBROUTINE BNDRY       76/76    OPT=1 ROUND=+-*/ TRACE          FTN 4.6+452         03/19/78  14.26.17        PAGE    15

             SUBROUTINE BNDRY(TIME,PH,U,UPH,DBDU,DBDUPH,DZDT,NPDE)                         BNDRY      2
             DIMENSION U(NPDE),UPH(NPDE),DZDT(NPDE)                                        BNDRY      3
             DIMENSION DBDU(NPDE,NPDE),DBDUPH(NPDE,NPDE)                                   BNDRY      4
             COMMON/ENDPT/PHL,PHR                                                          BNDRY      5
             DO 5 I=1,NPDE                                                                 BNDRY      6
             DZDT(I)=0.0                                                                   BNDRY      7
             DO 5 J=1,NPDE                                                                 BNDRY      8
             DBDU(I,J)=0.0                                                                 BNDRY      9
        5    DBDUPH(I,J)=0.0                                                               BNDRY     10
             IF(PH.EQ.PHR)GO TO 50                                                         BNDRY     11
             DO 10 J=1,NPDE                                                                BNDRY     12
       10    DBDU(J,J)=1.0                                                                 BNDRY     13
             RETURN                                                                        BNDRY     14
       50    DO 60 J=1,NPDE                                                                BNDRY     15
       60    DBDUPH(J,J)=1.0                                                               BNDRY     16
             RETURN                                                                        BNDRY     17
             END                                                                           BNDRY     18
```

```
      SUBROUTINE DECOMP(NN)                                        DECOMP   2
      DIMENSION SCALES(3)                                          DECOMP   3
      COMMON/DS/IPS(3)                                             DECOMP   4
      COMMON/TAB6/UL(3,3)                                          DECOMP   5
C  THIS IS A STANDARD ALGORITHM FOR SOLVING A SET OF LINEAR        DECOMP   6
C  ALGEBRAIC EQUATIONS USING AN LU DECOMPOSITION AND               DECOMP   7
C  BACK SUBSTITUTION.                                              DECOMP   8
C  REFERENCE.                                                      DECOMP   9
C  FORSYTHE AND MOLER.  COMPUTER SOLUTION OF LINEAR ALGEBRAIC SYSTEMS.  DECOMP  10
      N=NN                                                         DECOMP  11
      DO 5 I=1,N                                                   DECOMP  12
      IPS(I)=I                                                     DECOMP  13
      ROWNRM=0.0                                                   DECOMP  14
      DO 2 J=1,N                                                   DECOMP  15
      IF(ROWNRM-ABS(UL(I,J)))1,2,2                                 DECOMP  16
    1 ROWNRM=ABS(UL(I,J))                                          DECOMP  17
    2 CONTINUE                                                     DECOMP  18
      IF(ROWNRM)3,4,3                                              DECOMP  19
    3 SCALES(I)=1.0/ROWNRM                                         DECOMP  20
      GO TO 5                                                      DECOMP  21
    4 WRITE(3,100)                                                 DECOMP  22
  100 FORMAT(///10X,34HMATRIX WITH ZERO ROW IN DECOMPOSE.///)      DECOMP  23
      SCALES(I)=0.0                                                DECOMP  24
    5 CONTINUE                                                     DECOMP  25
      NM1=N-1                                                      DECOMP  26
      DO 17 K=1,NM1                                                DECOMP  27
      BIG=0.0                                                      DECOMP  28
      DO 11 I=K,N                                                  DECOMP  29
      IP=IPS(I)                                                    DECOMP  30
      SIZE=ABS(UL(IP,K))*SCALES(IP)                                DECOMP  31
      IF(SIZE-BIG)11,11,10                                         DECOMP  32
   10 BIG=SIZE                                                     DECOMP  33
      IDXPIV=I                                                     DECOMP  34
   11 CONTINUE                                                     DECOMP  35
      IF(BIG)13,12,13                                              DECOMP  36
   12 WRITE(3,110)                                                 DECOMP  37
  110 FORMAT(///10X,29HSINGULAR MATRIX IN DECOMPOSE.///)           DECOMP  38
      GO TO 17                                                     DECOMP  39
   13 IF(IDXPIV-K)14,15,14                                         DECOMP  40
   14 J=IPS(K)                                                     DECOMP  41
      IPS(K)=IPS(IDXPIV)                                           DECOMP  42
      IPS(IDXPIV)=J                                                DECOMP  43
   15 KP=IPS(K)                                                    DECOMP  44
      PIVOT=UL(KP,K)                                               DECOMP  45
      KP1=K+1                                                      DECOMP  46
      DO 16 I=KP1,N                                                DECOMP  47
      IP=IPS(I)                                                    DECOMP  48
      EM=-UL(IP,K)/PIVOT                                           DECOMP  49
      UL(IP,K)=-EM                                                 DECOMP  50
      DO 16 J=KP1,N                                                DECOMP  51
      UL(IP,J)=UL(IP,J)+EM*UL(KP,J)                                DECOMP  52
   16 CONTINUE                                                     DECOMP  53
   17 CONTINUE                                                     DECOMP  54
      KP=IPS(N)                                                    DECOMP  55
      IF(UL(KP,N))19,18,19                                         DECOMP  56
   18 WRITE(3,110)                                                 DECOMP  57
   19 RETURN                                                       DECOMP  58
```

```
      END                                                    DECOMP      59
```

```
              SUBROUTINE SOLVE(NN,B,X)                                 SOLVE     2
              DIMENSION B(3),X(3)                                      SOLVE     3
              COMMON/DS/IPS(3)                                         SOLVE     4
              COMMON/TAB6/UL(3,3)                                      SOLVE     5
              N=NN                                                     SOLVE     6
              NP1=N+1                                                  SOLVE     7
              IP=IPS(1)                                                SOLVE     8
              X(1)=B(IP)                                               SOLVE     9
              DO 2 I=2,N                                               SOLVE    10
              IP=IPS(I)                                                SOLVE    11
              IM1=I-1                                                  SOLVE    12
              SUM=0.0                                                  SOLVE    13
              DO 1 J=1,IM1                                             SOLVE    14
      1       SUM=SUM+UL(IP,J)*X(J)                                    SOLVE    15
      2       X(I)=B(IP)-SUM                                           SOLVE    16
              IP=IPS(N)                                                SOLVE    17
              X(N)=X(N)/UL(IP,N)                                       SOLVE    18
              DO 4 IBACK=2,N                                           SOLVE    19
              I=NP1-IBACK                                              SOLVE    20
              IP=IPS(I)                                                SOLVE    21
              IP1=I +1                                                 SOLVE    22
              SUM=0.0                                                  SOLVE    23
              DO 3 J=IP1,N                                             SOLVE    24
      3       SUM=SUM+UL(IP,J)*X(J)                                    SOLVE    25
      4       X(I)=(X(I)-SUM)/UL(IP,I)                                 SOLVE    26
              RETURN                                                   SOLVE    27
              END                                                      SOLVE    28
```

```
      SUBROUTINE PDECOL(T0,TOUT,DT,XBKPT,EPS,NINT,KORD,NCC,NPDE,MF,        PDECOL
     *                  INDEX,WORK,IWORK)                                  PDECOL
C                                                                          PDECOL
C------------------------------------------------------------------------- PDECOL
C------------------------------------------------------------------------- PDECOL
C                                                                          PDECOL
C THIS IS THE SEPT 23, 1977 VERSION OF PDECOL.                             PDECOL
C                                                                          PDECOL
C THIS PACKAGE WAS CONSTRUCTED SO AS TO CONFORM TO AS MANY ANSI-FORTRAN    PDECOL
C RULES AS WAS CONVIENTLY POSSIBLE.  THE FORTRAN USED VIOLATES ANSI        PDECOL
C STANDARDS IN THE TWO WAYS LISTED BELOW....                               PDECOL
C                                                                          PDECOL
C   1. SUBSCRIPTS OF THE GENERAL FORM C*V1 + V2 + V3 ARE USED              PDECOL
C      (POSSIBLY IN A PERMUTED ORDER), WHERE C IS AN INTEGER CONSTANT      PDECOL
C      AND V1, V2, AND V3 ARE INTEGER VARIABLES.                           PDECOL
C                                                                          PDECOL
C   2. ARRAY NAMES APPEAR SINGLY IN DATA STATEMENTS IN THE ROUTINES        PDECOL
C      BSPLVN AND COSET.                                                   PDECOL
C                                                                          PDECOL
C------------------------------------------------------------------------- PDECOL
C------------------------------------------------------------------------- PDECOL
C                                                                          PDECOL
C------------------------------------------------------------------------- PDECOL
C PDECOL IS THE DRIVER ROUTINE FOR A SOPHISTICATED PACKAGE OF              PDECOL
C SUBROUTINES WHICH IS DESIGNED TO SOLVE THE GENERAL SYSTEM OF             PDECOL
C NPDE NONLINEAR PARTIAL DIFFERENTIAL EQUATIONS OF AT MOST SECOND          PDECOL
C ORDER ON THE INTERVAL (XLEFT,XRIGHT) FOR T .GT. T0 WHICH IS OF THE       PDECOL
C FORM....                                                                 PDECOL
C                                                                          PDECOL
C      DU/DT  =  F( T, X, U, UX, UXX )                                     PDECOL
C                                                                          PDECOL
C WHERE                                                                    PDECOL
C                                                                          PDECOL
C           U  =  (  U(1),  U(2), ... ,  U(NPDE) )                         PDECOL
C          UX  =  ( UX(1), UX(2), ... , UX(NPDE) )                         PDECOL
C         UXX  =  (UXX(1),UXX(2), ... ,UXX(NPDE) ) .                       PDECOL
C                                                                          PDECOL
C EACH U(K) IS A FUNCTION OF THE SCALAR QUANTITIES T AND X.                PDECOL
C UX(K) REPRESENTS THE FIRST PARTIAL DERIVATIVE OF U(K) WITH RESPECT       PDECOL
C TO THE VARIABLE X.  UXX(K) REPRESENTS THE SECOND PARTIAL DERIVATIVE      PDECOL
C OF U(K) WITH RESPECT TO THE VARIABLE X, AND DU/DT IS THE VECTOR OF       PDECOL
C PARTIAL DERIVATIVES OF U WITH RESPECT TO THE TIME VARIABLE T.            PDECOL
C F  REPRESENTS AN ARBITRARY VECTOR VALUED FUNCTION WHOSE NPDE             PDECOL
C COMPONENTS DEFINE THE RESPECTIVE PARTIAL DIFFERENTIAL EQUATIONS OF       PDECOL
C THE PDE SYSTEM.  SEE SUBROUTINE F DESCRIPTION BELOW.                     PDECOL
C                                                                          PDECOL
C BOUNDARY CONDITIONS                                                      PDECOL
C                                                                          PDECOL
C   DEPENDING ON THE TYPE OF PDE(S), 0, 1, OR 2 BOUNDARY CONDITIONS        PDECOL
C   ARE REQUIRED FOR EACH PDE IN THE SYSTEM. THESE ARE IMPOSED AT XLEFT    PDECOL
C   AND/OR XRIGHT AND EACH MUST BE OF THE FORM....                         PDECOL
C                                                                          PDECOL
C        B(U,UX)  =  Z(T)                                                  PDECOL
C                                                                          PDECOL
C   WHERE  B  AND  Z  ARE ARBITRARY VECTOR VALUED FUNCTIONS WITH           PDECOL
C   NPDE COMPONENTS AND  U, UX, AND T  ARE AS ABOVE.  THESE BOUNDARY       PDECOL
C   CONDITIONS MUST BE CONSISTENT WITH THE INITIAL CONDITIONS WHICH ARE    PDECOL
```

```
C   DESCRIBED NEXT.                                                    PDECOL   59
C                                                                      PDECOL   60
C INITIAL CONDITIONS                                                   PDECOL   61
C                                                                      PDECOL   62
C   EACH SOLUTION COMPONENT  U(K)  IS ASSUMED TO BE A KNOWN (USER      PDECOL   63
C   PROVIDED) FUNCTION OF  X  AT THE INITIAL TIME T = T0.  THE         PDECOL   64
C   INITIAL CONDITION FUNCTIONS MUST BE CONSISTENT WITH THE BOUNDARY   PDECOL   6
C   CONDITIONS ABOVE, I.E. THE INITIAL CONDITION FUNCTIONS MUST        PDECOL   66
C   SATISFY THE BOUNDARY CONDITIONS FOR T = T0.  SEE SUBROUTINE UINIT  PDECOL   67
C   DESCRIPTION BELOW.                                                 PDECOL   68
C--------------------------------------------------------------------- PDECOL   69
C                                                                      PDECOL   70
C REQUIRED USER SUPPLIED SUBROUTINES                                   PDECOL   71
C                                                                      PDECOL   72
C THE USER IS REQUIRED TO CONSTRUCT THREE SUBPROGRAMS AND A MAIN       PDECOL   73
C PROGRAM WHICH DEFINE THE PDE PROBLEM WHOSE SOLUTION IS TO BE         PDECOL   74
C ATTEMPTED.  THE THREE SUBPROGRAMS ARE...                             PDECOL   75
C                                                                      PDECOL   76
C 1)   SUBROUTINE F( T, X, U, UX, UXX, FVAL, NPDE )                    PDECOL   77
C      DIMENSION U(NPDE), UX(NPDE), UXX(NPDE), FVAL(NPDE)              PDECOL   78
C         THIS ROUTINE DEFINES THE DESIRED PARTIAL DIFFERENTIAL        PDECOL   79
C         EQUATIONS TO BE SOLVED.  THE PACKAGE PROVIDES VALUES OF THE  PDECOL   80
C         INPUT SCALARS T AND X AND INPUT ARRAYS (LENGTH NPDE) U, UX,  PDECOL   81
C         AND UXX, AND THE USER MUST CONSTRUCT THIS ROUTINE TO COMPUTE PDECOL   82
C         THE OUTPUT ARRAY FVAL (LENGTH NPDE) WHICH CONTAINS THE       PDECOL   83
C         CORRESPONDING VALUES OF THE RIGHT HAND SIDES OF THE DESIRED  PDECOL   84
C         PARTIAL DIFFERENTIAL EQUATIONS, I.E.                         PDECOL   85
C                                                                      PDECOL   86
C         FVAL(K) = THE VALUE OF THE RIGHT HAND SIDE OF THE K-TH PDE IN PDECOL  87
C                   THE PDE SYSTEM ABOVE, FOR K = 1 TO NPDE.           PDECOL   88
C                                                                      PDECOL   89
C         THE INCOMING VALUE OF THE SCALAR QUANTITY X WILL BE A        PDECOL   90
C         COLLOCATION POINT VALUE (SEE INITAL AND COLPNT) AND THE      PDECOL   91
C         INCOMING VALUES IN THE ARRAYS U, UX AND UXX CORRESPOND TO THIS PDECOL 92
C         POINT X AND TIME T.                                          PDECOL   93
C      RETURN                                                          PDECOL   94
C      END                                                             PDECOL   95
C                                                                      PDECOL   96
C 2)   SUBROUTINE BNDRY( T, X, U, UX, DBDU, DBDUX, DZDT, NPDE )        PDECOL   97
C      DIMENSION U(NPDE), UX(NPDE), DZDT(NPDE)                         PDECOL   98
C      DIMENSION DBDU(NPDE,NPDE), DBDUX(NPDE,NPDE)                     PDECOL   99
C         THIS ROUTINE IS USED TO PROVIDE THE PDE PACKAGE WITH NEEDED  PDECOL  100
C         INFORMATION ABOUT THE BOUNDARY CONDITION FUNCTIONS B AND Z   PDECOL  101
C         ABOVE.  THE PACKAGE PROVIDES VALUES OF THE INPUT VARIABLES   PDECOL  102
C         T, X, U, AND UX, AND THE USER IS TO DEFINE THE CORRESPONDING PDECOL  103
C         OUTPUT VALUES OF THE DERIVATIVES OF THE FUNCTIONS B AND Z    PDECOL  104
C         WHERE....                                                    PDECOL  105
C            DBDU(K,J) = PARTIAL DERIVATIVE OF THE K-TH COMPONENT OF THE PDECOL 106
C                        VECTOR FUNCTION B(U,UX) ABOVE WITH RESPECT TO PDECOL  107
C                        THE J-TH VARIABLE U(J).                       PDECOL  108
C           DBDUX(K,J) = PARTIAL DERIVATIVE OF THE K-TH COMPONENT OF THE PDECOL 109
C                        VECTOR FUNCTION B(U,UX) ABOVE WITH RESPECT TO PDECOL  110
C                        THE J-TH VARIABLE UX(J).                      PDECOL  111
C              DZDT(K) = DERIVATIVE OF THE K-TH COMPONENT OF THE VECTOR PDECOL 112
C                        FUNCTION Z(T) ABOVE WITH RESPECT TO THE       PDECOL  113
C                        VARIABLE T.                                   PDECOL  114
C         NOTE... THE INCOMING VALUE OF X WILL BE EITHER XLEFT OR XRIGHT. PDECOL 115
```

```
C           IF NO BOUNDARY CONDITION IS DESIRED FOR SAY THE K-TH PDE AT      PDECOL     116
C           ONE OR BOTH OF THE ENDPOINTS XLEFT OR XRIGHT, THEN DBDU(K,K)     PDECOL     117
C           AND DBDUX(K,K) SHOULD BOTH BE SET TO ZERO WHEN BNDRY IS          PDECOL     118
C           CALLED FOR THAT POINT.  WE REFER TO THIS AS A NULL BOUNDARY      PDECOL     119
C           CONDITION.  THIS ROUTINE CAN BE STRUCTURED AS FOLLOWS...         PDECOL     120
C           THE COMMON BLOCK /ENDPT/ IS NOT A PART OF PDECOL AND             PDECOL     121
C           MUST BE SUPPLIED AND DEFINED BY THE USER.                        PDECOL     122
C        COMMON /ENDPT/ XLEFT                                                PDECOL     123
C        IF( X .NE. XLEFT ) GO TO 10                                         PDECOL     124
C           HERE DEFINE AND SET PROPER VALUES FOR DBDU(K,J), DBDUX(K,J),     PDECOL     125
C           AND DZDT(K) FOR K,J = 1 TO NPDE FOR THE LEFT BOUNDARY POINT      PDECOL     126
C           X = XLEFT.                                                       PDECOL     127
C        RETURN                                                              PDECOL     128
C     10 CONTINUE                                                            PDECOL     129
C           HERE DEFINE AND SET PROPER VALUES FOR DBDU(K,J), DBDUX(K,J),     PDECOL     130
C           AND DZDT(K) FOR K,J = 1 TO NPDE FOR THE RIGHT BOUNDARY POINT     PDECOL     131
C           X = XRIGHT.                                                      PDECOL     132
C        RETURN                                                              PDECOL     133
C        END                                                                 PDECOL     134
C                                                                            PDECOL     135
C 3)     SUBROUTINE UINIT( X, U, NPDE )                                      PDECOL     136
C        DIMENSION U(NPDE)                                                   PDECOL     137
C           THIS ROUTINE IS USED TO PROVIDE THE PDE PACKAGE WITH THE         PDECOL     138
C           NEEDED INITIAL CONDITION FUNCTION VALUES.  THE PACKAGE           PDECOL     139
C           PROVIDES A VALUE OF THE INPUT VARIABLE X, AND THE USER IS TO     PDECOL     140
C           DEFINE THE PROPER INITIAL VALUES (AT T = T0) FOR ALL OF THE      PDECOL     141
C           PDE COMPONENTS, I.E.                                             PDECOL     142
C              U(K) = DESIRED INITIAL VALUE OF PDE COMPONENT U(K) AT         PDECOL     143
C                     X AND T = T0 FOR K = 1 TO NPDE.                        PDECOL     144
C           NOTE... THE INCOMING VALUE OF X WILL BE A COLLOCATION POINT      PDECOL     145
C           VALUE.  THE INITIAL CONDITIONS AND BOUNDARY CONDITIONS           PDECOL     146
C           MUST BE CONSISTENT (SEE ABOVE).                                  PDECOL     147
C        RETURN                                                              PDECOL     148
C        END                                                                 PDECOL     149
C----------------------------------------------------------------------      PDECOL     150
C                                                                            PDECOL     151
C OPTIONAL USER SUPPLIED SUBROUTINE                                          PDECOL     152
C                                                                            PDECOL     153
C IF THE USER DESIRES TO USE THE MF = 11 OR 21 OPTION IN ORDER TO SAVE       PDECOL     154
C ABOUT 10-20 PERCENT IN EXECUTION TIME (SEE BELOW), THEN THE USER MUST      PDECOL     155
C PROVIDE THE FOLLOWING SUBROUTINE WHICH PROVIDES INFORMATION ABOUT THE      PDECOL     156
C DERIVATIVES OF THE FUNCTION F ABOVE. THIS PROVIDES FOR MORE EFFICIENT      PDECOL     157
C JACOBIAN MATRIX GENERATION.  ON MOST COMPUTER SYSTEMS, THE USER WILL       PDECOL     158
C BE REQUIRED TO SUPPLY THIS SUBROUTINE AS A DUMMY SUBROUTINE IF THE         PDECOL     159
C OPTIONS MF = 12 OR 22 ARE USED (SEE BELOW).                                PDECOL     160
C                                                                            PDECOL     161
C 1)     SUBROUTINE DERIVF( T, X, U, UX, UXX, DFDU, DFDUX, DFDUXX, NPDE )    PDECOL     162
C        DIMENSION U(NPDE), UX(NPDE), UXX(NPDE)                              PDECOL     163
C        DIMENSION DFDU(NPDE,NPDE), DFDUX(NPDE,NPDE), DFDUXX(NPDE,NPDE)      PDECOL     164
C           THE PACKAGE PROVIDES VALUES OF THE INPUT VARIABLES T, X, U, UX,  PDECOL     165
C           AND UXX, AND THE USER SHOULD CONSTRUCT THIS ROUTINE TO PROVIDE   PDECOL     166
C           THE FOLLOWING CORRESPONDING VALUES OF THE OUTPUT ARRAYS          PDECOL     167
C           DFDU, DFDUX, AND DFDUXX FOR K,J = 1 TO NPDE...                   PDECOL     168
C              DFDU(K,J) = PARTIAL DERIVATIVE OF THE K-TH COMPONENT OF THE   PDECOL     169
C                          PDE DEFINING FUNCTION  F  WITH RESPECT TO THE     PDECOL     170
C                          VARIABLE U(J).                                    PDECOL     171
C             DFDUX(K,J) = PARTIAL DERIVATIVE OF THE K-TH COMPONENT OF THE   PDECOL     172
```

```
C                       PDE DEFINING FUNCTION  F  WITH RESPECT TO THE    PDECOL     173
C                       VARIABLE UX(J).                                  PDECOL     174
C           DFDUXX(K,J) = PARTIAL DERIVATIVE OF THE K-TH COMPONENT OF THE PDECOL     175
C                       PDE DEFINING FUNCTION  F  WITH RESPECT TO THE    PDECOL     176
C                       VARIABLE UXX(J).                                 PDECOL     177
C        NOTE... THE INCOMING VALUE OF  X  WILL BE A COLLOCATION POINT   PDECOL     178
C        VALUE.                                                          PDECOL     179
C     RETURN                                                             PDECOL     180
C     END                                                                PDECOL     181
C------------------------------------------------------------------------ PDECOL     182
C                                                                        PDECOL     183
C METHODS USED                                                           PDECOL     184
C                                                                        PDECOL     185
C   THE PACKAGE PDECOL IS BASED ON THE METHOD OF LINES AND USES A        PDECOL     186
C   FINITE ELEMENT COLLOCATION PROCEDURE (WITH PIECEWISE POLYNOMIALS     PDECOL     187
C   AS THE TRIAL SPACE) FOR THE DISCRETIZATION OF THE SPATIAL VARIABLE   PDECOL     188
C   X.  THE COLLOCATION PROCEDURE REDUCES THE PDE SYSTEM TO A SEMI-      PDECOL     189
C   DISCRETE SYSTEM WHICH THEN DEPENDS ONLY ON THE TIME VARIABLE T.      PDECOL     190
C   THE TIME INTEGRATION IS THEN ACCOMPLISHED BY USE OF SLIGHTLY         PDECOL     191
C   MODIFIED STANDARD TECHNIQUES (SEE REFS. 1,2).                        PDECOL     192
C                                                                        PDECOL     193
C   PIECEWISE POLYNOMIALS                                                PDECOL     194
C                                                                        PDECOL     195
C   THE USER IS REQUIRED TO SELECT THE PIECEWISE POLYNOMIAL SPACE        PDECOL     196
C   WHICH IS TO BE USED TO COMPUTE HIS APPROXIMATE SOLUTION. FIRST, THE  PDECOL     197
C   ORDER, KORD, OF THE POLYNOMIALS TO BE USED MUST BE SPECIFIED         PDECOL     198
C   (KORD = POLYNOMIAL DEGREE + 1).  NEXT, THE NUMBER OF PIECES          PDECOL     199
C   (INTERVALS), NINT, INTO WHICH THE SPATIAL DOMAIN (XLEFT,XRIGHT) IS   PDECOL     200
C   TO BE DIVIDED, IS CHOSEN.  THE NINT + 1 DISTINCT BREAKPOINTS OF      PDECOL     201
C   THE DOMAIN MUST BE DEFINED AND SET INTO THE ARRAY XBKPT IN           PDECOL     202
C   STRICTLY INCREASING ORDER, I.E.                                      PDECOL     203
C   XLEFT=XBKPT(1) .LT. XBKPT(2) .LT. ... .LT. XBKPT(NINT+1)=XRIGHT.     PDECOL     204
C   THE APPROXIMATE SOLUTION AT ANY TIME T WILL BE A POLYNOMIAL OF       PDECOL     205
C   ORDER KORD OVER EACH SUBINTERVAL (XBKPT(I),XBKPT(I+1)).  THE         PDECOL     206
C   NUMBER OF CONTINUITY CONDITIONS, NCC, TO BE IMPOSED ACROSS ALL OF    PDECOL     207
C   THE BREAKPOINTS IS THE LAST PIECE OF USER SUPPLIED DATA WHICH IS     PDECOL     208
C   REQUIRED TO UNIQUELY DETERMINE THE DESIRED PIECEWISE POLYNOMIAL      PDECOL     209
C   SPACE.  FOR EXAMPLE, NCC = 2 WOULD REQUIRE THAT THE APPROXIMATE      PDECOL     210
C   SOLUTION (MADE UP OF THE SEPARATE POLYNOMIAL PIECES) AND ITS FIRST   PDECOL     211
C   SPATIAL DERIVATIVE BE CONTINUOUS AT THE BREAKPOINTS AND HENCE ON     PDECOL     212
C   THE ENTIRE DOMAIN (XLEFT,XRIGHT).  NCC = 3 WOULD REQUIRE THAT THE    PDECOL     213
C   APPROXIMATE SOLUTION AND ITS FIRST AND SECOND SPATIAL DERIVATIVES    PDECOL     214
C   BE CONTINUOUS AT THE BREAKPOINTS, ETC. THE DIMENSION OF THIS LINEAR  PDECOL     215
C   SPACE IS KNOWN AND FINITE AND IS NCPTS = KORD*NINT - NCC*(NINT-1).   PDECOL     216
C   THE WELL-KNOWN B-SPLINE BASIS (SEE REF. 3) FOR THIS SPACE IS USED    PDECOL     217
C   BY PDECOL AND IT CONSISTS OF NCPTS KNOWN PIECEWISE POLYNOMIAL        PDECOL     218
C   FUNCTIONS BF(I,X), FOR I=1 TO NCPTS, WHICH DO NOT DEPEND ON THE      PDECOL     219
C   TIME VARIABLE T. WE WISH TO EMPHASIZE THAT THE PIECEWISE POLYNOMIAL  PDECOL     220
C   SPACE USED IN PDECOL (WHICH IS SELECTED BY THE USER) WILL DETERMINE  PDECOL     221
C   THE MAGNITUDE OF THE SPATIAL DISCRETIZATION ERRORS IN THE COMPUTED   PDECOL     222
C   APPROXIMATE SOLUTION.  THE PACKAGE HAS NO CONTROL OVER ERRORS        PDECOL     223
C   INTRODUCED BY THE USERS CHOICE OF THIS SPACE.  SEE INPUT PARAMETERS  PDECOL     224
C   BELOW.                                                               PDECOL     225
C                                                                        PDECOL     226
C   COLLOCATION OVER PIECEWISE POLYNOMIALS                               PDECOL     227
C                                                                        PDECOL     228
C   THE BASIC ASSUMPTION MADE IS THAT THE APPROXIMATE SOLUTION           PDECOL     229
```

```
C   SATISFIES                                                        PDECOL   230
C                           NCPTS                                    PDECOL   231
C              U(T,X)   =   SUM   C(I,T) * BF(I,X)                   PDECOL   232
C                            I=1                                     PDECOL   233
C                                                                    PDECOL   234
C   WHERE THE UNKNOWN COEFFICIENTS C DEPEND ONLY ON THE TIME T AND   PDECOL   235
C   THE KNOWN BASIS FUNCTIONS DEPEND ONLY ON X (WE HAVE ASSUMED THAT PDECOL   236
C   NPDE = 1 FOR CONVENIENCE).  SO, AT ANY GIVEN TIME T THE APPROX-  PDECOL   237
C   IMATE SOLUTION IS A PIECEWISE POLYNOMIAL IN THE USER CHOSEN SPACE. PDECOL 238
C   THE SEMI-DISCRETE EQUATIONS (ACTUALLY ORDINARY DIFFERENTIAL      PDECOL   239
C   EQUATIONS) WHICH DETERMINE THE COEFFICIENTS C ARE OBTAINED BY    PDECOL   240
C   REQUIRING THAT THE ABOVE APPROXIMATE U(T,X) SATISFY THE PDE AND  PDECOL   241
C   BOUNDARY CONDITIONS EXACTLY AT A SET OF NCPTS COLLOCATION POINTS PDECOL   242
C   (SEE COLPNT).  THUS, PDECOL ACTUALLY COMPUTES THE BASIS FUNCTION PDECOL   243
C   COEFFICIENTS RATHER THAN SPECIFIC APPROXIMATE SOLUTION VALUES.   PDECOL   244
C                                                                    PDECOL   245
C   REFERENCES                                                       PDECOL   246
C                                                                    PDECOL   247
C   1. MADSEN, N.K. AND R.F. SINCOVEC, PDECOL - COLLOCATION SOFTWARE PDECOL   248
C        FOR PARTIAL DIFFERENTIAL EQUATIONS, ACM-TOMS, VOL.  , NO.  , PDECOL  249
C                                                                    PDECOL   250
C   2. SINCOVEC, R.F. AND N.K. MADSEN, SOFTWARE FOR NONLINEAR PARTIAL PDECOL  251
C        DIFFERENTIAL EQUATIONS, ACM-TOMS, VOL. 1, NO. 3,            PDECOL   252
C        SEPTEMBER 1975, PP. 232-260.                                PDECOL   253
C   3. HINDMARSH, A.C., PRELIMINARY DOCUMENTATION OF GEARIB.. SOLUTION PDECOL 254
C        OF IMPLICIT SYSTEMS OF ORDINARY DIFFERENTIAL EQUATIONS WITH PDECOL   255
C        BANDED JACOBIANS, LAWRENCE LIVERMORE LAB, UCID-30130, FEBRUARY PDECOL 256
C        1976.                                                       PDECOL   257
C   4. DEBOOR, C., PACKAGE FOR CALCULATING WITH B-SPLINES, SIAM J.   PDECOL   258
C        NUMER. ANAL., VOL. 14, NO. 3, JUNE 1977, PP. 441-472.       PDECOL   259
C----------------------------------------------------------------------- PDECOL 260
C                                                                    PDECOL   261
C USE OF PDECOL                                                      PDECOL   262
C                                                                    PDECOL   263
C PDECOL IS CALLED ONCE FOR EACH DESIRED OUTPUT VALUE (TOUT) OF THE  PDECOL   264
C TIME T, AND IT IN TURN MAKES REPEATED CALLS TO THE CORE INTEGRATOR, PDECOL  265
C STIFIB, WHICH ADVANCES THE TIME BY TAKING SINGLE STEPS UNTIL       PDECOL   266
C T .GE. TOUT.  INTERPOLATION TO THE EXACT TIME TOUT IS THEN DONE.   PDECOL   267
C SEE TOUT BELOW.                                                    PDECOL   268
C                                                                    PDECOL   269
C                                                                    PDECOL   270
C SUMMARY OF SUGGESTED INPUT VALUES                                  PDECOL   271
C                                                                    PDECOL   272
C   IT IS OF COURSE IMPOSSIBLE TO SUGGEST INPUT PARAMETER VALUES WHICH PDECOL 273
C   ARE APPROPRIATE FOR ALL PROBLEMS.  THE FOLLOWING SUGGESTIONS ARE TO PDECOL 274
C   BE USED ONLY IF YOU HAVE NO IDEA OF BETTER VALUES FOR YOUR PROBLEM. PDECOL 275
C                                                                    PDECOL   276
C   DT    = 1.E-10                                                   PDECOL   277
C   XBKPT = CHOOSE  NINT+1  EQUALLY SPACED VALUES SUCH THAT XBKPT(1) = PDECOL 278
C           XLEFT AND  XBKPT(NINT+1) = XRIGHT.                       PDECOL   279
C   EPS   = 1.E-4                                                    PDECOL   280
C   NINT  = ENOUGH SO THAT ANY FINE STRUCTURE OF THE PROBLEM MAY BE  PDECOL   281
C           RESOLVED.                                                PDECOL   282
C   KORD  = 4                                                        PDECOL   283
C   NCC   = 2                                                        PDECOL   284
C   MF    = 22                                                       PDECOL   285
C   INDEX = 1 (ON FIRST CALL ONLY, THEN 0 THEREAFTER).               PDECOL   286
```

```
C                                                                       PDECOL   287
C                                                                       PDECOL   288
C THE INPUT PARAMETERS ARE..                                            PDECOL   289
C   TO     =  THE INITIAL VALUE OF T, THE INDEPENDENT VARIABLE          PDECOL   290
C               (USED ONLY ON FIRST CALL).                              PDECOL   291
C   TOUT   =  THE VALUE OF T AT WHICH OUTPUT IS DESIRED NEXT.  SINCE    PDECOL   292
C               THE PACKAGE CHOOSES ITS OWN TIME STEP SIZES, THE        PDECOL   293
C               INTEGRATION WILL NORMALLY GO SLIGHTLY BEYOND TOUT       PDECOL   294
C               AND THE PACKAGE WILL INTERPOLATE TO T = TOUT.           PDECOL   295
C   DT     =  THE INITIAL STEP SIZE IN T, IF INDEX = 1, OR, THE         PDECOL   296
C               MAXIMUM STEP SIZE ALLOWED (MUST BE .GT. 0), IF INDEX = 3. PDECOL 297
C               USED FOR INPUT ONLY WHEN INDEX = 1 OR 3. SEE BELOW.     PDECOL   298
C   XBKPT  =  THE ARRAY OF PIECEWISE POLYNOMIAL BREAKPOINTS.            PDECOL   299
C               THE NINT+1 VALUES MUST BE STRICTLY INCREASING WITH      PDECOL   300
C               XBKPT(1) = XLEFT AND XBKPT(NINT+1) = XRIGHT (USED ONLY  PDECOL   301
C               ON FIRST CALL).                                         PDECOL   302
C   EPS    =  THE RELATIVE TIME ERROR BOUND  (USED ONLY ON THE          PDECOL   303
C               FIRST CALL, UNLESS INDEX = 4).  SINGLE STEP ERROR       PDECOL   304
C               ESTIMATES DIVIDED BY CMAX(I) WILL BE KEPT LESS THAN     PDECOL   305
C               EPS IN ROOT-MEAN-SQUARE NORM. THE VECTOR CMAX OF WEIGHTS PDECOL  306
C               IS COMPUTED IN PDECOL.  INITIALLY CMAX(I) IS SET TO     PDECOL   307
C               ABS(C(I)), WITH A DEFAULT VALUE OF 1 IF ABS(C(I)) .LT. 1. PDECOL 308
C               THEREAFTER, CMAX(I) IS THE LARGEST VALUE                PDECOL   309
C               OF ABS(C(I)) SEEN SO FAR, OR THE INITIAL CMAX(I) IF     PDECOL   310
C               THAT IS LARGER.  TO ALTER EITHER OF THESE, CHANGE THE   PDECOL   311
C               APPROPRIATE STATEMENTS IN THE DO-LOOPS ENDING AT        PDECOL   312
C               STATEMENTS 50 AND 130 BELOW.  THE USER SHOULD EXERCISE  PDECOL   313
C               SOME DISCRETION IN CHOOSING EPS.  IN GENERAL, THE       PDECOL   314
C               OVERALL RUNNING TIME FOR A PROBLEM WILL BE GREATER IF   PDECOL   315
C               EPS IS CHOSEN SMALLER. THERE IS USUALLY LITTLE REASON TO PDECOL  316
C               CHOOSE EPS MUCH SMALLER THAN THE ERRORS WHICH ARE BEING PDECOL   317
C               INTRODUCED BY THE USERS CHOICE OF THE POLYNOMIAL SPACE. PDECOL   318
C               SEE RELATED COMMENTS CONCERNING CMAX BELOW STATEMENT 40. PDECOL  319
C   NINT   =  THE NUMBER OF SUBINTERVALS INTO WHICH THE SPATIAL DOMAIN  PDECOL   320
C               (XLEFT,XRIGHT) IS TO BE DIVIDED (MUST BE .GE. 1)        PDECOL   321
C               (USED ONLY ON FIRST CALL).                              PDECOL   322
C   KORD   =  THE ORDER OF THE PIECEWISE POLYNOMIAL SPACE TO BE USED.   PDECOL   323
C               ITS VALUE MUST BE GREATER THAN 2 AND LESS THAN 21.  FOR PDECOL   324
C               FIRST ATTEMPTS WE RECOMMEND KORD = 4.  IF THE SOLUTION  PDECOL   325
C               IS SMOOTH AND MUCH ACCURACY IS DESIRED, HIGHER VALUES   PDECOL   326
C               MAY PROVE TO BE MORE EFFICIENT.  WE HAVE SELDOM USED    PDECOL   327
C               VALUES OF KORD IN EXCESS OF 8 OR 9, THOUGH THEY ARE     PDECOL   328
C               AVAILABLE FOR USE IN PDECOL (USED ONLY ON FIRST CALL).  PDECOL   329
C   NCC    =  THE NUMBER OF CONTINUITY CONDITIONS TO BE IMPOSED ON THE  PDECOL   330
C               APPROXIMATE SOLUTION AT THE BREAKPOINTS IN XBKPT.       PDECOL   331
C               NCC MUST BE GREATER THAN 1 AND LESS THAN KORD.  WE      PDECOL   332
C               RECOMMEND THE USE OF NCC = 2 (WITH NOGAUS = 0, SEE      PDECOL   333
C               BELOW), SINCE THEORY PREDICTS THAT DRAMATICALLY MORE    PDECOL   334
C               ACCURATE RESULTS CAN OFTEN BE OBTAINED USING THIS CHOICE PDECOL  335
C               (USED ONLY ON FIRST CALL).                              PDECOL   336
C   NPDE   =  THE NUMBER OF PARTIAL DIFFERENTIAL EQUATIONS IN THE SYSTEM PDECOL  337
C               TO BE SOLVED (USED ONLY ON FIRST CALL).                 PDECOL   338
C   MF     =  THE METHOD FLAG  (USED ONLY ON FIRST CALL, UNLESS         PDECOL   339
C               INDEX = 4).  ALLOWED VALUES ARE 11, 12, 21, 22.         PDECOL   340
C               FOR FIRST ATTEMPTS WE RECOMMEND THE USE OF MF = 22.     PDECOL   341
C               MF HAS TWO DECIMAL DIGITS, METH AND MITER               PDECOL   342
C               (MF = 10*METH + MITER).                                 PDECOL   343
```

