AD

MEMORANDUM REPORT ARBRL-MR-03165

A COMPUTER CODE FOR THE SOLUTION OF THE EQUATIONS GOVERNING A LAMINAR, PREMIXED, ONE-DIMENSIONAL FLAME

Terence P. Coffee

April 1982

US ARMY ARMAMENT RESEARCH AND DEVELOPMENT COMMAND
BALLISTIC RESEARCH LABORATORY
ABERDEEN PROVING GROUND, MARYLAND

Destroy this report when it is no longer needed.
Do not return it to the originator.

Secondary distribution of this report by originating or sponsoring activity is prohibited.

Additional copies of this report may be obtained from the National Technical Information Service, U.S. Department of Commerce, Springfield, Virginia 22161.

The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

The use of trade names or manufacturers' names in this report does not constitute indorsement of any commercial product.

SECURITY CLASSIFICATION OF THIS PAGE (When Date Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: Memorandum Report ARBRL-MR-03165	2. GOVT ACCESSION NO.: AD-A114041
3. RECIPIENT'S CATALOG NUMBER:	
4. TITLE (and Subtitle): A COMPUTER CODE FOR THE SOLUTION OF THE EQUATIONS GOVERNING A LAMINAR, PREMIXED, ONE-DIMENSIONAL FLAME	5. TYPE OF REPORT & PERIOD COVERED:
	6. PERFORMING ORG. REPORT NUMBER:
7. AUTHOR(s): Terence P. Coffee	8. CONTRACT OR GRANT NUMBER(s):
9. PERFORMING ORGANIZATION NAME AND ADDRESS: US Army Ballistic Research Laboratory ATTN: DRDAR-BLI Aberdeen Proving Ground, MD 21005	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 1L161102AH43
11. CONTROLLING OFFICE NAME AND ADDRESS: U.S. Army Armament Research & Development Command U.S. Army Ballistic Research Laboratory (DRDAR-BL) Aberdeen Proving Ground, MD 21005	12. REPORT DATE: April 1982 13. NUMBER OF PAGES: 141
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)

Laminar Flame
One Dimensional Flame
Premixed Flame
Finite Elements Method
Non-uniform Grid
Species Profiles
Temperature Profiles

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

A computer program for numerically solving the equations governing a laminar, premixed, one-dimensional flame is described. Both unbounded flames and burner stabilized flames are discussed. Some numerical considerations in successfully using the code are presented.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Date Entered)

TABLE OF CONTENTS

I. INTRODUCTION

We are interested in determining validated sets of elementary chemical reactions for use in predictive combustion models. For this purpose, a simulation of the laminar, premixed, one dimensional, steady state flame has been implemented. This has the advantage of being a relatively simple combustion problem that also yields predicted temperature and species profiles that can be compared with suitable burner experiments.

The basic idealism is of an infinite column of premixed gas. At some point the gas is ignited. A flame front forms and propagates along the column of gas. All effects are assumed to be one-dimensional. Eventually, the flame will reach steady state. A basic characteristic of the solution is the flame speed, the velocity with which the flame front moves with reference to the undisturbed gas mixture. The flame front is marked by steep temperature and chemical species profiles. The above formulation is referred to as an unbounded, or adiabatic flame.

In practice, a flame is usually stabilized on a burner. That is, the flame will propagate to the burner surface. Loss of heat to the burner will stabilize the flame. This creates different boundary conditions.

This report describes a method of numerically solving the equations governing a laminar, premixed, one-dimensional, steady state flame. Both unbounded flames and burner stabilized flames are discussed. To solve such systems, we have modified a standard package for integrating one-dimensional partial differential equations so that it efficiently handles flame equations. The basic procedure is to integrate the equations in time until the steady state solution is reached. Changes have been made in the basic structure of the code, as well as adding a number of subroutines. The program has been implemented for the ozone flame and the H_2-O_2-N_2 flame.

Section II describes the partial differential equations governing the flames. In Section III, the basic code used to solve the equations is introduced. Section IV describes the modifications made in the code, and Section V the additional subroutines added to the program. Section VI discusses the changes in the equations and the code necessary to model burner stabilized flames. Section VII reports some of the numerical considerations in successfully using the code. Finally, the Appendix gives a complete listing of the code, plus the output for a run modeling a typical H_2-O_2-N_2 flame.