```
C                METH IS THE BASIC METHOD INDICATOR..
C                  METH = 1  MEANS THE ADAMS METHODS (GENERALIZATIONS OF
C                            CRANK-NICOLSON).
C                  METH = 2  MEANS THE BACKWARD DIFFERENTIATION
C                            FORMULAS (BDF), OR STIFF METHODS OF GEAR.
C               MITER IS THE ITERATION METHOD INDICATOR
C                AND DETERMINES HOW THE JACOBIAN MATRIX IS
C                TO BE COMPUTED..
C                  MITER = 1 MEANS CHORD METHOD WITH ANALYTIC JACOBIAN.
C                            FOR THIS USER SUPPLIES SUBROUTINE DERIVF.
C                            SEE DESCRIPTION ABOVE.
C                  MITER = 2 MEANS CHORD METHOD WITH JACOBIAN CALCULATED
C                            INTERNALLY BY FINITE DIFFERENCES.  SEE
C                            SUBROUTINES PSETIB AND DIFFF.
C     INDEX =  INTEGER USED ON INPUT TO INDICATE TYPE OF CALL,
C                WITH THE FOLLOWING VALUES AND MEANINGS..
C                   1     THIS IS THE FIRST CALL FOR THIS PROBLEM.
C                   0     THIS IS NOT THE FIRST CALL FOR THIS PROBLEM,
C                         AND INTEGRATION IS TO CONTINUE.
C                   2     SAME AS 0 EXCEPT THAT TOUT IS TO BE HIT
C                         EXACTLY (NO INTERPOLATION IS DONE).  SEE NOTE
C                         BELOW.  ASSUMES TOUT .GE. THE CURRENT T.
C                         IF TOUT IS .LT. THE CURRENT TIME, THEN TOUT IS
C                         RESET TO THE CURRENT TIME AND CONTROL IS
C                         RETURNED TO THE USER.  A CALL TO VALUES WILL
C                         PRODUCE ANSWERS FOR THE NEW VALUE OF TOUT.
C                   3     SAME AS 0 EXCEPT CONTROL RETURNS TO CALLING
C                         PROGRAM AFTER ONE STEP.  TOUT IS IGNORED AND
C                         DT MUST BE SET .GT. 0 TO A MAXIMUM ALLOWED
C                         DT VALUE. SEE ABOVE.
C                   4     THIS IS NOT THE FIRST CALL FOR THE PROBLEM,
C                         AND THE USER HAS RESET EPS AND/OR MF.
C                SINCE THE NORMAL OUTPUT VALUE OF INDEX IS 0,
C                IT NEED NOT BE RESET FOR NORMAL CONTINUATION.
C
C NOTE.. THE PACKAGE MUST HAVE TAKEN AT LEAST ONE SUCCESSFUL TIME
C STEP BEFORE A CALL WITH INDEX = 2 OR 4 IS ALLOWED.
C AFTER THE INITIAL CALL, IF A NORMAL RETURN OCCURRED AND A NORMAL
C CONTINUATION IS DESIRED, SIMPLY RESET TOUT AND CALL AGAIN.
C ALL OTHER PARAMETERS WILL BE READY FOR THE NEXT CALL.
C A CHANGE OF PARAMETERS WITH INDEX = 4 CAN BE MADE AFTER
C EITHER A SUCCESSFUL OR AN UNSUCCESSFUL RETURN PROVIDED AT LEAST
C ONE SUCCESSFUL TIME STEP HAS BEEN MADE.
C
C     WORK  =  FLOATING POINT WORKING ARRAY FOR PDECOL.  WE RECOMMEND
C                THAT IT BE INITIALIZED TO ZERO BEFORE THE FIRST CALL
C                TO PDECOL.  ITS TOTAL LENGTH MUST BE AT LEAST
C
C                KORD + 4*NPDE + 9*NPDE**2 + NCPTS*(3*KORD + 2) +
C                NPDE*NCPTS*(3*ML + MAXDER + 7)
C
C                WHERE ML AND MAXDER ARE DEFINED BELOW (SEE STORAGE
C                ALLOCATION).
C     IWORK =  INTEGER WORKING ARRAY FOR PDECOL.  THE FIRST TWO
C                LOCATIONS MUST BE DEFINED AS FOLLOWS...
C                IWORK(1) = LENGTH OF USERS ARRAY WORK
C                IWORK(2) = LENGTH OF USERS ARRAY IWORK
```

```
      C               THE TOTAL LENGTH OF IWORK MUST BE AT LEAST           PDECOL   401
      C               NCPTS*(NPDE + 1).                                    PDECOL   402
      C OUTPUT                                                             PDECOL   403
      C                                                                    PDECOL   404
      C THE SOLUTION VALUES ARE NOT RETURNED DIRECTLY TO THE USER BY PDECOL.  PDECOL   405
      C THE METHODS USED IN PDECOL COMPUTE BASIS FUNCTION COEFFICIENTS, SO   PDECOL   406
      C THE USER (AFTER A RETURN FROM PDECOL) MUST CALL THE PACKAGE ROUTINE  PDECOL   407
      C VALUES TO OBTAIN HIS APPROXIMATE SOLUTION VALUES AT ANY DESIRED SPACE PDECOL  408
      C POINTS X AT THE TIME T = TOUT.  SEE THE COMMENTS IN SUBROUTINE VALUES PDECOL  409
      C FOR DETAILS ON HOW TO PROPERLY MAKE THE CALL.                        PDECOL   410
      C                                                                    PDECOL   411
      C THE COMMON BLOCK /GEAR0/ CAN BE ACCESSED EXTERNALLY BY THE USER      PDECOL   412
      C IF DESIRED.  IT CONTAINS THE STEP SIZE LAST USED (SUCCESSFULLY),    PDECOL   413
      C THE ORDER LAST USED (SUCCESSFULLY), THE NUMBER OF STEPS TAKEN       PDECOL   414
      C SO FAR, THE NUMBER OF RESIDUAL EVALUATIONS (RES CALLS) SO FAR,      PDECOL   415
      C AND THE NUMBER OF MATRIX EVALUATIONS (PSETIB CALLS) SO FAR.        PDECOL   416
      C DIFFUN CALLS ARE COUNTED IN WITH RESIDUAL EVALUATIONS.              PDECOL   417
      C                                                                    PDECOL   418
      C THE OUTPUT PARAMETERS ARE..                                        PDECOL   419
      C   DT    =  THE STEP SIZE USED LAST, WHETHER SUCCESSFULLY OR NOT.   PDECOL   420
      C   TOUT  =  THE OUTPUT VALUE OF T.  IF INTEGRATION WAS SUCCESSFUL,  PDECOL   421
      C               AND THE INPUT VALUE OF INDEX WAS NOT 3, TOUT IS      PDECOL   422
      C               UNCHANGED FROM ITS INPUT VALUE.  OTHERWISE, TOUT     PDECOL   423
      C               IS THE CURRENT VALUE OF T TO WHICH THE INTEGRATION   PDECOL   424
      C               HAS BEEN COMPLETED.                                  PDECOL   425
      C   INDEX =  INTEGER USED ON OUTPUT TO INDICATE RESULTS,             PDECOL   426
      C               WITH THE FOLLOWING VALUES AND MEANINGS..             PDECOL   427
      C         0     INTEGRATION WAS COMPLETED TO TOUT OR BEYOND.         PDECOL   428
      C        -1     THE INTEGRATION WAS HALTED AFTER FAILING TO PASS THE PDECOL   429
      C               ERROR TEST EVEN AFTER REDUCING DT BY A FACTOR OF     PDECOL   430
      C               1.E10 FROM ITS INITIAL VALUE.                        PDECOL   431
      C        -2     AFTER SOME INITIAL SUCCESS, THE INTEGRATION WAS      PDECOL   432
      C               HALTED EITHER BY REPEATED ERROR TEST FAILURES OR BY  PDECOL   433
      C               A TEST ON EPS.  TOO MUCH ACCURACY HAS BEEN REQUESTED. PDECOL  434
      C        -3     THE INTEGRATION WAS HALTED AFTER FAILING TO ACHIEVE  PDECOL   435
      C               CORRECTOR CONVERGENCE EVEN AFTER REDUCING DT BY A    PDECOL   436
      C               FACTOR OF 1.E10 FROM ITS INITIAL VALUE.              PDECOL   437
      C        -4     SINGULAR MATRIX ENCOUNTERED.  PROBABLY DUE TO STORAGE PDECOL  438
      C               OVERWRITES.                                          PDECOL   439
      C        -5     INDEX WAS 4 ON INPUT, BUT THE DESIRED CHANGES OF     PDECOL   440
      C               PARAMETERS WERE NOT IMPLEMENTED BECAUSE TOUT         PDECOL   441
      C               WAS NOT BEYOND T.  INTERPOLATION TO T = TOUT WAS     PDECOL   442
      C               PERFORMED AS ON A NORMAL RETURN.  TO TRY AGAIN,      PDECOL   443
      C               SIMPLY CALL AGAIN WITH INDEX = 4 AND A NEW TOUT.     PDECOL   444
      C        -6     ILLEGAL INDEX VALUE.                                 PDECOL   445
      C        -7     ILLEGAL EPS VALUE.                                   PDECOL   446
      C        -8     AN ATTEMPT TO INTEGRATE IN THE WRONG DIRECTION.  THE PDECOL   447
      C               SIGN OF DT IS WRONG RELATIVE TO T0 AND TOUT.         PDECOL   448
      C        -9     DT .EQ. 0.0.                                         PDECOL   449
      C       -10     ILLEGAL NINT VALUE.                                  PDECOL   450
      C       -11     ILLEGAL KORD VALUE.                                  PDECOL   451
      C       -12     ILLEGAL NCC VALUE.                                   PDECOL   452
      C       -13     ILLEGAL NPDE VALUE.                                  PDECOL   453
      C       -14     ILLEGAL MF VALUE.                                    PDECOL   454
      C       -15     ILLEGAL BREAKPOINTS - NOT STRICTLY INCREASING.       PDECOL   455
      C       -16     INSUFFICIENT STORAGE FOR WORK OR IWORK.              PDECOL   456
      C----------------------------------------------------------------- PDECOL   457
```

```
C                                                                                  PDECOL     458
C                                                                                  PDECOL     459
C SUMMARY OF ALL PACKAGE ROUTINES                                                  PDECOL     460
C                                                                                  PDECOL     461
C PDECOL - STORAGE ALLOCATION, ERROR CHECKING, INITIALIZATION, REPEATED            PDECOL     462
C          CALLS TO STIFIB TO ADVANCE THE TIME.                                    PDECOL     463
C                                                                                  PDECOL     464
C INTERP - INTERPOLATES COMPUTED BASIS FUNCTION COEFFICIENTS TO THE                PDECOL     465
C          DESIRED OUTPUT TIMES, TOUT, FOR USE BY VALUES.                          PDECOL     466
C                                                                                  PDECOL     467
C INITAL - INITIALIZATION, GENERATION AND STORAGE OF PIECEWISE POLY-               PDECOL     468
C          NOMIAL SPACE BASIS FUNCTION VALUES AND DERIVATIVES, DET-                PDECOL     469
C          ERMINES THE BASIS FUNCTION COEFFICINTS OF THE PIECEWISE                 PDECOL     470
C          POLYNOMIALS WHICH INTERPOLATE THE USERS INITIAL CONDITIONS.             PDECOL     471
C                                                                                  PDECOL     472
C COLPNT - GENERATION OF REQUIRED COLLOCATION POINTS.                              PDECOL     473
C                                                                                  PDECOL     474
C BSPLVD - B-SPLINE PACKAGE ROUTINES WHICH ALLOW FOR EVALUATION OF                 PDECOL     475
C BSPLVN   ANY B-SPLINE BASIS FUNCTION OR DERIVATIVE VALUE.                        PDECOL     476
C INTERV                                                                           PDECOL     477
C                                                                                  PDECOL     478
C VALUES - GENERATION AT ANY POINT(S) OF VALUES OF THE COMPUTED                    PDECOL     479
C          APPROXIMATE SOLUTION AND ITS DERIVATIVES WHICH ARE                      PDECOL     480
C          PIECEWISE POLYNOMIALS.  THE SUBROUTINE IS CALLED ONLY BY                PDECOL     481
C          THE USER.                                                               PDECOL     482
C                                                                                  PDECOL     483
C STIFIB - CORE INTEGRATOR, TAKES SINGLE TIME STEPS TO ADVANCE THE                 PDECOL     484
C          TIME.  ASSEMBLES AND SOLVES THE PROPER NONLINEAR EQUATIONS              PDECOL     485
C          WHICH ARE RELATED TO USE OF ADAMS OR GEAR TYPE INTEGRATION              PDECOL     486
C          FORMULAS.  CHOOSES PROPER STEP SIZE AND INTEGRATION FORMULA             PDECOL     487
C          ORDER TO MAINTAIN A DESIRED ACCURACY.  DESIGNED FOR ODE                 PDECOL     488
C          PROBLEMS OF THE FORM A * (DY/DT) = G(T,Y).                              PDECOL     489
C                                                                                  PDECOL     490
C COSET  - GENERATES INTEGRATION FORMULA AND ERROR CONTROL COEFFICIENTS.           PDECOL     491
C                                                                                  PDECOL     492
C RES    - COMPUTES RESIDUAL VECTORS USED IN SOLVING THE NONLINEAR                 PDECOL     493
C          EQUATIONS BY A MODIFIED NEWTON METHOD.                                  PDECOL     494
C                                                                                  PDECOL     495
C DIFFUN - COMPUTES A**-1 * G(T,Y) WHERE A AND G ARE AS ABOVE (STIFIB).            PDECOL     496
C                                                                                  PDECOL     497
C ADDA   - ADDS THE A MATRIX TO A GIVEN MATRIX IN BAND FORM.                       PDECOL     498
C                                                                                  PDECOL     499
C EVAL   - EVALUATES THE COMPUTED PIECEWISE POLYNOMIAL SOLUTION AND                PDECOL     500
C          DERIVATIVES AT COLLOCATION POINTS.                                      PDECOL     501
C                                                                                  PDECOL     502
C GFUN   - EVALUATES THE FUNCTION G(T,Y) BY CALLING EVAL AND THE USER              PDECOL     503
C          SUBROUTINES F AND BNDRY.                                                PDECOL     504
C                                                                                  PDECOL     505
C PSETIB - GENERATES PROPER JACOBIAN MATRICES REQUIRED BY THE MODIFIED             PDECOL     506
C          NEWTON METHOD.                                                          PDECOL     507
C                                                                                  PDECOL     508
C DIFFF  - PERFORMS SAME ROLE AS THE USER ROUTINE DERIVF. COMPUTES                 PDECOL     509
C          DERIVATIVE APPROXIMATIONS BY USE OF FINITE DIFFERENCES.                 PDECOL     510
C                                                                                  PDECOL     511
C DECB   - PERFORM AN LU DECOMPOSTION AND FORWARD AND BACKWARD                     PDECOL     512
C SOLB     SUBSTITUTION FOR SOLVING BANDED SYSTEMS OF LINEAR EQUATIONS.            PDECOL     513
C                                                                                  PDECOL     514
```

```
C------------------------------------------------------------------------ PDECOL   515
C                                                                         PDECOL   516
C                                                                         PDECOL   517
C STORAGE ALLOCATION                                                      PDECOL   518
C                                                                         PDECOL   519
C SINCE PDECOL IS A DYNAMICALLY DIMENSIONED PROGRAM, MOST OF ITS          PDECOL   520
C WORKING STORAGE IS PROVIDED BY THE USER IN THE ARRAYS WORK AND IWORK.   PDECOL   521
C THE FOLLOWING GIVES A LIST OF THE ARRAYS WHICH MAKE UP THE CONTENTS     PDECOL   522
C WORK AND IWORK, THEIR LENGTHS, AND THEIR USES.  WHEN MORE THAN ONE      PDECOL   523
C NAME IS GIVEN, IT INDICATES THAT DIFFERENT NAMES ARE USED FOR THE       PDECOL   524
C SAME ARRAY IN DIFFERENT PARTS OF THE PROGRAM.  THE DIFFERENT NAMES      PDECOL   525
C OCCUR BECAUSE PDECOL IS AN AMALGAMATION OF SEVERAL OTHER CODES          PDECOL   526
C WRITTEN BY DIFFERENT PEOPLE AND WE HAVE TRIED TO LEAVE THE SEPARATE     PDECOL   527
C PARTS AS UNCHANGED FROM THEIR ORIGINAL VERSIONS AS POSSIBLE.            PDECOL   528
C                                                                         PDECOL   529
C                                                                         PDECOL   530
C   NAMES       LENGTH        USE                                         PDECOL   531
C ---------     ------------  ------------------------------------        PDECOL   532
C                                                                         PDECOL   533
C BC            4*NPDE**2     BOUNDARY CONDITION INFORMATION.             PDECOL   534
C WORK                                                                    PDECOL   535
C                                                                         PDECOL   536
C A             3*KORD*NCPTS  BASIS FUNCTION VALUES AT COLLOCATION POINT  PDECOL   537
C WORK(IW1)                                                               PDECOL   538
C                                                                         PDECOL   539
C XT            NCPTS + KORD  BREAKPOINT SEQUENCE FOR GENERATION OF BASI  PDECOL   540
C WORK(IW2)                   FUNCTION VALUES.                            PDECOL   541
C                                                                         PDECOL   542
C XC            NCPTS         COLLOCATION POINTS.                         PDECOL   543
C WORK(IW3)                                                               PDECOL   544
C                                                                         PDECOL   545
C CMAX          NPDE*NCPTS    VALUES USED IN ESTIMATING TIME              PDECOL   546
C YMAX                        INTEGRATION ERRORS.                         PDECOL   547
C WORK(IW4)                                                               PDECOL   548
C                                                                         PDECOL   549
C ERROR         NPDE*NCPTS    TIME INTEGRATION ERRORS.                    PDECOL   550
C WORK(IW5)                                                               PDECOL   551
C                                                                         PDECOL   552
C SAVE1         NPDE*NCPTS    WORKING STORAGE FOR THE TIME INTEGRATION    PDECOL   553
C WORK(IW6)                   METHOD.                                     PDECOL   554
C                                                                         PDECOL   555
C SAVE2         NPDE*NCPTS    WORKING STORAGE FOR THE TIME INTEGRATION    PDECOL   556
C WORK(IW7)                   METHOD.                                     PDECOL   557
C                                                                         PDECOL   558
C SAVE3         NPDE*NCPTS    WORKING STORAGE FOR THE TIME INTEGRATION    PDECOL   559
C WORK(IW8)                   METHOD.                                     PDECOL   560
C                                                                         PDECOL   561
C UVAL          3*NPDE        WORKING STORAGE FOR VALUES OF U, UX, AND    PDECOL   562
C WORK(IW9)                   USS AT ONE POINT.                           PDECOL   563
C                                                                         PDECOL   564
C C             NPDE*NCPTS*   CURRENT BASIS FUNCTION COEFFICIENT VALUES   PDECOL   565
C Y               (MAXDER+1)  AND THEIR SCALED TIME DERIVATIVES.          PDECOL   566
C WORK(IW10)                                                              PDECOL   567
C                                                                         PDECOL   568
C DFDU          NPDE**2       WORKING STORAGE USED TO COMPUTE THE         PDECOL   569
C WORK(IW11)                  JACOBIAN MATRIX.                            PDECOL   570
C                                                                         PDECOL   571
```

```
C DFDUX          NPDE**2        WORKING STORAGE USED TO COMPUTE THE        PDECOL   572
C WORK(IW12)                    JACOBIAN MATRIX.                           PDECOL   573
C                                                                          PDECOL   574
C DFDUXX         NPDE**2        WORKING STORAGE USED TO COMPUTE THE        PDECOL   575
C WORK(IW13)                    JACOBIAN MATRIX.                           PDECOL   576
C                                                                          PDECOL   577
C DBDU           NPDE**2        BOUNDARY CONDITION INFORMATION.            PDECOL   578
C WORK(IW14)                                                               PDECOL   579
C                                                                          PDECOL   580
C DBDUX          NPDE**2        BOUNDARY CONDITION INFORMATION.            PDECOL   581
C WORK(IW15)                                                               PDECOL   582
C                                                                          PDECOL   583
C DZDT           NPDE           BOUNDARY CONDITION INFORMATION.            PDECOL   584
C WORK(IW16)                                                               PDECOL   585
C                                                                          PDECOL   586
C PW             NPDE*NCPTS*    STORAGE AND PROCESSING OF THE JACOBIAN     PDECOL   587
C WORK(IW17)       (3*ML+1)     MATRIX.                                    PDECOL   588
C                                                                          PDECOL   589
C ILEFT          NCPTS          POINTERS TO BREAKPOINT SEQUENCE FOR        PDECOL   590
C IWORK                         GENERATION OF BASIS FUNCTION VALUES.       PDECOL   591
C                                                                          PDECOL   592
C IPIV           NPDE*NCPTS     PIVOT INFORMATION FOR THE LU DECOMPOSED    PDECOL   593
C IWORK(IW18)                   JACOBIAN MATRIX PW.                        PDECOL   594
C                                                                          PDECOL   595
C WHERE...                                                                 PDECOL   596
C                                                                          PDECOL   597
C      NCPTS = KORD*NINT - NCC*(NINT-1)                                    PDECOL   598
C         ML = NPDE*(KORD+IQUAD-1) - 1                                     PDECOL   599
C      IQUAD = 1 IF KORD = 3 AND A NULL BOUNDARY CONDITION EXISTS          PDECOL   600
C      IQUAD = 0 OTHERWISE                                                 PDECOL   601
C     MAXDER = 5 UNLESS OTHERWISE SET BY THE USER INTO /OPTION/.           PDECOL   602
C                                                                          PDECOL   603
C THE COMMON BLOCK /OPTION/ CONTAINS THE VARIABLES NOGAUS AND MAXDER.     PDECOL   604
C NOGAUS IS SET .EQ. 0 IN THE BLOCK DATA.  IT CAN BE CHANGED TO BE         PDECOL   605
C SET .EQ. 1 IF THE GAUSS-LEGENDRE COLLOCATION POINTS ARE NOT             PDECOL   606
C DESIRED WHEN NCC = 2 (SEE ABOVE AND COLPNT).  MAXDER IS SET              PDECOL   607
C .EQ. 5 IN THE BLOCK DATA AND ITS VALUE REPRESENTS THE                    PDECOL   608
C MAXIMUM ORDER OF TIME INTEGRATION FORMULA ALLOWED.  ITS VALUE            PDECOL   609
C AFFECTS THE STORAGE REQUIRED IN WORK AND MAY BE CHANGED IF               PDECOL   610
C DESIRED.  SEE COSET FOR RESTRICTIONS.  THESE CHANGES MAY BE MADE BY      PDECOL   611
C THE USER BY ACCESSING /OPTION/ IN HIS CALLING PROGRAM (BEFORE THE        PDECOL   612
C FIRST CALL TO PDECOL) OR BY CHANGING THE DATA STATEMENT IN               PDECOL   613
C THE BLOCK DATA.                                                          PDECOL   614
C                                                                          PDECOL   615
C------------------------------------------------------------------------ PDECOL   616
C                                                                          PDECOL   617
C                                                                          PDECOL   618
C COMMUNICATION                                                            PDECOL   619
C                                                                          PDECOL   620
C EACH SUBROUTINE IN THE PACKAGE CONTAINS A COMMUNICATION SUMMARY          PDECOL   621
C AS INDICATED BELOW.                                                      PDECOL   622
C                                                                          PDECOL   623
C PACKAGE ROUTINES CALLED..   EVAL,INITAL,INTERP,STIFIB                    PDECOL   624
C USER ROUTINES CALLED..      BNDRY                                        PDECOL   625
C CALLED BY..                 USERS MAIN PROGRAM                           PDECOL   626
C FORTRAN FUNCTIONS USED..    ABS,AMAX1,FLOAT,SQRT                         PDECOL   627
C------------------------------------------------------------------------ PDECOL   628
```

```
      DIMENSION WORK(1),IWORK(1),XBKPT(1)                               PDECOL 629
C  SREC ALLOWS FOR THE MODIFICATION OF THE ERROR CONTROL.               TC1      1
C  AN ABSOLUTE ERROR CRITERION IS USED FOR QUANTITIES LESS THAN SREC.   TC1      2
C  XCPTS KEEPS TRACK OF THE COLLOCATION POINTS GENERATED BY PDECUL.     TC1      3
      COMMON/TABF/SREC,XCPTS(100)                                       TC1      4
      COMMON /GEAR0/ DTUSED,NQUSED,NSTEP,NFE,NJE                        PDECOL 630
      COMMON /GEAR1/ T,DTC,DTMN,DTMX,EPSC,UROUND,N,MFC,KFLAG,JSTART     PDECOL 631
      COMMON /GEAR9/ EPSJ,R0,ML,MU,MW,NM1,N0ML,N0W                      PDECOL 632
      COMMON /OPTION/ NOGAUS,MAXDER                                     PDECOL 633
      COMMON /SIZES/ NIN,KOR,NC,NPD,NCPTS,NEQN,IQUAD                    PDECOL 634
      COMMON /ISTART/ IW1,IW2,IW3,IW4,IW5,IW6,IW7,IW8,IW9,IW10,         PDECOL 635
     *                IW11,IW12,IW13,IW14,IW15,IW16,IW17,IW18           PDECOL 636
      COMMON /IOUNIT/ LOUT                                              PDECOL 637
      IF (INDEX .EQ. 0) GO TO 60                                        PDECOL 638
      IF (INDEX .EQ. 2) GO TO 70                                        PDECOL 639
      IF (INDEX .EQ. 4) GO TO 80                                        PDECOL 640
      IF (INDEX .EQ. 3) GO TO 90                                        PDECOL 641
C-----------------------------------------------------------------------PDECOL 642
C SEVERAL CHECKS ARE MADE HERE TO DETERMINE IF THE INPUT PARAMETERS     PDECOL 643
C HAVE LEGAL VALUES.  ERROR CHECKS ARE MADE ON INDEX, EPS, (T0-TOUT)*DT, PDECOL 644
C DT, NINT, KORD, NCC, NPDE, MF, WHETHER THE BREAKPOINT SEQUENCE IS     PDECOL 645
C STRICTLY INCREASING, AND WHETHER THERE IS SUFFICIENT STORAGE          PDECOL 646
C PROVIDED FOR WORK AND IWORK.  PROBLEM DEPENDENT PARAMETERS ARE        PDECOL 647
C CALCULATED AND PLACED IN COMMON.                                      PDECOL 648
C-----------------------------------------------------------------------PDECOL 649
      IERID = -6                                                        PDECOL 650
      IF (INDEX .NE. 1) GO TO 320                                       PDECOL 651
      IERID = IERID - 1                                                 PDECOL 652
      IF (EPS .LE. 0.) GO TO 320                                        PDECOL 653
      IERID = IERID - 1                                                 PDECOL 654
      IF ((T0-TOUT)*DT .GT. 0.) GO TO 320                               PDECOL 655
      IERID = IERID - 1                                                 PDECOL 656
      IF (DT .EQ. 0.0) GO TO 320                                        PDECOL 657
      IERID = IERID - 1                                                 PDECOL 658
      NIN = NINT                                                        PDECOL 659
      IF (NIN .LT. 1) GO TO 320                                         PDECOL 660
      IERID = IERID - 1                                                 PDECOL 661
      KOR = KORD                                                        PDECOL 662
      IF (KOR .LT. 3  .OR.  KOR .GT. 20) GO TO 320                      PDECOL 663
      IERID = IERID - 1                                                 PDECOL 664
      NC = NCC                                                          PDECOL 665
      IF (NCC .LT. 2  .OR.  NCC .GE. KOR) GO TO 320                     PDECOL 666
      IERID = IERID - 1                                                 PDECOL 667
      NPD = NPDE                                                        PDECOL 668
      NPDE2 = NPD*NPD                                                   PDECOL 669
      IF (NPDE .LT. 1) GO TO 320                                        PDECOL 670
      IERID = IERID - 1                                                 PDECOL 671
      IF (MF.NE.22.AND.MF.NE.21.AND.MF.NE.12.AND.MF.NE.11) GO TO 320    PDECOL 672
      IERID = IERID - 1                                                 PDECOL 673
      DO 10 K=1,NIN                                                     PDECOL 674
        IF(XBKPT(K) .GE. XBKPT(K+1)) GO TO 320                          PDECOL 675
   10 CONTINUE                                                          PDECOL 676
      NCPTS = KOR + (NIN - 1) * (KOR - NCC)                             PDECOL 677
      NEQN = NPDE * NCPTS                                               PDECOL 678
      ML = (KOR-1)*NPDE - 1                                             PDECOL 679
      MU = ML                                                           PDECOL 680
      MW = ML + ML + 1                                                  PDECOL 681
```

```
      NOW = NEQN*MW                                                     PDECOL     682
      IWSAVE = IWORK(1)                                                 PDECOL     683
      IISAVE = IWORK(2)                                                 PDECOL     684
      IW1 = 4*NPDE2 + 1                                                 PDECOL     685
      IW2 = IW1 + 3*KORD*NCPTS                                          PDECOL     686
      IW3 = IW2 + NCPTS + KORD                                          PDECOL     687
      IW4 = IW3 + NCPTS                                                 PDECOL     688
      IW5 = IW4 + NEQN                                                  PDECOL     689
      IW6 = IW5 + NEQN                                                  PDECOL     690
      IW7 = IW6 + NEQN                                                  PDECOL     691
      IW8 = IW7 + NEQN                                                  PDECOL     692
      IW9 = IW8 + NEQN                                                  PDECOL     693
      IW10 = IW9 + 3*NPDE                                               PDECOL     694
      IW11 = IW10 + NEQN*(MAXDER+1)                                     PDECOL     695
      IW12 = IW11 + NPDE2                                               PDECOL     696
      IW13 = IW12 + NPDE2                                               PDECOL     697
      IW14 = IW13 + NPDE2                                               PDECOL     698
      IW15 = IW14 + NPDE2                                               PDECOL     699
      IW16 = IW15 + NPDE2                                               PDECOL     700
      IW17 = IW16 + NPDE                                                PDECOL     701
      IW18 = NCPTS + 1                                                  PDECOL     702
      IERID = IERID - 1                                                 PDECOL     703
      IWSTOR = IW17 + NEQN*(3*ML+1) - 1                                 PDECOL     704
      IISTOR = IW18 + NEQN - 1                                          PDECOL     705
      IF ( IWSAVE .LT. IWSTOR  .OR.  IISAVE .LT. IISTOR ) GO TO 335     PDECOL     706
C-----------------------------------------------------------------------PDECOL     707
C PERFORM INITIALIZATION TASKS.  IF KORD .EQ. 3 THEN CALCULATE THE BAND-PDECOL     708
C WIDTH OF THE ASSOCIATED MATRIX PROBLEM BY DETERMINING THE TYPE OF     PDECOL     709
C BOUNDARY CONDITIONS, THEN CHECK FOR SUFFICIENT STORAGE AGAIN.         PDECOL     710
C-----------------------------------------------------------------------PDECOL     711
      CALL INITAL(KOR,WORK(IW1),WORK(IW6),XBKPT,WORK(IW2),WORK(IW3),    PDECOL     712
     *            WORK(IW17),IWORK(IW18),IWORK)                         PDECOL     713
      DO 5 K=1,NCPTS                                                    TC1          5
      KP=IW3+K-1                                                        TC1          6
    5 XCPTS(K)=WORK(KP)                                                 TC1          7
      IF(IQUAD .NE. 0) GO TO 280                                        PDECOL     714
      IF( KOR .NE. 3 ) GO TO 40                                         PDECOL     715
      CALL EVAL(1,NPDE,WORK(IW6),WORK(IW9),WORK(IW1),IWORK)             PDECOL     716
      CALL BNDRY(T0,WORK(IW3),WORK(IW9),WORK(IW9+NPDE),WORK(IW14),      PDECOL     717
     *           WORK(IW15),WORK(IW16),NPDE)                            PDECOL     718
      DO 20 K=1,NPDE                                                    PDECOL     719
        I = K + NPDE*(K-1) - 1                                          PDECOL     720
        IF(WORK(IW14+I) .EQ. 0.0 .AND. WORK(IW15+I) .EQ. 0.0)           PDECOL     721
     *     IQUAD = 1                                                    PDECOL     722
   20 CONTINUE                                                          PDECOL     723
      CALL EVAL(NCPTS,NPDE,WORK(IW6),WORK(IW9),WORK(IW1),IWORK)         PDECOL     724
      CALL BNDRY(T0,WORK(IW3+NCPTS-1),WORK(IW9),WORK(IW9+NPDE),         PDECOL     725
     *           WORK(IW14),WORK(IW15),WORK(IW16),NPDE)                 PDECOL     726
      DO 30 K=1,NPDE                                                    PDECOL     727
        I = K + NPDE*(K-1) - 1                                          PDECOL     728
        IF(WORK(IW14+I) .EQ. 0.0 .AND. WORK(IW15+I) .EQ. 0.0)           PDECOL     729
     *     IQUAD = 1                                                    PDECOL     730
   30 CONTINUE                                                          PDECOL     731
      ML = ML + IQUAD*NPDE                                              PDECOL     732
      MU = ML                                                           PDECOL     733
      MW = ML + ML + 1                                                  PDECOL     734
      NOW = NEQN*MW                                                     PDECOL     735
```

```
   40 CONTINUE                                                           PDECOL   736
      IWSTOR = IW17 + NEQN*(3*ML+1) - 1                                  PDECOL   737
      IF ( IWSAVE .LT. IWSTOR ) GO TO 335                                PDECOL   738
C----------------------------------------------------------------------- PDECOL   739
C IF INITIAL VALUES OF CMAX OTHER THAN THOSE SET BELOW ARE DESIRED,      PDECOL   740
C THEY SHOULD BE SET HERE.  ALL CMAX(I) MUST BE POSITIVE.                PDECOL   741
C HAVING PROPER VALUES OF CMAX FOR THE PROBLEM BEING SOLVED IS AS        PDECOL   742
C IMPORTANT AS CHOOSING EPS (SEE ABOVE), SINCE ERRORS ARE                PDECOL   743
C MEASURED RELATIVE TO CMAX.  IF VALUES FOR DTMN OR DTMX, THE            PDECOL   744
C BOUNDS ON ABS(DT), OTHER THAN THOSE BELOW ARE DESIRED, THEY            PDECOL   745
C SHOULD BE SET BELOW.                                                   PDECOL   746
C----------------------------------------------------------------------- PDECOL   747
      DO 50 I = 1,NEQN                                                   PDECOL   748
        I1 = I - 1                                                       PDECOL   749
      WORK(IW4+I1)=AMAX1(ABS(WORK(IW6+I1)),SHEC)                         TC1        8
   50   WORK(IW10+I1) = WORK(IW6+I1)                                     PDECOL   752
      N = NEQN                                                           PDECOL   753
      T = T0                                                             PDECOL   754
      DTC = DT                                                           PDECOL   755
      DTMN = ABS(DT)                                                     PDECOL   756
      DTUSED = 0.                                                        PDECOL   757
      EPSC = EPS                                                         PDECOL   758
      MFC = MF                                                           PDECOL   759
      JSTART = 0                                                         PDECOL   760
      EPSJ =  SQRT(UROUND)                                               PDECOL   761
      NM1 = NEQN - 1                                                     PDECOL   762
      NQML = NEQN*ML                                                     PDECOL   763
      NHCUT = 0                                                          PDECOL   764
      KFLAG = 0                                                          PDECOL   765
      TOUTP = T0                                                         PDECOL   766
      IF ( T0 .EQ. TOUT ) GO TO 360                                      PDECOL   767
   60 DTMX = ABS(TOUT-TOUTP)*10.                                         PDECOL   768
      GO TO 140                                                          PDECOL   769
C                                                                        PDECOL   770
   70 DTMX = ABS(TOUT-TOUTP)*10.                                         PDECOL   771
      IF ((T-TOUT)*DTC .GE. 0.) GO TO 340                                PDECOL   772
      GO TO 150                                                          PDECOL   773
C                                                                        PDECOL   774
   80 IF ((T-TOUT)*DTC .GE. 0.) GO TO 300                                PDECOL   775
      JSTART = -1                                                        PDECOL   776
      EPSC = EPS                                                         PDECOL   777
      MFC = MF                                                           PDECOL   778
      GO TO 100                                                          PDECOL   779
C                                                                        PDECOL   780
   90 DTMX = DT                                                          PDECOL   781
  100 IF ((T+DTC) .EQ. T) WRITE(LOUT,110)                                PDECOL   782
  110 FORMAT(36H WARNING.,  T + DT = T ON NEXT STEP.)                    PDECOL   783
C----------------------------------------------------------------------- PDECOL   784
C TAKE A TIME STEP BY CALLING THE INTEGRATOR.                            PDECOL   785
C----------------------------------------------------------------------- PDECOL   786
      CALL STIFIB (NEQN,WORK(IW10),WORK(IW4),WORK(IW5),WORK(IW6),        PDECOL   787
     *        WORK(IW7),WORK(IW8),WORK(IW17),IWORK(IW18),WORK,IWORK)     PDECOL   788
C                                                                        PDECOL   789
      KGO = 1 - KFLAG                                                    PDECOL   790
      GO TO (120, 160, 220, 260, 280), KGO                               PDECOL   791
C KFLAG  =   0,  -1,  -2,  -3   -4                                       PDECOL   792
C                                                                        PDECOL   793
```

```
      120 CONTINUE                                                        PDECOL 794
C-----------------------------------------------------------------------  PDECOL 795
C NORMAL RETURN FROM INTEGRATOR.                                          PDECOL 796
C                                                                         PDECOL 797
C THE WEIGHTS CMAX(I) ARE UPDATED.  IF DIFFERENT VALUES ARE DESIRED,      PDECOL 798
C THEY SHOULD BE SET HERE.  A TEST IS MADE FOR EPS BEING TOO SMALL        PDECOL 799
C FOR THE MACHINE PRECISION.                                              PDECOL 800
C                                                                         PDECOL 801
C ANY OTHER TESTS OR CALCULATIONS THAT ARE REQUIRED AFTER EVERY           PDECOL 802
C STEP SHOULD BE INSERTED HERE.                                           PDECOL 803
C                                                                         PDECOL 804
C IF INDEX = 3, SAVE1 IS SET TO THE CURRENT C VALUES ON RETURN.           PDECOL 805
C IF INDEX = 2, DT IS CONTROLLED TO HIT TOUT (WITHIN ROUNDOFF             PDECOL 806
C ERROR), AND THEN THE CURRENT C VALUES ARE PUT IN SAVE1 ON RETURN.       PDECOL 807
C FOR ANY OTHER VALUE OF INDEX, CONTROL RETURNS TO THE INTEGRATOR         PDECOL 808
C UNLESS TOUT HAS BEEN REACHED.  THEN INTERPOLATED VALUES OF C ARE        PDECOL 809
C COMPUTED AND STORED IN SAVE1 ON RETURN.                                 PDECOL 810
C IF INTERPOLATION IS NOT DESIRED, THE CALL TO INTERP SHOULD BE           PDECOL 811
C REMOVED AND CONTROL TRANSFERRED TO STATEMENT 340 INSTEAD OF 360.        PDECOL 812
C-----------------------------------------------------------------------  PDECOL 813
      D = 0.                                                              PDECOL 814
      DO 130 I = 1,NEQN                                                   PDECOL 815
        I1 = I - 1                                                        PDECOL 816
        AYI = ABS(WORK(IW10+I1))                                          PDECOL 817
      WORK(IW4+I1)=AMAX1(SREC,AYI)                                        TC2      1
  130   D = D + (AYI/WORK(IW4+I1))**2                                     PDECOL 819
      D = D*(UROUND/EPS)**2                                               PDECOL 820
      IF (D .GT.  FLOAT(NEQN)) GO TO 240                                  PDECOL 821
      IF (INDEX .EQ. 3) GO TO 340                                         PDECOL 822
      IF (INDEX .EQ. 2) GO TO 150                                         PDECOL 823
  140 IF ((T-TOUT)*DTC .LT. 0.) GO TO 100                                 PDECOL 824
      CALL INTERP(TOUT,WORK(IW10),NEQN,WORK(IW6))                         PDECOL 825
      GO TO 360                                                           PDECOL 826
C                                                                         PDECOL 827
  150 IF (((T+DTC)-TOUT)*DTC .LE. 0.) GO TO 100                           PDECOL 828
      IF (ABS(T-TOUT) .LE. 100.*UROUND*DTMX) GO TO 340                    PDECOL 829
      IF ((T-TOUT)*DTC .GE. 0.) GO TO 340                                 PDECOL 830
      DTC = (TOUT - T)*(1. - 4.*UROUND)                                   PDECOL 831
      JSTART = -1                                                         PDECOL 832
      GO TO 100                                                           PDECOL 833
C-----------------------------------------------------------------------  PDECOL 834
C ON AN ERROR RETURN FROM INTEGRATOR, AN IMMEDIATE RETURN OCCURS IF       PDECOL 835
C KFLAG = -2 OR -4, AND RECOVERY ATTEMPTS ARE MADE OTHERWISE.             PDECOL 836
C TO RECOVER, DT AND DTMN ARE REDUCED BY A FACTOR OF .1 UP TO 10          PDECOL 837
C TIMES BEFORE GIVING UP.                                                 PDECOL 838
C-----------------------------------------------------------------------  PDECOL 839
  160 WRITE (LOUT,170) T                                                  PDECOL 840
  170 FORMAT(//35H KFLAG = -1 FROM INTEGRATOR AT T = ,E16.8/              PDECOL 841
     *   40H  ERROR TEST FAILED WITH ABS(DT) = DTMIN/)                    PDECOL 842
  180 IF (NHCUT .EQ. 10) GO TO 200                                        PDECOL 843
      NHCUT = NHCUT + 1                                                   PDECOL 844
      DTMN = .1*DTMN                                                      PDECOL 845
      DTC = .1*DTC                                                        PDECOL 846
      WRITE (LOUT,190) DTC                                                PDECOL 847
  190 FORMAT(25H  DT HAS BEEN REDUCED TO ,E16.8,                          PDECOL 848
     *   26H  AND STEP WILL BE RETRIED//)                                 PDECOL 849
      JSTART = -1                                                         PDECOL 850
```

```
      GO TO 100
C
  200 WRITE (LOUT,210)
  210 FORMAT(//44H PROBLEM APPEARS UNSOLVABLE WITH GIVEN INPUT//)
      GO TO 340
C
  220 WRITE (LOUT,230) T,DTC
  230 FORMAT(//35H KFLAG = -2 FROM INTEGRATOR AT T = ,E16.8,6H  DT =,
     *  E16.8/52H  THE REQUESTED ERROR IS SMALLER THAN CAN BE HANDLED//)
      GO TO 340
C
  240 WRITE (LOUT,250) T
  250 FORMAT(//37H INTEGRATION HALTED BY DRIVER AT T = ,E16.8/
     *   56H  EPS TOO SMALL TO BE ATTAINED FOR THE MACHINE PRECISION/)
      KFLAG = -2
      GO TO 340
C
  260 WRITE (LOUT,270) T
  270 FORMAT(//35H KFLAG = -3 FROM INTEGRATOR AT T = ,E16.8/
     *   45H  CORRECTOR CONVERGENCE COULD NOT BE ACHIEVED/)
      GO TO 180
C
  280 WRITE (LOUT,290)
  290 FORMAT(//28H SINGULAR MATRIX ENCOUNTERED,
     *           35H PROBABLY DUE TO STORAGE OVERWRITES//)
      KFLAG = -4
      GO TO 340
C
  300 WRITE(LOUT,310) T,TOUT,DTC
  310 FORMAT(//45H INDEX = -1 ON INPUT WITH (T-TOUT)*DT .GE. 0./
     *   4H T =,E16.8,9H   TOUT =,E16.8,6H   DTC =,E16.8/
     *   44H INTERPOLATION WAS DONE AS ON NORMAL RETURN./
     *   41H DESIRED PARAMETER CHANGES WERE NOT MADE.)
      CALL INTERP(TOUT,WORK(IW10),NEQN,WORK(IW6))
      INDEX = -5
      RETURN
C
  320 WRITE(LOUT,330) IERID
  330 FORMAT(//24H ILLEGAL INPUT...INDEX= ,I3//)
      INDEX = IERID
      RETURN
C
  335 WRITE(LOUT,336) IWSTOR,IWSAVE,IISTOR,IISAVE
  336 FORMAT(//21H INSUFFICIENT STORAGE/24H  WORK MUST BE OF LENGTH,
     * I10,5X,12HYOU PROVIDED,I10/24H IWORK MUST BE OF LENGTH,I10,5X,
     * 12HYOU PROVIDED,I10//)
      INDEX = IERID
      RETURN
C
  340 TOUT = T
      DO 350 I = 1,NEQN
        I1 = I - 1
  350   WORK(IW6+I1) = WORK(IW10+I1)
  360 INDEX = KFLAG
      TOUTP = TOUT
      DT = DTUSED
      IF (KFLAG .NE. 0) DT = DTC
```

```
      RETURN                                                    PDECOL     908
      END                                                       PDECOL     909
```

```
CARD NR. SEVERITY  DETAILS    DIAGNOSIS OF PROBLEM

   687      I      IWORK      ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
```

```
      SUBROUTINE VALUES(X,USOL,SCTCH,NDIM1,NDIM2,NPTS,NDERV,WORK)        VALUES     2
C-----------------------------------------------------------------------  VALUES     3
C SUBROUTINE VALUES COMPUTES THE SOLUTION U AND THE FIRST NDERV          VALUES     4
C DERIVATIVES OF U AT THE NPTS POINTS X AND AT TIME TOUT AND RETURNS     VALUES     5
C THEM IN THE ARRAY USOL.  THIS ROUTINE MUST BE USED TO OBTAIN           VALUES     6
C SOLUTION VALUES SINCE PDECOL DOES NOT RETURN ANY SOLUTION VALUES       VALUES     7
C TO THE USER.  SEE PDECOL.                                              VALUES     8
C                                                                        VALUES     9
C THE CALLING PARAMETERS ARE...                                          VALUES    10
C   X      =  AN ARBITRARY VECTOR OF SPATIAL POINTS OF LENGTH NPTS AT    VALUES    11
C             WHICH THE SOLUTION AND THE FIRST NDERV DERIVATIVE VALUES   VALUES    12
C             ARE TO BE CALCULATED.  IF X .LT. XLEFT ( X .GT. XRIGHT )   VALUES    13
C             THEN THE PIECEWISE POLYNOMIAL OVER THE LEFTMOST ( RIGHT-   VALUES    14
C             MOST ) INTERVAL IS EVALUATED TO CALCULATE THE SOLUTION     VALUES    15
C             VALUES AT THIS UNUSUAL VALUE OF  X.  SEE PDECOL.           VALUES    16
C                                                                        VALUES    17
C   USOL  =  AN ARRAY WHICH CONTAINS THE SOLUTION AND THE FIRST          VALUES    18
C             NDERV DERIVATIVES OF THE SOLUTION AT ALL THE POINTS IN     VALUES    19
C             THE INPUT VECTOR X.  IN PARTICULAR, USOL(J,I,K) CONTAINS   VALUES    20
C             THE VALUE OF THE (K-1)-ST DERIVATIVE OF THE J-TH PDE       VALUES    21
C             COMPONENT AT THE I-TH POINT OF THE X VECTOR FOR            VALUES    22
C             J = 1 TO NPDE, I = 1 TO NPTS, AND K = 1 TO NDERV+1.        VALUES    23
C                                                                        VALUES    24
C   SCTCH =  A USER SUPPLIED WORKING STORAGE ARRAY OF LENGTH AT LEAST    VALUES    25
C             KORD*(NDERV+1).  SEE BELOW AND PDECOL FOR DEFINITIONS OF   VALUES    26
C             THESE PARAMETERS.                                          VALUES    27
C                                                                        VALUES    28
C   NDIM1 =  THE FIRST DIMENSION OF THE OUTPUT ARRAY USOL IN THE CALLING VALUES    29
C             PROGRAM.  NDIM1 MUST BE .GE. NPDE.                         VALUES    30
C                                                                        VALUES    31
C   NDIM2 =  THE SECOND DIMENSION OF THE OUTPUT ARRAY USOL IN THE        VALUES    32
C             CALLING PROGRAM.  NDIM2 MUST BE .GE. NPTS.                 VALUES    33
C                                                                        VALUES    34
C   NPTS  =  THE NUMBER OF POINTS IN THE X VECTOR.                       VALUES    35
C                                                                        VALUES    36
C   NDERV =  THE NUMBER OF DERIVATIVE VALUES OF THE SOLUTION THAT ARE    VALUES    37
C             TO BE CALCULATED.  NDERV SHOULD BE LESS THAN KORD SINCE    VALUES    38
C             THE KORD-TH DERIVATIVE OF A POLYNOMIAL OF DEGREE KORD-1    VALUES    39
C             IS EQUAL TO ZERO.  SEE PDECOL.                             VALUES    40
C                                                                        VALUES    41
C   WORK  =  THE USERS WORKING STORAGE ARRAY WHICH IS USED IN THIS CASE  VALUES    42
C             TO PROVIDE THE CURRENT BASIS FUNCTION COEFFICIENTS AND THE VALUES    43
C             PIECEWISE POLYNOMIAL BREAKPOINT SEQUENCE.                  VALUES    44
C                                                                        VALUES    45
C PACKAGE ROUTINES CALLED..  BSPLVD,INTERV                               VALUES    46
C USER ROUTINES CALLED..     NONE                                        VALUES    47
C CALLED BY..                USERS MAIN PROGRAM                          VALUES    48
C FORTRAN FUNCTIONS USED..   NONE                                        VALUES    49
C                                                                        VALUES    50
C-----------------------------------------------------------------------  VALUES    51
      DIMENSION USOL(NDIM1,NDIM2,NDERV),X(NPTS),SCTCH(1),WORK(1)         VALUES    52
      COMMON /SIZES/ NINT,KORD,NCC,NPDE,NCPTS,NEQN,IQUAD                 VALUES    53
      COMMON /ISTART/ IW1,IW2,IW3,IW4,IW5,IW6,IDUM(12)                   VALUES    54
      DATA ILEFT/0/, MFLAG/0/                                            VALUES    55
      NDERV1 = NDERV + 1                                                 VALUES    56
      DO 20 IPTS=1,NPTS                                                  VALUES    57
        CALL INTERV(WORK(IW2),NCPTS,X(IPTS),ILEFT,MFLAG)                 VALUES    58
```

```
      CALL BSPLVD(WORK(IW2),KORD,X(IPTS),ILEFT,SCTCH,NDERV1)           VALUES     59
      IK = ILEFT - KORD                                                 VALUES     60
      DO 10 M=1,NDERV1                                                  VALUES     61
        I1 = (M-1)*KORD                                                 VALUES     62
        DO 10 K=1,NPDE                                                  VALUES     63
          USOL(K,IPTS,M) = 0.                                           VALUES     64
          DO 10 I=1,KORD                                                VALUES     65
            I2 = (I+IK-1)*NPDE + IW6 - 1                                VALUES     66
            USOL(K,IPTS,M) = USOL(K,IPTS,M) + WORK(I2+K) * SCTCH(I+I1) VALUES     67
   10   CONTINUE                                                        VALUES     68
   20 CONTINUE                                                          VALUES     69
      RETURN                                                            VALUES     70
      END                                                               VALUES     71
```

```
      BLOCK DATA                                                        BLOCK       2
C---------------------------------------------------------------------- BLOCK       3
C IN THE FOLLOWING DATA STATEMENT, SET..                                BLOCK       4
C    LOUT   = THE LOGICAL UNIT NUMBER FOR THE OUTPUT OF MESSAGES DURING BLOCK       5
C             THE INTEGRATION.                                          BLOCK       6
C    NOGAUS = SET .EQ. 1 IF THE GAUSS-LEGENDRE COLLOCATION POINTS ARE   BLOCK       7
C             NOT DESIRED WHEN NCC = 2 (SEE PDECOL AND COLPNT).         BLOCK       8
C    MAXDER = SET .EQ. 5.  ITS VALUE REPRESENTS THE MAXIMUM ORDER OF    BLOCK       9
C             THE TIME INTEGRATION ALLOWED.  ITS VALUE AFFECTS THE STOR- BLOCK     10
C             AGE REQUIRED IN /HISTRY/ AND MAY BE CHANGED IF DESIRED    BLOCK      11
C             (SEE COSET FOR RESTRICTIONS).                             BLOCK      12
C    UROUND = THE UNIT ROUNDOFF OF THE MACHINE, I.E. THE SMALLEST       BLOCK      13
C             POSITIVE U SUCH THAT 1. + U .NE. 1. ON THE MACHINE.       BLOCK      14
C---------------------------------------------------------------------- BLOCK      15
      COMMON /GEAR1/ DUM(5),UROUND,IDUM(4)                              BLOCK      16
      COMMON /OPTION/ NOGAUS,MAXDER                                     BLOCK      17
      COMMON /IOUNIT/ LOUT                                              BLOCK      18
      DATA LOUT,NOGAUS,MAXDER,UROUND/3,0,5,7.1E-15/                     BLOCK      19
      END                                                               BLOCK      20
```

```
      SUBROUTINE INITAL(K,A,RHS,X,XT,XC,PW,IPIV,ILEFT)                  INITAL     2
C-----------------------------------------------------------------------INITAL     3
C INITAL IS CALLED ONLY ONCE BY PDECOL TO PERFORM INITIALIZATION TASKS.  INITAL     4
C THESE TASKS INCLUDE - 1) DEFINING THE PIECEWISE POLYNOMIAL SPACE      INITAL     5
C BREAKPOINT SEQUENCE, 2) CALLING THE SUBROUTINE COLPNT TO DEFINE THE   INITAL     6
C REQUIRED COLLOCATION POINTS, 3) DEFING THE PIECEWISE POLYNOMIAL SPACE INITAL     7
C BASIS FUNCTION VALUES (PLUS FIRST AND SECOND DERIVATIVE VALUES) AT    INITAL     8
C THE COLLOCATION POINTS, AND 4) DEFINING THE INITIAL BASIS FUNCTION    INITAL     9
C COEFFICIENTS WHICH DETERMINE THE PIECEWISE POLYNOMIAL WHICH           INITAL    10
C INTERPOLATES THE USER SUPPLIED (UINIT) INITIAL CONDITION FUNCTION(S)  INITAL    11
C AT THE COLLOCATION POINTS.                                            INITAL    12
C                                                                       INITAL    13
C K      = ORDER OF PIECEWISE POLYNOMIAL SPACE.                         INITAL    14
C A      = BASIS FUNCTION VALUES GENERATED BY INITAL.                   INITAL    15
C RHS    = TEMPORARY STORAGE USED TO RETURN INITIAL CONDITION COEFFICIENT INITAL  16
C          VALUES.                                                      INITAL    17
C X      = USER DEFINED PIECEWISE POLYNOMIAL BREAKPOINTS.               INITAL    18
C XT     = PIECEWISE POLYNOMIAL BREAKPOINT SEQUENCE GENERATED BY INITAL. INITAL   19
C XC     = COLLOCATION POINTS GENERATED BY INITAL.                      INITAL    20
C PW     = STORAGE FOR BAND MATRIX USED TO GENERATE INITIAL             INITAL    21
C          COEFFICIENT VALUES.                                          INITAL    22
C IPIV   = PIVOT INFORMATION FOR LINEAR EQUATION SOLVER DECB-SOLB.      INITAL    23
C ILEFT  = POINTERS TO BREAKPOINT SEQUENCE GENERATED BY INITAL.          INITAL    24
C                                                                       INITAL    25
C PACKAGE ROUTINES CALLED..  BSPLVD,COLPNT,DECB,INTERV,SOLB             INITAL    26
C USER ROUTINES CALLED..     UINIT                                      INITAL    27
C CALLED BY..                PDECOL                                     INITAL    28
C FORTRAN FUNCTIONS USED..   MAX0,MIN0                                  INITAL    29
C-----------------------------------------------------------------------INITAL    30
      DIMENSION A(K,3,1),RHS(1),X(1),XT(1),XC(1),PW(1),IPIV(1),ILEFT(1) INITAL    31
      COMMON /SIZES/ NINT,KORD,NCC,NPDE,NCPTS,NEQN,IER                  INITAL    32
      COMMON /GEAR9/ EPSJ,R0,ML,MU,IDUM(3),NOW                          INITAL    33
      MFLAG = -2                                                        INITAL    34
      IER = 0                                                           INITAL    35
C-----------------------------------------------------------------------INITAL    36
C SET UP THE PIECEWISE POLYNOMIAL SPACE BREAKPOINT SEQUENCE.            INITAL    37
C-----------------------------------------------------------------------INITAL    38
      KRPT = KORD - NCC                                                 INITAL    39
      DO 10 I=1,KORD                                                    INITAL    40
        XT(NCPTS+I) = X(NINT+1)                                         INITAL    41
   10   XT(I) = X(1)                                                    INITAL    42
      DO 20 I=2,NINT                                                    INITAL    43
        I1 = (I-2)*KRPT + KORD                                          INITAL    44
        DO 20 J=1,KRPT                                                  INITAL    45
   20     XT(I1+J) = X(I)                                               INITAL    46
C-----------------------------------------------------------------------INITAL    47
C SET UP COLLOCATION POINTS ARRAY XC.                                   INITAL    48
C-----------------------------------------------------------------------INITAL    49
      CALL COLPNT(X, XC, XT)                                            INITAL    50
C-----------------------------------------------------------------------INITAL    51
C GENERATE THE ILEFT ARRAY.  STORE THE BASIS FUNCTION VALUES IN THE     INITAL    52
C ARRAY A.  THE ARRAY A IS DIMENSIONED A(KORD,3,NCPTS) AND A(K,J,I)     INITAL    53
C CONTAINS THE VALUE OF THE (J-1)-ST DERIVATIVE (J = 1,2,3) OF THE K-TH INITAL    54
C NONZERO BASIS FUNCTION (K = 1, ... ,KORD) AT THE I-TH COLLOCATION     INITAL    55
C POINT (I = 1, ... ,NCPTS).  SET UP RHS FOR INTERPOLATING THE INITIAL  INITAL    56
C CONDITIONS AT THE COLLOCATION POINTS.  SET THE INTERPOLATION MATRIX   INITAL    57
C INTO THE BANDED MATRIX PW.                                            INITAL    58
```

```
C-----------------------------------------------------------------------
      DO 30 I=1,NOW
   30   PW(I) = 0.
      DO 40 I=1,NCPTS
        CALL INTERV(XT,NCPTS,XC(I),ILEFT(I),MFLAG)
        CALL BSPLVD(XT,KORD,XC(I),ILEFT(I),A(1,1,I),3)
        I1 = NPDE * (I-1)
        CALL UINIT(XC(I),RHS(I1+1),NPDE)
        ICOL = ILEFT(I) - I - 1
        JL = MAX0(1,I+2-NCPTS)
        JU = MIN0(KORD,KORD+I-2)
        DO 40 J=JL,JU
          J1 = I1 + NEQN * (NPDE * (ICOL + J) - 1)
          DO 40 JJ=1,NPDE
   40       PW(JJ+J1) = A(J,1,I)
C-----------------------------------------------------------------------
C LU DECOMPOSE THE MATRIX PW.
C-----------------------------------------------------------------------
      CALL DECB (NEQN,NEQN,ML,MU,PW,IPIV,IER)
      IF ( IER .NE. 0 ) RETURN
C-----------------------------------------------------------------------
C SOLVE THE LINEAR SYSTEM   PW*Z = RHS.  THIS GIVES THE BASIS FUNCTION
C COEFFICIENTS FOR THE INITIAL CONDITIONS.
C-----------------------------------------------------------------------
      CALL SOLB (NEQN,NEQN,ML,MU,PW,RHS,IPIV)
      RETURN
      END
```

```
      SUBROUTINE COLPNT(X, XC, XT)
C-----------------------------------------------------------------------
C COLPNT IS CALLED ONLY ONCE BY INITAL TO DEFINE THE REQUIRED COLLOCA-
C TION POINTS WHICH ARE TO BE USED WITH THE USER SELECTED PIECEWISE
C POLYNOMIAL SPACE.  THE COLLOCATION POINTS ARE CHOSEN SUCH THAT THEY
C ARE EITHER THE POINTS AT WHICH THE PIECEWISE POLYNOMIAL SPACE BASIS
C FUNCTIONS ATTAIN THEIR UNIQUE MAXIMUM VALUES, OR, THE GAUSS-LEGENDRE
C QUADRATURE POINTS WITHIN EACH PIECEWISE POLYNOMIAL SPACE SUBINTERVAL,
C DEPENDING UPON THE SPACE BEING USED AND THE DESIRE OF THE USER.
C
C X  = USER DEFINED PIECEWISE POLYNOMIAL BREAKPOINTS.
C XC = COLLOCATION POINTS DEFINED BY COLPNT.
C XT = PIECEWISE POLYNOMIAL BREAKPOINT SEQUENCE.
C
C PACKAGE ROUTINES CALLED..  BSPLVD,INTERV
C USER ROUTINES CALLED..     NONE
C CALLED BY..                INITAL
C FORTRAN FUNCTIONS USED..   NONE
C-----------------------------------------------------------------------
      DIMENSION RHO(40),X(1),XC(1),XT(1)
      COMMON /SIZES/ NINT,KORD,NCC,NPDE,NCPTS,NEQN,IQUAD
      COMMON /OPTION/ NOGAUS,MAXDER
      DATA ILEFT/0/
C-----------------------------------------------------------------------
C IF THE VARIABLE NOGAUS IN THE COMMON BLOCK /OPTION/ IS SET .EQ. 1,
C THE USE OF THE GAUSS-LEGENDRE POINTS IS PROHIBITED FOR ALL CASES.
C NOGAUS IS CURRENTLY SET .EQ. 0 BY A DATA STATEMENT IN THE BLOCK DATA.
C THE USER MAY CHANGE THIS AS DESIRED.
C-----------------------------------------------------------------------
      IF ( NCC .NE. 2  .OR. NOGAUS .EQ. 1 ) GO TO 200
C-----------------------------------------------------------------------
C COMPUTE THE COLLOCATION POINTS TO BE AT THE GAUSS-LEGENDRE POINTS IN
C EACH PIECEWISE POLYNOMIAL SPACE SUBINTERVAL.  THE ARRAY RHO IS SET TO
C CONTAIN THE GAUSS-LEGENDRE POINTS FOR THE STANDARD INTERVAL (-1,1).
C-----------------------------------------------------------------------
      IPTS = KORD - 2
      GO TO (10,20,30,40,50,60,70,80,90,100,110,120,130,140,150,160,170,
     *   180),IPTS
   10 RHO(1) = 0.
      GO TO 190
   20 RHO(2) = .57735026918962E-00
      RHO(1) = - RHO(2)
      GO TO 190
   30 RHO(3) = .77459666924148E-00
      RHO(1) = - RHO(3)
      RHO(2) = 0.
      GO TO 190
   40 RHO(3) = .33998104358486E-00
      RHO(2) = - RHO(3)
      RHO(4) = .86113631159405E-00
      RHO(1) = - RHO(4)
      GO TO 190
   50 RHO(4) = .53846931010568E-00
      RHO(2) = - RHO(4)
      RHO(5) = .90617984593866E-00
      RHO(1) = - RHO(5)
      RHO(3) = 0.
```

```
      GO TO 190                                                  COLPNT     59
   60 RHO(4) = .238619186083197E-00                              COLPNT     60
      RHO(3) = - RHO(4)                                          COLPNT     61
      RHO(5) = .661209386466265E-00                              COLPNT     62
      RHO(2) = - RHO(5)                                          COLPNT     63
      RHO(6) = .932469514203152E-00                              COLPNT     64
      RHO(1) = - RHO(6)                                          COLPNT     65
      GO TO 190                                                  COLPNT     66
   70 RHO(5) = .405845151377397E-00                              COLPNT     67
      RHO(3) = - RHO(5)                                          COLPNT     68
      RHO(6) = .741531185599394E-00                              COLPNT     69
      RHO(2) = - RHO(6)                                          COLPNT     70
      RHO(7) = .949107912342759E-00                              COLPNT     71
      RHO(1) = - RHO(7)                                          COLPNT     72
      RHO(4) = 0.                                                COLPNT     73
      GO TO 190                                                  COLPNT     74
   80 RHO(5) = .183434642495650E-00                              COLPNT     75
      RHO(4) = - RHO(5)                                          COLPNT     76
      RHO(6) = .525532409916329E-00                              COLPNT     77
      RHO(3) = - RHO(6)                                          COLPNT     78
      RHO(7) = .796666477413627E-00                              COLPNT     79
      RHO(2) = - RHO(7)                                          COLPNT     80
      RHO(8) = .960289856497536E-00                              COLPNT     81
      RHO(1) = - RHO(8)                                          COLPNT     82
      GO TO 190                                                  COLPNT     83
   90 RHO( 5) = .0                                               COLPNT     84
      RHO( 6) = .324253423403809E-00                             COLPNT     85
      RHO( 7) = .613371432700590E-00                             COLPNT     86
      RHO( 8) = .836031107326636E-00                             COLPNT     87
      RHO( 9) = .968160239507626E-00                             COLPNT     88
      DO 95 I=1,4                                                COLPNT     89
   95   RHO(I) = -RHO(10-I)                                      COLPNT     90
      GO TO 190                                                  COLPNT     91
  100 RHO( 6) = .148874338981631E-00                             COLPNT     92
      RHO( 7) = .433395394129247E-00                             COLPNT     93
      RHO( 8) = .679409568299024E-00                             COLPNT     94
      RHO( 9) = .865063366688984E-00                             COLPNT     95
      RHO(10) = .973906528517172E-00                             COLPNT     96
      DO 105 I=1,5                                               COLPNT     97
  105   RHO(I) = -RHO(11-I)                                      COLPNT     98
      GO TO 190                                                  COLPNT     99
  110 RHO( 6) = .0                                               COLPNT    100
      RHO( 7) = .269543155952345E-00                             COLPNT    101
      RHO( 8) = .519096129206812E-00                             COLPNT    102
      RHO( 9) = .730152005574049E-00                             COLPNT    103
      RHO(10) = .887062599768095E-00                             COLPNT    104
      RHO(11) = .978228658146057E-00                             COLPNT    105
      DO 115 I=1,5                                               COLPNT    106
  115   RHO(I) = -RHO(12-I)                                      COLPNT    107
      GO TO 190                                                  COLPNT    108
  120 RHO( 7) = .125233408511469E-00                             COLPNT    109
      RHO( 8) = .367831498998180E-00                             COLPNT    110
      RHO( 9) = .587317954286617E-00                             COLPNT    111
      RHO(10) = .769902674194305E-00                             COLPNT    112
      RHO(11) = .904117256370475E-00                             COLPNT    113
      RHO(12) = .981560634246719E-00                             COLPNT    114
      DO 125 I=1,6                                               COLPNT    115
```

```
  125   RHO(I) = -RHO(13-I)                                COLPNT   116
        GO TO 190                                          COLPNT   117
  130 RHO( 7) = .0                                         COLPNT   118
      RHO( 8) = .23045831595551356E-00                     COLPNT   119
      RHO( 9) = .44849275103644 7E-00                      COLPNT   120
      RHO(10) = .64234933944034 0E-00                      COLPNT   121
      RHO(11) = .80157809073331 0E-00                      COLPNT   122
      RHO(12) = .91759839922297 6E-00                      COLPNT   123
      RHO(13) = .98418305471858 8E-00                      COLPNT   124
      DO 135 I=1,6                                         COLPNT   125
  135   RHO(I) = -RHO(14-I)                                COLPNT   126
      GO TO 190                                            COLPNT   127
  140 RHO( 8) = .10805494870734 4E-00                      COLPNT   128
      RHO( 9) = .31911236892789 0E-00                      COLPNT   129
      RHO(10) = .51524863635815 4E-00                      COLPNT   130
      RHO(11) = .68729290481168 5E-00                      COLPNT   131
      RHO(12) = .82720131506976 5E-00                      COLPNT   132
      RHO(13) = .92843488366357 4E-00                      COLPNT   133
      RHO(14) = .98628380869681 2E-00                      COLPNT   134
      DO 145 I=1,7                                         COLPNT   135
  145   RHO(I) = -RHO(15-I)                                COLPNT   136
      GO TO 190                                            COLPNT   137
  150 RHO( 8) = .0                                         COLPNT   138
      RHO( 9) = .20119409399743 5E-00                      COLPNT   139
      RHO(10) = .39415134707756 3E-00                      COLPNT   140
      RHO(11) = .57097217260853 9E-00                      COLPNT   141
      RHO(12) = .72441773136017 0E-00                      COLPNT   142
      RHO(13) = .84820658341042 7E-00                      COLPNT   143
      RHO(14) = .93727339240070 6E-00                      COLPNT   144
      RHO(15) = .98799251802048 5E-00                      COLPNT   145
      DO 155 I = 1,7                                       COLPNT   146
  155   RHO(I) = -RHO(16-I)                                COLPNT   147
      GO TO 190                                            COLPNT   148
  160 RHO( 9) = .95012509837637 4E-01                      COLPNT   149
      RHO(10) = .28160355077925 9E-00                      COLPNT   150
      RHO(11) = .45801677765722 7E-00                      COLPNT   151
      RHO(12) = .61787624440264 4E-00                      COLPNT   152
      RHO(13) = .75540440835500 3E-00                      COLPNT   153
      RHO(14) = .86563120238783 2E-00                      COLPNT   154
      RHO(15) = .94457502307323 3E-00                      COLPNT   155
      RHO(16) = .98940093499165 0E-00                      COLPNT   156
      DO 165 I=1,8                                         COLPNT   157
  165   RHO(I) = -RHO(17-I)                                COLPNT   158
      GO TO 190                                            COLPNT   159
  170 RHO( 9) = .0                                         COLPNT   160
      RHO(10) = .17848418149584 6E-00                      COLPNT   161
      RHO(11) = .35123176345387 6E-00                      COLPNT   162
      RHO(12) = .51269053708647 7E-00                      COLPNT   163
      RHO(13) = .65767115921669 1E-00                      COLPNT   164
      RHO(14) = .78151400389680 1E-00                      COLPNT   165
      RHO(15) = .88023915372698 6E-00                      COLPNT   166
      RHO(16) = .95067552176876 6E-00                      COLPNT   167
      RHO(17) = .99057547531441 7E-00                      COLPNT   168
      DO 175 I=1,8                                         COLPNT   169
  175   RHO(I) = -RHO(18-I)                                COLPNT   170
      GO TO 190                                            COLPNT   171
  180 RHO(10) = .84775013041735 3E-01                      COLPNT   172
```

202420

```
      RHO(11) = .251886225691506E-00                                    COLPNT   173
      RHO(12) = .411751161462843E-00                                    COLPNT   174
      RHO(13) = .559770831073948E-00                                    COLPNT   175
      RHO(14) = .691687043060353E-00                                    COLPNT   176
      RHO(15) = .803704958972523E-00                                    COLPNT   177
      RHO(16) = .892602466497556E-00                                    COLPNT   178
      RHO(17) = .955823949571398E-00                                    COLPNT   179
      RHO(18) = .991565168420931E-00                                    COLPNT   180
      DO 185 I=1,9                                                      COLPNT   181
  185   RHO(I) = -RHO(19-I)                                             COLPNT   182
C-----------------------------------------------------------------------COLPNT   183
C COMPUTE THE GAUSS-LEGENDRE COLLOCATION POINTS IN EACH SUBINTERVAL.    COLPNT   184
C-----------------------------------------------------------------------COLPNT   185
  190 DO 195 I=1,NINT                                                   COLPNT   186
        FAC = ( X(I+1) - X(I) ) * .5                                    COLPNT   187
        DO 195 J = 1,IPTS                                               COLPNT   188
          KNOT = IPTS * (I-1) + J + 1                                   COLPNT   189
  195     XC(KNOT) = X(I) + FAC * ( RHO(J) + 1. )                       COLPNT   190
      XC(1) = X(1)                                                      COLPNT   191
      XC(NCPTS) = X(NINT+1)                                             COLPNT   192
      RETURN                                                            COLPNT   193
C-----------------------------------------------------------------------COLPNT   194
C COMPUTE THE COLLOCATION POINTS TO BE AT THE POINTS WHERE THE BASIS    COLPNT   195
C FUNCTIONS ATTAIN THEIR MAXIMA.  A BISECTION METHOD IS USED TO FIND    COLPNT   196
C THE POINTS TO MACHINE PRECISION.  THIS PROCESS COULD BE SPEEDED UP    COLPNT   197
C BY USING A SECANT METHOD IF DESIRED.                                  COLPNT   198
C-----------------------------------------------------------------------COLPNT   199
  200 ITOP = NCPTS - 1                                                  COLPNT   200
      MFLAG = -2                                                        COLPNT   201
      XC(1) = X(1)                                                      COLPNT   202
      XC(NCPTS) = X(NINT+1)                                             COLPNT   203
      DO 240 I=2,ITOP                                                   COLPNT   204
        XOLD = 1.E+20                                                   COLPNT   205
        XL = XT(I)                                                      COLPNT   206
        XR = XT(I+KORD)                                                 COLPNT   207
  210   XNEW = .5 * (XL + XR)                                           COLPNT   208
        IF( XOLD .EQ. XNEW ) GO TO 240                                  COLPNT   209
        CALL INTERV(XT,NCPTS,XNEW,ILEFT,MFLAG)                          COLPNT   210
        CALL BSPLVD(XT,KORD,XNEW,ILEFT,RHO,2)                           COLPNT   211
        DO 220 J=1,KORD                                                 COLPNT   212
          IF( I .EQ. J + ILEFT - KORD ) GO TO 230                       COLPNT   213
  220   CONTINUE                                                        COLPNT   214
  230   XVAL = RHO(KORD+J)                                              COLPNT   215
        IF( XVAL .EQ. 0.0 ) XR = XNEW                                   COLPNT   216
        IF( XVAL .GT. 0.0 ) XL = XNEW                                   COLPNT   217
        IF( XVAL .LT. 0.0 ) XR = XNEW                                   COLPNT   218
        XOLD = XNEW                                                     COLPNT   219
        GO TO 210                                                       COLPNT   220
  240 XC(I) = XR                                                        COLPNT   221
      RETURN                                                            COLPNT   222
      END                                                               COLPNT   223
```

```
        SUBROUTINE COLPNT      76/76    OPT=1 ROUND=+-*/ TRACE          FTN 4.6+452        03/19/78  14.26.15        PAGE     36

   CARD NR. SEVERITY  DETAILS    DIAGNOSIS OF PROBLEM

        41      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
        44      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
        48      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
        50      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
        53      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
        55      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
        61      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
        63      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
        66      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
        68      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
        70      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
        76      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
        78      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
        80      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
        84      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
        85      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
        86      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
        87      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
        92      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
        93      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
        94      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
        95      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
       101      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
       102      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
       103      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
       104      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
       109      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
       110      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
       111      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
       112      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
       113      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
       119      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
       120      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
       121      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
       122      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
       123      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
       128      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
       129      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
       130      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
       131      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
       132      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
       133      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
       139      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
       140      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
       141      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
       142      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
       143      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
       144      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
       148      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
       149      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
       150      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
       151      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
       152      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
       153      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
       154      I                CONSTANT TOO LONG.  HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
155	I		CONSTANT TOO LONG. HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
161	I		CONSTANT TOO LONG. HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
162	I		CONSTANT TOO LONG. HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
163	I		CONSTANT TOO LONG. HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
164	I		CONSTANT TOO LONG. HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
165	I		CONSTANT TOO LONG. HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
166	I		CONSTANT TOO LONG. HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
167	I		CONSTANT TOO LONG. HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
171	I		CONSTANT TOO LONG. HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
173	I		CONSTANT TOO LONG. HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
174	I		CONSTANT TOO LONG. HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
175	I		CONSTANT TOO LONG. HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
176	I		CONSTANT TOO LONG. HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
177	I		CONSTANT TOO LONG. HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
178	I		CONSTANT TOO LONG. HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.
179	I		CONSTANT TOO LONG. HIGH ORDER DIGITS RETAINED, BUT SOME PRECISION LOST.

```
      SUBROUTINE BSPLVD ( XT, K, X, ILEFT, VNIKX, NDERIV )                    BSPLVD   2
C-----------------------------------------------------------------------  BSPLVD   3
C THIS SUBROUTINE IS PART OF THE B-SPLINE PACKAGE FOR THE STABLE          BSPLVD   4
C EVALUATION OF ANY B-SPLINE BASIS FUNCTION OR DERIVATIVE VALUE.          BSPLVD   5
C SEE REFERENCE BELOW.                                                    BSPLVD   6
C                                                                         BSPLVD   7
C CALCULATES THE VALUE AND THE FIRST NDERIV-1 DERIVATIVES OF ALL          BSPLVD   8
C B-SPLINES WHICH DO NOT VANISH AT X.  THE ROUTINE FILLS THE TWO-         BSPLVD   9
C DIMENSIONAL ARRAY VNIKX(J,IDERIV), J=IDERIV, ... ,K WITH NONZERO        BSPLVD  10
C VALUES OF B-SPLINES OF ORDER K+1-IDERIV, IDERIV=NDERIV, ... ,1, BY      BSPLVD  11
C REPEATED CALLS TO BSPLVN.                                               BSPLVD  12
C                                                                         BSPLVD  13
C XT     = PIECEWISE POLYNOMIAL BREAKPOINT SEQUENCE.                      BSPLVD  14
C K      = ORDER OF THE PIECEWISE POLYNOMIAL SPACE.                       BSPLVD  15
C X      = POINT AT WHICH THE B-SPLINE IS TO BE EVALUATED.                BSPLVD  16
C ILEFT  = POINTER TO THE BREAKPOINT SEQUENCE.                            BSPLVD  17
C VNIKX  = TABLE OF B-SPLINE VALUES AND DERIVATIVES.                      BSPLVD  18
C NDERIV = DETERMINES NUMBER OF DERIVATIVES TO BE GENERATED.              BSPLVD  19
C                                                                         BSPLVD  20
C REFERENCE                                                               BSPLVD  21
C                                                                         BSPLVD  22
C    DEBOOR, C., PACKAGE FOR CALCULATING WITH B-SPLINES, SIAM J.          BSPLVD  23
C      NUMER. ANAL., VOL. 14, NO. 3, JUNE 1977, PP. 441-472.              BSPLVD  24
C                                                                         BSPLVD  25
C PACKAGE ROUTINES CALLED..  BSPLVN                                       BSPLVD  26
C USER ROUTINES CALLED..     NONE                                         BSPLVD  27
C CALLED BY..                COLPNT,INITAL,VALUES                         BSPLVD  28
C FORTRAN FUNCTIONS USED..   FLOAT,MAX0                                   BSPLVD  29
C-----------------------------------------------------------------------  BSPLVD  30
      DIMENSION XT(1),VNIKX(K,NDERIV)                                     BSPLVD  31
      DIMENSION A(20,20)                                                  BSPLVD  32
      KO = K + 1 - NDERIV                                                 BSPLVD  33
      CALL BSPLVN(XT,KO,1,X,ILEFT,VNIKX(NDERIV,NDERIV))                   BSPLVD  34
      IF (NDERIV .LE. 1) GO TO 120                                        BSPLVD  35
      IDERIV = NDERIV                                                     BSPLVD  36
      DO 20 I=2,NDERIV                                                    BSPLVD  37
        IDERVM = IDERIV-1                                                 BSPLVD  38
        DO 10 J=IDERIV,K                                                  BSPLVD  39
   10     VNIKX(J-1,IDERVM) = VNIKX(J,IDERIV)                             BSPLVD  40
        IDERIV = IDERVM                                                   BSPLVD  41
        CALL BSPLVN(XT,0,2,X,ILEFT,VNIKX(IDERIV,IDERIV))                  BSPLVD  42
   20   CONTINUE                                                          BSPLVD  43
      DO 40 I=1,K                                                         BSPLVD  44
        DO 30 J=1,K                                                       BSPLVD  45
   30     A(I,J) = 0.                                                     BSPLVD  46
   40   A(I,I) = 1.                                                       BSPLVD  47
      KMD = K                                                             BSPLVD  48
      DO 110 M=2,NDERIV                                                   BSPLVD  49
        KMD = KMD - 1                                                     BSPLVD  50
        FKMD =  FLOAT(KMD)                                                BSPLVD  51
        I = ILEFT                                                         BSPLVD  52
        J = K                                                             BSPLVD  53
   50   JM1 = J-1                                                         BSPLVD  54
        IPKMD = I + KMD                                                   BSPLVD  55
        DIFF = XT(IPKMD) -XT(I)                                           BSPLVD  56
        IF (JM1 .EQ. 0) GO TO 80                                          BSPLVD  57
        IF (DIFF .EQ. 0.) GO TO 70                                        BSPLVD  58
```

```
          DO 60 L=1,J                                                 BSPLVD     59
   60       A(L,J) = (A(L,J) - A(L,J-1))/DIFF*FKMD                    BSPLVD     60
   70     J = JM1                                                     BSPLVD     61
          I = I - 1                                                   BSPLVD     62
          GO TO 50                                                    BSPLVD     63
   80     IF (DIFF .EQ. 0.) GO TO 90                                  BSPLVD     64
          A(1,1) = A(1,1)/DIFF*FKMD                                   BSPLVD     65
   90     DO 110 I=1,K                                                BSPLVD     66
            V = 0.                                                    BSPLVD     67
            JLOW = MAX0(I,M)                                          BSPLVD     68
            DO 100 J=JLOW,K                                           BSPLVD     69
  100         V = A(I,J)*VNIKX(J,M) + V                               BSPLVD     70
  110       VNIKX(I,M) = V                                            BSPLVD     71
  120 RETURN                                                          BSPLVD     72
      END                                                             BSPLVD     73
```

```
      SUBROUTINE BSPLVN ( XT, JHIGH, INDEX, X, ILEFT, VNIKX )            BSPLVN   2
C----------------------------------------------------------------------- BSPLVN   3
C THIS SUBROUTINE IS PART OF THE B-SPLINE PACKAGE FOR THE STABLE        BSPLVN   4
C EVALUATION OF ANY B-SPLINE BASIS FUNCTION OR DERIVATIVE VALUE.        BSPLVN   5
C SEE REFERENCE BELOW.                                                   BSPLVN   6
C                                                                        BSPLVN   7
C CALCULATES THE VALUE OF ALL POSSIBLY NONZERO B-SPLINES AT THE         BSPLVN   8
C POINT X OF ORDER MAX(JHIGH,(J+1)(INDEX-1)) FOR THE BREAKPOINT SEQ-    BSPLVN   9
C UENCE XT.  ASSUMING THAT XT(ILEFT) .LE. X .LE. XT(ILEFT+1), THE ROUT- BSPLVN  10
C INE RETURNS THE B-SPLINE VALUES IN THE ONE DIMENSIONAL ARRAY VNIKX.   BSPLVN  11
C                                                                        BSPLVN  12
C FOR DEFINITIONS OF CALLING ARGUMENTS SEE ABOVE AND BSPLVD.            BSPLVN  13
C                                                                        BSPLVN  14
C REFERENCE                                                              BSPLVN  15
C                                                                        BSPLVN  16
C    DEBOOR, C., PACKAGE FOR CALCULATING WITH B-SPLINES, SIAM J.         BSPLVN  17
C      NUMER. ANAL., VOL. 14, NO. 3, JUNE 1977, PP. 441-472.             BSPLVN  18
C                                                                        BSPLVN  19
C PACKAGE ROUTINES CALLED..  NONE                                        BSPLVN  20
C USER ROUTINES CALLED..     NONE                                        BSPLVN  21
C CALLED BY..                BSPLVD                                      BSPLVN  22
C FORTRAN FUNCTIONS USED..   NONE                                        BSPLVN  23
C----------------------------------------------------------------------- BSPLVN  24
      DIMENSION XT(1),VNIKX(1)                                           BSPLVN  25
      DIMENSION DELTAM(20),DELTAP(20)                                    BSPLVN  26
      DATA J/1/,DELTAM/20*0.E-00/,DELTAP/20*0.E-00/                      BSPLVN  27
      GO TO (10,20),INDEX                                                BSPLVN  28
   10 J = 1                                                              BSPLVN  29
      VNIKX(1) = 1.                                                      BSPLVN  30
      IF (J .GE. JHIGH) GO TO 40                                         BSPLVN  31
   20 IPJ = ILEFT+J                                                      BSPLVN  32
      DELTAP(J) = XT(IPJ) - X                                            BSPLVN  33
      IMJP1 = ILEFT-J+1                                                  BSPLVN  34
      DELTAM(J) = X - XT(IMJP1)                                          BSPLVN  35
      VMPREV = 0.                                                        BSPLVN  36
      JP1 = J+1                                                          BSPLVN  37
      DO 30 L=1,J                                                        BSPLVN  38
        JP1ML = JP1-L                                                    BSPLVN  39
        VM = VNIKX(L)/(DELTAP(L) + DELTAM(JP1ML))                        BSPLVN  40
        VNIKX(L) = VM*DELTAP(L) + VMPREV                                 BSPLVN  41
   30   VMPREV = VM*DELTAM(JP1ML)                                        BSPLVN  42
      VNIKX(JP1) = VMPREV                                                BSPLVN  43
      J = JP1                                                            BSPLVN  44
      IF (J .LT. JHIGH) GO TO 20                                         BSPLVN  45
   40 RETURN                                                             BSPLVN  46
      END                                                                BSPLVN  47
```