II. THE UNBOUNDED FLAME EQUATIONS

The equations for a multicomponent reacting ideal gas mixture can be found in the literature.[1-4] We are interested in the equations that describe a one-dimensional, laminar, premixed flame that propagates in an unbounded medium. The effects of radiation, viscosity and body forces are ignored. The momentum equation

$$\hat{\rho}\,\frac{\partial \hat{v}}{\partial \hat{t}} = -\hat{\rho}\,\hat{v}\,\frac{\partial \hat{v}}{\partial \hat{x}} - \frac{\partial \hat{p}}{\partial \hat{x}}$$

can be eliminated. Here $\hat{v}$ is the fluid velocity, $\hat{\rho}$ is the density, and $\hat{p}$ is the pressure. The independent variables are time $\hat{t}$ (sec) and distance $\hat{x}$ (cm). For flames, the fluid velocity is much less than the speed of sound, that is, $v^2 << \hat{p}/\hat{\rho}$. For steady state flow, this implies that the pressure gradient is negligibly small. So the pressure is assumed to be constant.

The pertinent equations are then, overall continuity.

$$\frac{\partial \hat{\rho}}{\partial \hat{t}} = -\frac{\partial (\hat{\rho}\hat{v})}{\partial \hat{x}} \quad . \tag{1}$$

That is, any change in density with respect to time is due to the overall convection of the mixture, which equals the gradient of the total mass flux $\hat{\rho}\hat{v}$.

Continuity of species:

$$\hat{\rho}\,\frac{\partial \hat{Y}_k}{\partial \hat{t}} = -\hat{\rho}\hat{v}\,\frac{\partial \hat{Y}_k}{\partial \hat{x}} - \frac{\partial}{\partial \hat{x}}\,(\hat{\rho}\hat{Y}_k\hat{V}_k) + \hat{R}_k M_k, \quad k = 1,2,\ldots N \quad . \tag{2}$$

---

[1] *F. A. Williams, Combustion Theory, Addison-Wesley, Reading, MA, Chapter 1. 1965.*

[2] *R. B. Bird, W. E. Stewart and E. N. Lightfoot, Transport Phenomena, John Wiley and Sons, NY, Chapter 18, 1960.*

[3] *R. M. Fristrom and A. A. Westenberg, Flame Structure, McGraw-Hill, NY, Chapter V-1, 1965*

[4] *J.O. Hirshfelder, C. F. Curtiss and R. B. Bird, Molecular Theory of Gases and Liquids, 2nd Printing corrected, with notes, John Wiley and Sons, NY, Chapter 11.1, 1964.*

where $\hat{Y}_k$ is the mass fraction of the k^{th} species, $\hat{V}_k$ is its diffusion velocity, M_k is its molecular weight, and $\hat{R}_k$ is the rate at which the species is produced or consumed by the chemical reactions. So changes in the concentration of the k^{th} species can be due to the convection of the species, the diffusion of the species in the mixture, or the production or consumption of the species by chemistry. The term $\rho \hat{Y}_k \hat{V}_k$ is the mass flux of the k^{th} species (relative to $\hat{v}$). Note that the total mass flux for the k^{th} species is $\hat{\rho}(\hat{v} + \hat{V}_k)\,\hat{Y}_k$.

Conservation of Energy.

$$\hat{\rho}\,\hat{c}_p \frac{\partial \hat{T}}{\partial \hat{t}} = -\hat{\rho}\hat{v}\hat{c}_p \frac{\partial \hat{T}}{\partial \hat{x}} + \frac{\partial}{\partial \hat{x}}\left(\hat{\lambda} \frac{\partial \hat{T}}{\partial \hat{x}}\right) - \sum_{k=1}^{N} \hat{R}_k M_k \hat{h}_k$$

$$-\hat{\rho} \sum_{k=1}^{N} \hat{c}_{pk}\,\hat{Y}_k \hat{V}_k \frac{\partial \hat{T}}{\partial \hat{x}} \qquad (3)$$

where $\hat{T}$ is the temperature, $\hat{c}_{pk}$ is the specific heat of the k^{th} species, $\hat{c}_p$ is the specific heat of the mixture, $\hat{\lambda}$ is the thermal conductivity of the mixture, and $\hat{h}_k$ is the specific enthalpy of the k^{th} species. So changes in the temperature can be due to the convection of heat, the conduction of heat, the production or consumption of energy by the chemical reactions, and a small amount due to the diffusion of species with different specific heats. We have not written a negligibly small term due to the species gradients (Dufour effect).