CARD NR. SEVERITY DETAILS   DIAGNOSIS OF PROBLEM

27   I   AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.

```
      SUBROUTINE INTERV ( XT, LXT, X, ILEFT, MFLAG )                    INTERV    2
C---------------------------------------------------------------------- INTERV    3
C THIS SUBROUTINE IS PART OF THE B-SPLINE PACKAGE FOR THE STABLE        INTERV    4
C EVALUATION OF ANY B-SPLINE BASIS FUNCTION OR DERIVATIVE VALUE.        INTERV    5
C SEE REFERENCE BELOW.                                                  INTERV    6
C                                                                       INTERV    7
C COMPUTES LARGEST ILEFT IN (1,LXT) SUCH THAT XT(ILEFT) .LE. X.  THE    INTERV    8
C PROGRAM STARTS THE SEARCH FOR ILEFT WITH THE VALUE OF ILEFT THAT WAS  INTERV    9
C RETURNED AT THE PREVIOUS CALL (AND WAS SAVED IN THE LOCAL VARIABLE    INTERV   10
C ILO) TO MINIMIZE THE WORK IN THE COMMON CASE THAT THE VALUE OF X ON   INTERV   11
C THIS CALL IS CLOSE TO THE VALUE OF X ON THE PREVIOUS CALL.  SHOULD    INTERV   12
C THIS ASSUMPTION NOT BE VALID, THEN THE PROGRAM LOCATES ILO AND IHI    INTERV   13
C SUCH THAT XT(ILO) .LE. X .LT. XT(IHI) AND, ONCE THEY ARE FOUND USES   INTERV   14
C BISECTION TO FIND THE CORRECT VALUE FOR ILEFT. MFLAG IS AN ERROR FLAG. INTERV  15
C                                                                       INTERV   16
C FOR DEFINITIONS OF CALLING ARGUMENTS SEE ABOVE AND BSPLVD.            INTERV   17
C                                                                       INTERV   18
C REFERENCE                                                             INTERV   19
C                                                                       INTERV   20
C    DEBOOR, C., PACKAGE FOR CALCULATING WITH B-SPLINES, SIAM J.        INTERV   21
C      NUMER. ANAL., VOL. 14, NO. 3, JUNE 1977, PP. 441-472.            INTERV   22
C                                                                       INTERV   23
C PACKAGE ROUTINES CALLED..  NONE                                       INTERV   24
C USER ROUTINES CALLED..     NONE                                       INTERV   25
C CALLED BY..                COLPNT,INITAL,VALUES                       INTERV   26
C FORTRAN FUNCTIONS USED..   NONE                                       INTERV   27
C---------------------------------------------------------------------- INTERV   28
      DIMENSION XT(LXT)                                                 INTERV   29
      IF(MFLAG.EQ.-2) ILO = 1                                           INTERV   30
      IHI = ILO + 1                                                     INTERV   31
      IF (IHI .LT. LXT) GO TO 20                                        INTERV   32
      IF (X .GE. XT(LXT)) GO TO 110                                     INTERV   33
      IF (LXT .LE. 1) GO TO 90                                          INTERV   34
      ILO = LXT - 1                                                     INTERV   35
      GO TO 21                                                          INTERV   36
   20 IF (X .GE. XT(IHI)) GO TO 40                                      INTERV   37
   21 IF (X .GE. XT(ILO)) GO TO 100                                     INTERV   38
C---------------------------------------------------------------------- INTERV   39
C NOW X .LT. XT(IHI).  FIND LOWER BOUND.                                INTERV   40
C---------------------------------------------------------------------- INTERV   41
   30 ISTEP = 1                                                         INTERV   42
   31 IHI = ILO                                                         INTERV   43
      ILO = IHI - ISTEP                                                 INTERV   44
      IF (ILO .LE. 1) GO TO 35                                          INTERV   45
      IF (X .GE. XT(ILO)) GO TO 50                                      INTERV   46
      ISTEP = ISTEP*2                                                   INTERV   47
      GO TO 31                                                          INTERV   48
   35 ILO = 1                                                           INTERV   49
      IF (X .LT. XT(1)) GO TO 90                                        INTERV   50
      GO TO 50                                                          INTERV   51
C---------------------------------------------------------------------- INTERV   52
C NOW X .GE. XT(ILO).  FIND UPPER BOUND.                                INTERV   53
C---------------------------------------------------------------------- INTERV   54
   40 ISTEP = 1                                                         INTERV   55
   41 ILO = IHI                                                         INTERV   56
      IHI = ILO + ISTEP                                                 INTERV   57
      IF (IHI .GE. LXT) GO TO 45                                        INTERV   58
```

```
      IF (X .LT. XT(IHI)) GO TO 50                                      INTERV   59
      ISTEP = ISTEP*2                                                   INTERV   60
      GO TO 41                                                          INTERV   61
   45 IF (X .GE. XT(LXT)) GO TO 110                                     INTERV   62
      IHI = LXT                                                         INTERV   63
C---------------------------------------------------------------------- INTERV   64
C NOW XT(ILO) .LE. X .LT. XT(IHI).  NARROW THE INTERVAL.                INTERV   65
C---------------------------------------------------------------------- INTERV   66
   50 MIDDLE = (ILO + IHI)/2                                            INTERV   67
      IF (MIDDLE .EQ. ILO) GO TO 100                                    INTERV   68
C---------------------------------------------------------------------- INTERV   69
C NOTE..  IT IS ASSUMED THAT MIDDLE = ILO IN CASE IHI = ILO+1.          INTERV   70
C---------------------------------------------------------------------- INTERV   71
      IF (X .LT. XT(MIDDLE)) GO TO 53                                   INTERV   72
      ILO = MIDDLE                                                      INTERV   73
      GO TO 50                                                          INTERV   74
   53 IHI = MIDDLE                                                      INTERV   75
      GO TO 50                                                          INTERV   76
C---------------------------------------------------------------------- INTERV   77
C SET OUTPUT AND RETURN.                                                INTERV   78
C---------------------------------------------------------------------- INTERV   79
   90 MFLAG = -1                                                        INTERV   80
      ILEFT = 1                                                         INTERV   81
      RETURN                                                            INTERV   82
  100 MFLAG = 0                                                         INTERV   83
      ILEFT = ILO                                                       INTERV   84
      RETURN                                                            INTERV   85
  110 MFLAG = 1                                                         INTERV   86
      ILEFT = LXT                                                       INTERV   87
      RETURN                                                            INTERV   88
      END                                                               INTERV   89
```

```
      SUBROUTINE STIFIB (N0,Y,YMAX,ERROR,SAVE1,SAVE2,SAVE3,             STIFIB     2
     *                   PW,IPIV,WORK,IWORK)                            STIFIB     3
C---------------------------------------------------------------------- STIFIB     4
C STIFIB PERFORMS ONE STEP OF THE INTEGRATION OF AN INITIAL VALUE       STIFIB     5
C PROBLEM FOR A SYSTEM OF ORDINARY DIFFERENTIAL EQUATIONS OF THE FORM,  STIFIB     6
C   A(Y,T)*(DY/DT) = G(Y,T),    WHERE   Y = (Y(1),Y(2), ... ,Y(N)).     STIFIB     7
C STIFIB IS FOR USE WHEN THE MATRICES A AND DG/DY HAVE BANDED OR NEARLY STIFIB     8
C BANDED FORM.  THE DEPENDENCE OF A(Y,T) ON Y IS ASSUMED TO BE WEAK.    STIFIB     9
C                                                                       STIFIB    10
C REFERENCE                                                             STIFIB    11
C                                                                       STIFIB    12
C   HINDMARSH, A.C., PRELIMINARY DOCUMENTATION OF GEARIB.. SOLUTION     STIFIB    13
C     OF IMPLICIT SYSTEMS OF ORDINARY DIFFERENTIAL EQUATIONS WITH       STIFIB    14
C     BANDED JACOBIANS, LAWRENCE LIVERMORE LAB, UCID-30130, FEBRUARY    STIFIB    15
C     1976.                                                             STIFIB    16
C                                                                       STIFIB    17
C COMMUNICATION WITH STIFIB IS DONE WITH THE FOLLOWING VARIABLES..      STIFIB    18
C                                                                       STIFIB    19
C   Y       AN N0 BY LMAX ARRAY CONTAINING THE DEPENDENT VARIABLES      STIFIB    20
C             AND THEIR SCALED DERIVATIVES.  LMAX IS 13 FOR THE ADAMS   STIFIB    21
C             METHODS AND 6 FOR THE GEAR METHODS.  LMAX - 1 = MAXDER    STIFIB    22
C             IS THE MAXIMUM ORDER AVAILABLE.  SEE SUBROUTINE COSET.    STIFIB    23
C             Y(I,J+1) CONTAINS THE J-TH DERIVATIVE OF Y(I), SCALED BY  STIFIB    24
C             H**J/FACTORIAL(J)  (J = 0,1,...,NQ).                      STIFIB    25
C   N0      A CONSTANT INTEGER .GE. N, USED FOR DIMENSIONING PURPOSES.  STIFIB    26
C   T       THE INDEPENDENT VARIABLE, T IS UPDATED ON EACH STEP TAKEN.  STIFIB    27
C   H       THE STEP SIZE TO BE ATTEMPTED ON THE NEXT STEP.             STIFIB    28
C             H IS ALTERED BY THE ERROR CONTROL ALGORITHM DURING THE    STIFIB    29
C             PROBLEM.  H CAN BE EITHER POSITIVE OR NEGATIVE, BUT ITS   STIFIB    30
C             SIGN MUST REMAIN CONSTANT THROUGHOUT THE PROBLEM.         STIFIB    31
C   HMIN,   THE MINIMUM AND MAXIMUM ABSOLUTE VALUE OF THE STEP SIZE     STIFIB    32
C    HMAX     TO BE USED FOR THE STEP.  THESE MAY BE CHANGED AT ANY     STIFIB    33
C             TIME, BUT WILL NOT TAKE EFFECT UNTIL THE NEXT H CHANGE.   STIFIB    34
C   EPS     THE RELATIVE ERROR BOUND.  SEE DESCRIPTION IN PDECOL.       STIFIB    35
C   UROUND  THE UNIT ROUNDOFF OF THE MACHINE.                           STIFIB    36
C   N       THE NUMBER OF FIRST-ORDER DIFFERENTIAL EQUATIONS.           STIFIB    37
C   MF      THE METHOD FLAG.  SEE DESCRIPTION IN PDECOL.                STIFIB    38
C   KFLAG   A COMPLETION CODE WITH THE FOLLOWING MEANINGS..             STIFIB    39
C                        0  THE STEP WAS SUCCESFUL.                     STIFIB    40
C                       -1  THE REQUESTED ERROR COULD NOT BE ACHIEVED   STIFIB    41
C                             WITH ABS(H) = HMIN.                       STIFIB    42
C                       -2  THE REQUESTED ERROR IS SMALLER THAN CAN     STIFIB    43
C                             BE HANDLED FOR THIS PROBLEM.              STIFIB    44
C                       -3  CORRECTOR CONVERGENCE COULD NOT BE          STIFIB    45
C                             ACHIEVED FOR ABS(H) = HMIN.               STIFIB    46
C                       -4  SINGULAR A-MATRIX ENCOUNTERED.              STIFIB    47
C             ON A RETURN WITH KFLAG NEGATIVE, THE VALUES OF T AND      STIFIB    48
C             THE Y ARRAY ARE AS OF THE BEGINNING OF THE LAST           STIFIB    49
C             STEP, AND H IS THE LAST STEP SIZE ATTEMPTED.              STIFIB    50
C   JSTART  AN INTEGER USED ON INPUT AND OUTPUT.                        STIFIB    51
C             ON INPUT, IT HAS THE FOLLOWING VALUES AND MEANINGS..      STIFIB    52
C                        0  PERFORM THE FIRST STEP.                     STIFIB    53
C                    .GT.0  TAKE A NEW STEP CONTINUING FROM THE LAST.   STIFIB    54
C                    .LT.0  TAKE THE NEXT STEP WITH A NEW VALUE OF      STIFIB    55
C                             H, EPS, N, AND/OR MF.                     STIFIB    56
C             ON EXIT, JSTART IS NQ, THE CURRENT ORDER OF THE METHOD.   STIFIB    57
C   YMAX    AN ARRAY OF N ELEMENTS WITH WHICH THE ESTIMATED LOCAL       STIFIB    58
```

```
C            ERRORS IN Y ARE COMPARED.
C   ERROR    AN ARRAY OF N ELEMENTS.  ERROR(I)/TQ(2) IS THE ESTIMATED
C            ONE-STEP ERROR IN Y(I).
C   SAVE1,SAVE2,SAVE3   THREE WORKING STORAGE ARRAYS, EACH OF LENGTH N.
C   PW       A BLOCK OF LOCATIONS USED FOR THE CHORD ITERATION
C            MATRIX.  SEE DESCRIPTION IN PDECOL.
C   IPIV     AN INTEGER ARRAY OF LENGTH N FOR PIVOT INFORMATION.
C   ML,MU    THE LOWER AND UPPER HALF BANDWIDTHS, RESPECTIVELY, OF
C            THE CHORD ITERATION MATRIX.  SEE DESCRIPTION IN PDECOL.
C   WORK,IWORK   WORKING ARRAYS WHICH ARE USED TO PASS APPROPRIATE
C                ARRAYS TO OTHER SUBROUTINES.
C
C PACKAGE ROUTINES CALLED..  COSET,DIFFUN,PSETIB,RES,SOLB
C USER ROUTINES CALLED..     NONE
C CALLED BY..                PDECOL
C FORTRAN FUNCTIONS USED..   ABS,AMAX1,AMIN1,FLOAT
C-----------------------------------------------------------------------
      DIMENSION Y(NO,1),YMAX(NO),ERROR(NO),SAVE1(NO),SAVE2(NO),
     *          SAVE3(NO),PW(1),IPIV(1),WORK(1),IWORK(1)
      COMMON /SIZES/ NINT,KORD,NCC,NPDE,NCPTS,NEQN,IQUAD
      COMMON /ISTART/ IW1,IW2,IW3,IW4,IW5,IW6,IW7,IW8,IW9,IW10,IW11,
     *                IW12,IW13,IW14,IW15,IW16,IW17,IW18
      COMMON /GEAR1/ T,H,HMIN,HMAX,EPS,UROUND,N,MF,KFLAG,JSTART
      COMMON /GEAR9/ EPSJ,R0,ML,MU,MW,NM1,N0ML,N0W
      COMMON /GEAR0/ HUSED,NQUSED,NSTEP,NFE,NJE
      COMMON /OPTION/ NOGAUS,MAADER
      DIMENSION EL(13),TQ(4)
      DATA EL(2)/1./, OLDL0/1./, TQ(1)/0./, IER/0/
      KFLAG = 0
      TOLD = T
      IF (JSTART .GT. 0) GO TO 200
      IF (JSTART .NE. 0) GO TO 120
C-----------------------------------------------------------------------
C ON THE FIRST CALL, THE ORDER IS SET TO 1 AND THE INITIAL YDOT IS
C CALCULATED.  RMAX IS THE MAXIMUM RATIO BY WHICH H CAN BE INCREASED
C IN A SINGLE STEP.  IT IS INITIALLY 1.E4 TO COMPENSATE FOR THE SMALL
C INITIAL H, BUT THEN IS NORMALLY EQUAL TO 10.  IF A FAILURE
C OCCURS (IN CORRECTOR CONVERGENCE OR ERROR TEST), RMAX IS SET AT 2
C FOR THE NEXT INCREASE.
C-----------------------------------------------------------------------
      NQ = 1
      IER = 0
      CALL DIFFUN (N, T, Y, SAVE1, IER, PW, IPIV, WORK, IWORK)
      IF ( IER .NE. 0 ) GO TO 685
      DO 110 I = 1,N
  110   Y(I,2) = H*SAVE1(I)
      METH = MF/10
      MITER = MF - 10*METH
      L = 2
      IDOUB = 3
      RMAX = 1.E+04
      RC = 0.
      CRATE = 1.
      EPSOLD = EPS
      HOLD = H
      MFOLD = MF
      NOLD = N
```

```
      NSTEP = 0                                                       STIFIB  116
      NSTEPJ = 0                                                      STIFIB  117
      NFE = 0                                                         STIFIB  118
      NJE = 1                                                         STIFIB  119
      IRET = 3                                                        STIFIB  120
      GO TO 130                                                       STIFIB  121
C-------------------------------------------------------------------- STIFIB  122
C IF THE CALLER HAS CHANGED METH, COSET IS CALLED TO SET              STIFIB  123
C THE COEFFICIENTS OF THE METHOD.  IF THE CALLER HAS CHANGED          STIFIB  124
C N, EPS, OR METH, THE CONSTANTS E, EDN, EUP, AND BND MUST BE RESET.  STIFIB  125
C E IS A COMPARISON FOR ERRORS OF THE CURRENT ORDER NQ. EUP IS        STIFIB  126
C TO TEST FOR INCREASING THE ORDER, EDN FOR DECREASING THE ORDER.     STIFIB  127
C BND IS USED TO TEST FOR CONVERGENCE OF THE CORRECTOR ITERATES.      STIFIB  128
C IF THE CALLER HAS CHANGED H, Y MUST BE RESCALED.                    STIFIB  129
C IF H OR METH HAS BEEN CHANGED, IDOUB IS RESET TO L + 1 TO PREVENT   STIFIB  130
C FURTHER CHANGES IN H FOR THAT MANY STEPS.                           STIFIB  131
C-------------------------------------------------------------------- STIFIB  132
  120 IF (MF .EQ. MFOLD) GO TO 150                                    STIFIB  133
      MEO = METH                                                      STIFIB  134
      MIO = MITER                                                     STIFIB  135
      METH = MF/10                                                    STIFIB  136
      MITER = MF - 10*METH                                            STIFIB  137
      MFOLD = MF                                                      STIFIB  138
      IF (MITER .NE. MIO) IWEVAL = MITER                              STIFIB  139
      IF (METH .EQ. MEO) GO TO 150                                    STIFIB  140
      IDOUB = L + 1                                                   STIFIB  141
      IRET = 1                                                        STIFIB  142
  130 CALL COSET (METH, NQ, EL, TQ)                                   STIFIB  143
      LMAX = MAXDER + 1                                               STIFIB  144
      RC = RC*EL(1)/OLDLO                                             STIFIB  145
      OLDLO = EL(1)                                                   STIFIB  146
  140 FN =  FLOAT(N)                                                  STIFIB  147
      EDN = FN*(TQ(1)*EPS)**2                                         STIFIB  148
      E   = FN*(TQ(2)*EPS)**2                                         STIFIB  149
      EUP = FN*(TQ(3)*EPS)**2                                         STIFIB  150
      BND = FN*(TQ(4)*EPS)**2                                         STIFIB  151
      GO TO (160, 170, 200), IRET                                     STIFIB  152
  150 IF ((EPS .EQ. EPSOLD) .AND. (N .EQ. NOLD)) GO TO 160            STIFIB  153
      EPSOLD = EPS                                                    STIFIB  154
      NOLD = N                                                        STIFIB  155
      IRET = 1                                                        STIFIB  156
      GO TO 140                                                       STIFIB  157
  160 IF (H .EQ. HOLD) GO TO 200                                      STIFIB  158
      RH = H/HOLD                                                     STIFIB  159
      H = HOLD                                                        STIFIB  160
      IREDO = 3                                                       STIFIB  161
      GO TO 175                                                       STIFIB  162
  170 RH = AMAX1(RH,HMIN/ ABS(H))                                     STIFIB  163
  175 RH = AMIN1(RH,HMAX/ ABS(H),RMAX)                                STIFIB  164
      R1 = 1.                                                         STIFIB  165
      DO 180 J = 2,L                                                  STIFIB  166
        R1 = R1*RH                                                    STIFIB  167
        DO 180 I = 1,N                                                STIFIB  168
  180     Y(I,J) = Y(I,J)*R1                                          STIFIB  169
      H = H*RH                                                        STIFIB  170
      RC = RC*RH                                                      STIFIB  171
      IDOUB = L + 1                                                   STIFIB  172
```

```
      IF (IREDO .EQ. 0) GO TO 690
C-----------------------------------------------------------------------
C THIS SECTION COMPUTES THE PREDICTED VALUES BY EFFECTIVELY
C MULTIPLYING THE Y ARRAY BY THE PASCAL TRIANGLE MATRIX.
C RC IS THE RATIO OF NEW TO OLD VALUES OF THE COEFFICIENT  H*EL(1).
C WHEN RC DIFFERS FROM 1 BY MORE THAN 30 PERCENT, OR THE CALLER HAS
C CHANGED MITER, IWEVAL IS SET TO MITER TO FORCE PW TO BE UPDATED.
C IN ANY CASE, PW IS UPDATED AT LEAST EVERY 40-TH STEP.
C PW IS THE CHORD ITERATION MATRIX A - H*EL(1)*(DG/DY).
C-----------------------------------------------------------------------
  200 IF ( ABS(RC-1.) .GT. 0.3) IWEVAL = MITER
      IF (NSTEP .GE. NSTEPJ+40) IWEVAL = MITER
      T = T + H
      DO 210 J1 = 1,NQ
        DO 210 J2 = J1,NQ
          J = (NQ + J1) - J2
          DO 210 I = 1,N
  210       Y(I,J) = Y(I,J) + Y(I,J+1)
C-----------------------------------------------------------------------
C UP TO 3 CORRECTOR ITERATIONS ARE TAKEN.  A CONVERGENCE TEST IS
C MADE ON THE R.M.S. NORM OF EACH CORRECTION, USING RND, WHICH
C IS DEPENDENT ON EPS.  THE SUM OF THE CORRECTIONS IS ACCUMULATED
C IN THE VECTOR ERROR(I).  THE Y ARRAY IS NOT ALTERED IN THE CORRECTOR
C LOOP.  THE UPDATED Y VECTOR IS STORED TEMPORARILY IN SAVE1.
C THE UPDATED H*YDOT IS STORED IN SAVE2.
C-----------------------------------------------------------------------
  220 DO 230 I = 1,N
        SAVE2(I) = Y(I,2)
  230   ERROR(I) = 0.
      M = 0
      CALL RES (T, H, Y, SAVE2, SAVE3, NPDE, NCPTS, WORK(IW1), IWORK,
     *  WORK, WORK(IW14), WORK(IW15), WORK(IW16), WORK(IW3), WORK(IW9))
      NFE = NFE + 1
      IF (IWEVAL .LE. 0) GO TO 350
C-----------------------------------------------------------------------
C IF INDICATED, THE MATRIX PW IS REEVALUATED BEFORE STARTING THE
C CORRECTOR ITERATION.  IWEVAL IS SET TO 0 AS AN INDICATOR
C THAT THIS HAS BEEN DONE.  PW IS COMPUTED AND PROCESSED IN PSETIB.
C-----------------------------------------------------------------------
      IWEVAL = 0
      RC = 1.
      NJE = NJE + 1
      NSTEPJ = NSTEP
      CON = -H*EL(1)
      CALL PSETIB (Y, PW, N0, CON, MITER, IER, WORK(IW1), IWORK,
     *     WORK(IW3),WORK(IW9),SAVE2,IPIV,YMAX,WORK(IW11),WORK(IW12),
     *     WORK(IW13),WORK(IW16),WORK(IW14),WORK(IW15),WORK,NPDE)
      IF (IER .NE. 0) GO TO 420
C-----------------------------------------------------------------------
C COMPUTE THE CORRECTOR ERROR, R SUB M, AND SOLVE THE LINEAR SYSTEM
C WITH THAT AS RIGHT-HAND SIDE AND PW AS COEFFICIENT MATRIX,
C USING THE LU DECOMPOSITION OF PW.
C-----------------------------------------------------------------------
  350 CALL SOLB (N0, N, ML, MU, PW, SAVE3, IPIV)
  370 D = 0.
      DO 380 I = 1,N
        ERROR(I) = ERROR(I) + SAVE3(I)
```

```
      D = D + (SAVE3(I)/YMAX(I))**2                                  STIFIB  230
      SAVE1(I) = Y(I,1) + EL(1)*ERROR(I)                             STIFIB  231
  380 SAVE2(I) = Y(I,2) + ERROR(I)                                   STIFIB  232
C-------------------------------------------------------------------- STIFIB  233
C TEST FOR CONVERGENCE.  IF M.GT.0, AN ESTIMATE OF THE CONVERGENCE    STIFIB  234
C RATE CONSTANT IS STORED IN CRATE, AND THIS IS USED IN THE TEST.     STIFIB  235
C-------------------------------------------------------------------- STIFIB  236
  400 IF (M .NE. 0) CRATE = AMAX1(.9*CRATE,D/D1)                      STIFIB  237
      IF ((D*AMIN1(1.E0,2.*CRATE)) .LE. BND) GO TO 450                STIFIB  238
      D1 = D                                                          STIFIB  239
      M = M + 1                                                       STIFIB  240
      IF (M .EQ. 3) GO TO 410                                         STIFIB  241
      CALL RES(T, H, SAVE1, SAVE2, SAVE3, NPDE, NCPTS, WORK(IW1), IWORK, STIFIB  242
     *  WORK, WORK(IW14), WORK(IW15), WORK(IW16), WORK(IW3), WORK(IW9)) STIFIB  243
      GO TO 350                                                       STIFIB  244
C-------------------------------------------------------------------- STIFIB  245
C THE CORRECTOR ITERATION FAILED TO CONVERGE IN 3 TRIES.              STIFIB  246
C IF THE MATRIX PW IS NOT UP TO DATE, IT IS REEVALUATED FOR THE       STIFIB  247
C NEXT TRY.  OTHERWISE THE Y ARRAY IS RETRACTED TO ITS VALUES         STIFIB  248
C BEFORE PREDICTION, AND H IS REDUCED, IF POSSIBLE.  IF NOT, A        STIFIB  249
C NO-CONVERGENCE EXIT IS TAKEN.                                       STIFIB  250
C-------------------------------------------------------------------- STIFIB  251
  410 NFE = NFE + 2                                                   STIFIB  252
      IF (IWEVAL .EQ. -1) GO TO 440                                   STIFIB  253
  420 T = TOLD                                                        STIFIB  254
      RMAX = 2.                                                       STIFIB  255
      DO 430 J1 = 1,NQ                                                STIFIB  256
        DO 430 J2 = J1,NQ                                             STIFIB  257
          J = (NQ + J1) - J2                                          STIFIB  258
          DO 430 I = 1,N                                              STIFIB  259
  430       Y(I,J) = Y(I,J) - Y(I,J+1)                                STIFIB  260
      IF ( ABS(H) .LE. HMIN*1.00001) GO TO 680                        STIFIB  261
      RH = .25                                                        STIFIB  262
      IREDO = 1                                                       STIFIB  263
      GO TO 170                                                       STIFIB  264
  440 IWEVAL = MITER                                                  STIFIB  265
      GO TO 220                                                       STIFIB  266
C-------------------------------------------------------------------- STIFIB  267
C THE CORRECTOR HAS CONVERGED.  IWEVAL IS SET TO -1 TO SIGNAL         STIFIB  268
C THAT PW MAY NEED UPDATING ON SUBSEQUENT STEPS.  THE ERROR TEST      STIFIB  269
C IS MADE AND CONTROL PASSES TO STATEMENT 500 IF IT FAILS.            STIFIB  270
C-------------------------------------------------------------------- STIFIB  271
 450  IWEVAL = -1                                                     STIFIB  272
      NFE = NFE + M                                                   STIFIB  273
      D = 0.                                                          STIFIB  274
      DO 460 I = 1,N                                                  STIFIB  275
  460   D = D + (ERROR(I)/YMAX(I))**2                                 STIFIB  276
      IF (D .GT. E) GO TO 500                                         STIFIB  277
C-------------------------------------------------------------------- STIFIB  278
C AFTER A SUCCESSFUL STEP, UPDATE THE Y ARRAY.                        STIFIB  279
C CONSIDER CHANGING H IF IDOUB = 1.  OTHERWISE DECREASE IDOUB BY 1.   STIFIB  280
C IF IDOUB IS THEN 1 AND NQ .LT. MAXDER, THEN ERROR IS SAVED FOR      STIFIB  281
C USE IN A POSSIBLE ORDER INCREASE ON THE NEXT STEP.                  STIFIB  282
C IF A CHANGE IN H IS CONSIDERED, AN INCREASE OR DECREASE IN ORDER    STIFIB  283
C BY ONE IS CONSIDERED ALSO.  A CHANGE IN H IS MADE ONLY IF IT IS BY A STIFIB  284
C FACTOR OF AT LEAST 1.1.  IF NOT, IDOUB IS SET TO 10 TO PREVENT      STIFIB  285
C TESTING FOR THAT MANY STEPS.                                        STIFIB  286
```

```
      C-----------------------------------------------------------------------
            KFLAG = 0
            IREDO = 0
            NSTEP = NSTEP + 1
            HUSED = H
            NQUSED = NQ
            DO 470 J = 1,L
              DO 470 I = 1,N
        470     Y(I,J) = Y(I,J) + EL(J)*ERROR(I)
            IF (IDOUB .EQ. 1) GO TO 520
            IDOUB = IDOUB - 1
            IF (IDOUB .GT. 1) GO TO 700
            IF (L .EQ. LMAX) GO TO 700
            DO 490 I = 1,N
        490   Y(I,LMAX) = ERROR(I)
            GO TO 700
      C-----------------------------------------------------------------------
      C THE ERROR TEST FAILED.  KFLAG KEEPS TRACK OF MULTIPLE FAILURES.
      C RESTORE T AND THE Y ARRAY TO THEIR PREVIOUS VALUES, AND PREPARE
      C TO TRY THE STEP AGAIN.  COMPUTE THE OPTIMUM STEP SIZE FOR THIS OR
      C ONE LOWER ORDER.
      C-----------------------------------------------------------------------
        500 KFLAG = KFLAG - 1
            T = TOLD
            DO 510 J1 = 1,NQ
              DO 510 J2 = J1,NQ
                J = (NQ + J1) - J2
                DO 510 I = 1,N
        510       Y(I,J) = Y(I,J) - Y(I,J+1)
            RMAX = 2.
            IF ( ABS(H) .LE. HMIN*1.00001) GO TO 660
            IF (KFLAG .LE. -3) GO TO 640
            IREDO = 2
            PR3 = 1.E+20
            GO TO 540
      C-----------------------------------------------------------------------
      C REGARDLESS OF THE SUCCESS OR FAILURE OF THE STEP, FACTORS
      C PR1, PR2, AND PR3 ARE COMPUTED, BY WHICH H COULD BE DIVIDED
      C AT ORDER NQ - 1, ORDER NQ, OR ORDER NQ + 1, RESPECTIVELY.
      C IN THE CASE OF FAILURE, PR3 = 1.E20 TO AVOID AN ORDER INCREASE.
      C THE SMALLEST OF THESE IS DETERMINED AND THE NEW ORDER CHOSEN
      C ACCORDINGLY.  IF THE ORDER IS TO BE INCREASED, WE COMPUTE ONE
      C ADDITIONAL SCALED DERIVATIVE.
      C-----------------------------------------------------------------------
        520 PR3 = 1.E+20
            IF (L .EQ. LMAX) GO TO 540
            D1 = 0.
            DO 530 I = 1,N
        530   D1 = D1 + ((ERROR(I) - Y(I,LMAX))/YMAX(I))**2
            ENQ3 = .5/ FLOAT(L+1)
            PR3 = ((D1/EUP)**ENQ3)*1.4 + 1.4E-06
        540 ENQ2 = .5/ FLOAT(L)
            PR2 = ((D/E)**ENQ2)*1.2 + 1.2E-06
            PR1 = 1.E+20
            IF (NQ .EQ. 1) GO TO 560
            D = 0.
            DO 550 I = 1,N
```

```
  550   D = D + (Y(I,L)/YMAX(I))**2
        ENQ1 = .5/ FLOAT(NQ)
        PR1 = ((D/EDN)**ENQ1)*1.3 + 1.3E-06
  560 IF (PR2 .LE. PR3) GO TO 570
        IF (PR3 .LT. PR1) GO TO 590
        GO TO 580
  570 IF (PR2 .GT. PR1) GO TO 580
        NEWQ = NQ
        RH = 1./PR2
        GO TO 620
  580 NEWQ = NQ - 1
        RH = 1./PR1
        GO TO 620
  590 NEWQ = L
        RH = 1./PR3
        IF (RH .LT. 1.1) GO TO 610
        DO 600 I = 1,N
  600   Y(I,NEWQ+1) = ERROR(I)*EL(L)/ FLOAT(L)
        GO TO 630
  610 IDOUB = 10
        GO TO 700
  620 IF ((KFLAG .EQ. 0) .AND. (RH .LT. 1.1)) GO TO 610
C-----------------------------------------------------------------------
C IF THERE IS A CHANGE OF ORDER, RESET NQ, L, AND THE COEFFICIENTS.
C IN ANY CASE H IS RESET ACCORDING TO RH AND THE Y ARRAY IS RESCALED.
C THEN EXIT FROM 690 IF THE STEP WAS OK, OR REDO THE STEP OTHERWISE.
C-----------------------------------------------------------------------
        IF (NEWQ .EQ. NQ) GO TO 170
  630 NQ = NEWQ
        L = NQ + 1
        IRET = 2
        GO TO 130
C-----------------------------------------------------------------------
C CONTROL REACHES THIS SECTION IF 3 OR MORE FAILURES HAVE OCCURED.
C IT IS ASSUMED THAT THE DERIVATIVES THAT HAVE ACCUMULATED IN THE
C Y ARRAY HAVE ERRORS OF THE WRONG ORDER.  HENCE THE FIRST
C DERIVATIVE IS RECOMPUTED, AND THE ORDER IS SET TO 1.  THEN
C H IS REDUCED BY A FACTOR OF 10, AND THE STEP IS RETRIED.
C AFTER A TOTAL OF 7 FAILURES, AN EXIT IS TAKEN WITH KFLAG = -2.
C-----------------------------------------------------------------------
  640 IF (KFLAG .EQ. -7) GO TO 670
        RH = .1
        RH = AMAX1(HMIN/ ABS(H),RH)
        H = H*RH
        IER = 0
        CALL DIFFUN (N, T, Y, SAVE1, IER, PW, IPIV, WORK, IWORK)
        IF ( IER .NE. 0 ) GO TO 665
        NJE = NJE + 1
        DO 650 I = 1,N
  650   Y(I,2) = H*SAVE1(I)
        IWEVAL = MITER
        IDOUB = 10
        IF (NQ .EQ. 1) GO TO 200
        NQ = 1
        L = 2
        IRET = 3
        GO TO 130
```

```
         C----------------------------------------------------------------- STIFIB     401
         C ALL RETURNS ARE MADE THROUGH THIS SECTION.  H IS SAVED IN HOLD   STIFIB     402
         C TO ALLOW THE CALLER TO CHANGE H ON THE NEXT STEP.                STIFIB     403
         C----------------------------------------------------------------- STIFIB     404
           660 KFLAG = -1                                                   STIFIB     405
               GO TO 700                                                    STIFIB     406
           670 KFLAG = -2                                                   STIFIB     407
               GO TO 700                                                    STIFIB     408
           680 KFLAG = -3                                                   STIFIB     409
               GO TO 700                                                    STIFIB     410
           685 KFLAG = -4                                                   STIFIB     411
               GO TO 700                                                    STIFIB     412
           690 RMAX = 10.                                                   STIFIB     413
           700 HOLD = H                                                     STIFIB     414
               JSTART = NQ                                                  STIFIB     415
               RETURN                                                       STIFIB     416
               END                                                          STIFIB     417
```

```
CARD NR. SEVERITY  DETAILS    DIAGNOSIS OF PROBLEM

   151     I                  AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
```

```
      SUBROUTINE GFUN (T,C,UDOT,NPDE,NCPTS,A,BC,DBDU,DBDUX,DZDT,
     *                 XC,UVAL,ILEFT)
C-------------------------------------------------------------------
C CALLING ARGUMENTS ARE DEFINED BELOW AND IN PDECOL.
C
C SUBROUTINE GFUN COMPUTES THE FUNCTION UDOT=G(C,T), THE RIGHT-
C HAND SIDE OF THE SEMI-DISCRETE APPROXIMATION TO THE ORIGINAL
C SYSTEM OF PARTIAL DIFFERENTIAL EQUATIONS AND UPDATES THE BOUNDARY
C CONDITION INFORMATION.
C
C PACKAGE ROUTINES CALLED..  EVAL
C USER ROUTINES CALLED..     BNDRY,F
C CALLED BY..                DIFFUN,PSETIB,RES
C FORTRAN FUNCTIONS USED..   NONE
C-------------------------------------------------------------------
      DIMENSION C(NPDE,NCPTS),UDOT(NPDE,NCPTS)
      DIMENSION A(1),BC(NPDE,NPDE,4),XC(1),UVAL(NPDE,3),ILEFT(1)
      DIMENSION DZDT(NPDE),DBDU(NPDE,NPDE),DBDUX(NPDE,NPDE)
      COMMON /SIZES/ NINT,KORD,IDUM(4),IQUAD
      DO 10 K=1,4
        DO 10 J=1,NPDE
          DO 10 I=1,NPDE
            BC(I,J,K) = 0.0
   10 CONTINUE
C-------------------------------------------------------------------
C UPDATE THE LEFT BOUNDARY VALUES.  SAVE LEFT BOUNDARY CONDITION
C INFORMATION IN THE FIRST 2*NPDE*NPDE LOCATIONS OF BC.
C
C NOTE.. UVAL(K,1) = U(K), UVAL(K,2) = UX(K), AND UVAL(K,3) = UXX(K).
C-------------------------------------------------------------------
      CALL EVAL(1,NPDE,C,UVAL,A,ILEFT)
      CALL BNDRY(T,XC(1),UVAL,UVAL(1,2),DBDU,DBDUX,DZDT,NPDE)
      CALL F(T,XC(1),UVAL,UVAL(1,2),UVAL(1,3),UDOT,NPDE)
      ILIM = KORD + 2
      DO 30 K=1,NPDE
        BC(K,K,1) = 1.
        IF( DBDU(K,K) .EQ. 0.0  .AND.  DBDUX(K,K) .EQ. 0.0 ) GO TO 30
        UDOT(K,1) = DZDT(K)
        DO 20 J=1,NPDE
          BC(K,J,2) = A(ILIM) * DBDUX(K,J)
          BC(K,J,1) = DBDU(K,J) - BC(K,J,2)
   20   CONTINUE
   30 CONTINUE
C-------------------------------------------------------------------
C MAIN LOOP TO FORM RIGHT SIDE OF ODES AT THE COLLOCATION POINTS.
C-------------------------------------------------------------------
      ILIM = NCPTS - 1
      DO 40 I=2,ILIM
        CALL EVAL(I,NPDE,C,UVAL,A,ILEFT)
        CALL F(T,XC(I),UVAL,UVAL(1,2),UVAL(1,3),UDOT(1,I),NPDE)
   40 CONTINUE
C-------------------------------------------------------------------
C UPDATE THE RIGHT BOUNDARY VALUES.  SAVE THE RIGHT BOUNDARY CONDITION
C INFORMATION IN THE LAST 2*NPDE*NPDE LOCATIONS IN BC.
C-------------------------------------------------------------------
      CALL EVAL(NCPTS,NPDE,C,UVAL,A,ILEFT)
      CALL F(T,XC(NCPTS),UVAL,UVAL(1,2),UVAL(1,3),UDOT(1,NCPTS),NPDE)
```

```
      CALL BNDRY(T,XC(NCPTS),UVAL,UVAL(1,2),DBDU,DBDUX,DZDT,NPDE)
      ILIM = NCPTS * 3 * KORD - KORD - 1
      DO 60 K=1,NPDE
        BC(K,K,4) = 1.
        IF( DBDU(K,K) .EQ. 0.0  .AND.  DBDUX(K,K) .EQ. 0.0 ) GO TO 60
        UDOT(K,NCPTS) = DZDT(K)
        DO 50 J=1,NPDE
          BC(K,J,3) = A(ILIM) * DBDUX(K,J)
          BC(K,J,4) = DBDU(K,J) - BC(K,J,3)
   50   CONTINUE
   60 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE EVAL(ICPT,NPDE,C,UVAL,A,ILEFT)
C-------------------------------------------------------------------
C CALLING ARGUMENTS ARE DEFINED BELOW AND IN PDECOL.
C
C SUBROUTINE EVAL EVALUATES U(K), UX(K), AND UXX(K), K=1 TO NPDE,
C AT THE COLLOCATION POINT WITH INDEX ICPT USING THE VALUES OF
C THE BASIS FUNCTION COEFFICIENTS IN C AND THE BASIS FUNCTION VALUES
C STORED IN A.  THE RESULTS ARE STORED IN UVAL AS FOLLOWS..
C UVAL(K,1) = U(K), UVAL(K,2) = UX(K), AND UVAL(K,3) = UXX(K).
C
C PACKAGE ROUTINES CALLED..  NONE
C USER ROUTINES CALLED..     NONE
C CALLED BY..                GFUN,PDECOL,PSETIB
C FORTRAN FUNCTIONS USED..   NONE
C-------------------------------------------------------------------
      DIMENSION C(NPDE,1),UVAL(NPDE,3),A(1),ILEFT(1)
      COMMON /SIZES/ NINT,KORD,IDUM(5)
      IK = ILEFT(ICPT) - KORD
      IC = 3*KORD*(ICPT-1)
      DO 10 M=1,3
        ICC = IC + KORD*(M-1)
        DO 10 J=1,NPDE
          UVAL(J,M) = 0.
          DO 10 I=1,KORD
            UVAL(J,M) = UVAL(J,M) + C(J,I+IK)*A(I+ICC)
   10 CONTINUE
      RETURN
      END
```

AD-A082 803 ARMY ARMAMENT RESEARCH AND DEVELOPMENT COMMAND ABERD--ETC F/G 21/2
A METHOD FOR COMPUTING THE FLAME SPEED OR A LAMINAR, PREMIXED, --ETC(U)
JAN 80 T P COFFEE, J M HEIMERL

UNCLASSIFIED ARBRL-TR-02212 SBIE -AD-E430 386 NL

2 OF 2
AD A082803

END
DATE
FILMED
5-80
DTIC

```
      SUBROUTINE DIFFUN (N, T, Y, YDOT, IER, PW, IPIV, WORK, IWORK)
C-----------------------------------------------------------------------
C CALLING ARGUMENTS ARE DEFINED BELOW AND IN PDECOL.
C
C THIS ROUTINE COMPUTES YDOT = A(Y,T)**-1 * G(Y,T) BY USE OF
C THE ROUTINES GFUN, ADDA, DECB, AND SOLB.
C
C PACKAGE ROUTINES CALLED..  ADDA,DECB,GFUN,SOLB
C USER ROUTINES CALLED..     NONE
C CALLED BY..                STIFIB
C FORTRAN FUNCTIONS USED..   NONE
C-----------------------------------------------------------------------
      DIMENSION Y(N),YDOT(N),PW(1),IPIV(1),WORK(1),IWORK(1)
      COMMON /GEAR9/ EPSJ,R0,ML,MU,MW,NM1,NUML,NOW
      COMMON /SIZES/ NINT,KORD,NCC,NPDE,NCPTS,NEQN,IQUAD
      COMMON /ISTART/ IW1,IW2,IW3,IDUM(5),IW9,IW10,IW11,IW12,IW13,IW14,
     *                IW15,IW16,IW17,IW18
      CALL GFUN (T, Y, YDOT, NPDE, NCPTS, WORK(IW1), WORK, WORK(IW14),
     *           WORK(IW15), WORK(IW16), WORK(IW3), WORK(IW9), IWORK)
      DO 10 I = 1,NOW
   10   PW(I) = 0.
      NO = NM1 + 1
      CALL ADDA (PW, NO, WORK(IW1), IWORK, WORK, NPDE)
      CALL DECB (NO, N, ML, MU, PW, IPIV, IER)
      IF ( IER .NE. 0 ) RETURN
      CALL SOLB (NO, N, ML, MU, PW, YDOT, IPIV)
      RETURN
      END
```

202440

```
      SUBROUTINE ADDA(PW,NO,A,ILEFT,BC,NPDE)
C-----------------------------------------------------------------------
C CALLING ARGUMENTS ARE DEFINED BELOW AND IN PDECOL AND STIFIB.
C
C SUBROUTINE ADDA ADDS THE MATRIX  A  TO THE MATRIX STORED IN  PW  IN
C BAND FORM.  PW IS STORED BY DIAGONALS WITH THE LOWERMOST DIAGONAL
C STORED IN THE FIRST COLUMN OF THE ARRAY.
C
C PACKAGE ROUTINES CALLED..  NONE
C USER ROUTINES CALLED..     NONE
C CALLED BY..                DIFFUN,PSETIB
C FORTRAN FUNCTIONS USED..   NONE
C-----------------------------------------------------------------------
      DIMENSION PW(NO,1),A(1),ILEFT(1),BC(NPDE,NPDE,4)
      COMMON /SIZES/ NINT,KORD,NCC,NPD,NCPTS,NEQN,IQUAD
C-----------------------------------------------------------------------
C ADD THE BOUNDARY CONDITION PORTIONS OF THE A MATRIX TO PW ( THE FIRST
C AND LAST BLOCK ROWS).
C-----------------------------------------------------------------------
      ICOL = (ILEFT(1) + IQUAD - 1) * NPDE
      DO 10 I=1,NPDE
        IBOT = NEQN - NPDE + I
        DO 10 J=1,NPDE
          IND = ICOL + J - I
          PW(I,IND) = PW(I,IND) + BC(I,J,1)
          PW(I,IND+NPDE) = PW(I,IND+NPDE) + BC(I,J,2)
          PW(IBOT,IND-NPDE) = PW(IBOT,IND-NPDE) + BC(I,J,3)
          PW(IBOT,IND) = PW(IBOT,IND) + BC(I,J,4)
   10 CONTINUE
C-----------------------------------------------------------------------
C UPDATE THE REMAINING ROWS OF PW BY ADDING THE APPROPRIATE VALUES
C IN A TO PW.
C-----------------------------------------------------------------------
      IND = NCPTS - 1
      DO 20 I=2,IND
        I1 = (I-1) * NPDE
        I2 = (I-1) * KORD * 3
        ICOL = ILEFT(I) - I + IQUAD - 1
        DO 20 J=1,KORD
          J1 = (ICOL+J) * NPDE
          J2 = I2 + J
          DO 20 JJ=1,NPDE
   20       PW(I1+JJ,J1) = PW(I1+JJ,J1) + A(J2)
      RETURN
      END
```

```
      SUBROUTINE RES(T,H,C,V,R,NPDE,NCPTS,A,ILEFT,BC,DBDU,DBDUX,DZDT,    RES    2
     *               XC,UVAL)                                            RES    3
C----------------------------------------------------------------------- RES    4
C CALLING ARGUMENTS ARE DEFINED BELOW AND IN PDECOL.                     RES    5
C                                                                        RES    6
C SUBROUTINE RES COMPUTES THE RESIDUAL VECTOR R = H*G(C,T) - A(C,T)*V    RES    7
C WHERE H IS THE CURRENT TIME STEP SIZE, G IS A VECTOR, A IS A           RES    8
C MATRIX, V IS A VECTOR, AND T IS THE CURRENT TIME.                      RES    9
C                                                                        RES   10
C PACKAGE ROUTINES CALLED..  GFUN                                        RES   11
C USER ROUTINES CALLED..     NONE                                        RES   12
C CALLED BY..                STIFIB                                      RES   13
C FORTRAN FUNCTIONS USED..   NONE                                        RES   14
C----------------------------------------------------------------------- RES   15
      DIMENSION C(NPDE,NCPTS),R(NPDE,1),V(NPDE,1)                        RES   16
      DIMENSION A(1),ILEFT(1),BC(NPDE,NPDE,4),XC(1),UVAL(1)              RES   17
      DIMENSION DBDU(NPDE,NPDE),DBDUX(NPDE,NPDE),DZDT(NPDE)              RES   18
      COMMON /SIZES/ NINT,KORD,NCC,IDUM(3),IQUAD                         RES   19
C----------------------------------------------------------------------- RES   20
C FORM G(C,T) AND STORE IN R.                                            RES   21
C----------------------------------------------------------------------- RES   22
      CALL GFUN(T,C,R,NPDE,NCPTS,A,BC,DBDU,DBDUX,DZDT,XC,UVAL,ILEFT)     RES   23
C----------------------------------------------------------------------- RES   24
C FORM THE FIRST AND LAST BLOCK ROWS OF THE RESIDUAL VECTOR              RES   25
C WHICH ARE DEPENDENT ON THE BOUNDARY CONDITIONS.                        RES   26
C----------------------------------------------------------------------- RES   27
      ILIM = NCPTS - 1                                                   RES   28
      DO 20 I=1,NPDE                                                     RES   29
        SUM1 = 0.0                                                       RES   30
        SUM2 = 0.0                                                       RES   31
        DO 10 J=1,NPDE                                                   RES   32
          SUM1 = SUM1 + BC(I,J,1) * V(J,1) + BC(I,J,2) * V(J,2)          RES   33
          SUM2 = SUM2 + BC(I,J,3) * V(J,ILIM) + BC(I,J,4) * V(J,NCPTS)   RES   34
   10   CONTINUE                                                         RES   35
        R(I,1) = H * R(I,1) - SUM1                                       RES   36
        R(I,NCPTS) = H * R(I,NCPTS) - SUM2                               RES   37
   20 CONTINUE                                                           RES   38
C----------------------------------------------------------------------- RES   39
C FORM THE REMAINING COMPONENTS OF THE RESIDUAL VECTOR.                  RES   40
C----------------------------------------------------------------------- RES   41
      DO 50 ICPTS=2,ILIM                                                 RES   42
        I2 = (ICPTS-1) * KORD * 3                                        RES   43
        ICOL = ILEFT(ICPTS) - KORD                                       RES   44
        DO 40 JJ=1,NPDE                                                  RES   45
          SUM1 = 0.                                                      RES   46
          DO 30 J=1,KORD                                                 RES   47
            SUM1 = SUM1 + A(I2+J) * V(JJ,ICOL+J)                         RES   48
   30       CONTINUE                                                     RES   49
          R(JJ,ICPTS) = H*R(JJ,ICPTS) - SUM1                             RES   50
   40     CONTINUE                                                       RES   51
   50   CONTINUE                                                         RES   52
      RETURN                                                             RES   53
      END                                                                RES   54
```

```
      SUBROUTINE PSETIB (C, PW, NO, CON, MITER, IER, A, ILEFT, XC, UVAL,
     *     SAVE2,IPIV,CMAX,DFDU,DFDUX,DFDUXX,DZDT,DBDU,DBDUX,BC,NPDE)
C-----------------------------------------------------------------------
C CALLING ARGUMENTS ARE DEFINED BELOW AND IN PDECOL AND STIFIB.
C
C PSETIB IS CALLED BY STIFIB TO COMPUTE AND PROCESS THE MATRIX
C PW = A - H*EL(1)*(DG/DC), WHERE A AND DG/DC ARE TREATED IN BAND
C FORM.  DG/DC IS COMPUTED, EITHER WITH THE AID OF THE USER-SUPPLIED
C ROUTINE DERIVF IF MITER = 1, OR BY FINITE DIFFERENCING WITH THE AID
C OF THE PACKAGE-SUPPLIED ROUTINE DIFFF IF MITER = 2.  FINALLY,
C PW IS SUBJECTED TO LU DECOMPOSITION IN PREPARATION FOR LATER
C SOLUTION OF LINEAR SYSTEMS WITH PW AS COEFFICIENT MATRIX.
C SEE SUBROUTINES DECB AND SOLB.
C
C IN ADDITION TO VARIABLES DESCRIBED PREVIOUSLY, COMMUNICATION
C WITH PSETIB USES THE FOLLOWING..
C   EPSJ    = SQRT(UROUND), USED IN THE NUMERICAL JACOBIAN INCREMENTS.
C   MW      = ML + MU + 1.
C   NM1     = NO - 1.
C   NOML    = NO*ML.
C   NOW     = NO*MW.
C
C PACKAGE ROUTINES CALLED..  ADDA,DECB,DIFFF,EVAL,GFUN
C USER ROUTINES CALLED..     BNDRY,DERIVF
C CALLED BY..                STIFIB
C FORTRAN FUNCTIONS USED..   ABS,FLOAT,MAX0,MIN0,SQRT
C-----------------------------------------------------------------------
      DIMENSION PW(NO,1),C(1),CMAX(1)
      DIMENSION A(1),ILEFT(1),BC(1),XC(1),UVAL(NPDE,3),SAVE2(1),IPIV(1)
      DIMENSION DFDU(NPDE,NPDE),DFDUX(NPDE,NPDE),DFDUXX(NPDE,NPDE)
      DIMENSION DZDT(NPDE),DBDU(NPDE,NPDE),DBDUX(NPDE,NPDE)
      COMMON /SIZES/ NINT,KORD,NCC,NPD,NCPTS,NEQN,IQUAD
      COMMON /GEAR1/ T,H,DUMMY(3),UROUND,N,IDUMMY(3)
      COMMON /GEAR9/ EPSJ,R0,ML,MU,MW,NM1,NOML,NOW
      DO 10 I=1,NOW
   10   PW(I,1) = 0.
      IF ( MITER .EQ. 1 ) GO TO 25
      CALL GFUN (T, C, SAVE2, NPDE, NCPTS,A,BC,DBDU,DBDUX,DZDT,XC,
     *           UVAL,ILEFT)
      D = 0.
      DO 20 I = 1,N
   20   D = D + SAVE2(I)**2
      R0 =  ABS(H)* SQRT(D/FLOAT(NO))*1.E+03*UROUND
C-----------------------------------------------------------------------
C COMPUTE BLOCK ROWS OF JACOBIAN.
C-----------------------------------------------------------------------
   25 DO 30 I=1,NCPTS
        I1 = (I-1)*NPDE
        I2 = (I-1)*KORD*3
        CALL EVAL(I,NPDE,C,UVAL,A,ILEFT)
        IF ( MITER .EQ. 1 )
     *      CALL DERIVF(T,XC(I),UVAL,UVAL(1,2),UVAL(1,3),
     *                  DFDU,DFDUX,DFDUXX,NPDE)
        IF ( MITER .EQ. 2 )
     *       CALL DIFFF(T,XC(I),I,UVAL,UVAL(1,2),UVAL(1,3),
     *                  DFDU,DFDUX,DFDUXX,NPDE,CMAX,SAVE2)
        ICOL = ILEFT(I) - I + IQUAD - 1
```

```
      KLOW = MAX0(1,I+2-NCPTS)
      KUP = MIN0(KORD,KORD+I-2)
      DO 30 KBLK=KLOW,KUP
        J1 = (ICOL+KBLK)*NPDE
        J2 = I2 + KBLK
        J3 = J2 + KORD
        J4 = J3 + KORD
        DO 30 L=1,NPDE
          DO 30 K=1,NPDE
            PW(I1+K,J1-K+L) = DFDU(K,L)*A(J2) + DFDUX(K,L)*A(J3)
     *                        + DFDUXX(K,L)*A(J4)
   30 CONTINUE
C-----------------------------------------------------------------------
C MODIFY THE LAST AND THE FIRST BLOCK ROWS FOR THE BOUNDARY CONDITIONS.
C CURRENT INFORMATION FOR THE RIGHT BOUNDARY CONDITION IS ALREADY IN
C THE ARRAYS DBDU, DBDUX AS A RESULT OF A PREVIOUS CALL TO GFUN.
C-----------------------------------------------------------------------
      IROW = NEQN - NPDE
      DO 50 K=1,NPDE
        IROW = IROW + 1
        IF(DBDU(K,K) .EQ. 0.0  .AND.  DBDUX(K,K) .EQ. 0.0) GO TO 50
        DO 40 J=1,MW
          PW(IROW,J) = 0.0
   40   CONTINUE
   50 CONTINUE
      CALL EVAL(1,NPDE,C,UVAL,A,ILEFT)
      CALL BNDRY(T,XC(1),UVAL,UVAL(1,2),DBDU,DBDUX,DZDT,NPDE)
      DO 70 K=1,NPDE
        IF(DBDU(K,K) .EQ. 0.0  .AND.  DBDUX(K,K) .EQ. 0.0) GO TO 70
      DO 60 J=1,MW
          PW(K,J) = 0.0
   60   CONTINUE
   70 CONTINUE
      DO 80 I = 1,NOW
   80   PW(I,1) = PW(I,1)*CON
C-----------------------------------------------------------------------
C ADD MATRIX A(C,T) TO PW.
C-----------------------------------------------------------------------
      CALL ADDA (PW, NO, A, ILEFT, HC, NPDE)
C-----------------------------------------------------------------------
C DO LU DECOMPOSITION ON PW.
C-----------------------------------------------------------------------
      CALL DECB (NO, N, ML, MU, PW, IPIV, IER)
      RETURN
      END
```

```
      SUBROUTINE DIFFF(T,X,IPT,U,UX,UXX,DFDU,DFDUX,DFDUXX,NPDE,CMAX,    DIFFF     2
     *                 SAVE2)                                           DIFFF     3
C-------------------------------------------------------------------   DIFFF     4
C CALLING ARGUMENTS ARE DEFINED BELOW AND IN PDECOL.                    DIFFF     5
C                                                                       DIFFF     6
C SUBROUTINE DIFFF IS USED IF MITER=2 TO PROVIDE FINITE DIFFERENCE      DIFFF     7
C APPROXIMATIONS FOR THE PARTIAL DERIVATIVES OF THE K-TH USER DEFINED   DIFFF     8
C FUNCTION IN THE F ROUTINE WITH RESPECT TO THE VARIABLES U, UX, AND    DIFFF     9
C UXX.  THESE PARTIALS WITH RESPECT TO U, UX, AND UXX ARE COMPUTED,     DIFFF    10
C STORED, AND RETURNED IN THE NPDE BY NPDE ARRAYS DFDU, DFDUX, AND      DIFFF    11
C DFDUXX, RESPECTIVELY, AT COLLOCATION POINT NUMBER IPT.                DIFFF    12
C                                                                       DIFFF    13
C PACKAGE ROUTINES CALLED..  NONE                                       DIFFF    14
C USER ROUTINES CALLED..     F                                          DIFFF    15
C CALLED BY..                PSETIB                                     DIFFF    16
C FORTRAN FUNCTIONS USED..   AMAX1                                      DIFFF    17
C-------------------------------------------------------------------   DIFFF    18
      DIMENSION U(NPDE),UX(NPDE),UXX(NPDE),DFDU(NPDE,NPDE),             DIFFF    19
     *   DFDUX(NPDE,NPDE),DFDUXX(NPDE,NPDE),CMAX(1),SAVE2(1)            DIFFF    20
      COMMON /GEAR9/ EPSJ,R0,ML,MU,MW,NMI,NOML,NOW                      DIFFF    21
      ID = (IPT-1) * NPDE                                               DIFFF    22
      DO 40 J=1,NPDE                                                    DIFFF    23
        UJ = U(J)                                                       DIFFF    24
        R = EPSJ * CMAX(J)                                              DIFFF    25
        R = AMAX1(R,R0)                                                 DIFFF    26
        U(J) = U(J) + R                                                 DIFFF    27
        RINV = 1. / R                                                   DIFFF    28
        CALL F(T,X,U,UX,UXX,DFDU(1,J),NPDF)                             DIFFF    29
        DO 10 I=1,NPDE                                                  DIFFF    30
   10     DFDU(I,J) = ( DFDU(I,J) - SAVE2(I+ID) ) * RINV                DIFFF    31
        U(J) = UJ                                                       DIFFF    32
        UJ = UX(J)                                                      DIFFF    33
        UX(J) = UX(J) + R                                               DIFFF    34
        CALL F(T,X,U,UX,UXX,DFDUX(1,J),NPDE)                            DIFFF    35
        DO 20 I=1,NPDE                                                  DIFFF    36
   20     DFDUX(I,J) = ( DFDUX(I,J) - SAVE2(I+ID) ) * RINV              DIFFF    37
        UX(J) = UJ                                                      DIFFF    38
        UJ = UXX(J)                                                     DIFFF    39
        UXX(J) = UXX(J) + R                                             DIFFF    40
        CALL F(T,X,U,UX,UXX,DFDUAX(1,J),NPDE)                           DIFFF    41
        DO 30 I=1,NPDE                                                  DIFFF    42
   30     DFDUXX(I,J) = ( DFDUAX(I,J) - SAVE2(I+ID) ) * RINV            DIFFF    43
        UXX(J) = UJ                                                     DIFFF    44
   40 CONTINUE                                                          DIFFF    45
      RETURN                                                            DIFFF    46
      END                                                               DIFFF    47
```

```
          SUBROUTINE INTERP (TOUT, Y, NO, YO)                             INTERP   2
C-----------------------------------------------------------------------  INTERP   3
C CALLING ARGUMENTS ARE DEFINED BELOW AND IN STIFIH                       INTERP   4
C                                                                         INTERP   5
C SUBROUTINE INTERP COMPUTES INTERPOLATED VALUES OF THE DEPENDENT         INTERP   6
C VARIABLE Y AND STORES THEM IN YO.  THE INTERPOLATION IS TO THE          INTERP   7
C POINT T = TOUT, AND USES THE NORDSIECK HISTORY ARRAY Y, AS FOLLOWS..    INTERP   8
C                              NQ                                         INTERP   9
C                  YO(I)  =  SUM  Y(I,J+1)*S**J ,                         INTERP  10
C                             J=0                                         INTERP  11
C WHERE S = -(T-TOUT)/H.                                                  INTERP  12
C                                                                         INTERP  13
C PACKAGE ROUTINES CALLED..  NONE                                         INTERP  14
C USER ROUTINES CALLED..     NONE                                         INTERP  15
C CALLED BY..                PDECOL                                       INTERP  16
C FORTRAN FUNCTIONS USED..   NONE                                         INTERP  17
C-----------------------------------------------------------------------  INTERP  18
      DIMENSION YO(NO),Y(NO,1)                                            INTERP  19
      COMMON /GEAR1/ T,H,DUMMY(4),N,IDUMMY(2),JSTART                      INTERP  20
      DO 10 I = 1,N                                                       INTERP  21
   10   YO(I) = Y(I,1)                                                    INTERP  22
      L = JSTART + 1                                                      INTERP  23
      S = (TOUT - T)/H                                                    INTERP  24
      S1 = 1.                                                             INTERP  25
      DO 30 J = 2,L                                                       INTERP  26
        S1 = S1*S                                                         INTERP  27
        DO 20 I = 1,N                                                     INTERP  28
   20     YO(I) = YO(I) + S1*Y(I,J)                                       INTERP  29
   30 CONTINUE                                                            INTERP  30
      RETURN                                                              INTERP  31
      END                                                                 INTERP  32
```

```
      SUBROUTINE COSET (METH, NQ, EL, TQ)                               COSET    2
C-----------------------------------------------------------------------COSET    3
C COSET IS CALLED BY THE INTEGRATOR AND SETS COEFFICIENTS USED THERE.   COSET    4
C THE VECTOR EL, OF LENGTH NQ + 1, DETERMINES THE BASIC METHOD.         COSET    5
C THE VECTOR TQ, OF LENGTH 4, IS INVOLVED IN ADJUSTING THE STEP SIZE    COSET    6
C IN RELATION TO TRUNCATION ERROR.  ITS VALUES ARE GIVEN BY THE         COSET    7
C PERTST ARRAY.                                                         COSET    8
C                                                                       COSET    9
C THE VECTORS EL AND TQ DEPEND ON METH AND NQ.                          COSET   10
C THE MAXIMUM ORDER, MAXDER, OF THE METHODS AVAILABLE IS CURRENTLY      COSET   11
C 12 FOR THE ADAMS METHODS AND 5 FOR THE BDF METHODS.  MAXDER DEFAULTS  COSET   12
C TO 5 UNLESS THE USER SETS MAXDER TO SOME OTHER LEGITIMATE VALUE       COSET   13
C THROUGH THE COMMON BLOCK /OPTION/.  SEE PDECOL FOR ADDITIONAL DETAILS. COSET   14
C LMAX = MAXDER + 1 IS THE NUMBER OF COLUMNS IN THE Y ARRAY (SEE STIFIB COSET   15
C AND THE VARIABLE C, Y, OR WORK(IW10) IN PDECOL.                       COSET   16
C                                                                       COSET   17
C THE COEFFICIENTS IN PERTST NEED BE GIVEN TO ONLY ABOUT                COSET   18
C ONE PERCENT ACCURACY.  THE ORDER IN WHICH THE GROUPS APPEAR BELOW     COSET   19
C IS..  COEFFICIENTS FOR ORDER NQ - 1, COEFFICIENTS FOR ORDER NQ,       COSET   20
C COEFFICIENTS FOR ORDER NQ + 1.  WITHIN EACH GROUP ARE THE             COSET   21
C COEFFICIENTS FOR THE ADAMS METHODS, FOLLOWED BY THOSE FOR THE         COSET   22
C BDF METHODS.                                                          COSET   23
C                                                                       COSET   24
C REFERENCE                                                             COSET   25
C                                                                       COSET   26
C   GEAR, C.W., NUMERICAL INITIAL VALUE PROBLEMS IN ORDINARY            COSET   27
C     DIFFERENTIAL EQUATIONS, PRENTICE-HALL, ENGLEWOOD CLIFFS,          COSET   28
C     N. J., 1971.                                                      COSET   29
C                                                                       COSET   30
C PACKAGE ROUTINES CALLED..  NONE                                       COSET   31
C USER ROUTINES CALLED..     NONE                                       COSET   32
C CALLED BY..                STIFIB                                     COSET   33
C FORTRAN FUNCTIONS USED..   FLOAT                                      COSET   34
C-----------------------------------------------------------------------COSET   35
      DIMENSION PERTST(12,2,3),EL(13),TQ(4)                             COSET   36
      DATA  PERTST / 1.,1.,2.,1.,.3158,.07407,.01391,.002182,           COSET   37
     *                 .0002945,.00003492,.000003692,.0000003524,       COSET   38
     *               1.,1.,.5,.1667,.04167,1.,1.,1.,1.,1.,1.,1.,        COSET   39
     *               2.,12.,24.,37.69,53.33,70.08,87.97,106.9,          COSET   40
     *                 126.7,147.4,168.8,191.0,                         COSET   41
     *               2.0,4.5,7.333,10.42,13.7,1.,1.,1.,1.,1.,1.,1.,     COSET   42
     *               12.0,24.0,37.89,53.33,70.08,87.97,106.9,           COSET   43
     *                 126.7,147.4,168.8,191.0,1.,                      COSET   44
     *               3.0,6.0,9.167,12.5,1.,1.,1.,1.,1.,1.,1.,1. /       COSET   45
C                                                                       COSET   46
      GO TO (1,2),METH                                                  COSET   47
    1 GO TO (101,102,103,104,105,106,107,108,109,110,111,112),NQ        COSET   48
    2 GO TO (201,202,203,204,205),NQ                                    COSET   49
C-----------------------------------------------------------------------COSET   50
C THE FOLLOWING COEFFICIENTS SHOULD BE DEFINED TO MACHINE ACCURACY.     COSET   51
C FOR A GIVEN ORDER NQ, THEY CAN BE CALCULATED BY USE OF THE            COSET   52
C GENERATING POLYNOMIAL L(T), WHOSE COEFFICIENTS ARE EL(I)..            COSET   53
C      L(T) = EL(1) + EL(2)*T + ... + EL(NQ+1)*T**NQ.                   COSET   54
C FOR THE IMPLICIT ADAMS METHODS, L(T) IS GIVEN BY                      COSET   55
C      DL/DT = (T+1)*(T+2)* ... *(T+NQ-1)/K,    L(-1) = 0,              COSET   56
C WHERE                K = FACTORIAL(NQ-1).                             COSET   57
C FOR THE BDF METHODS,                                                  COSET   58
```

```
C       L(T) = (T+1)*(T+2)* ... *(T+NQ)/K,                                COSET
C WHERE         K = FACTORIAL(NQ)*(1 + 1/2 + ... + 1/NQ).                 COSET
C                                                                         COSET
C THE ORDER IN WHICH THE GROUPS APPEAR BELOW IS..                         COSET
C IMPLICIT ADAMS METHODS OF ORDERS 1 TO 12.                               COSET
C BDF METHODS OF ORDERS 1 TO 5.                                           COSET
C------------------------------------------------------------------------ COSET
  101 EL(1) = 1.0E-00                                                     COSET
      GO TO 900                                                           COSET
  102 EL(1) = 0.5E-00                                                     COSET
      EL(3) = 0.5E-00                                                     COSET
      GO TO 900                                                           COSET
  103 EL(1) = 4.1666666666667E-01                                         COSET
      EL(3) = 0.75E-00                                                    COSET
      EL(4) = 1.6666666666667E-01                                         COSET
      GO TO 900                                                           COSET
  104 EL(1) = 0.375E-00                                                   COSET
      EL(3) = 9.1666666666667E-01                                         COSET
      EL(4) = 3.3333333333333E-01                                         COSET
      EL(5) = 4.1666666666667E-02                                         COSET
      GO TO 900                                                           COSET
  105 EL(1) = 3.4861111111111E-01                                         COSET
      EL(3) = 1.0416666666667E-00                                         COSET
      EL(4) = 4.8611111111111E-01                                         COSET
      EL(5) = 1.0416666666667E-01                                         COSET
      EL(6) = 8.3333333333333E-03                                         COSET
      GO TO 900                                                           COSET
  106 EL(1) = 3.2986111111111E-01                                         COSET
      EL(3) = 1.1416666666667E-00                                         COSET
      EL(4) = 0.625E-00                                                   COSET
      EL(5) = 1.7708333333333E-01                                         COSET
      EL(6) = 0.025E-00                                                   COSET
      EL(7) = 1.3888888888889E-03                                         COSET
      GO TO 900                                                           COSET
  107 EL(1) = 3.1559193121693E-01                                         COSET
      EL(3) = 1.225E-00                                                   COSET
      EL(4) = 7.5185185185185E-01                                         COSET
      EL(5) = 2.5520833333333E-01                                         COSET
      EL(6) = 4.8611111111111E-02                                         COSET
      EL(7) = 4.8611111111111E-03                                         COSET
      EL(8) = 1.9841269841270E-04                                         COSET
      GO TO 900                                                           COSET
  108 EL(1) = 3.0422453703704E-01                                         COSET
      EL(3) = 1.2964285714286E-00                                         COSET
      EL(4) = 8.6851851851852E-01                                         COSET
      EL(5) = 3.3576388888889E-01                                         COSET
      EL(6) = 7.7777777777778E-02                                         COSET
      EL(7) = 1.0648148148148E-02                                         COSET
      EL(8) = 7.9365079365079E-04                                         COSET
      EL(9) = 2.4801587301587E-05                                         COSET
      GO TO 900                                                           COSET
  109 EL(1) = 2.9486800044092E-01                                         COSET
      EL(3) = 1.3589285714286E-00                                         COSET
      EL(4) = 9.7655423280423E-01                                         COSET
      EL(5) = 0.4171875E-00                                               COSET
      EL(6) = 1.1135416666667E-01                                         COSET
      EL(7) = 0.01875E-00                                                 COSET
```

```
      EL(8) = 1.9345238095238E-03                            COSET     116
      EL(9) = 1.1160714285714E-04                            COSET     117
      EL(10)= 2.7557319223986E-06                            COSET     118
      GO TO 900                                              COSET     119
  110 EL(1) = 2.8697544642857E-01                            COSET     120
      EL(3) = 1.4144841269841E-00                            COSET     121
      EL(4) = 1.0772156084656E-00                            COSET     122
      EL(5) = 4.9856701940035E-01                            COSET     123
      EL(6) = 0.1484375E-00                                  COSET     124
      EL(7) = 2.9060570987654E-02                            COSET     125
      EL(8) = 3.7202380952381E-03                            COSET     126
      EL(9) = 2.9968584656085E-04                            COSET     127
      EL(10)= 1.3778659611993E-05                            COSET     128
      EL(11)= 2.7557319223986E-07                            COSET     129
      GO TO 900                                              COSET     130
  111 EL(1) = 2.8018959644394E-01                            COSET     131
      EL(3) = 1.4644841269841E-00                            COSET     132
      EL(4) = 1.1715145502646E-00                            COSET     133
      EL(5) = 5.7935819003527E-01                            COSET     134
      EL(6) = 1.8832286155203E-01                            COSET     135
      EL(7) = 4.1430362654321E-02                            COSET     136
      EL(8) = 6.2111441798942E-03                            COSET     137
      EL(9) = 6.2520667989418E-04                            COSET     138
      EL(10)= 4.0417401528513E-05                            COSET     139
      EL(11)= 1.5156525573192E-06                            COSET     140
      EL(12)= 2.5052108385442E-08                            COSET     141
      GO TO 900                                              COSET     142
  112 EL(1) = 2.7426554003160E-01                            COSET     143
      EL(3) = 1.5099386724387E-00                            COSET     144
      EL(4) = 1.2602711640212E-00                            COSET     145
      EL(5) = 6.5923418209877E-01                            COSET     146
      EL(6) = 2.3045800264550E-01                            COSET     147
      EL(7) = 5.5697246105232E-02                            COSET     148
      EL(8) = 9.4394841269841E-03                            COSET     149
      EL(9) = 1.1192749669312E-03                            COSET     150
      EL(10)= 9.0939153439153E-05                            COSET     151
      EL(11)= 4.8225308641975E-06                            COSET     152
      EL(12)= 1.5031265031265E-07                            COSET     153
      EL(13)= 2.0876756987868E-09                            COSET     154
      GO TO 900                                              COSET     155
  201 EL(1) = 1.0E-00                                        COSET     156
      GO TO 900                                              COSET     157
  202 EL(1) = 6.6666666666667E-01                            COSET     158
      EL(3) = 3.3333333333333E-01                            COSET     159
      GO TO 900                                              COSET     160
  203 EL(1) = 5.4545454545455E-01                            COSET     161
      EL(3) = EL(1)                                          COSET     162
      EL(4) = 9.0909090909091E-02                            COSET     163
      GO TO 900                                              COSET     164
  204 EL(1) = 0.48E-00                                       COSET     165
      EL(3) = 0.7E-00                                        COSET     166
      EL(4) = 0.2E-00                                        COSET     167
      EL(5) = 0.02E-00                                       COSET     168
      GO TO 900                                              COSET     169
  205 EL(1) = 4.3795620437956E-01                            COSET     170
      EL(3) = 8.2116788321168E-01                            COSET     171
      EL(4) = 3.1021897810219E-01                            COSET     172
```

```
                                                                  COSET      173
                  EL(5) = 5.4744525547445E-02                     COSET      174
                  EL(6) = 3.6496350364964E-03                     COSET      175
               C                                                  COSET      176
175        900    DO 910 K = 1,3                                  COSET      177
           910      TQ(K) = PERTST(NQ,METH,K)                     COSET      178
                  TQ(4) = .5E-00*TQ(2)/ FLOAT(NQ+2)               COSET      179
                  RETURN                                          COSET      180
                  END
```

```
CARD NR. SEVERITY  DETAILS    DIAGNOSIS OF PROBLEM

    46     I                  AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
```

```
      SUBROUTINE DECB (NDIM, N, ML, MU, B, IPIV, IER)                   DECB       2
C---------------------------------------------------------------------- DECB       3
C SUBROUTINES DECB AND SOLB FORM A TWO SUBROUTINE PACKAGE FOR THE       DECB       4
C DIRECT SOLUTION OF A SYSTEM OF LINEAR EQUATIONS IN WHICH THE          DECB       5
C COEFFICIENT MATRIX IS REAL AND BANDED.                                DECB       6
C                                                                       DECB       7
C    LU DECOMPOSITION OF BAND MATRIX A..  L*U = P*A , WHERE P IS A      DECB       8
C       PERMUTATION MATRIX, L IS A UNIT LOWER TRIANGULAR MATRIX,        DECB       9
C       AND U IS AN UPPER TRIANGULAR MATRIX.                            DECB      10
C    N      =  ORDER OF MATRIX.                                         DECB      11
C    B      =  N BY (2*ML+MU+1) ARRAY CONTAINING THE MATRIX A ON INPUT  DECB      12
C             AND ITS FACTORED FORM ON OUTPUT.                          DECB      13
C             ON INPUT, B(I,K) (1.LE.I.LE.N) CONTAINS THE K-TH          DECB      14
C             DIAGONAL OF A, OR A(I,J) IS STORED IN B(I,J-I+ML+1).      DECB      15
C             ON OUTPUT, B CONTAINS THE L AND U FACTORS, WITH           DECB      16
C             U IN COLUMNS 1 TO ML+MU+1, AND L IN COLUMNS               DECB      17
C             ML+MU+2 TO 2*ML+MU+1.                                     DECB      18
C    ML,MU =  WIDTHS OF THE LOWER AND UPPER PARTS OF THE BAND, NOT      DECB      19
C             COUNTING THE MAIN DIAGONAL. TOTAL BANDWIDTH IS ML+MU+1.   DECB      20
C    NDIM  =  THE FIRST DIMENSION (COLUMN LENGTH) OF THE ARRAY B.       DECB      21
C             NDIM MUST BE .GE. N.                                      DECB      22
C    IPIV  =  ARRAY OF LENGTH N CONTAINING PIVOT INFORMATION.           DECB      23
C    IER   =  ERROR INDICATOR..                                         DECB      24
C          =  0  IF NO ERROR,                                           DECB      25
C          =  K  IF THE K-TH PIVOT CHOSEN WAS ZERO (A IS SINGULAR).     DECB      26
C    THE INPUT ARGUMENTS ARE  NDIM, N, ML, MU, B.                       DECB      27
C    THE OUTPUT ARGUMENTS ARE  B, IPIV, IER.                            DECB      28
C                                                                       DECB      29
C PACKAGE ROUTINES CALLED..  NONE                                       DECB      30
C USER ROUTINES CALLED..     NONE                                       DECB      31
C CALLED BY..                DIFFUN,INITAL,PSETIB                       DECB      32
C FORTRAN FUNCTIONS USED..   ABS,MIN0                                   DECB      33
C---------------------------------------------------------------------- DECB      34
      DIMENSION B(NDIM,1),IPIV(N)                                       DECB      35
      IER = 0                                                           DECB      36
      IF (N .EQ. 1) GO TO 92                                            DECB      37
      LL = ML + MU + 1                                                  DECB      38
      N1 = N - 1                                                        DECB      39
      IF (ML .EQ. 0) GO TO 32                                           DECB      40
      DO 30 I = 1,ML                                                    DECB      41
        II = MU + I                                                     DECB      42
        K = ML + 1 - I                                                  DECB      43
        DO 10 J = 1,II                                                  DECB      44
   10     B(I,J) = B(I,J+K)                                             DECB      45
        K = II + 1                                                      DECB      46
        DO 20 J = K,LL                                                  DECB      47
   20     B(I,J) = 0.                                                   DECB      48
   30   CONTINUE                                                        DECB      49
   32 LR = ML                                                           DECB      50
      DO 90 NR = 1,N1                                                   DECB      51
        NP = NR + 1                                                     DECB      52
        IF (LR .NE. N) LR = LR + 1                                      DECB      53
        MX = NR                                                         DECB      54
        XM =  ABS(B(NR,1))                                              DECB      55
        IF (ML .EQ. 0) GO TO 42                                         DECB      56
        DO 40 I = NP,LR                                                 DECB      57
          IF ( ABS(B(I,1)) .LE. XM) GO TO 40                            DECB      58
```

```
            MX = I                                             DECB    59
            XM = ABS(B(I,1))                                   DECB    60
   40       CONTINUE                                           DECB    61
   42     IPIV(NR) = MX                                        DECB    62
          IF (MX .EQ. NR) GO TO 60                             DECB    63
          DO 50 I = 1,LL                                       DECB    64
            XX = B(NR,I)                                       DECB    65
            B(NR,I) = B(MX,I)                                  DECB    66
   50       B(MX,I) = XX                                       DECB    67
   60     XM = B(NR,1)                                         DECB    68
          IF (XM .EQ. 0.) GO TO 100                            DECB    69
          B(NR,1) = 1./XM                                      DECB    70
          IF (ML .EQ. 0) GO TO 90                              DECB    71
          XM = -B(NR,1)                                        DECB    72
          KK = MIN0(N-NR,LL-1)                                 DECB    73
          DO 80 I = NP,LR                                      DECB    74
            J = LL + I - NR                                    DECB    75
            XX = B(I,1)*XM                                     DECB    76
            B(NR,J) = XX                                       DECB    77
            DO 70 II = 1,KK                                    DECB    78
   70         B(I,II) = B(I,II+1) + XX*B(NR,II+1)              DECB    79
   80       B(I,LL) = 0.                                       DECB    80
   90     CONTINUE                                             DECB    81
   92   NR = N                                                 DECB    82
        IF (B(N,1) .EQ. 0.) GO TO 100                          DECB    83
        B(N,1) = 1./B(N,1)                                     DECB    84
        RETURN                                                 DECB    85
  100   IER = NR                                               DECB    86
        RETURN                                                 DECB    87
        END                                                    DECB    88
```

```
      SUBROUTINE SOLB (NDIM, N, ML, MU, B, Y, IPIV)
C-----------------------------------------------------------------------
C SUBROUTINES DECB AND SOLB FORM A TWO SUBROUTINE PACKAGE FOR THE
C DIRECT SOLUTION OF A SYSTEM OF LINEAR EQUATIONS IN WHICH THE
C COEFFICIENT MATRIX IS REAL AND BANDED.
C
C    SOLUTION OF  A*X = C  GIVEN LU DECOMPOSITION OF A FROM DECB.
C    Y  =  RIGHT-HAND VECTOR C, OF LENGTH N, ON INPUT,
C       =  SOLUTION VECTOR X ON OUTPUT.
C    ALL THE ARGUMENTS ARE INPUT ARGUMENTS.
C    THE OUTPUT ARGUMENT IS  Y.
C
C PACKAGE ROUTINES CALLED..  NONE
C USER ROUTINES CALLED..     NONE
C CALLED BY..                DIFFUN,INITAL,STIFIB
C FORTRAN FUNCTIONS USED..   MINO
C-----------------------------------------------------------------------
      DIMENSION B(NDIM,1),Y(N),IPIV(N)
      IF (N .EQ. 1) GO TO 60
      N1 = N - 1
      LL = ML + MU + 1
      IF (ML .EQ. 0) GO TO 32
      DO 30 NR = 1,N1
        IF (IPIV(NR) .EQ. NR) GO TO 10
        J = IPIV(NR)
        XX = Y(NR)
        Y(NR) = Y(J)
        Y(J) = XX
 10     KK = MINO(N-NR,ML)
        DO 20 I = 1,KK
 20       Y(NR+I) = Y(NR+I) + Y(NR)*B(NR,LL+I)
 30     CONTINUE
 32   LL = LL - 1
      Y(N) = Y(N)*B(N,1)
      KK = 0
      DO 50 NB = 1,N1
        NR = N - NB
        IF (KK .NE. LL) KK = KK + 1
        DP = 0.
        IF (LL .EQ. 0) GO TO 50
        DO 40 I = 1,KK
 40       DP = DP + B(NR,I+1)*Y(NR+I)
 50     Y(NR) = (Y(NR) - DP)*B(NR,1)
      RETURN
 60   Y(1) = Y(1)*B(1,1)
      RETURN
      END
```

```
         S C O P E   2   L O A D   M A P                                        LOADER VERSION 1.0    03/19/78  14.26.46.  PAGE    1

PROGRAM  WILL BE ENTERED AT  MAIN        ( 54355)                               SCM LENGTH    101346          LCM LENGTH          0

         BLOCK          ADDRESS    LENGTH       FILE

          /TABSP/          110          4       CNEW
          /ENDPT/          114          2       CNEW
          /TABP/           116          2       CNEW
          /GEAR0/          120          5       CNEW
          /TABF/           125        145       CNEW
          /START/          272          1       CNEW
          /TABAB/          273          5       CNEW
          /MAIN/           300      51402       CNEW
          /TABIN/        51702         12       CNEW
          /TABW/         51714          3       CNEW
          /OUT/          51717        361       CNEW
          /OUTP/         52300       1704       CNEW
         MAIN            54204       1762       CNEW
          /TABRY/        56166          5       CNEW
          /TABUV/        56173          4       CNEW
          /TAB6/         56177         11       CNEW
         F               56210       1134       CNEW
         RT              57344        212       CNEW
         UINIT           57556        313       CNEW
         FLSP            60071       1332       CNEW
         BNDRY           61423        114       CNEW
          /DS/           61537          3       CNEW
         DECOMP          61542        207       CNEW
         SOLVE           61751        132       CNEW
         OUTPUT          62103          3       CNEW
          /GEAR1/        62106         12       CNEW
          /GEAR9/        62120         10       CNEW
          /OPTION/       62130          2       CNEW
          /SIZES/        62132          7       CNEW
          /ISTART/       62141         22       CNEW
          /IOUNIT/       62163          1       CNEW
         PDECOL          62164       1321       CNEW
         VALUES          63505        165       CNEW
         BLKDAT.         63672          0       CNEW
         INITAL          63672        312       CNEW
         COLPNT          64204        751       CNEW
         BSPLVD          65155       1110       CNEW
         BSPLVN          66265        135       CNEW
         INTERV          66422        116       CNEW
         STIFIB          66540       1407       CNEW
         GFUN            70147        451       CNEW
         EVAL            70620        107       CNEW
         DIFFUN          70727        137       CNEW
         ADDA            71066        213       CNEW
         RES             71301        264       CNEW
         PSETIB          71565        632       CNEW
         DIFFF           72417        302       CNEW
         INTERP          72721         76       CNEW
         COSET           73017        544       CNEW
         DECH            73563        255       CNEW
         SOLB            74040        166       CNEW
         DERIVF          74226          3       CNEW
```

```
S C O P E   2   L O A D   M A P                                                    LOADER VERSION 1.0    03/19/78  14.26.46.  PAGE   2
  /STP.END/     74231          1    SL-FTNLIB
  /FCL.C./      74232         23    SL-FTNLIB
  /Q8.IO./      74255        136    SL-FTNLIB
 Q8NTRY=        74413          1    SL-FTNLIB
 COMIO=         74414         44    SL-FTNLIB
 FECMSK=        74460         41    SL-FTNLIB
 FLTIN=         74521        156    SL-FTNLIB
 FLTOUT=        74677        315    SL-FTNLIB
 FMTAP=         75214        373    SL-FTNLIB
 FORSYS=        75607        533    SL-FTNLIB
 FORUTL=        76342         44    SL-FTNLIB
 GETFIT=        76406         43    SL-FTNLIB
 INCOM=         76451        256    SL-FTNLIB
 INPC=          76727        173    SL-FTNLIB
 KODER=         77122        467    SL-FTNLIB
 KRAKER=        77611        454    SL-FTNLIB
 OUTC=         100265        171    SL-FTNLIB
 OUTCOM=       100456        204    SL-FTNLIB
 CLOCK=        100662         43    SL-FTNLIB
 GOTOER=       100725         14    SL-FTNLIB
 ALOG          100741         77    SL-FTNLIB
 EXP           101040        100    SL-FTNLIB
 SQRT          101140         46    SL-FTNLIB
 SYSAID=       101206          1    SL-FTNLIB
 SYS=1ST       101207         62    SL-FTNLIB
 XTOY=         101271         55    SL-FTNLIB

 PRESSURE =    1.0000E+00  ATM.


 INITIAL DENSITY =  1.4625E-03

6.0840E+00  7.9205E+00  7.5000E+00
0.          3.0613E+00  6.1226E+00  8.8774E+00  1.1939E+01  1.5000E+01
0.          1.0204E+00  2.0409E+00  3.0613E+00  3.5715E+00  4.0818E+00  4.5920E+00  5.1022E+00  5.6124E+00  6.1226E+00
6.3522E+00  6.5818E+00  6.8113E+00  7.0409E+00  7.2704E+00  7.5000E+00  7.7296E+00  7.9591E+00  8.1887E+00  8.4183E+00
8.6478E+00  8.8774E+00  9.3876E+00  9.8978E+00  1.0408E+01  1.0918E+01  1.1428E+01  1.1939E+01  1.2959E+01  1.3980E+01
1.5000E+01
0.          2.5511E-01  5.1022E-01  7.6533E-01  1.0204E+00  1.2755E+00  1.5307E+00  1.7858E+00  2.0409E+00  2.2960E+00
2.5511E+00  2.8062E+00  3.0613E+00  3.1889E+00  3.3164E+00  3.4440E+00  3.5715E+00  3.6991E+00  3.8266E+00  3.9542E+00
4.0818E+00  4.2093E+00  4.3369E+00  4.4644E+00  4.5920E+00  4.7195E+00  4.8471E+00  4.9746E+00  5.1022E+00  5.2297E+00
5.3573E+00  5.4849E+00  5.6124E+00  5.7400E+00  5.8675E+00  5.9951E+00  6.1226E+00  6.1800E+00  6.2374E+00  6.2948E+00
6.3522E+00  6.4096E+00  6.4670E+00  6.5244E+00  6.5818E+00  6.6391E+00  6.6965E+00  6.7539E+00  6.8113E+00  6.8687E+00
6.9261E+00  6.9835E+00  7.0409E+00  7.0983E+00  7.1557E+00  7.2130E+00  7.2704E+00  7.3278E+00  7.3852E+00  7.4426E+00
7.5000E+00  7.5574E+00  7.6148E+00  7.6722E+00  7.7296E+00  7.7870E+00  7.8443E+00  7.9017E+00  7.9591E+00  8.0165E+00
8.0739E+00  8.1313E+00  8.1887E+00  8.2461E+00  8.3035E+00  8.3609E+00  8.4183E+00  8.4756E+00  8.5330E+00  8.5904E+00
8.6478E+00  8.7052E+00  8.7626E+00  8.8200E+00  8.8774E+00  9.0049E+00  9.1325E+00  9.2600E+00  9.3876E+00  9.5151E+00
9.6427E+00  9.7703E+00  9.8978E+00  1.0025E+01  1.0153E+01  1.0280E+01  1.0408E+01  1.0536E+01  1.0663E+01  1.0791E+01
1.0918E+01  1.1046E+01  1.1173E+01  1.1301E+01  1.1428E+01  1.1556E+01  1.1684E+01  1.1811E+01  1.1939E+01  1.2194E+01
1.2449E+01  1.2704E+01  1.2959E+01  1.3214E+01  1.3469E+01  1.3724E+01  1.3980E+01  1.4235E+01  1.4490E+01  1.4745E+01
1.5000E+01

      T =  3.0000E-07    DT =  3.0000E-06    TOTAL STEPS =       4


      RUN TIME =  8.3400E-01


      TEMP/1000
```