The thermal equation of state is given by the ideal gas law

$$\hat{p} = \hat{\rho}\, R\, \hat{T} \sum_{k=1}^{N} \hat{Y}_k / M_k \,, \qquad (4)$$

where R is the gas constant. The caloric equation of state is

$$\hat{h}_k = \hat{h}_k^{\,o} + \int_{\hat{T}_o}^{\hat{T}} \hat{c}_{pk}\, d\,\hat{T} \,, \qquad (5)$$

where $\hat{h}_k^{\,o}$ is the specific enthalpy of the k^{th} species at some reference temperature $\hat{T}_o$. The specific heat of the mixture is given by

$$\hat{c}_p = \sum_{k=1}^{N} \hat{c}_{pk} \hat{Y}_k. \tag{6}$$

From conservation of mass we have the relation

$$\sum_{k=1}^{N} \hat{Y}_k = 1 \tag{7}$$

and

$$\sum_{k=1}^{N} \hat{i}_k \hat{V}_k = 0. \tag{8}$$

The boundary conditions are the following. For $x = -\infty$.

$$\hat{T} = \hat{T}_U \text{ and } \hat{Y}_k = \hat{Y}_{kU}, \ (k=1,2,\ldots N), \tag{9}$$

where $\hat{T}_U$ is the temperature and the $\hat{Y}_{kU}$ are the mass fractions of the original, undisturbed mixture. For $x = \infty$

$$\hat{T} = \hat{T}_B \text{ and } \hat{Y}_k = \hat{Y}_{kB}, \ k=1,2,\ldots N \ . \tag{10}$$

where $\hat{T}_B$ is the temperature and $\hat{Y}_{kB}$ are the mass fractions of the burned mixture. $\hat{T}_B$ is called the adiabatic temperature. Since we are assuming no heat loss to the surroundings, T_B and Y_{kB} depend only on the chemistry, and can be calculated in advance.

In practice, the integration will be over a finite interval $(\hat{x}_L, \hat{x}_R)$, where the flame front will be located roughly in the center of the interval. The boundary conditions (9) will be applied to $\hat{x}_L$. However, the conditions (10) are not convenient. After the flame front, there is a long recombination period before the mixture reaches adiabatic conditions. Choosing the interval of integration long enough to cover the entire recombination zone would be computationally expensive. So normally the flame will be cut off before it reaches adiabatic conditions,

and the weaker boundary conditions

$$\frac{\partial \hat{T}}{\partial \hat{x}} = \frac{\partial Y_k}{\partial \hat{x}} = 0, \quad k=1,2,\ldots N , \tag{11}$$

will be applied at $\hat{x}_R$.

At this stage the partial differential Eq. (1) for the density $\hat{\rho}$ can be eliminated by introducing a new coordinate $\hat{\psi}$ such that

$$\hat{\psi}(\hat{x},\hat{t}) = \int_{\hat{x}_L}^{\hat{x}} \hat{\rho}(\hat{x}', \hat{t})dx' . \tag{12}$$

Then $\frac{\partial \hat{\psi}}{\partial \hat{x}} = \hat{\rho}$ and $\frac{\partial \hat{\psi}}{\partial \hat{t}} = -\hat{\rho}\hat{v} + \hat{m}_o(\hat{t})$, where $m_o(\hat{t}) = \hat{\rho}\hat{v}|_{x=o}$. With this notation Eqs. (2) and (3) become

$$\frac{\partial \tilde{Y}_k}{\partial \tilde{t}} = -\tilde{m}_o \frac{\partial \tilde{Y}_k}{\partial \tilde{\psi}} - \frac{\partial}{\partial \tilde{\psi}} (\tilde{P}\tilde{Y}_k\tilde{V}_k) + \tilde{R}_k M_k/\tilde{\rho} \tag{13}$$

and

$$\frac{\partial \tilde{T}}{\partial \tilde{t}} = -\tilde{m}_o \frac{\partial \tilde{T}}{\partial \tilde{\psi}} + \frac{1}{\tilde{c}_p}\left\{ \frac{\partial}{\partial \tilde{\psi}} \left(\tilde{\rho}\tilde{\lambda} \frac{\partial \tilde{T}}{\partial \tilde{\psi}} \right) - \sum_{k=1}^{N} \tilde{R}_k M_k \tilde{h}_k/\tilde{\rho} - \sum_{k=1}^{N} \tilde{c}_{pk} \tilde{Y}_k \tilde{V}_k \frac{\partial \tilde{T}}{\partial \tilde{\psi}} \right\} , \tag{14}$$

where the tilde variables are functions of $\tilde{t} = \hat{t}$ (sec) and $\tilde{\psi}$ (gm/cm^2).