3.0000E-01	3.0002E-01	3.0006E-01	3.0012E-01	3.0019E-01	3.0029E-01	3.0041E-01	3.0058E-01	3.0081E-01	3.0110E-01
3.0149E-01	3.0202E-01	3.0274E-01	3.0318E-01	3.0369E-01	3.0427E-01	3.0495E-01	3.0574E-01	3.0665E-01	3.0770E-01
3.0892E-01	3.1033E-01	3.1196E-01	3.1385E-01	3.1605E-01	3.1859E-01	3.2153E-01	3.2495E-01	3.2893E-01	3.3355E-01
3.3893E-01	3.4522E-01	3.5256E-01	3.6110E-01	3.7113E-01	3.8293E-01	3.9684E-01	4.0387E-01	4.1144E-01	4.1962E-01
4.2846E-01	4.3802E-01	4.4835E-01	4.5953E-01	4.7163E-01	4.8475E-01	4.9900E-01	5.1450E-01	5.3137E-01	5.4972E-01
5.6970E-01	5.9150E-01	6.1529E-01	6.4126E-01	6.6957E-01	7.0032E-01	7.3363E-01	7.6960E-01	8.0790E-01	8.4801E-01
8.8940E-01	9.3118E-01	9.7206E-01	1.0108E+00	1.0463E+00	1.0774E+00	1.1038E+00	1.1259E+00	1.1437E+00	1.1578E+00
1.1686E+00	1.1768E+00	1.1831E+00	1.1879E+00	1.1915E+00	1.1942E+00	1.1963E+00	1.1979E+00	1.1992E+00	1.2003E+00
1.2011E+00	1.2018E+00	1.2024E+00	1.2029E+00	1.2034E+00	1.2043E+00	1.2051E+00	1.2059E+00	1.2066E+00	1.2073E+00
1.2080E+00	1.2086E+00	1.2092E+00	1.2098E+00	1.2104E+00	1.2110E+00	1.2116E+00	1.2121E+00	1.2126E+00	1.2131E+00
1.2137E+00	1.2142E+00	1.2147E+00	1.2151E+00	1.2156E+00	1.2161E+00	1.2165E+00	1.2169E+00	1.2174E+00	1.2182E+00
1.2190E+00	1.2198E+00	1.2205E+00	1.2212E+00	1.2219E+00	1.2225E+00	1.2231E+00	1.2236E+00	1.2240E+00	1.2244E+00
1.2245E+00									

O

1.0000E-10	4.5652E-11	1.8752E-11	8.8288E-12	5.4092E-12	2.6290E-12	1.2618E-12	7.9369E-13	7.1030E-13	5.9213E-13
7.4202E-13	1.7294E-12	4.1235E-12	6.0895E-12	9.3550E-12	1.4903E-11	2.3715E-11	3.6523E-11	5.7490E-11	9.2753E-11
1.4845E-10	2.2926E-10	3.6058E-10	5.7957E-10	9.2341E-10	1.4201E-09	2.2203E-09	3.5454E-09	5.6165E-09	8.5785E-09
1.3356E-08	2.1339E-08	3.3918E-08	5.1387E-08	8.0927E-08	1.3426E-07	2.2309E-07	2.7887E-07	3.5117E-07	4.4660E-07
5.7177E-07	7.3317E-07	9.5078E-07	1.2515E-06	1.6620E-06	2.2067E-06	2.9809E-06	4.1185E-06	5.7534E-06	7.9936E-06
1.1357E-05	1.6592E-05	2.4448E-05	3.5629E-05	5.2900E-05	8.0096E-05	1.2105E-04	1.8096E-04	2.6827E-04	3.9167E-04
5.5986E-04	7.8242E-04	1.0546E-03	1.3638E-03	1.6973E-03	2.0346E-03	2.3553E-03	2.6472E-03	2.8978E-03	3.1004E-03
3.2598E-03	3.3817E-03	3.4714E-03	3.5351E-03	3.5776E-03	3.6039E-03	3.6174E-03	3.6221E-03	3.6198E-03	3.6120E-03
3.6003E-03	3.5859E-03	3.5696E-03	3.5517E-03	3.5327E-03	3.4888E-03	3.4437E-03	3.3982E-03	3.3533E-03	3.3092E-03
3.2661E-03	3.2240E-03	3.1830E-03	3.1430E-03	3.1037E-03	3.0653E-03	3.0280E-03	2.9916E-03	2.9558E-03	2.9209E-03
2.8869E-03	2.8537E-03	2.8212E-03	2.7895E-03	2.7583E-03	2.7278E-03	2.6981E-03	2.6691E-03	2.6406E-03	2.5859E-03
2.5329E-03	2.4828E-03	2.4353E-03	2.3902E-03	2.3478E-03	2.3084E-03	2.2725E-03	2.2403E-03	2.2131E-03	2.1925E-03
2.1802E-03									

O3

3.3333E-01	3.3333E-01	3.3333E-01	3.3333E-01	3.3333E-01	3.3332E-01	3.3331E-01	3.3330E-01	3.3329E-01	3.3327E-01
3.3324E-01	3.3321E-01	3.3315E-01	3.3311E-01	3.3306E-01	3.3301E-01	3.3295E-01	3.3288E-01	3.3279E-01	3.3268E-01
3.3255E-01	3.3239E-01	3.3221E-01	3.3199E-01	3.3172E-01	3.3141E-01	3.3102E-01	3.3056E-01	3.3000E-01	3.2934E-01
3.2853E-01	3.2755E-01	3.2635E-01	3.2491E-01	3.2315E-01	3.2099E-01	3.1832E-01	3.1692E-01	3.1539E-01	3.1370E-01
3.1184E-01	3.0978E-01	3.0749E-01	3.0496E-01	3.0216E-01	2.9906E-01	2.9559E-01	2.9172E-01	2.8739E-01	2.8255E-01
2.7712E-01	2.7101E-01	2.6412E-01	2.5635E-01	2.4756E-01	2.3762E-01	2.2637E-01	2.1366E-01	1.9945E-01	1.8378E-01
1.6667E-01	1.4829E-01	1.2923E-01	1.1016E-01	9.1776E-02	7.4854E-02	5.9831E-02	4.6854E-02	3.6072E-02	2.7461E-02
2.0698E-02	1.5482E-02	1.1516E-02	8.5093E-03	6.2629E-03	4.6164E-03	3.4089E-03	2.5043E-03	1.8380E-03	1.3569E-03
1.0078E-03	7.4732E-04	5.5676E-04	4.2047E-04	3.2276E-04	1.7916E-04	1.0371E-04	7.1701E-05	5.8404E-05	4.8887E-05
4.3662E-05	4.1179E-05	3.9887E-05	3.8845E-05	3.8082E-05	3.7503E-05	3.7011E-05	3.6545E-05	3.6105E-05	3.5685E-05
3.5280E-05	3.4886E-05	3.4501E-05	3.4125E-05	3.3755E-05	3.3394E-05	3.3042E-05	3.2697E-05	3.2359E-05	3.1704E-05
3.1076E-05	3.0477E-05	2.9906E-05	2.9362E-05	2.8847E-05	2.8365E-05	2.7923E-05	2.7522E-05	2.7178E-05	2.6917E-05
2.6765E-05									

O2

6.6667E-01	6.6667E-01	6.6667E-01	6.6667E-01	6.6667E-01	6.6668E-01	6.6669E-01	6.6670E-01	6.6671E-01	6.6673E-01
6.6676E-01	6.6679E-01	6.6685E-01	6.6689E-01	6.6694E-01	6.6699E-01	6.6705E-01	6.6712E-01	6.6721E-01	6.6732E-01
6.6745E-01	6.6761E-01	6.6779E-01	6.6801E-01	6.6828E-01	6.6859E-01	6.6898E-01	6.6944E-01	6.7000E-01	6.7066E-01
6.7147E-01	6.7245E-01	6.7365E-01	6.7509E-01	6.7685E-01	6.7901E-01	6.8168E-01	6.8308E-01	6.8461E-01	6.8630E-01
6.8816E-01	6.9022E-01	6.9251E-01	6.9504E-01	6.9783E-01	7.0094E-01	7.0440E-01	7.0828E-01	7.1260E-01	7.1744E-01
7.2287E-01	7.2898E-01	7.3586E-01	7.4362E-01	7.5239E-01	7.6230E-01	7.7351E-01	7.8616E-01	8.0028E-01	8.1583E-01
8.3277E-01	8.5093E-01	8.6972E-01	8.8847E-01	9.0653E-01	9.2311E-01	9.3781E-01	9.5050E-01	9.6103E-01	9.6944E-01
9.7604E-01	9.8114E-01	9.8501E-01	9.8796E-01	9.9016E-01	9.9178E-01	9.9297E-01	9.9387E-01	9.9454E-01	9.9503E-01
9.9539E-01	9.9567E-01	9.9587E-01	9.9603E-01	9.9614E-01	9.9633E-01	9.9645E-01	9.9653E-01	9.9659E-01	9.9664E-01
9.9669E-01	9.9673E-01	9.9678E-01	9.9682E-01	9.9686E-01	9.9690E-01	9.9694E-01	9.9697E-01	9.9701E-01	9.9704E-01
9.9708E-01	9.9711E-01	9.9714E-01	9.9718E-01	9.9721E-01	9.9724E-01	9.9727E-01	9.9730E-01	9.9733E-01	9.9738E-01
9.9744E-01	9.9749E-01	9.9753E-01	9.9758E-01	9.9762E-01	9.9766E-01	9.9770E-01	9.9773E-01	9.9776E-01	9.9778E-01
9.9779E-01									

```
INTEGRALS =  3.0813E-02  2.0422E-04 -3.1017E-02
DIFF TERMS =  5.8317E-07 -2.3712E-06  1.7880E-06
FLAME SPEED =  6.3628E+01  6.3302E+01  6.3625E+01
       FLAME FRONT FROM PHI* =  6.0839E+00  TO PHI* =  7.9205E+00

       FLAME FRONT FROM X =  2.1707E-02  TO X =  3.6405E-02

       FLAME THICKNESS =  1.4698E-02  CM


       T =  6.0000E-01    DT =  8.4758E-01    TOTAL STEPS =      25

       RUN TIME =  3.7250E+00

INTEGRALS =  3.0814E-02  2.0459E-04 -3.1018E-02
DIFF TERMS =  2.1225E-06 -2.3712E-06  2.4864E-07
FLAME SPEED =  6.3632E+01  6.3415E+01  6.3631E+01
       FLAME FRONT FROM PHI* =  6.0840E+00  TO PHI* =  7.9205E+00

       FLAME FRONT FROM X =  2.1707E-02  TO X =  3.6404E-02

       FLAME THICKNESS =  1.4697E-02  CM

   MO = -9.3061E-02      FLSP = -6.3631E+01

   ORIGIN SPEED = -9.3060E-01      CHANGE =  3.0913E-05


       T =  3.0000E+00    DT =  8.4758E-01    TOTAL STEPS =      27

       RUN TIME =  3.8410E+00

INTEGRALS =  3.0814E-02  2.0458E-04 -3.1018E-02
DIFF TERMS =  2.1224E-06 -2.3711E-06  2.4874E-07
FLAME SPEED =  6.3632E+01  6.3413E+01  6.3631E+01
       FLAME FRONT FROM PHI* =  6.0840E+00  TO PHI* =  7.9205E+00

       FLAME FRONT FROM X =  2.1707E-02  TO X =  3.6405E-02

       FLAME THICKNESS =  1.4698E-02  CM
```

```
 MO = -9.3063E-02       FLSP = -6.3632E+01

 ORIGIN SPEED = -9.3065E-01       CHANGE = -4.9660E-05

      T =  9.0000E+00    DT =  2.5639E+00    TOTAL STEPS =       30

      RUN TIME =  4.3830E+00

INTEGRALS =  3.0814E-02  2.0458E-04 -3.1018E-02
DIFF TERMS =  2.1237E-06 -2.3712E-06  2.4742E-07
FLAME SPEED =  6.3632E+01  6.3411E+01  6.3631E+01
      FLAME FRONT FROM PHI* =  6.0842E+00  TO PHI* =  7.9206E+00

      FLAME FRONT FROM X =  2.1707E-02  TO X =  3.6405E-02

      FLAME THICKNESS =  1.4698E-02  CM

 MO = -9.3061E-02       FLSP = -6.3630E+01

 ORIGIN SPEED = -9.3058E-01       CHANGE =  6.6552E-05

      T =  1.5000E+01    DT =  5.5397E-01    TOTAL STEPS =       43

      RUN TIME =  6.3060E+00

INTEGRALS =  3.0814E-02  2.0458E-04 -3.1018E-02
DIFF TERMS =  2.1239E-06 -2.3712E-06  2.4725E-07
FLAME SPEED =  6.3632E+01  6.3411E+01  6.3631E+01
      FLAME FRONT FROM PHI* =  6.0840E+00  TO PHI* =  7.9205E+00

      FLAME FRONT FROM X =  2.1707E-02  TO X =  3.6405E-02

      FLAME THICKNESS =  1.4698E-02  CM

 MO = -9.3062E-02       FLSP = -6.3631E+01

 ORIGIN SPEED = -9.3063E-01       CHANGE = -4.8309E-05
```

T = 2.1000E+01 DT = 3.1632E+00 TOTAL STEPS = 46

RUN TIME = 6.8140E+00

INTEGRALS = 3.0814E-02 2.0458E-04 -3.1018E-02

DIFF TERMS = 2.1239E-06 -2.3712E-06 2.4724E-07

FLAME SPEED = 6.3632E+01 6.3411E+01 6.3631E+01

FLAME FRONT FROM PHI* = 6.0840E+00 TO PHI* = 7.9205E+00

FLAME FRONT FROM X = 2.1707E-02 TO X = 3.6405E-02

FLAME THICKNESS = 1.4698E-02 CM

MO = -9.3061E-02 FLSP = -6.3630E+01

ORIGIN SPEED = -9.3060E-01 CHANGE = 3.2170E-05

T = 2.7000E+01 DT = 3.1632E+00 TOTAL STEPS = 46

RUN TIME = 6.9980E+00

TEMP/1000

3.0000E-01	3.0003E-01	3.0007E-01	3.0012E-01	3.0019E-01	3.0029E-01	3.0041E-01	3.0058E-01	3.0080E-01	3.0110E-01
3.0149E-01	3.0203E-01	3.0274E-01	3.0318E-01	3.0369E-01	3.0428E-01	3.0496E-01	3.0574E-01	3.0665E-01	3.0770E-01
3.0892E-01	3.1033E-01	3.1196E-01	3.1365E-01	3.1605E-01	3.1858E-01	3.2153E-01	3.2490E-01	3.2894E-01	3.3356E-01
3.3893E-01	3.4522E-01	3.5256E-01	3.6111E-01	3.7113E-01	3.8294E-01	3.9684E-01	4.0387E-01	4.1146E-01	4.1964E-01
4.2848E-01	4.3803E-01	4.4835E-01	4.5953E-01	4.7165E-01	4.8478E-01	4.9903E-01	5.1453E-01	5.3139E-01	5.4973E-01
5.6973E-01	5.9154E-01	6.1533E-01	6.4131E-01	6.6961E-01	7.0035E-01	7.3364E-01	7.6962E-01	8.0795E-01	8.4808E-01
8.8945E-01	9.3123E-01	9.7212E-01	1.0109E+00	1.0464E+00	1.0774E+00	1.1039E+00	1.1259E+00	1.1437E+00	1.1577E+00
1.1686E+00	1.1768E+00	1.1831E+00	1.1879E+00	1.1915E+00	1.1943E+00	1.1963E+00	1.1980E+00	1.1993E+00	1.2003E+00
1.2011E+00	1.2018E+00	1.2024E+00	1.2029E+00	1.2034E+00	1.2043E+00	1.2052E+00	1.2059E+00	1.2066E+00	1.2073E+00
1.2079E+00	1.2086E+00	1.2092E+00	1.2098E+00	1.2104E+00	1.2110E+00	1.2115E+00	1.2121E+00	1.2126E+00	1.2131E+00
1.2137E+00	1.2142E+00	1.2146E+00	1.2151E+00	1.2156E+00	1.2160E+00	1.2165E+00	1.2169E+00	1.2174E+00	1.2182E+00
1.2190E+00	1.2198E+00	1.2205E+00	1.2212E+00	1.2219E+00	1.2225E+00	1.2231E+00	1.2236E+00	1.2241E+00	1.2244E+00
1.2245E+00									

O

1.0000E-10	4.5653E-11	1.6753E-11	8.8288E-12	5.4089E-12	2.6267E-12	1.2618E-12	7.9369E-13	7.1029E-13	5.9214E-13
7.4223E-13	1.7299E-12	4.1244E-12	6.0896E-12	9.3545E-12	1.4405E-11	2.3726E-11	3.6535E-11	5.7493E-11	9.2762E-11
1.4850E-10	2.2933E-10	3.6061E-10	5.7958E-10	9.2349E-10	1.4206E-09	2.2214E-09	3.5467E-09	5.6171E-09	8.5807E-09
1.3362E-08	2.1346E-08	3.3919E-08	5.1394E-08	8.0956E-08	1.3430E-07	2.2311E-07	2.7890E-07	3.5126E-07	4.4677E-07
5.7204E-07	7.3339E-07	9.5085E-07	1.2515E-06	1.6626E-06	2.2080E-06	2.9827E-06	4.1203E-06	5.7545E-06	7.9957E-06
1.1363E-05	1.6603E-05	2.4461E-05	3.5650E-05	5.2931E-05	8.0122E-05	1.2104E-04	1.8097E-04	2.6838E-04	3.9188E-04
5.6009E-04	7.8269E-04	1.0551E-03	1.3645E-03	1.6980E-03	2.0351E-03	2.3556E-03	2.6475E-03	2.8980E-03	3.1005E-03
3.2600E-03	3.3819E-03	3.4716E-03	3.5352E-03	3.5779E-03	3.6039E-03	3.6174E-03	3.6221E-03	3.6198E-03	3.6120E-03
3.6002E-03	3.5859E-03	3.5695E-03	3.5516E-03	3.5327E-03	3.4888E-03	3.4437E-03	3.3982E-03	3.3532E-03	3.3092E-03
3.2662E-03	3.2241E-03	3.1829E-03	3.1428E-03	3.1036E-03	3.0653E-03	3.0280E-03	2.9915E-03	2.9558E-03	2.9210E-03

2.8869E-03	2.8537E-03	2.8211E-03	2.7893E-03	2.7582E-03	2.7278E-03	2.6981E-03	2.6690E-03	2.6406E-03	2.5855E-03
2.5330E-03	2.4829E-03	2.4353E-03	2.3902E-03	2.3477E-03	2.3084E-03	2.2724E-03	2.2403E-03	2.2131E-03	2.1926E-03
2.1803E-03									

O3

3.3333E-01	3.3333E-01	3.3333E-01	3.3333E-01	3.3332E-01	3.3332E-01	3.3331E-01	3.3330E-01	3.3329E-01	3.3327E-01
3.3324E-01	3.3320E-01	3.3315E-01	3.3311E-01	3.3307E-01	3.3301E-01	3.3295E-01	3.3288E-01	3.3279E-01	3.3268E-01
3.3255E-01	3.3239E-01	3.3221E-01	3.3199E-01	3.3172E-01	3.3140E-01	3.3102E-01	3.3056E-01	3.3000E-01	3.2934E-01
3.2853E-01	3.2755E-01	3.2635E-01	3.2491E-01	3.2315E-01	3.2098E-01	3.1831E-01	3.1692E-01	3.1539E-01	3.1370E-01
3.1183E-01	3.0977E-01	3.0749E-01	3.0496E-01	3.0216E-01	2.9905E-01	2.9558E-01	2.9171E-01	2.8738E-01	2.8255E-01
2.7711E-01	2.7100E-01	2.6410E-01	2.5633E-01	2.4754E-01	2.3760E-01	2.2636E-01	2.1366E-01	1.9944E-01	1.8375E-01
1.6665E-01	1.4827E-01	1.2920E-01	1.1012E-01	9.1735E-02	7.4827E-02	5.9812E-02	4.6840E-02	3.6063E-02	2.7455E-02
2.0690E-02	1.5474E-02	1.1510E-02	8.5061E-03	6.2614E-03	4.6153E-03	3.4074E-03	2.5025E-03	1.8367E-03	1.3563E-03
1.0076E-03	7.4709E-04	5.5651E-04	4.2021E-04	3.2255E-04	1.7910E-04	1.0370E-04	7.1679E-05	5.6384E-05	4.8865E-05
4.3664E-05	4.1178E-05	3.9883E-05	3.8841E-05	3.8080E-05	3.7503E-05	3.7011E-05	3.6545E-05	3.6105E-05	3.5686E-05
3.5281E-05	3.4886E-05	3.4500E-05	3.4124E-05	3.3755E-05	3.3395E-05	3.3042E-05	3.2697E-05	3.2359E-05	3.1704E-05
3.1077E-05	3.0478E-05	2.9906E-05	2.9361E-05	2.8846E-05	2.8365E-05	2.7923E-05	2.7522E-05	2.7179E-05	2.6918E-05
2.6766E-05									

O2

6.6667E-01	6.6667E-01	6.6667E-01	6.6667E-01	6.6668E-01	6.6668E-01	6.6669E-01	6.6670E-01	6.6671E-01	6.6673E-01
6.6676E-01	6.6680E-01	6.6685E-01	6.6689E-01	6.6693E-01	6.6699E-01	6.6705E-01	6.6712E-01	6.6721E-01	6.6732E-01
6.6745E-01	6.6761E-01	6.6779E-01	6.6801E-01	6.6828E-01	6.6860E-01	6.6898E-01	6.6944E-01	6.7000E-01	6.7066E-01
6.7147E-01	6.7245E-01	6.7365E-01	6.7509E-01	6.7685E-01	6.7902E-01	6.8169E-01	6.8308E-01	6.8461E-01	6.8630E-01
6.8817E-01	6.9023E-01	6.9251E-01	6.9503E-01	6.9784E-01	7.0095E-01	7.0442E-01	7.0829E-01	7.1261E-01	7.1745E-01
7.2287E-01	7.2899E-01	7.3587E-01	7.4363E-01	7.5240E-01	7.6232E-01	7.7351E-01	7.8616E-01	8.0029E-01	8.1585E-01
8.3279E-01	8.5095E-01	8.6975E-01	8.8852E-01	9.0657E-01	9.2314E-01	9.3783E-01	9.5051E-01	9.6104E-01	9.6944E-01
9.7605E-01	9.8114E-01	9.8502E-01	9.8796E-01	9.9016E-01	9.9178E-01	9.9298E-01	9.9388E-01	9.9454E-01	9.9503E-01
9.9539E-01	9.9567E-01	9.9587E-01	9.9603E-01	9.9614E-01	9.9633E-01	9.9645E-01	9.9653E-01	9.9659E-01	9.9664E-01
9.9669E-01	9.9673E-01	9.9678E-01	9.9682E-01	9.9686E-01	9.9690E-01	9.9694E-01	9.9697E-01	9.9701E-01	9.9704E-01
9.9706E-01	9.9711E-01	9.9714E-01	9.9718E-01	9.9721E-01	9.9724E-01	9.9727E-01	9.9730E-01	9.9733E-01	9.9738E-01
9.9744E-01	9.9749E-01	9.9753E-01	9.9758E-01	9.9762E-01	9.9766E-01	9.9770E-01	9.9773E-01	9.9776E-01	9.9778E-01
9.9779E-01									

INTEGRALS = 3.0814E-02 2.0457E-04 -3.1018E-02

DIFF TERMS = 2.1239E-06 -2.3712E-06 2.4728E-07

FLAME SPEED = 6.3632E+01 6.3411E+01 6.3631E+01

X IN CM

0.	8.7220E-04	1.7445E-03	2.6169E-03	3.4895E-03	4.3624E-03	5.2356E-03	6.1091E-03	6.9833E-03	7.8582E-03
8.7342E-03	9.6114E-03	1.0491E-02	1.0931E-02	1.1372E-02	1.1814E-02	1.2257E-02	1.2701E-02	1.3146E-02	1.3592E-02
1.4041E-02	1.4491E-02	1.4943E-02	1.5398E-02	1.5856E-02	1.6318E-02	1.6783E-02	1.7254E-02	1.7729E-02	1.8212E-02
1.8701E-02	1.9199E-02	1.9707E-02	2.0228E-02	2.0761E-02	2.1312E-02	2.1881E-02	2.2144E-02	2.2413E-02	2.2686E-02
2.2966E-02	2.3252E-02	2.3544E-02	2.3844E-02	2.4152E-02	2.4468E-02	2.4794E-02	2.5131E-02	2.5478E-02	2.5838E-02
2.6211E-02	2.6600E-02	2.7004E-02	2.7426E-02	2.7868E-02	2.8331E-02	2.8818E-02	2.9330E-02	2.9870E-02	3.0441E-02
3.1043E-02	3.1678E-02	3.2347E-02	3.3048E-02	3.3761E-02	3.4542E-02	3.5329E-02	3.6137E-02	3.6963E-02	3.7803E-02
3.8655E-02	3.9516E-02	4.0383E-02	4.1255E-02	4.2132E-02	4.3011E-02	4.3893E-02	4.4776E-02	4.5660E-02	4.6546E-02
4.7432E-02	4.8319E-02	4.9206E-02	5.0094E-02	5.0983E-02	5.2958E-02	5.4935E-02	5.6913E-02	5.8892E-02	6.0872E-02
6.2854E-02	6.4836E-02	6.6819E-02	6.8803E-02	7.0788E-02	7.2774E-02	7.4761E-02	7.6748E-02	7.8737E-02	8.0726E-02
8.2716E-02	8.4707E-02	8.6698E-02	8.8691E-02	9.0684E-02	9.2677E-02	9.4672E-02	9.6667E-02	9.8662E-02	1.0266E-01
1.0665E-01	1.1065E-01	1.1465E-01	1.1865E-01	1.2266E-01	1.2666E-01	1.3067E-01	1.3468E-01	1.3869E-01	1.4271E-01
1.4672E-01									

FLAME FRONT FROM PHI* = 6.0639E+00 TO PHI* = 7.9204E+00

FLAME FRONT FROM X = 2.1707E-02 TO X = 3.6404E-02

FLAME THICKNESS = 1.4698E-02 CM

MO = -9.3063E-02 FLSP = -6.3632E+01

ORIGIN SPEED = -9.3067E-01 CHANGE = -7.2119E-05

T = 3.0000E+01 DT = 1.4258E+00 TOTAL STEPS = 50

RUN TIME = 7.5690E+00

TEMP/1000

3.0000E-01	3.0003E-01	3.0007E-01	3.0012E-01	3.0019E-01	3.0029E-01	3.0041E-01	3.0058E-01	3.0080E-01	3.0110E-01
3.0149E-01	3.0203E-01	3.0274E-01	3.0318E-01	3.0369E-01	3.0428E-01	3.0496E-01	3.0574E-01	3.0665E-01	3.0770E-01
3.0892E-01	3.1033E-01	3.1196E-01	3.1385E-01	3.1605E-01	3.1858E-01	3.2153E-01	3.2495E-01	3.2894E-01	3.3355E-01
3.3893E-01	3.4522E-01	3.5256E-01	3.6111E-01	3.7113E-01	3.8293E-01	3.9684E-01	4.0387E-01	4.1145E-01	4.1964E-01
4.2848E-01	4.3802E-01	4.4835E-01	4.5953E-01	4.7164E-01	4.8477E-01	4.9902E-01	5.1452E-01	5.3138E-01	5.4972E-01
5.6972E-01	5.9153E-01	6.1532E-01	6.4129E-01	6.6959E-01	7.0033E-01	7.3363E-01	7.6960E-01	8.0793E-01	8.4806E-01
8.8943E-01	9.3121E-01	9.7211E-01	1.0109E+00	1.0464E+00	1.0774E+00	1.1039E+00	1.1259E+00	1.1437E+00	1.1577E+00
1.1686E+00	1.1768E+00	1.1831E+00	1.1879E+00	1.1915E+00	1.1943E+00	1.1963E+00	1.1980E+00	1.1993E+00	1.2003E+00
1.2011E+00	1.2018E+00	1.2024E+00	1.2029E+00	1.2034E+00	1.2043E+00	1.2052E+00	1.2059E+00	1.2066E+00	1.2073E+00
1.2079E+00	1.2086E+00	1.2092E+00	1.2098E+00	1.2104E+00	1.2110E+00	1.2115E+00	1.2121E+00	1.2126E+00	1.2131E+00
1.2137E+00	1.2142E+00	1.2146E+00	1.2151E+00	1.2156E+00	1.2160E+00	1.2165E+00	1.2169E+00	1.2174E+00	1.2182E+00
1.2190E+00	1.2198E+00	1.2205E+00	1.2212E+00	1.2219E+00	1.2225E+00	1.2231E+00	1.2236E+00	1.2241E+00	1.2244E+00
1.2245E+00									

O

1.0000E-10	4.5653E-11	1.8753E-11	8.8290E-12	5.4090E-12	2.6288E-12	1.2618E-12	7.9371E-13	7.1029E-13	5.9213E-13
7.4219E-13	1.7297E-12	4.1240E-12	6.0890E-12	9.3536E-12	1.4403E-11	2.3724E-11	3.6531E-11	5.7488E-11	9.2753E-11
1.4848E-10	2.2931E-10	3.6057E-10	5.7952E-10	9.2340E-10	1.4204E-09	2.2212E-09	3.5464E-09	5.6165E-09	8.5798E-09
1.3361E-08	2.1344E-08	3.3915E-08	5.1388E-08	8.0947E-08	1.3428E-07	2.2309E-07	2.7867E-07	3.5122E-07	4.4672E-07
5.7197E-07	7.3330E-07	9.5073E-07	1.2514E-06	1.6623E-06	2.2076E-06	2.9822E-06	4.1197E-06	5.7535E-06	7.9943E-06
1.1361E-05	1.6600E-05	2.4457E-05	3.5643E-05	5.2920E-05	8.0106E-05	1.2102E-04	1.8094E-04	2.6833E-04	3.9181E-04
5.5999E-04	7.8257E-04	1.0550E-03	1.3643E-03	1.6978E-03	2.0350E-03	2.3556E-03	2.6474E-03	2.8979E-03	3.1005E-03
3.2600E-03	3.3819E-03	3.4716E-03	3.5351E-03	3.5776E-03	3.6034E-03	3.6174E-03	3.6221E-03	3.6198E-03	3.6120E-03
3.6002E-03	3.5859E-03	3.5696E-03	3.5516E-03	3.5327E-03	3.4888E-03	3.4437E-03	3.3982E-03	3.3532E-03	3.3092E-03
3.2662E-03	3.2241E-03	3.1829E-03	3.1428E-03	3.1036E-03	3.0653E-03	3.0280E-03	2.9915E-03	2.9558E-03	2.9210E-03
2.8869E-03	2.8537E-03	2.8211E-03	2.7893E-03	2.7582E-03	2.7278E-03	2.6981E-03	2.6690E-03	2.6406E-03	2.5856E-03
2.5330E-03	2.4829E-03	2.4353E-03	2.3902E-03	2.3477E-03	2.3084E-03	2.2725E-03	2.2403E-03	2.2131E-03	2.1926E-03
2.1803E-03									

O3

3.3333E-01	3.3333E-01	3.3333E-01	3.3333E-01	3.3332E-01	3.3332E-01	3.3331E-01	3.3330E-01	3.3329E-01	3.3327E-01
3.3324E-01	3.3320E-01	3.3315E-01	3.3311E-01	3.3307E-01	3.3301E-01	3.3295E-01	3.3288E-01	3.3279E-01	3.3268E-01
3.3255E-01	3.3240E-01	3.3221E-01	3.3199E-01	3.3172E-01	3.3140E-01	3.3102E-01	3.3056E-01	3.3000E-01	3.2934E-01
3.2853E-01	3.2755E-01	3.2635E-01	3.2491E-01	3.2315E-01	3.2098E-01	3.1831E-01	3.1692E-01	3.1539E-01	3.1370E-01
3.1183E-01	3.0977E-01	3.0749E-01	3.0497E-01	3.0216E-01	2.9905E-01	2.9558E-01	2.9171E-01	2.8739E-01	2.8255E-01
2.7712E-01	2.7100E-01	2.6411E-01	2.5634E-01	2.4755E-01	2.3761E-01	2.2637E-01	2.1367E-01	1.9945E-01	1.8376E-01
1.6666E-01	1.4828E-01	1.2920E-01	1.1013E-01	9.1744E-02	7.4834E-02	5.9818E-02	4.6846E-02	3.6068E-02	2.7458E-02
2.0693E-02	1.5476E-02	1.1512E-02	8.5074E-03	6.2623E-03	4.6160E-03	3.4079E-03	2.5029E-03	1.8370E-03	1.3565E-03
1.0077E-03	7.4720E-04	5.5659E-04	4.2027E-04	3.2260E-04	1.7912E-04	1.0371E-04	7.1684E-05	5.8387E-05	4.8666E-05
4.3665E-05	4.1179E-05	3.9863E-05	3.8841E-05	3.8080E-05	3.7503E-05	3.7011E-05	3.6545E-05	3.6105E-05	3.5666E-05
3.5281E-05	3.4888E-05	3.4500E-05	3.4124E-05	3.3755E-05	3.3395E-05	3.3042E-05	3.2697E-05	3.2359E-05	3.1704E-05
3.1077E-05	3.0478E-05	2.9906E-05	2.9361E-05	2.8840E-05	2.8365E-05	2.7923E-05	2.7522E-05	2.7179E-05	2.6918E-05

```
2.6766E-05

      O2

6.6667E-01  6.6667E-01  6.6667E-01  6.6667E-01  6.6668E-01  6.6668E-01  6.6669E-01  6.6670E-01  6.6671E-01  6.6673E-01
6.6676E-01  6.6680E-01  6.6685E-01  6.6689E-01  6.6693E-01  6.6699E-01  6.6705E-01  6.6712E-01  6.6721E-01  6.6732E-01
6.6745E-01  6.6760E-01  6.6779E-01  6.6801E-01  6.6828E-01  6.6860E-01  6.6898E-01  6.6944E-01  6.7000E-01  6.7066E-01
6.7147E-01  6.7245E-01  6.7365E-01  6.7509E-01  6.7685E-01  6.7902E-01  6.8169E-01  6.8308E-01  6.8461E-01  6.8630E-01
6.8817E-01  6.9023E-01  6.9251E-01  6.9503E-01  6.9784E-01  7.0095E-01  7.0441E-01  7.0828E-01  7.1261E-01  7.1744E-01
7.2287E-01  7.2898E-01  7.3587E-01  7.4363E-01  7.5240E-01  7.6231E-01  7.7351E-01  7.8615E-01  8.0028E-01  8.1585E-01
8.3278E-01  8.5094E-01  8.6974E-01  8.8851E-01  9.0656E-01  9.2313E-01  9.3783E-01  9.5051E-01  9.6103E-01  9.6944E-01
9.7605E-01  9.8114E-01  9.8502E-01  9.8796E-01  9.9016E-01  9.9178E-01  9.9297E-01  9.9387E-01  9.9454E-01  9.9503E-01
9.9539E-01  9.9567E-01  9.9587E-01  9.9603E-01  9.9614E-01  9.9633E-01  9.9645E-01  9.9653E-01  9.9659E-01  9.9664E-01
9.9669E-01  9.9673E-01  9.9678E-01  9.9682E-01  9.9686E-01  9.9690E-01  9.9694E-01  9.9697E-01  9.9701E-01  9.9704E-01
9.9708E-01  9.9711E-01  9.9714E-01  9.9718E-01  9.9721E-01  9.9724E-01  9.9727E-01  9.9730E-01  9.9733E-01  9.9738E-01
9.9744E-01  9.9749E-01  9.9753E-01  9.9758E-01  9.9762E-01  9.9766E-01  9.9770E-01  9.9773E-01  9.9776E-01  9.9778E-01
9.9779E-01

INTEGRALS =  3.0814E-02  2.0458E-04 -3.1018E-02

DIFF TERMS =  2.1239E-06 -2.3712E-06  2.4726E-07

FLAME SPEED =  6.3632E+01  6.3411E+01  6.3631E+01

      X IN CM

0.          8.7220E-04  1.7445E-03  2.6169E-03  3.4895E-03  4.3624E-03  5.2356E-03  6.1091E-03  6.9833E-03  7.8582E-03
8.7342E-03  9.6114E-03  1.0491E-02  1.0931E-02  1.1372E-02  1.1814E-02  1.2257E-02  1.2701E-02  1.3146E-02  1.3592E-02
1.4041E-02  1.4491E-02  1.4943E-02  1.5398E-02  1.5856E-02  1.6318E-02  1.6783E-02  1.7254E-02  1.7729E-02  1.8212E-02
1.8701E-02  1.9199E-02  1.9707E-02  2.0228E-02  2.0761E-02  2.1312E-02  2.1881E-02  2.2144E-02  2.2413E-02  2.2686E-02
2.2966E-02  2.3252E-02  2.3544E-02  2.3844E-02  2.4152E-02  2.4468E-02  2.4794E-02  2.5131E-02  2.5478E-02  2.5838E-02
2.6211E-02  2.6599E-02  2.7004E-02  2.7426E-02  2.7868E-02  2.8331E-02  2.8817E-02  2.9330E-02  2.9870E-02  3.0441E-02
3.1043E-02  3.1678E-02  3.2346E-02  3.3048E-02  3.3781E-02  3.4542E-02  3.5328E-02  3.6136E-02  3.6962E-02  3.7803E-02
3.8655E-02  3.9515E-02  4.0383E-02  4.1255E-02  4.2131E-02  4.3011E-02  4.3892E-02  4.4775E-02  4.5660E-02  4.6545E-02
4.7432E-02  4.8319E-02  4.9206E-02  5.0094E-02  5.0982E-02  5.2958E-02  5.4935E-02  5.6913E-02  5.8892E-02  6.0872E-02
6.2853E-02  6.4836E-02  6.6819E-02  6.8803E-02  7.0788E-02  7.2774E-02  7.4761E-02  7.6748E-02  7.8737E-02  8.0726E-02
8.2716E-02  8.4707E-02  8.6698E-02  8.8690E-02  9.0683E-02  9.2677E-02  9.4671E-02  9.6666E-02  9.8662E-02  1.0266E-01
1.0665E-01  1.1065E-01  1.1465E-01  1.1865E-01  1.2266E-01  1.2666E-01  1.3067E-01  1.3468E-01  1.3869E-01  1.4271E-01
1.4672E-01

      FLAME FRONT FROM PHI* =  6.0840E+00  TO PHI* =  7.9205E+00


      FLAME FRONT FROM X =  2.1707E-02  TO X =  3.6405E-02


      FLAME THICKNESS =  1.4698E-02  CM


   MO = -9.3062E-02       FLSP = -6.3631E+01

 6.0840E+00  7.9205E+00  7.5000E+00
-9.3067E-01

      COLLOCATION POINTS

    62
     1  0.          3.0000E-01  1.0000E-10  3.3333E-01
     2  2.1564E-01  3.0002E-01  5.2002E-11  3.3333E-01
     3  8.0479E-01  3.0013E-01  8.1382E-12  3.3333E-01
     4  1.2361E+00  3.0027E-01  2.9536E-12  3.3332E-01
     5  1.9252E+00  3.0061E-01  7.6859E-13  3.3330E-01
     6  2.2565E+00  3.0105E-01  6.0720E-13  3.3327E-01
```

7	2.8457E+00	3.0212E-01	1.9938E-12	3.3320E-01
8	3.1691E+00	3.0311E-01	5.7249E-12	3.3312E-01
9	3.4637E+00	3.0437E-01	1.6029E-11	3.3300E-01
10	3.6794E+00	3.0561E-01	3.4172E-11	3.3289E-01
11	3.9739E+00	3.0788E-01	9.9881E-11	3.3260E-01
12	4.1896E+00	3.1010E-01	2.1444E-10	3.3242E-01
13	4.4841E+00	3.1417E-01	6.2361E-10	3.3195E-01
14	4.6998E+00	3.1817E-01	1.3292E-09	3.3146E-01
15	4.9944E+00	3.2553E-01	3.8124E-09	3.3048E-01
16	5.2100E+00	3.3279E-01	8.0374E-09	3.2945E-01
17	5.5046E+00	3.4628E-01	2.2954E-08	3.2738E-01
18	5.7202E+00	3.5970E-01	4.8189E-08	3.2515E-01
19	6.0148E+00	3.8494E-01	1.4541E-07	3.2061E-01
20	6.1711E+00	4.0275E-01	2.6932E-07	3.1714E-01
21	6.3037E+00	4.2096E-01	4.6399E-07	3.1342E-01
22	6.4007E+00	4.3650E-01	7.0535E-07	3.1010E-01
23	6.5332E+00	4.6134E-01	1.3071E-06	3.0455E-01
24	6.6303E+00	4.8267E-01	2.1116E-06	2.9955E-01
25	6.7628E+00	5.1704E-01	4.3366E-06	2.9107E-01
26	6.8598E+00	5.4678E-01	7.5928E-06	2.8333E-01
27	6.9924E+00	5.9507E-01	1.7625E-05	2.6999E-01
28	7.0894E+00	6.3713E-01	3.3611E-05	2.5760E-01
29	7.2219E+00	7.0531E-01	8.5442E-05	2.3596E-01
30	7.3189E+00	7.6387E-01	1.7009E-04	2.1573E-01
31	7.4515E+00	8.5439E-01	4.1469E-04	1.8121E-01
32	7.5485E+00	9.2478E-01	7.4456E-04	1.5119E-01
33	7.6811E+00	1.0166E+00	1.4147E-03	1.0722E-01
34	7.7781E+00	1.0729E+00	1.9836E-03	7.7330E-02
35	7.9106E+00	1.1290E+00	2.6891E-03	4.5032E-02
36	8.0076E+00	1.1558E+00	3.0721E-03	2.8662E-02
37	8.1402E+00	1.1779E+00	3.3977E-03	1.4788E-02
38	8.2372E+00	1.1872E+00	3.5268E-03	8.9185E-03
39	8.3697E+00	1.1946E+00	3.6067E-03	4.4045E-03
40	8.4668E+00	1.1977E+00	3.6219E-03	2.6259E-03
41	8.5993E+00	1.2004E+00	3.6104E-03	1.2953E-03
42	8.6963E+00	1.2017E+00	3.5863E-03	7.8254E-04
43	8.8289E+00	1.2030E+00	3.5488E-03	4.0303E-04
44	8.9852E+00	1.2042E+00	3.4957E-03	1.9625E-04
45	9.2798E+00	1.2060E+00	3.3912E-03	6.9023E-05
46	9.4954E+00	1.2072E+00	3.3160E-03	5.0030E-05
47	9.7900E+00	1.2087E+00	3.2176E-03	4.0939E-05
48	1.0006E+01	1.2097E+00	3.1469E-03	3.8981E-05
49	1.0300E+01	1.2111E+00	3.0595E-03	3.7424E-05
50	1.0516E+01	1.2120E+00	2.9971E-03	3.6615E-05
51	1.0810E+01	1.2132E+00	2.9157E-03	3.5623E-05
52	1.1026E+01	1.2141E+00	2.8588E-03	3.4947E-05
53	1.1321E+01	1.2152E+00	2.7845E-03	3.4066E-05
54	1.1536E+01	1.2160E+00	2.7325E-03	3.3450E-05
55	1.1831E+01	1.2170E+00	2.6646E-03	3.2644E-05
56	1.2154E+01	1.2181E+00	2.5939E-03	3.1804E-05
57	1.2743E+01	1.2199E+00	2.4754E-03	3.0388E-05
58	1.3175E+01	1.2211E+00	2.3970E-03	2.9444E-05
59	1.3764E+01	1.2226E+00	2.3026E-03	2.8294E-05
60	1.4195E+01	1.2235E+00	2.2450E-03	2.7581E-05
61	1.4784E+01	1.2244E+00	2.1901E-03	2.6866E-05
62	1.5000E+01	1.2245E+00	2.1803E-03	2.6766E-05

INDEX = 0

************   TERRYVO   0001576 LINES PRINTED

```
      PROGRAM LOADF(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE3)        MAIN       2
C  LOADF CREATES THE SUBROUTINE F IN PDECOL-FICT.                       MAIN       3
      DIMENSION II(72)                                                  MAIN       4
      DIMENSION LB(25),W(25),AU(6,25),AL(6,25)                          MAIN       5
      DIMENSION ALD(25),BLD(25),ABD(25,25),BBD(25,25)                   MAIN       6
      DIMENSION LBPR(25)                                                MAIN       7
      READ(5,12)NLINF                                                   MAIN       8
   12 FORMAT(10I4)                                                      MAIN       9
C  NLINE IS THE NUMBER OF COMMENT LINES TO BE READ AND PRINTED.         MAIN      10
      DO 20 K=1,NLINF                                                   MAIN      11
      READ(5,22)II                                                      MAIN      12
      WRITE(3,22)II                                                     MAIN      13
      WRITE(6,24)II                                                     MAIN      14
   20 CONTINUE                                                          MAIN      15
   22 FORMAT(72A1)                                                      MAIN      16
   24 FORMAT(/1X,72A1/)                                                 MAIN      17
      READ(5,12)NSP,NSPC                                                MAIN      18
      DO 30 K=1,NSPC                                                    MAIN      19
      READ(5,32)LB(K),W(K)                                              MAIN      20
      WRITE(6,34)K,LB(K),W(K)                                           MAIN      21
      READ(5,36)(AU(L,K),L=1,6)                                         MAIN      22
      READ(5,36)(AL(L,K),L=1,6)                                         MAIN      23
   30 CONTINUE                                                          MAIN      24
   32 FORMAT(A4,6F6.0)                                                  MAIN      25
   34 FORMAT(/2X,I4,4X,A4,4X,F8.2/)                                     MAIN      26
   36 FORMAT(1P3E16.8)                                                  MAIN      27
      READ(5,12)NLINF                                                   MAIN      28
      DO 40 K=1,NLINF                                                   MAIN      29
      READ(5,22)II                                                      MAIN      30
      WRITE(6,24)II                                                     MAIN      31
      WRITE(3,22)II                                                     MAIN      32
   40 CONTINUE                                                          MAIN      33
C  *************************                                            MAIN      34
C  ENTHALPIES AND HEAT CAPACITIES.                                      MAIN      35
C  *************************                                            MAIN      36
      WRITE(3,42)                                                       MAIN      37
      WRITE(6,42)                                                       MAIN      38
   42 FORMAT(6X,24HIF(T.GT.1000.)GO TO 2000)                            MAIN      39
      DO 50 K=1,NSPC                                                    MAIN      40
      WRITE(3,52)K,AL(1,K),AL(2,K)                                      MAIN      41
      WRITE(6,52)K,AL(1,K),AL(2,K)                                      MAIN      42
      WRITE(3,54)(AL(L,K),L=3,5)                                        MAIN      43
      WRITE(6,54)(AL(L,K),L=3,5)                                        MAIN      44
      WRITE(3,62)K,AL(6,K),AL(1,K)                                      MAIN      45
      WRITE(6,62)K,AL(6,K),AL(1,K)                                      MAIN      46
      WRITE(3,64)AL(2,K),AL(3,K)                                        MAIN      47
      WRITE(6,64)AL(2,K),AL(3,K)                                        MAIN      48
      WRITE(3,66)AL(4,K),AL(5,K)                                        MAIN      49
      WRITE(6,66)AL(4,K),AL(5,K)                                        MAIN      50
   50 CONTINUE                                                          MAIN      51
   52 FORMAT(6X,1HC,I2,9H=1.9872*(,1PE16.8,4H+T*(,1PE16.8)              MAIN      52
   54 FORMAT(5X,1H*,1X,3(4H+T*(,1PE16.8),5H)))))                        MAIN      53
      WRITE(3,56)                                                       MAIN      54
      WRITE(6,56)                                                       MAIN      55
   56 FORMAT(6X,10HGO TO 3000)                                          MAIN      56
      WRITE(3,58)                                                       MAIN      57
      WRITE(6,58)                                                       MAIN      58
```

```
 58    FORMAT(4H2000,2X,8HCONTINUE)
       DO 60 K=1,NSPC
       WRITE(3,52)K,AU(1,K),AU(2,K)
       WRITE(6,52)K,AU(1,K),AU(2,K)
       WRITE(3,54)(AU(L,K),L=3,5)
       WRITE(6,54)(AU(L,K),L=3,5)
       WRITE(3,62)K,AU(6,K),AU(1,K)
       WRITE(6,62)K,AU(6,K),AU(1,K)
       WRITE(3,64)AU(2,K),AU(3,K)
       WRITE(6,64)AU(2,K),AU(3,K)
       WRITE(3,66)AU(4,K),AU(5,K)
       WRITE(6,66)AU(4,K),AU(5,K)
 60    CONTINUE
 62    FORMAT(6X,1HH,I2,9H=1.9872*(,1PE16.8,4H+T*(,1PE16.8)
 64    FORMAT(5X,1H*,1X,4H+T*(,1PE16.8,4H/2.0,4H+T*(,1PE16.8,4H/3.0)
 66    FORMAT(5X,1H*,1X,4H+T*(,1PE16.8,4H/4.0,4H+T*(,1PE16.8,
      * 4H/5.0,6H))))))
       WRITE(3,68)
       WRITE(6,68)
 68    FORMAT(4H3000,2X,8HCONTINUE)
 C  **************************
 C  THERMAL CONDUCTIVITIES AND BINARY DIFFUSION COEFFICIENTS.
 C  **************************
       READ(5,12)NLINE
       DO 70 K=1,NLINE
       READ(5,22)II
       WRITE(3,22)II
       WRITE(6,24)II
 70    CONTINUE
       DO 80 K=1,NSPC
       READ(5,82)ALD(K),BLD(K)
       WRITE(6,84)K,LB(K),ALD(K),BLD(K)
 80    CONTINUE
 82    FORMAT(1P8E14.6)
 84    FORMAT(/2X,I4,4X,A4,4X,1P2E14.6/)
       DO 90 K=1,NSPC
       WRITE(3,92)K,ALD(K),BLD(K)
       WRITE(6,92)K,ALD(K),BLD(K)
 90    CONTINUE
 92    FORMAT(6X,2HBL,I2,1H=,1PE14.6,5H*(T**,1PE14.6,1H))
       READ(5,12)NLINE
       DO 100 K=1,NLINE
       READ(5,22)II
       WRITE(3,22)II
       WRITE(6,24)II
 100   CONTINUE
       NM=NSPC-1
       DO 110 I=1,NM
       IP=I+1
       DO 110 J=IP,NSPC
       READ(5,82)ABD(I,J),BBD(I,J)
       WRITE(6,112)I,J,LB(I),LB(J),ABD(I,J),BBD(I,J)
 110   CONTINUE
 112   FORMAT(/2X,I4,4X,I4,4X,A4,4X,A4,4X,1P2E14.6/)
       READ(5,114)PRESS
 114   FORMAT(F6.0)
       WRITE(3,116)PRESS
```

```
          WRITE(6,116)PRESS                                              MAIN
   116    FORMAT(1HC,2X,7HPRESS =,F6.2)                                  MAIN
          DO 120 I=1,NP                                                  MAIN
          IP=I+1                                                         MAIN
          DO 120 J=IP,NSPC                                               MAIN
          ABD(I,J)=ABD(I,J)/PRESS                                        MAIN
          WRITE(3,122)I,J,ABD(I,J),HBD(I,J)                              MAIN
          WRITE(6,122)I,J,ABD(I,J),BBD(I,J)                              MAIN
   120    CONTINUE                                                       MAIN
   122    FORMAT(6X,1HD,2I2,1H=,1PE14.6,5H*(T**,1PF14.6,1H))             MAIN
   C  ************************                                           MAIN
   C  SPECIFIC HEATS AND SPECIFIC ENTHALPIES.                            MAIN
   C  ************************                                           MAIN
          WRITE(3,124)                                                   MAIN
          WRITE(6,124)                                                   MAIN
   124    FORMAT(1HC,2X,44HSPECIFIC HEAT OF THE MIXTURE.  NTIS FORMULA.) MAIN
          NWR= MIN0(NSPC,8)                                              MAIN
          LBC=2H*C                                                       MAIN
          LBY=2H*Y                                                       MAIN
          WRITE(3,132)((LBC,K,LBY,K),K=1,NWR)                            MAIN
          WRITE(6,132)((LBC,K,LHY,K),K=1,NWR)                            MAIN
   132    FORMAT(6X,3HCM=,16(A2,I2))                                     MAIN
   131    IF(NWR.GE.NSPC)GO TO 135                                       MAIN
          NP=NWR+1                                                       MAIN
          NWR= MIN0(NSPC,NWR+8)                                          MAIN
          WRITE(3,136)((LBC,K,LBY,K),K=NP,NWR)                           MAIN
          WRITE(6,136)((LBC,K,LHY,K),K=NP,NWR)                           MAIN
   136    FORMAT(5X,1H*,16(A2,I2))                                       MAIN
          GO TO 131                                                      MAIN
   135    CONTINUE                                                       MAIN
          WPITE(6,134)                                                   MAIN
          WRITE(3,134)                                                   MAIN
   134    FORMAT(1HC,2X,39HSPECIFIC HEATS AND SPECIFIC ENTHALPIES.)      MAIN
          DO 140 K=1,NSPC                                                MAIN
          WRITE(3,142)K,K,W(K)                                           MAIN
          WRITE(6,142)K,K,W(K)                                           MAIN
          WRITE(3,144)K,K,W(K)                                           MAIN
          WRITE(6,144)K,K,W(K)                                           MAIN
   140    CONTINUE                                                       MAIN
   142    FORMAT(6X,1HC,I2,2H=C,I2,1H/,F6.2)                             MAIN
   144    FORMAT(6X,1HH,I2,2H=H,I2,1H/,F6.2)                             MAIN
   C  ************************                                           MAIN
   C  MOLE FRACTIONS AND VARIOUS SPACE DERIVATIVES.                      MAIN
   C  ************************                                           MAIN
          WRITE(3,148)                                                   MAIN
          WRITE(6,148)                                                   MAIN
   148    FORMAT(1HC,2X,45HMOLE FRACTIONS AND VARIOUS SPACE DERIVATIVES.) MAIN
          NS=NSPC+1                                                      MAIN
          WRITE(3,149)NS                                                 MAIN
          WRITE(6,149)NS                                                 MAIN
   149    FORMAT(6X,4HYS=Y,I2)                                           MAIN
          DO 150 K=1,NSPC                                                MAIN
          WRITE(3,152)K,K                                                MAIN
          WRITE(6,152)K,K                                                MAIN
   150    CONTINUE                                                       MAIN
   152    FORMAT(6X,1HX,I2,2H=Y,I2,3H/YS)                                MAIN
          DO 160 K=2,NSPC                                                MAIN
```

4829460

```
           WRITE(3,162)K,K
           WRITE(6,162)K,K
           WRITE(3,164)K,K
           WRITE(6,164)K,K
           WRITE(3,166)K,K
           WRITE(6,166)K,K
     160   CONTINUE
     162   FORMAT(6X,1HU,I2,3H=U(,I2,1H))
     164   FORMAT(6X,2HDU,I2,5H=UPH(,I2,1H))
     166   FORMAT(6X,3HDDU,I2,6H=UPH2(,I2,1H))
           LBU=2H-U
           NWR= MIN0(NSPC,8)
           WRITE(3,172)((LBU,K),K=2,NWR)
           WRITE(6,172)((LBU,K),K=2,NWR)
     172   FORMAT(6X,4HU 1=,3H1.0,8(A2,I2))
     177   IF(NWR.GE.NSPC)GO TO 179
           NP=NWR+1
           NWR= MIN0(NSPC,NWR+8)
           WRITE(3,182)((LBU,K),K=NP,NWR)
           WRITE(6,182)((LBU,K),K=NP,NWR)
           GO TO 177
     179   CONTINUE
     182   FORMAT(5X,1H*,1X,8(A2,I2))
           LBDU=3H-DU
           NWR=MIN0(NSPC,8)
           WRITE(3,174)((LBDU,K),K=2,NWR)
           WRITE(6,174)((LBDU,K),K=2,NWR)
     174   FORMAT(6X,5HDU 1=,8(A3,I2))
     181   IF(NWR.GE.NSPC)GO TO 183
           NP=NWR+1
           NWR=MIN0(NSPC,NWR+8)
           WRITE(3,184)((LBDU,K),K=NP,NWR)
           WRITE(6,184)((LBDU,K),K=NP,NWR)
           GO TO 181
     183   CONTINUE
     184   FORMAT(5X,1H*,1X,8(A3,I2))
           LBDDU=4H-DDU
           NWR=MIN0(NSPC,8)
           WRITE(3,176)((LBDDU,K),K=2,NWR)
           WRITE(6,176)((LBDDU,K),K=2,NWR)
     176   FORMAT(6X,6HDDU 1=,8(A4,I2))
     185   IF(NWR.GE.NSPC)GO TO 187
           NP=NWR+1
           NWR=MIN0(NSPC,NWR+8)
           WRITE(3,186)((LBDDU,K),K=NP,NWR)
           WRITE(6,186)((LBDDU,K),K=NP,NWR)
           GO TO 185
     187   CONTINUE
     186   FORMAT(5X,1H*,1X,8(A4,I2))
           LBDU=3H+DU
           NWR=MIN0(NSPC,4)
           WRITE(3,212)((LBDU,K,(K)),K=1,NWR)
           WRITE(6,212)((LBDU,K,(K)),K=1,NWR)
     212   FORMAT(6X,4HDYS=,4(A3,I2,1H/,F6.2))
     220   IF(NWR.GE.NSPC)GO TO 230
           NP=NWR+1
           NWR= MIN0(NSPC,NWR+4)
```

```
             WRITE(3,222)((LHDU,K,W(K)),K=NP,NWR)                                 MAIN    230
             WRITE(6,222)((LRDU,K,W(K)),K=NP,NWR)                                 MAIN    231
      222    FORMAT(5X,1H*,1X,4(A3,I2,1H/,F6.2))                                  MAIN    232
             GO TO 220                                                            MAIN    233
      230    CONTINUE                                                             MAIN    234
             LBDDU=4H+DDU                                                         MAIN    235
             NWR=MIN0(NSPC,4)                                                     MAIN    236
             WRITE(3,214)((LBDDU,K,W(K)),K=1,NWR)                                 MAIN    237
             WRITE(6,214)((LBDDU,K,W(K)),K=1,NWR)                                 MAIN    238
      214    FORMAT(6X,5HDDYS=,4(A4,I2,1H/,F6.2))                                 MAIN    239
      215    IF(NWR.GE.NSPC)GO TO 225                                             MAIN    240
             NP=NWR+1                                                             MAIN    241
             NWR=MIN0(NSPC,NWR+4)                                                 MAIN    242
             WRITE(3,224)((LBDDU,K,W(K)),K=NP,NWR)                                MAIN    243
             WRITE(6,224)((LBDDU,K,W(K)),K=NP,NWR)                                MAIN    244
             GO TO 215                                                            MAIN    245
      225    CONTINUE                                                             MAIN    246
      224    FORMAT(5X,1H*,1X,4(A4,I2,1H/,F6.2))                                  MAIN    247
             READ(5,232)TPN,PHN,TMN                                               MAIN    248
      232    FORMAT(1P3E12.4)                                                     MAIN    249
             PSR=PRESS/82.05                                                      MAIN    250
             WRITE(3,234)TPN,PHN,TMN                                              MAIN    251
             WRITE(6,234)TPN,PHN,TMN                                              MAIN    252
      234    FORMAT(1HC,2X,5HTPN =,1PE12.4,6X,5HPHN =,1PE12.4,6X,                 MAIN    253
            * 5HTMN =,1PE12.4)                                                    MAIN    254
             WRITE(3,226)TPN                                                      MAIN    255
             WRITE(6,226)TPN                                                      MAIN    256
      226    FORMAT(6X,10HDT=UPH(1)*,1PE12.4)                                     MAIN    257
             WRITE(3,242)PSR                                                      MAIN    258
             WRITE(6,242)PSR                                                      MAIN    259
      242    FORMAT(6X,5HDRH=-,1PE18.10,21H*(DT/T+DYS/YS)/(T*YS))                 MAIN    260
             DO 250 K=1,NSPC                                                      MAIN    261
             A=ALD(K)*BLD(K)                                                      MAIN    262
             B=BLD(K)-1.0                                                         MAIN    263
             WRITE(3,252)K,A,B                                                    MAIN    264
             WRITE(6,252)K,A,B                                                    MAIN    265
      250    CONTINUE                                                             MAIN    266
      252    FORMAT(6X,2HDL,I2,1H=,1PE14.6,3H*DT,6H*(T**(,1PE14.6,2H)))           MAIN    267
             DO 260 K=1,NSPC                                                      MAIN    268
             WRITE(3,262)K,K,W(K),K                                               MAIN    269
             WRITE(6,262)K,K,W(K),K                                               MAIN    270
      260    CONTINUE                                                             MAIN    271
      262    FORMAT(6X,2HDX,I2,3H=DU,I2,2H/(,F6.2,4H*YS),2H-Y,I2,                 MAIN    272
            * 12H*DYS/(YS*YS))                                                    MAIN    273
             DO 270 I=1,NM                                                        MAIN    274
             IP=I+1                                                               MAIN    275
             DO 270 J=IP,NSPC                                                     MAIN    276
             A=ARD(I,J)*BRD(I,J)                                                  MAIN    277
             B=BRD(I,J)-1.0                                                       MAIN    278
             WRITE(3,268)I,J,A,B                                                  MAIN    279
             WRITE(6,268)I,J,A,B                                                  MAIN    280
      270    CONTINUE                                                             MAIN    281
      268    FORMAT(6X,2HDD,I2,I2,1H=,1PE14.6,5H*(T**,1PE14.6,4H)*DT)             MAIN    282
      C   ************************                                                MAIN    283
      C   THERMAL CONDUCTIVITY OF THE MIXTURE.                                    MAIN    284
      C   ************************                                                MAIN    285
             WRITE(3,272)                                                         MAIN    286
```

```
          WRITE(6,272)
          WRITE(3,274)
          WRITE(6,274)
  272     FORMAT(1HC,2X,36HTHERMAL CONDUCTIVITY OF THE MIXTURE.)
  274     FORMAT(1HC,2X,42HBROKAW METHOD.  P. 466. REID AND SHERWOOD.)
          NWR=MIN0(NSPC,4)
          LBX=3H +X
          LBRL=3H*RL
          WRITE(3,282)((LBX,K,LBRL,K),K=1,NWR)
          WRITE(6,282)((LBX,K,LBRL,K),K=1,NWR)
  282     FORMAT(6X,5HRLM =,8(A3,I2))
  290     IF(NWR.GE.NSPC)GO TO 300
          NP=NWR+1
          NWR=MIN0(NSPC,NWR+4)
          WRITE(3,292)((LBX,K,LBRL,K),K=NP,NWR)
          WRITE(6,292)((LBX,K,LBRL,K),K=NP,NWR)
          GO TO 290
  300     CONTINUE
  292     FORMAT(5X,1H+,1X,8(A3,I2))
          LBRL=3H/RL
          NWR=MIN0(NSPC,4)
          WRITE(3,284)((LBX,K,LBRL,K),K=1,NWR)
          WRITE(6,284)((LBX,K,LBRL,K),K=1,NWR)
  284     FORMAT(6X,5HRLMV=,8(A3,I2))
  295     IF(NWR.GE.NSPC)GO TO 305
          NP=NWR+1
          NWR=MIN0(NSPC,NWR+4)
          WRITE(3,294)((LBX,K,LBRL,K),K=NP,NWR)
          WRITE(6,294)((LBX,K,LBRL,K),K=NP,NWR)
          GO TO 295
  305     CONTINUE
  294     FORMAT(5X,1H+,1X,8(A3,I2))
          WRITE(3,302)
          WRITE(6,302)
  302     FORMAT(6X,13HRLMV=1.0/RLMV)
          WRITE(3,304)
          WRITE(6,304)
  304     FORMAT(6X,18HRLM=0.5*(RLM+RLMV))
C  ************************
C  DERIVATIVE OF THE THERMAL CONDUCTIVITY OF THE MIXTURE.
C  ************************
          WRITE(3,312)
          WRITE(6,312)
  312     FORMAT(1HC,2X,45HSPACE DERIVATIVE OF THE THERMAL CONDUCTIVITY.)
          LBRL=3H+RL
          LBDX=3H*DX
          LBDL=3H+DL
          LBX=3H *X
          NWR=MIN0(NSPC,3)
          WRITE(3,322)((LBRL,K,LBDX,K,LBDL,K,LBX,K),K=1,NWR)
          WRITE(6,322)((LBRL,K,LBDX,K,LBDL,K,LBX,K),K=1,NWR)
  322     FORMAT(6X,5HDRLM=,12(A3,I2))
  325     IF(NWR.GE.NSPC)GO TO 330
          NP=NWR+1
          NWR=MIN0(NSPC,NWR+3)
          WRITE(3,328)((LBRL,K,LBDX,K,LBDL,K,LBX,K),K=NP,NWR)
          WRITE(6,328)((LBRL,K,LBDX,K,LBDL,K,LBX,K),K=NP,NWR)
```

```
328    FORMAT(5X,1H*,1X,12(A3,I2))                                   MAIN
       GO TO 325                                                     MAIN
330    CONTINUE                                                      MAIN
       LBDX=4H +DX                                                   MAIN
       LBRL=4H /RL                                                   MAIN
       LBX=4H  -X                                                    MAIN
       LBDL=4H *DL                                                   MAIN
       LBLD=4H/(RL                                                   MAIN
       LBLD2=4H *RL                                                  MAIN
       LBP=2H)                                                       MAIN
       WRITE(3,332)                                                  MAIN
       WRITE(6,332)                                                  MAIN
332    FORMAT(6X,18HDRLMV=-RLMV*RLMV*()                              MAIN
       DO 340 K=1,NSPC                                               MAIN
       IF(K.EQ.NSPC)LBP=2H))                                         MAIN
       WRITE(3,342)LBDX,K,LBRL,K,LBX,K,LBDL,K,LBLD,K,LBLD2,K,LBP     MAIN
       WRITE(6,342)LBDX,K,LBRL,K,LBX,K,LBDL,K,LBLD,K,LBLD2,K,LBP     MAIN
340    CONTINUE                                                      MAIN
342    FORMAT(5X,1H*,1X,6(A4,I2),A2)                                 MAIN
       WRITE(3,346)                                                  MAIN
       WRITE(6,346)                                                  MAIN
346    FORMAT(6X,21HDRLM=0.5*(DRLM+DRLMV))                           MAIN
C  ************************                                          MAIN
C  SOLVE FOR UV.                                                     MAIN
C  ************************                                          MAIN
       LBU=2H-U                                                      MAIN
       LBS=2H/(                                                      MAIN
       LBD=2H*D                                                      MAIN
       LBP=1H)                                                       MAIN
       WRITE(3,400)                                                  MAIN
       WRITE(6,400)                                                  MAIN
400    FORMAT(1HC,2X,13HSOLVE FOR UV.)                               MAIN
       DO 410 K=1,NSPC                                               MAIN
       J=1                                                           MAIN
       WRITE(3,412)J,K                                               MAIN
       WRITE(6,412)J,K                                               MAIN
410    CONTINUE                                                      MAIN
412    FORMAT(6X,2HZ(,I2,1H,,I2,5H)=1.0)                             MAIN
       WRITE(3,422)                                                  MAIN
       WRITE(6,422)                                                  MAIN
422    FORMAT(6X,8HW(1)=0.0)                                         MAIN
       DO 430 I=2,NSPC                                               MAIN
       DO 430 J=1,NSPC                                               MAIN
       IF(I.EQ.J)GO TO 430                                           MAIN
       IMIN=MIN0(I,J)                                                MAIN
       IMAX=MAX0(I,J)                                                MAIN
       WRITE(3,432)I,J,I,W(J),IMIN,IMAX                              MAIN
       WRITE(6,432)I,J,I,W(J),IMIN,IMAX                              MAIN
430    CONTINUE                                                      MAIN
432    FORMAT(6X,2HZ(,I2,1H,,I2,3H)=U,I2,2H/(,F6.2,2H*D,2I2,1H))     MAIN
       DO 500 I=2,NSPC                                               MAIN
       J=1                                                           MAIN
       IM=I-1                                                        MAIN
       WRITE(3,442)I,I,J,W(J),J,I                                    MAIN
       WRITE(6,442)I,I,J,W(J),J,I                                    MAIN
442    FORMAT(6X,2HZ(,I2,1H,,I2,4H)=-U,I2,2H/(,F6.2,2H*D,2I2,1H))    MAIN
       NWR=1                                                         MAIN
```

```
 450  IF(NWR.EQ.IM)GO TO 470
      NP=NWR+1
      NWR=MINO(IM,NWR+3)
      WRITE(3,452)((LBU,J,LBS,'(J),LBD,J,I,LBP),J=NP,NWR)
      WRITE(6,452)((LBU,J,LBS,W(J),LBD,J,I,LBP),J=NP,NWR)
 452  FORMAT(5X,1H*,1X,4(A2,I2,A2,F6.2,A2,2I2,A1))
      GO TO 450
 470  CONTINUF
      NWR=I
 480  IF(NWR.EQ.NSPC)GO TO 490
      NP=NWR+1
      NWR=MINO(NSPC,NWR+3)
      WRITF(3,452)((LBU,J,LBS,W(J),LBD,I,J,LBP),J=NP,NWR)
      WRITE(6,452)((LBU,J,LBS,W(J),LBD,I,J,LBP),J=NP,NWR)
      GO TO 480
 490  CONTINUE
 500  CONTINUE
      DO 510 K=2,NSPC
      WRITE(3,512)K,K,K
      WRITE(6,512)K,K,K
 510  CONTINUF
 512  FORMAT(6X,2HW(,I2,8H)=RH*(DU,I2,5H*YS-U,I2,9H*DYS)/PHN)
      WRITE(3,514)
      WRITE(6,514)
 514  FORMAT(6X,17HCALL DECOMP(NPDE))
      WRITE(3,516)
      WRITE(6,516)
 516  FORMAT(6X,21HCALL SOLVE(NPDE,W,UV))
C ************************
C SOLVE FOR THE SPACE DERIVATIVE OF UV.
C ************************
      WRITE(3,522)
      WRITE(6,522)
 522  FORMAT(1HC,2X,28HSOLVE FOR THE PARTIAL OF UV.)
      WRITE(3,532)
      WRITF(6,532)
 532  FORMAT(6X,8HW(1)=0.0)
      DO 600 I=2,NSPC
      WRITE(3,542)I,I,I
      WRITE(6,542)I,I,I
 542  FORMAT(6X,2HW(,I2,9H)=RH*(DDU,I2,5H*YS-U,I2,10H*DDYS)/PHN)
      WRITE(3,544)I,I
      WRITE(6,544)I,I
 544  FORMAT(5X,1H*,1X,8H+DRH*(DU,I2,5H*YS-U,I2,9H*DYS)/PHN)
      DO 600 J=1,NSPC
      IF(I.EQ.J)GO TO 600
      IMIN=MINO(I,J)
      IMAX=MAXO(I,J)
      WRITF(3,552)I,I,I,J,I,J,W(J),IMIN,IMAX
      WRITF(6,552)I,I,I,J,I,J,W(J),IMIN,IMAX
      WRITE(3,554)I,J,I,J,IMIN,IMAX,W(J),IMIN,IMAX,IMIN,IMAX
      WRITE(6,554)I,J,I,J,IMIN,IMAX,W(J),IMIN,IMAX,IMIN,IMAX
 552  FORMAT(6X,2HW(,I2,4H)=W(,I2,1H),5H+(-DU,I2,4H*UV(,I2,5H)+UV(,
     * I2,4H)*DU,I2,3H)/(,F6.2,2H*D,2I2,1H))
 554  FORMAT(5X,1H*,1X,3H+(U,I2,4H*UV(,I2,5H)-UV(,I2,3H)*U,I2,
     * 4H)*DD,2I2,2H/(,F6.2,2H*D,2I2,2H*D,2I2,1H))
 600  CONTINUF
```

```
      WRITE(3,602)
      WRITE(6,602)
  602 FORMAT(6X,22HCALL SOLVE(NPDE,*,DUV))
C   *************************
C   FIND THE TIME DERIVATIVES.
C   ************** **********
      WRITE(3,612)
      WRITE(6,612)
  612 FORMAT(1HC,2X,26HFIND THE TIME DERIVATIVES.)
      WRITE(3,622)
      WRITE(6,622)
  622 FORMAT(6X,15HSP=ASP+RSP*TIME)
      TMSPH2=TMN/(PHN*PHN)
      WRITE(3,632)TMSPH2
      WRITE(6,632)TMSPH2
  632 FORMAT(6X,3HTL=,1PE18.10,
     * 42H*(RH*RLM*UPH2(1)+UPH(1)*(DRH*RLM+RH*DRLM)))
      NWR=1
      DO 645 K=1,25
  645 LBPR(K)=1H
      LBPR(NSPC)=1H)
      TMSTP=TMN/TPN
      LBR=3H+R(
      LBH=3H)*H
      K=1
      WRITE(3,642)TMSTP,LBR,K,LBH,K
      WRITE(6,642)TMSTP,LBR,K,LBH,K
  642 FORMAT(6X,4HTR=-,1PE18.10,2H*(,A3,I2,A3,I2)
  650 IF(NWR.EQ.NSPC)GO TO 660
      NP=NWR+1
      NWR=MIN0(NSPC,NWR+3)
      WRITE(3,652)((LBR,K,LBH,K,LBPR(K)),K=NP,NWR)
      WRITE(6,652)((LBR,K,LBH,K,LBPR(K)),K=NP,NWR)
      GO TO 650
  660 CONTINUE
  652 FORMAT(5X,1H*,1X,3(A3,I2,A3,I2,A1))
      NWR=1
      TMSPH=TMN/PHN
      LBUV=4H+UV(
      LBC=3H)*C
      K=1
      WRITE(3,662)TMSPH,LBUV,K,LBC,K
      WRITE(6,662)TMSPH,LBUV,K,LBC,K
  662 FORMAT(6X,4HTD=-,1PE18.10,12H*UPH(1)*PH*(,A4,I2,A3,I2)
  670 IF(NWR.EQ.NSPC)GO TO 680
      NP=NWR+1
      NWR=MIN0(NSPC,NWR+5)
      WRITE(3,672)((LBUV,K,LBC,K,LBPR(K)),K=NP,NWR)
      WRITE(6,672)((LBUV,K,LBC,K,LBPR(K)),K=NP,NWR)
      GO TO 670
  680 CONTINUE
  672 FORMAT(5X,1H*,1X,6(A4,I2,A3,I2,A1))
      WRITE(3,682)
      WRITE(6,682)
  682 FORMAT(6X,31HFVAL(1)=SP*UPH(1)+(TL+TR+TD)/CM)
      DO 700 K=2,NSPC
      WRITE(3,692)K,TMSPH,K,K
```

4823466

```
          WRITE(6,692)K,TMSPH,K,K
     692  FORMAT(6X,2HDY,I2,2H=-,1PE18.10,9H*(DRH*DUV(,I2,9H)+RH*DUV(,
         * I2,2H)))
          WRITE(3,694)K,TMN,K
          WRITE(6,694)K,TMN,K
     694  FORMAT(6X,2HRY,I2,1H=,1PE16.10,3H*R(,I2,1H))
          WRITE(3,696)K,K,K,K
          WRITE(6,696)K,K,K,K
     696  FORMAT(6X,5HFVAL(,I2,9H)=SP*UPH(,I2,4H)+DY,I2,3H+RY,I2)
     700  CONTINUE
          STOP
          END
```

```
                                                                                          4829467
       S C O P E   2   L O A D   M A P                         LOADER VERSION 1.0   05/16/79  10.48.58.  PAGE   1
PROGRAM  WILL BE ENTERED AT  LOADF      (   177)               SCM LENGTH     15740          LCM LENGTH        0

       BLOCK         ADDRESS    LENGTH      FILE

       LOADF            110      11176       LGO
        /STP.END/     11306          1    SL-FTNLIB
        /FCL.C./      11307         23    SL-FTNLIB
        /Q8.IO./      11332        136    SL-FTNLIB
       Q8NTRY=        11470          1    SL-FTNLIB
       COMIO=         11471         44    SL-FTNLIB
       FECMSK=        11535         41    SL-FTNLIB
       FLTIN=         11576        156    SL-FTNLIB
       FLTOUT=        11754        315    SL-FTNLIB
       FMTAP=         12271        373    SL-FTNLIB
       FORSYS=        12664        533    SL-FTNLIB
       FORUTL=        13417         44    SL-FTNLIB
       GETFIT=        13463         43    SL-FTNLIB
       INCOM=         13526        256    SL-FTNLIB
       INPC=          14004        173    SL-FTNLIB
       KODER=         14177        467    SL-FTNLIB
       KRAKER=        14666        454    SL-FTNLIB
       OUTC=          15342        171    SL-FTNLIB
       OUTCOM=        15533        204    SL-FTNLIB
       SYSAID=        15737          1    SL-FTNLIB

C  OZONE FLAME.

C  U=MASS FRACTION.  Y=MASS FRACTION/MOLECULAR WEIGHT.

    1     O2      32.00

    2      O      16.00

    3     O3      48.00

C  NTIS ENTHALPIES AND HEAT CAPACITIES.

      IF(T.GT.1000.)GO TO 2000
      C 1=1.9872*(  3.62559850E+00+T*( -1.87821840E-03
     * +T*(  7.05545440E-06+T*( -6.76351370E-09+T*(  2.15559930E-12)))))
      H 1=1.9872*( -1.04752260E+03+T*(  3.62559850E+00
     * +T*( -1.87821840E-03/2.0+T*(  7.05545440E-06/3.0
     * +T*( -6.76351370E-09/4.0+T*(  2.15559930E-12/5.0))))))
      C 2=1.9872*(  2.95586620E+00+T*( -1.70615360E-03
     * +T*(  2.59251540E-06+T*( -1.78379800E-09+T*(  4.57090120E-13)))))
      H 2=1.9872*(  2.91436540E+04+T*(  2.95586620E+00
     * +T*( -1.70615360E-03/2.0+T*(  2.59251540E-06/3.0
     * +T*( -1.78379800E-09/4.0+T*(  4.57090120E-13/5.0))))))
      C 3=1.9872*(  2.46606170E+00+T*(  9.17032090E-03
     * +T*( -4.96984800E-06+T*( -2.06342300E-09+T*(  2.00155950E-12)))))
      H 3=1.9872*(  1.60595560E+04+T*(  2.46606170E+00
     * +T*(  9.17032090E-03/2.0+T*( -4.96984800E-06/3.0
     * +T*( -2.06342300E-09/4.0+T*(  2.00155950E-12/5.0))))))
      GO TO 3000
```

4229468

```
000  CONTINUE
      C 1=1.9872*(  3.42195350E+00+T*(  7.36182640E-04
     * +T*( -1.96522280E-07+T*(  3.62015580E-11+T*( -2.89456270E-15)))))
      H 1=1.9872*( -1.20198250E+03+T*(  3.62195350E+00
     * +T*(  7.36182640E-04/2.0+T*( -1.96522280E-07/3.0
     * +T*(  3.62015580E-11/4.0+T*( -2.89456270E-15/5.0))))))
      C 2=1.9872*(  2.53526380E+00+T*( -1.43718980E-05
     * +T*( -1.13601390E-08+T*(  6.60051310E-12+T*( -6.11816260E-16)))))
      H 2=1.9872*(  2.92302650E+04+T*(  2.53526380E+00
     * +T*( -1.43718980E-05/2.0+T*( -1.13601390E-08/3.0
     * +T*(  6.60051310E-12/4.0+T*( -6.11816260E-16/5.0))))))
      C 3=1.9872*(  5.46652390E+00+T*(  1.73260310E-03
     * +T*( -7.22048890E-07+T*(  1.37216600E-10+T*( -9.62338260E-15)))))
      H 3=1.9872*(  1.52140960E+04+T*(  5.46652390E+00
     * +T*(  1.73260310E-03/2.0+T*( -7.22048890E-07/3.0
     * +T*(  1.37216600E-10/4.0+T*( -9.62338280E-15/5.0))))))
000  CONTINUE

C  THERMAL CONDUCTIVITY.

C  LO  DALGARNO AND SMITH FORMULA.  T=100,2000.

C  LO2 LEAST SQUAPES FIT OF HANLEY AND FLY DATA.  T=300,2000.

C  LO3 BROMLEY CORRELATION.  T=300,2000.

C  O3  LENNARD-JONES PARAMETERS FROM A FIT OF TRETTER VISCOSITY DATA.

     1      O2      5.743800E-07  8.267700E-01

     2       O      1.603700E-06  7.100000E-01

     3      O3      3.899200E-07  8.424200E-01

      RL 1=  5.743800E-07*(T**  8.267700E-01)
      RL 2=  1.603700E-06*(T**  7.100000E-01)
      RL 3=  3.899200E-07*(T**  8.424200E-01)

C  BINARY DIFFUSION COEFFICIENTS.

C  O-O2  FIT BY MARRERO AND MASON.

C  O2-O3 AND O-O3  LENNARD JONES POTENTIAL.

C  O   LENNARD JONES PARAMETERS FROM DALGARNO AND SMITH VISCOSITY DATA.

C  O2  LENNARD JONES PARAMETERS FROM HANLEY AND ELY OVISCOSITY DATA.

C  O3  LENNARD JONES PARAMETERS FROM A FIT OF TRETTER VISCOSITY DATA.
```

```
C  A LEAST SQUARES FIT OF THE RESULTING DIFFUSIONS COEFFICIENTS IS
```

4829469

```
C  MADE FOR T=300,2000.

  1       2      02       0      1.320000E-05  1.774000E+00

  1       3      02      03      1.175200E-05  1.665400E+00

  2       3       0      03      1.659600E-05  1.665400E+00
 PRESS =  1.00
    D 1 2=  1.320000E-05*(T**  1.774000E+00)
    D 1 3=  1.175200E-05*(T**  1.665400E+00)
    D 2 3=  1.659600E-05*(T**  1.665400E+00)
 SPECIFIC HEAT OF THE MIXTURE.  NTIS FORMULA.
    CM=+C 1*Y 1+C 2*Y 2+C 3*Y 3
 SPECIFIC HEATS AND SPECIFIC ENTHALPIES.
    C 1=C 1/ 32.00
    H 1=H 1/ 32.00
    C 2=C 2/ 16.00
    H 2=H 2/ 16.00
    C 3=C 3/ 48.00
    H 3=H 3/ 48.00
 MOLE FRACTIONS AND VARIOUS SPACE DERIVATIVES.
    YS=Y 4
    X 1=Y 1/YS
    X 2=Y 2/YS
    X 3=Y 3/YS
    U 2=U( 2)
    DU 2=UPH( 2)
    DDU 2=UPH2( 2)
    U 3=U( 3)
    DU 3=UPH( 3)
    DDU 3=UPH2( 3)
    U 1=1.0-U 2-U 3
    DU 1=-DU 2-DU 3
    DDU 1=-DDU 2-DDU 3
    DYS=+DU 1/ 32.00+DU 2/ 16.00+DU 3/ 48.00
    DDYS=+DDU 1/ 32.00+DDU 2/ 16.00+DDU 3/ 48.00
 TPN =  1.0000E+03       PHN =  5.0000E-06       TMN =  5.0000E-05
    DT=UPH(1)*  1.0000E+03
    DRH=-  1.2187690433E-02*(DT/T+DYS/YS)/(T*YS)
    DL 1=  4.748802E-07*DT*(T**( -1.732300E-01))
    DL 2=  1.138627E-06*DT*(T**( -2.900000E-01))
    DL 3=  3.284764E-07*DT*(T**( -1.575800E-01))
    DX 1=DU 1/( 32.00*YS)-Y 1*DYS/(YS*YS)
    DX 2=DU 2/( 16.00*YS)-Y 2*DYS/(YS*YS)
    DX 3=DU 3/( 48.00*YS)-Y 3*DYS/(YS*YS)
    DD 1 2=  2.341680E-05*(T**  7.740000E-01)*DT
    DD 1 3=  1.957178E-05*(T**  6.654000E-01)*DT
    DD 2 3=  2.763898E-05*(T**  6.654000E-01)*DT
 THERMAL CONDUCTIVITY OF THE MIXTURE.
 BROKAW METHOD.  P. 486. REID AND SHERWOOD.
    RLM = +X 1*RL 1 +X 2*RL 2 +X 3*RL 3
    RLMV= +X 1/RL 1 +X 2/RL 2 +X 3/RL 3
    RLMV=1.0/RLMV
    RLM=0.5*(RLM+RLMV)
 SPACE DERIVATIVE OF THE THERMAL CONDUCTIVITY.
    DRLM=+RL 1*DX 1+DL 1 *X 1+RL 2*DX 2+DL 2 *X 2+RL 3*DX 3+DL 3 *X 3
```

4823470

```
      DPLMV=-RLMV*RLMV*(
     *  +DX 1 /RL 1  -X 1 *DL 1/(RL 1 *RL 1)
     *  +DX 2 /RL 2  -X 2 *DL 2/(RL 2 *RL 2)
     *  +DX 3 /RL 3  -X 3 *DL 3/(RL 3 *RL 3))
      DRLM=0.5*(DRLM+DPLMV)
C   SOLVE FOR UV.
      Z( 1, 1)=1.0
      Z( 1, 2)=1.0
      Z( 1, 3)=1.0
      W(1)=0.0
      Z( 2, 1)=U 2/( 32.00*D 1 2)
      Z( 2, 3)=U 2/( 48.00*D 2 3)
      Z( 3, 1)=U 3/( 32.00*D 1 3)
      Z( 3, 2)=U 3/( 16.00*D 2 3)
      Z( 2, 2)=-U 1/( 32.00*D 1 2)
     * -U 3/( 48.00*D 2 3)
      Z( 3, 3)=-U 1/( 32.00*D 1 3)
     * -U 2/( 16.00*D 2 3)
      W( 2)=RH*(DU 2*YS-U 2*DYS)/PHN
      W( 3)=RH*(DU 3*YS-U 3*DYS)/PHN
      CALL DECOMP(NPDE)
      CALL SOLVE(NPDE,W,UV)
C   SOLVE FOR THE PARTIAL OF UV.
      W(1)=0.0
      W( 2)=RH*(DDU 2*YS-U 2*DDYS)/PHN
     * +DRH*(DU 2*YS-U 2*DYS)/PHN
      W( 2)=W( 2)+(-DU 2*UV( 1)+UV( 2)*DU 1)/( 32.00*D 1 2)
     * +(U 2*UV( 1)-UV( 2)*U 1)*DD 1 2/( 32.00*D 1 2*D 1 2)
      W( 2)=W( 2)+(-DU 2*UV( 3)+UV( 2)*DU 3)/( 48.00*D 2 3)
     * +(U 2*UV( 3)-UV( 2)*U 3)*DD 2 3/( 48.00*D 2 3*D 2 3)
      W( 3)=RH*(DDU 3*YS-U 3*DDYS)/PHN
     * +DRH*(DU 3*YS-U 3*DYS)/PHN
      W( 3)=W( 3)+(-DU 3*UV( 1)+UV( 3)*DU 1)/( 32.00*D 1 3)
     * +(U 3*UV( 1)-UV( 3)*U 1)*DD 1 3/( 32.00*D 1 3*D 1 3)
      W( 3)=W( 3)+(-DU 3*UV( 2)+UV( 3)*DU 2)/( 16.00*D 2 3)
     * +(U 3*UV( 2)-UV( 3)*U 2)*DD 2 3/( 16.00*D 2 3*D 2 3)
      CALL SOLVE(NPDE,W,DUV)
C   FIND THE TIME DERIVATIVES.
      SP=ASP+BSP*TIME
      TL=  2.0000000000E+06*(RH*RLM*UPH2(1)+UPH(1)*(DRH*RLM+RH*DRLM))
      TR=-  5.0000000000E-08*(+R( 1)*H 1
     * +R( 2)*H 2 +R( 3)*H 3)
      TD=-  1.0000000000E+01*UPH(1)*RH*(+UV( 1)*C 1
     * +UV( 2)*C 2 +UV( 3)*C 3)
      FVAL(1)=SP*UPH(1)+(TL+TR+TD)/CM
      DY 2=-  1.0000000000E+01*(DRH*UV( 2)+RH*DUV( 2))
      RY 2=  5.0000000000E-05*R( 2)
      FVAL( 2)=SP*UPH( 2)+DY 2+RY 2
      DY 3=-  1.0000000000E+01*(DRH*UV( 3)+RH*DUV( 3))
      RY 3=  5.0000000000E-05*R( 3)
      FVAL( 3)=SP*UPH( 3)+DY 3+RY 3

***********    TEKRYDE    0000875 LINES PRINTED
```

```
      PROGRAM LOAD(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE3)
C  LOAD CREATES THE SUBROUTINE DIFFUN IN EPISODE( IF KCR = 1)
C  OR THE SUBROUTINE HT IN PDECOL-FLCT (IF KCR = 2).
      DIMENSION A(150),B(150),C(150)
      DIMENSION LS(25),W(25),LSR(7,150),IR(150)
      DIMENSION II(72)
C  READ AND WRITE A COMMENT LINE.
      READ(5,310)II
310   FORMAT(72A1)
      WRITE(3,310)II
      WRITE(6,312)II
312   FORMAT(//1X,72A1//)
      KCR=2
      LBL=4H
      LBL1=1H
      L01=1H0
      READ(5,8)NSP,NSPG,NR
C  NSP = THE NUMBER OF SPECIES PARTICIPATING IN REACTIONS.
C  NSPG - NSP = THE NUMBER OF SPECIES NOT PARTICIPATING IN REACTIONS
C  PLUS ANY THIRD BODY TERMS.
C  NR = THE NUMBER OF REACTIONS.
8     FORMAT(3I4)
      WRITE(6,608)NSP,NSPG,NR
608   FORMAT(/10X,5HNSP =,I4,4,,6HNSPG =,I4,4X,4HNR =,I4)
C  READ IN THE LABELS AND MOLECULAR WEIGHTS OF THE SPECIES.
      DO 10 K=1,NSP
      READ(5,12)LS(K),W(K)
      WRITE(6,11)K,LS(K),W(K)
11    FORMAT(/10X,I8,4X,A4,4X,F4.0/)
12    FORMAT(A4,F4.0)
10    CONTINUE
      NP=NSP+1
      DO 20 K=NP,NSPG
      READ(5,12)LS(K)
      WRITE(6,11)K,LS(K)
20    CONTINUE
C  WRITE THE QUANTITIES YK = MASS FRACTION/ MOLECULAR WEIGHT.
C  IF KCR = 2, ALSO WRITE A TEMPERATURE EQUATION.
      IF(KCR.EQ.2)WRITE(6,120)
      IF(KCR.EQ.2)WRITE(3,120)
120   FORMAT(6X,10HT=U(1)*TPN)
      IF(KCR.EQ.1)WRITE(6,14)1,1,W(1)
      IF(KCR.EQ.1)WRITE(3,14)1,1,W(1)
      IF(KCR.EQ.2)WRITE(6,314)1,1,W(1)
      IF(KCR.EQ.2)WRITE(3,314)1,1,W(1)
      DO 18 K=2,NSP
      IF(KCR.EQ.1)WRITE(6,14)K,K,W(K)
      IF(KCR.EQ.1)WRITE(3,14)K,K,W(K)
      IF(KCR.EQ.2)WRITE(6,316)K,K,W(K)
      IF(KCR.EQ.2)WRITE(3,316)K,K,W(K)
18    CONTINUE
314   FORMAT(6X,1HY,I2,2H=Y,I2,1H/,F6.2)
14    FORMAT(6X,1HY,I2,3H=Y(,I2,2H)/,F6.2)
316   FORMAT(6X,1HY,I2,3H=U(,I2,2H)/,F6.2)
C  THE DEFINITION OF GENERALIZED YK DEPEND ON THE REACTION SET,
C  SO THEY ARE READ AND WRITTEN LINE BY LINE.
      DO 22 K=NP,NSPG
```

```
      READ(5,310)II                                                      MAIN
      WRITE(6,310)II                                                     MAIN
      WRITE(3,310)II                                                     MAIN
   22 CONTINUE                                                           MAIN
C  NSUM IS THE I VALUE CORRESPONDING TO THE SUM OF THE                   MAIN
C  MASS FRACTIONS/ MOLECULAR 'EIGHTS.                                    MAIN
C  IT IS NEEDED TO WRITE THE DENSITY EQUATION.                           MAIN
      READ(5,322)NSUM                                                    MAIN
  322 FORMAT(I2)                                                         MAIN
      WRITE(6,26)NSUM                                                    MAIN
      WRITE(3,26)NSUM                                                    MAIN
   26 FORMAT(6X,11HRH=PSP/(T*Y,I2,1H))                                   MAIN
      WRITE(6,28)                                                        MAIN
      WRITE(3,28)                                                        MAIN
   28 FORMAT(6X,9HRH2=RH*RH)                                             MAIN
C  READ IN THE EQUATIONS.  THERE ARE A MAXIMUM OF THREE REACTIONS        MAIN
C  (LSR(1,K),LSR(2,K), AND LSR(3,K)) AND THREE PRODUCTS (LSR(5,K),       MAIN
C  LSR(6,K), AND LSR(7,K)).                                              MAIN
C  ALSO READ IN THE ARRHENIUS CONSTANTS FOR THE RATE CONSTANT            MAIN
C  RK = A*(T**B)*EXP(C/T).                                               MAIN
C  THE RATE CONSTANTS ARE EITHER IN CM**3/MOLE-SEC OR                    MAIN
C  (CM**2/MOLE-SEC)**2.                                                  MAIN
C  IF A OR C ARE NOT IN THE APPROPRIATE UNITS, READ IN                   MAIN
C  CONVERSION FACTORS.                                                   MAIN
      READ(5,31)CONVA,CONVC                                              MAIN
   31 FORMAT(1P2E14.6)                                                   MAIN
      DO 30 K=1,NR                                                       MAIN
      READ(5,32)(LSR(L,K),L=1,7),A(K),B(K),C(K)                          MAIN
      WRITE(6,33)K,(LSR(L,K),L=1,7),A(K),B(K),C(K)                       MAIN
      A(K)=A(K)*CONVA                                                    MAIN
      IF(LSR(3,K).NE.LBL)A(K)=A(K)*CONVA                                 MAIN
      C(K)=C(K)*CONVC                                                    MAIN
   30 CONTINUE                                                           MAIN
   32 FORMAT(7A4,5X,1PE8.2,F4.0,F8.0)                                    MAIN
   33 FORMAT(/2X,I4,2X,7(A4,2X),10X,1PE8.2,4X,F6.2,4X,F8.0/)             MAIN
      IF(KCR.EQ.1)GO TO 350                                              MAIN
C  IF KCR=2, WRITE THE RATE CONSTANTS FOR THE VARIOUS                    MAIN
C  REACTIONS, (CM**3/MOLE-SEC) OR (CM**3/MOLE-SEC)**2.                   MAIN
      DO 340 K=1,NR                                                      MAIN
      IF(C(K).EQ.0.0)GO TO 330                                           MAIN
      IF(B(K).NE.0.0)WRITE(6,34)K,A(K),B(K),C(K)                         MAIN
      IF(B(K).NE.0.0)WRITE(3,34)K,A(K),B(K),C(K)                         MAIN
   34 FORMAT(6X,2HRK,I2,1H=,1PE12.4,6H*(T**(,F6.2,7H))*EXP(,             MAIN
     * F10.2,3H/T))                                                      MAIN
      IF(B(K).EQ.0.0)WRITE(6,36)K,A(K),C(K)                              MAIN
      IF(B(K).EQ.0.0)WRITE(3,36)K,A(K),C(K)                              MAIN
   36 FORMAT(6X,2HRK,I2,1H=,1PE12.4,5H*EXP(,F10.2,3H/T))                 MAIN
      GO TO 340                                                          MAIN
  330 CONTINUE                                                           MAIN
      IF(B(K).NE.0.0)WRITE(3,334)K,A(K),B(K)                             MAIN
      IF(B(K).NE.0.0)WRITE(6,334)K,A(K),B(K)                             MAIN
  334 FORMAT(6X,2HRK,I2,1H=,1PE12.4,6H*(T**(,F6.2,2H)))                  MAIN
      IF(B(K).EQ.0.0)WRITE(3,336)K,A(K)                                  MAIN
      IF(B(K).EQ.0.0)WRITE(6,336)K,A(K)                                  MAIN
  336 FORMAT(6X,2HRK,I2,1H=,1PE12.4)                                     MAIN
  340 CONTINUE                                                           MAIN
  350 CONTINUE                                                           MAIN
```

4829448

```
C  WRITE THE TERMS RK = RATE OF THE K'TH REACTION/DENSITY (MOLE/GR-SEC)
C  THESE TERMS DEPEND ON THE REACTANTS (LS(1,K),LS(2,K),LS(3,K))
      DO 100 K=1,NR
      DO 55 J=1,NSPG
55    IF(LS(J).EQ.LSR(1,K))IP1=J
      IF(LSR(2,K).NE.LBL)GO TO 60
      WRITE(6,58)K,K,IP1
      WRITE(3,58)K,K,IP1
58    FORMAT(6X,1HR,I2,3H=RK,I2,2H*Y,I2)
      GO TO 90
60    DO 65 J=1,NSPG
65    IF(LS(J).EQ.LSR(2,K))IP2=J
      IF(LSR(3,K).NE.LBL)GO TO 70
      WRITE(6,68)K,K,IP1,IP2
      WRITE(3,68)K,K,IP1,IP2
68    FORMAT(6X,1HR,I2,6H=RH*RK,I2,2H*Y,I2,2H*Y,I2)
      GO TO 90
70    DO 75 J=1,NSPG
75    IF(LS(J).EQ.LSR(3,K))IP3=J
      WRITE(6,78)K,K,IP1,IP2,IP3
      WRITE(3,78)K,K,IP1,IP2,IP3
78    FORMAT(6X,1HR,I2,7H=RH2*RK,I2,2H*Y,I2,2H*Y,I2,2H*Y,I2)
90    CONTINUE
100   CONTINUE
C  WRITE THE TERMS FOR THE RATE OF PRODUCTION OF THE J'TH
C  SPECIES (1/SEC).
      DO 300 J=1,NSP
C  IR(K)=THE NUMBER OF MOLECULES OF SPECIES J EITHER PRODUCED OR
C  DESTROYED BY REACTION K.
      DO 275 K=1,NR
      IR(K)=0
      DO 260 I=1,3
260   IF(LS(J).EQ.LSR(I,K))IR(K)=IR(K)-1
      DO 270 I=4,7
270   IF(LS(J).EQ.LSR(I,K))IR(K)=IR(K)+1
275   CONTINUE
      DO 280 K=1,NR
280   IF(IR(K).NE.0)GO TO 290
      RATE=0.0
      WRITE(6,282)J,RATE
      WRITE(3,282)J,RATE
282   FORMAT(6X,2HP(,I2,2H)=,F6.2)
      GO TO 300
290   CONTINUE
      WRITE(6,202)J,W(J)
      WRITE(3,202)J,W(J)
202   FORMAT(6X,2HP(,I2,2H)=,F6.2,1H*)
C  PUT THE RATES FOR THE APPROPRIATE REACTIONS IN THE VECTOR II.
C  THE RATES WILL BE WRITTEN LINE BY LINE.
      DO 210 I=1,72
210   II(I)=1H
      II(6)=1H*
      II(7)=1H(
      IPL=8
      DO 250 K=1,NR
      JSP=0
      IF(IPL.LT.65)GO TO 215
```

```
      WRITE(6,212)II
      WRITE(3,212)II
  212 FORMAT(72A1)
      DO 213 I=1,72
  213 II(I)=1H
      II(6)=1H*
      II(7)=1H
      IPL=8
  215 JSP=IR(K)
      IF(JSP.EQ.0)GO TO 235
      NR1=K/10
      NR2=K-10*NR1
      IF(JSP.GT.0)II(IPL)=1H+
      IF(JSP.LT.0)II(IPL)=1H-
      IPL=IPL+1
      IF(IABS(JSP).EQ.1)GO TO 230
      II(IPL)=IABS(JSP)
C  THE SUBROUTINE IHOL CHANGES AN INTEGER TO ALPHANUMERIC FORM.
      CALL IHOL(II(IPL),L01)
      IPL=IPL+1
      II(IPL)=1H.
      IPL=IPL+1
      II(IPL)=1H*
      IPL=IPL+1
  230  II(IPL)=1HP
      IPL=IPL+1
      II(IPL)=NR1
      CALL IHOL(II(IPL),LBL1)
      IPL=IPL+1
      II(IPL)=NR2
      CALL IHOL(II(IPL),L01)
      IPL=IPL+1
  235 CONTINUE
  250 CONTINUE
      II(IPL)=1H)
      WRITE(6,212)II
      WRITE(3,212)II
  300 CONTINUE
      WRITE(6,112)
      WRITE(3,112)
  112 FORMAT(6X,6HRETURN)
      IF(KCR.EQ.2)GO TO 400
C  IF KCR=1, CREATE THE ENTRY POINT FOR DETERMINING THE ARRHENIUS
C  RATE CONSTANTS.
      WRITE(3,114)
      WRITE(6,114)
  114 FORMAT(6X,8HENTRY RT)
      DO 150 K=1,NR
      IF(C(K).EQ.0.0)GO TO 140
      IF(B(K).NE.0.0)WRITE(6,34)K,A(K),B(K),C(K)
      IF(B(K).NE.0.0)WRITE(3,34)K,A(K),B(K),C(K)
      IF(B(K).EQ.0.0)WRITE(6,36)K,A(K),C(K)
      IF(B(K).EQ.0.0)WRITE(3,36)K,A(K),C(K)
      GO TO 150
  140 CONTINUE
      IF(B(K).NE.0.0)WRITE(3,334)K,A(K),B(K)
      IF(B(K).NE.0.0)WRITE(6,334)K,A(K),B(K)
```

```
      IF(B(K).EQ.0.0)WRITE(3,336)K,A(K)               MAIN
      IF(B(K).EQ.0.0)WRITE(6,336)K,A(K)               MAIN
  150 CONTINUE                                        MAIN
  400 CONTINUE                                        MAIN
      STOP                                            MAIN
      END                                             MAIN
```

```
SUBROUTINE IHOL        76/76    OPT=1 ROUND=+-*/ TRACE         FTN 4.6+452         05/16/79  10.44.03         PAGE

                    SUBROUTINE IHOL(K,LB)                                      IHOL      2
                    IF(K.EQ.0)K=LB                                             IHOL      3
                    IF(K.EQ.1)K=1H1                                            IHOL      4
                    IF(K.EQ.2)K=1H2                                            IHOL      5
                    IF(K.EQ.3)K=1H3                                            IHOL      6
                    IF(K.EQ.4)K=1H4                                            IHOL      7
                    IF(K.EQ.5)K=1H5                                            IHOL      8
                    IF(K.EQ.6)K=1H6                                            IHOL      9
                    IF(K.EQ.7)K=1H7                                            IHOL     10
                    IF(K.EQ.8)K=1H8                                            IHOL     11
                    IF(K.EQ.9)K=1H9                                            IHOL     12
                    RETURN                                                     IHOL     13
                    END                                                        IHOL     14
```

4829451

4829452

```
        S C O P E   2   L O A D   M A P                              LOADER VERSION 1.0   05/16/79  10.48.06.  PAGE   1
PROGRAM  WILL BE ENTERED AT  LOAD      (   177)                  SCM LENGTH     12172          LCM LENGTH        0

        BLOCK          ADDRESS    LENGTH      FILE

        LOAD              110       5353       LGO
        IMOL             5463         55       LGO
         /STP.END/       5540          1    SL-FTNLIB
         /FCL.C./        5541         23    SL-FTNLIB
         /DB.ID./        5564        136    SL-FTNLIB
        Q8NTRY=          5722          1    SL-FTNLIB
        COMIO=           5723         44    SL-FTNLIB
        FECMSK=          5767         41    SL-FTNLIB
        FLTIN=           6030        156    SL-FTNLIB
        FLTOUT=          6206        315    SL-FTNLIB
        FMTAP=           6523        373    SL-FTNLIB
        FORSYS=          7116        533    SL-FTNLIB
        FORUTL=          7651         44    SL-FTNLIB
        GETFIT=          7715         43    SL-FTNLIB
        INCOM=           7760        256    SL-FTNLIB
        INPC=           10236        173    SL-FTNLIB
        KODER=          10431        467    SL-FTNLIB
        KRAKER=         11120        454    SL-FTNLIB
        OUTC=           11574        171    SL-FTNLIB
        OUTCOM=         11765        204    SL-FTNLIB
        SYSAIO=         12171          1    SL-FTNLIB

C  OZONE KINETICS.

        NSP =   3    NSPG =   6    NR =   6
              1      02     32.

              2       0     16.

              3      03     48.

              4       M

              5     MBA

              6      MB
      T=U(1)*TPN
      Y 1=Y 1/ 32.00
      Y 2=U( 2)/ 16.00
      Y 3=U( 3)/ 48.00
      Y4=Y1+Y2+Y3
      Y5=0.44*(Y1+Y2)+Y3
      Y6=Y1+3.6*Y2+Y3
      RH=PSR/(T*Y 4)
      RH2=RH*RH

      1    03   MBA          =       0    02    MBA          4.31E+14      0.00     -11161.
```

4829453

```
2    0    02   MMA       =    03   MMA                      1.20E+13    0.00      976.

3    03   0              =    02   02                       1.14E+13    0.00    -2300.

4    02   02             =    03   0                        1.19E+13    0.00   -50600.

5    0    0    MG        =    02   MG                       1.38E+18   -1.00     -171.

6    02   MG             =    0    0    MG                  2.75E+19   -1.00   -59732.
 RK 1=  4.3100E+14*EXP( -11161.00/T)
 RK 2=  1.2000E+13*EXP(    976.00/T)
 RK 3=  1.1400E+13*EXP(  -2300.00/T)
 RK 4=  1.1900E+13*EXP( -50600.00/T)
 RK 5=  1.3800E+18*(T**( -1.00))*EXP(   -171.00/T)
 RK 6=  2.7500E+19*(T**( -1.00))*EXP( -59732.00/T)
  R 1=RH*RK 1*Y 3*Y 5
  R 2=RH2*RK 2*Y 2*Y 1*Y 5
  R 3=RH*RK 3*Y 3*Y 2
  R 4=RH*RK 4*Y 1*Y 1
  R 5=RH2*RK 5*Y 2*Y 2*Y 6
  R 6=RH*RK 6*Y 1*Y 6
  R( 1)= 32.00*
 *(+R 1-R 2+2.*R 3-2.*R 4+R 5-R 6)
  R( 2)= 16.00*
 *(+R 1-R 2-R 3+R 4-2.*R 5+2.*R 6)
  R( 3)= 48.00*
 *(-R 1+R 2-R 3+R 4)
  RETURN
***********     TERRYDD    0000442 LINES PRINTED
```

GLOSSARY

c_p — Specific heat capacity at constant pressure of the fluid mixture (cal/gm-K)

c_{pk} — Specific heat capacity at constant pressure of the k'th species (cal/gm-K)

D_{ij} — Binary diffusion coefficient of species i in a bath of species j (cm^2/sec)

h_k — Specific enthalpy of the k'th species (cal/gm)

M_k — Molecular weight of k'th species (gm/mole)

N — The number of species

P — Pressure (atm)

R — Universal gas constant
$R = 1.9872$ cal/mole-K
$R = 82.05$ cm^3-atm/mole-K

R_k — Rate of production of k'th species by chemical reactions (moles/cm^3-sec)

t — Time coordinate (sec)

T — Temperature of the fluid (K)

u — Velocity of fluid mixture (cm/sec)

V_k — Diffusion velocity of k'th species (cm/sec)

x — Space coordinate (cm)

X_k — Mole fraction of k'th species

Y_k — Mass fraction of k'th species

λ — Thermal conductivity of the fluid mixture (cal/cm-sec-K)

ρ — Density (gm/cm^3)

DISTRIBUTION LIST

No. of Copies	Organization
12	Commander Defense Technical Info Center ATTN: DDC-DDA Cameron Station Alexandria, VA 22314
1	Director Defense Advanced Research Projects Agency ATTN: LTC C. Buck 1400 Wilson Boulevard Arlington, VA 22209
2	Director Institute for Defense Analyses ATTN: H. Wolfhard R.T. Oliver 400 Army-Navy Drive Arlington, VA 22202
1	Commander US Army Materiel Development and Readiness Command ATTN: DRCDMD-ST 5001 Eisenhower Avenue Alexandria, VA 22333
2	Commander US Army Armament Research & Development Command ATTN: DRDAR-TSS Dover, NJ 07801
5	Commander US Army Armament Research and Development Command ATTN: DRDAR-LCA, J. Lannon DRDAR-LCA, C. Lenchitz DRDAR-LC, J.P. Picard DRDAR-LCE, R.F. Walker DRDAR-SCA, L. Stiefel Dover, NJ 07801

No. of Copies	Organization
1	Commander US Army Armament Materiel Readiness Command ATTN: DRSAR-LEP-L, Tech Lib Rock Island, IL 61299
1	Director US Army ARRADCOM Benet Weapons Laboratory ATTN: DRDAR-LCB-TL Watervliet, NY 12189
1	Commander US Army Watervliet Arsenal ATTN: Code SARWV-RD, R. Thierry Watervliet, NY 12189
1	Commander US Army Aviation Research and Development Command ATTN: DRSAV-E P.O. Box 209 St. Louis, MO 63166
1	Director US Army Air Mobility Research and Development Laboratory Ames Research Center Moffett Field, CA 94035
1	Commander US Army Communications Rsch and Development Command ATTN: DRDCO-PPS-SA Ft. Monmouth, NJ 07703
1	Commander US Army Electronics Research & Development Command Technical Support Activity ATTN: DELSD-L Ft. Monmouth, NJ 07703

DISTRIBUTION LIST

No. of Copies	Organization
2	Commander US Army Missile Command ATTN: DRDMI-R DRDMI-YDL Redstone Arsenal, AL 35809
1	Commander US Army Natick Research and Development Command ATTN: DRXRE, D. Sieling Natick, MA 01762
1	Commander US Army Tank Automotive Rsch and Development Command ATTN: DRDTA-UL Warren, MI 48090
1	Commander US Army White Sands Missile Range ATTN: STEWS-VT White Sands Missile Range, NM 88002
1	Commander US Army Materials & Mechanics Research Center ATTN: DRXMR-ATL Watertown, MA 02172
1	Commander US Army Research Office ATTN: Tech Lib P.O. Box 12211 Research Triangle Park, NC 27706
1	Director US Army TRADOC Systems Analysis Activity ATTN: ATAA-SL, Tech Lib White Sands Missile Range, NM 88002
1	Office of Naval Research ATTN: Code 473 800 N. Quincy Street Arlington, VA 22217
1	Commander Naval Sea Systems Command ATTN: J.W. Murrin (NAVSEA-0331) National Center Bldg 2, Room 6E09 Washington, DC 20360
2	Commander Naval Surface Weapons Center ATTN: S.J. Jacobs/Code 240 Code 730 Silver Spring, MD 20910
1	Commander Naval Surface Weapons Center ATTN: Library Br, DX-21 Dahlgren, VA 22448
1	Commander Naval Underwater Systems Center Energy Conversion Department ATTN: R.S. Lazar/Code 5B331 Newport, RI 02840
2	Commander Naval Weapons Center ATTN: R. Derr C. Thelen China Lake, CA 93555
1	Commander Naval Research Laboratory ATTN: Code 6180 Washington, DC 20375
3	Superintendent Naval Postgraduate School ATTN: Tech Lib D. Netzer A. Fuhs Monterey, CA 93940

DISTRIBUTION LIST

No. of Copies	Organization
2	Commander Naval Ordnance Station ATTN: A. Roberts Tech Lib Indian Head, MD 20640
1	AFOSR ATTN: B.T. Wolfson Bolling AFB, DC 20332
2	AFRPL (DYSC) ATTN: D. George J.N. Levine Edwards AFB, CA 93523
1	Lockheed Palo Alto Rsch Labs ATTN: Tech Info Ctr 3521 Hanover Street Palo Alto, CA 94304
1	Aerojet Solid Propulsion Co. ATTN: P. Micheli Sacramento, CA 95813
1	ARO Incorporated ATTN: N. Dougherty Arnold AFS, TN 37389
1	Atlantic Research Corporation ATTN: M.K. King 5390 Cherokee Avenue Alexandria, VA 22314
1	AVCO Corporation AVCO Everett Research Lab Div ATTN: D. Stickler 2385 Revere Beach Parkway Everett, MA 02149
2	Calspan Corporation ATTN: E.B. Fisher A.P. Trippe P.O. Box 235 Buffalo, NY 14221

No. of Copies	Organization
1	ENKI Corporation ATTN: M.I. Madison 9015 Fulbright Avenue Chatsworth, CA 91311
1	Foster Miller Associates, Inc. ATTN: A.J. Erickson 135 Second Avenue Waltham, MA 02154
1	General Electric Company Armament Department ATTN: M.J. Bulman Lakeside Avenue Burlington, VT 05402
1	General Electric Company Flight Propulsion Division ATTN: Tech Lib Cincinnati, OH 45215
2	Hercules Incorporated Alleghany Ballistic Lab ATTN: R. Miller Tech Lib Cumberland, MD 21501
2	Hercules Incorporated Bacchus Works ATTN: M. Beckstead R. Simmons Magna, UT 84044
1	IITRI ATTN: M.J. Klein 10 West 35th Street Chicago, IL 60615
1	Olin Corporation Badger Army Ammunition Plant ATTN: J. Ramnarace Baraboo, WI 53913

DISTRIBUTION LIST

No. of Copies	Organization	No. of Copies	Organization
2	Olin Corporation New Haven Plant ATTN: R.L. Cook D.W. Riefler 275 Winchester Avenue New Haven, CT 06504	1	Thiokol Corporation Elkton Division ATTN: E. Sutton Elkton, MD 21921
1	Paul Gough Associates, Inc. ATTN: P.S. Gough P.O. Box 1614 Portsmouth, NJ 03801	3	Thiokol Corporation Huntsville Division ATTN: D. Flanigan R. Glick Tech Lib Huntsville, AL 35807
1	Physics International Company 2700 Merced Street San Leandro, CA 94577	2	Thiokol Corporation Wasatch Division ATTN: J. Peterson Tech Lib P.O. Box 524 Brigham City, UT 84302
1	Pulsepower Systems, Inc. ATTN: L.C. Elmore 815 American Street San Carlos, CA 94070	1	TRW Systems Group ATTN: H. Korman One Space Park Redondo Beach, CA 90278
2	Rockwell International Corp. Rocketdyne Division ATTN: C. Obert J.E. Flanagan 6633 Canoga Avenue Canoga Park, CA 91304	2	United Technology Center ATTN: R. Brown Tech Lib Sunnyvale, CA 94088
2	Rockwell International Corp. Rocketdyne Division ATTN: W. Haymes Tech Lib McGregor, TX 76657	1	Universal Propulsion Co. ATTN: H.J. McSpadden P.O. Box 546 Riverside, CA 92502
1	Science Applications, Inc. ATTN: R.B. Edelman 23146 Cumorah Crest Woodland Hills, CA 91364	1	Battelle Memorial Institute ATTN: Tech Lib 505 King Avenue Columbus, OH 43201
1	Shock Hydrodynamics, Inc. ATTN: W.H. Anderson 4710-16 Vineland Avenue North Hollywood, CA 91602	1	Brigham Young University Dept of Chemical Engineering ATTN: Dr. M. Beckstead Provo, UT 84601

DISTRIBUTION LIST

No. of Copies	Organization
1	California Institute of Tech 204 Karmar Lab Mail Stop 301-46 ATTN: F.E.C. Culick 1201 E. California Street Pasadena, CA 91125
1	Case Western Reserve Univ. Division of Aerospace Sciences ATTN: J. Tien Cleveland, OH 44135
3	Georgia Institute of Tech School of Aerospace Engineering ATTN: B.T.Zinn E. Price W.C. Strahle Atlanta, GA 30332
1	Institute of Gas Technology ATTN: D. Gidaspow 3424 S. State Street Chicago, IL 60616
1	Johns Hopkins University/APL Chemical Propulsion Information Agency ATTN: T. Christian Johns Hopkins Road Laurel, MD 20810
1	Massachusetts Institute of Technology Dept of Mechanical Engineering ATTN: T. Toong Cambridge, MA 02139
1	Director Graduate Center of Applied Science New York University ATTN: M. Summerfield 26-36 Stuyvesant New York, NY 10003

No. of Copies	Organization
1	Pennsylvania State University Applied Research Lab ATTN: G.M. Faeth P.O. Box 30 State College, PA 16801
1	Pennsylvania State University Dept of Mechanical Engineering ATTN: K. Kuo University Park, PA 16802
2	Princeton University Forrestal Campus Library ATTN: L. Caveny Tech Lib P.O. Box 710 Princeton, NJ 08540
2	Purdue University School of Mechanical Engineering ATTN: J. Osborn S.N.B. Murthy TSPC Chaffee Hall West Lafayette, IN 47906
1	Rutgers State University Dept of Mechanical and Aerospace Engineering ATTN: S. Temkin University Heights Campus New Brunswick, NJ 08903
1	SRI International ATTN: Tech Lib 333 Ravenswood Avenue Menlo Park, CA 94025
1	Stevens Institute of Technology Davidson Laboratory ATTN: R. McAlevy, III Hoboken, NJ 07030

DISTRIBUTION LIST

No. of Copies	Organization
1	University of California San Diego AMES Department ATTN: F. Williams P.O. Box 109 La Jolla, CA 92037
1	University of Illinois Dept of Aeronautical Engineering ATTN: H. Krier Transportation Bldg, Rm 105 Urbana, IL 61801
1	University of Minnesota Dept of Mechanical Engineering ATTN: E. Fletcher Minneapolis, MN 55455
2	University of Utah Dept of Chemical Engineering ATTN: A. Baer G. Flandro Salt Lake City, UT 84112

Aberdeen Proving Ground

Dir, USAMSAA
ATTN: DRXSY-D
DRXSY-MP, H. Cohen

Cdr, USATECOM
ATTN: DRSTE-TO-F

Dir, Wpns Sys Concepts Team,
Bldg E3516, EA
ATTN: DRDAR-ACW

USER EVALUATION OF REPORT

Please take a few minutes to answer the questions below; tear out this sheet and return it to Director, US Army Ballistic Research Laboratory, ARRADCOM, ATTN: DRDAR-TSB, Aberdeen Proving Ground, Maryland 21005. Your comments will provide us with information for improving future reports.

1. BRL Report Number ________________________

2. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which report will be used.)

____

____

____

3. How, specifically, is the report being used? (Information source, design data or procedure, management procedure, source of ideas, etc.) ________________________

____

____

4. Has the information in this report led to any quantitative savings as far as man-hours/contract dollars saved, operating costs avoided, efficiencies achieved, etc.? If so, please elaborate.

____

____

5. General Comments (Indicate what you think should be changed to make this report and future reports of this type more responsive to your needs, more usable, improve readability, etc.) ____________

____

____

____

6. If you would like to be contacted by the personnel who prepared this report to raise specific questions or discuss the topic, please fill in the following information.

Name: ________________________

Telephone Number: ________________________

Organization Address: ________________________

____

____