For numerical convenience, dimensionless forms of Eqs. (13) and (14) are integrated. That is, we will define $T = \tilde{T}/T_\infty$, $t = \tilde{t}/t_\infty$ and $\psi = \tilde{\psi}/\psi_\infty$, where t_∞ and ψ_∞ are chosen so as to obtain reasonable time and space scales for a given flame, and T_∞ is chosen so that T is the same order of magnitude as the larger Y_k's . Then the equations are

$$\frac{\partial Y_k}{\partial t} = - \frac{t_\infty}{\psi_\infty} m_o \frac{\partial Y_k}{\partial \psi} - \frac{t_\infty}{\psi_\infty} \frac{\partial}{\partial \psi} (\rho Y_k V_k) + t_\infty R_k M_k/\rho \quad , \quad (15)$$

and

$$\frac{\partial T}{\partial t} = - \frac{t_\infty}{\psi_\infty} m_o \frac{\partial T}{\partial \psi} + \frac{t_\infty}{c_p} \{ \frac{1}{\psi_\infty^2} \frac{\partial}{\partial \psi} (\rho \lambda \frac{\partial T}{\partial \psi}) \quad (16)$$

$$- \sum_{k=1}^{N} R_k M_k h_k/(\rho T_\infty) - \frac{1}{\psi_\infty} \sum_{k=1}^{N} c_{pk} Y_k V_k \frac{\partial T}{\partial \psi} \} ,$$

where the variables are functions of t and ψ. The boundary conditions are

$$T = T_U \quad , \quad Y_k = Y_{k\,U} \quad , \quad k = 1,2...N \quad (17)$$

at $\psi = \psi_L$ and

$$\frac{\partial T}{\partial \psi} = \frac{\partial Y_k}{\partial \psi} = 0 \quad , \quad k = 1,2...N \quad (18)$$

at $\psi = \psi_R$.

The integration can start from any initial profiles, as long as there is enough energy available to begin the combustion. As shown in Section V, the mass flux m_o through the origin is modified as the integration proceeds so as to keep the flame front in the center of the interval of integration (ψ_L, ψ_R). The integration proceeds until m_o approaches a constant, and the terms $\frac{\partial Y_K}{\partial t}$ and $\frac{\partial T}{\partial t}$ approach zero. The equation for species N is not integrated; Y_N is found from the relation (7).

The burning velocity, S_u, can be calculated at steady state. From the continuity equation $\partial(\rho v)/\partial\psi = 0$, or ρv is constant with respect to ψ. Then we can take any of the equations (15) and integrate over any interval (a,b) to obtain

$$\rho v_k \left[Y_k(b) - Y_k(a)\right] = \psi_\infty \int_a^b \rho^{-1} R_k M_k \, d\psi - \rho Y_k V_k \Big|_a^b . \qquad (19)$$

Then

$$v_k(-\infty) = \frac{\psi_\infty \int_a^b \rho^{-1} R_k M_k \, d\psi - \rho Y_k V_k \Big|_a^b}{\rho(-\infty)\left[Y_k(b) - Y_k(a)\right]} . \qquad (20)$$

At steady state, all of the v_k are equal. Since in the present coordinate system the flame does not move, $v_k(-\infty)$ is the speed at which the unburned gas is approaching the flame. Conversely, $S_v = v_k(-\infty)$ is the speed at which the flame is propagating into the unburned mixture.

The parameters of the equations (15) and (16) are the specific heats c_{pk}, the specific enthalpies h_k, the chemistry production terms R_k, and the diffusion velocities V_k for each species, plus the thermal conductivity λ of the mixture. The c_{pk} and h_k are functions only of the temperature, and can be evaluated very accurately using sixth degree polynomial fits[5]. The chemistry is generally the least well known of the input data. We need to know which species are involved, which reactions can occur, and the rate constant for each reaction. The rate constants k_j will be of the form $a\hat{T}^b \exp(c/\hat{T})$, where either b or c can be zero. We can then find the rate r_j for each reaction by multiplying the rate constant times the concentrations of the reactants (concentrations of the k^{th} species = $\rho Y_k / M_k$). Each R_k is then found by adding the rates of the reactions in which the k^{th} species is a reactant. The transport parameters λ and V_k are in general very complicated functions of temperature and species concentrations. Because of the numerical complexity, various approximations to these quantities have been used. We have the capability to run the code with a number of different levels of approximation. This is discussed in detail in another paper[6].

---

[5] *S. Gordon and B. J. McBride, "Computer Program for Calculation of Complex Chemical Equilibrium Compositions, Rocket Performance, Incident and Reflected Shocks, and Chapman-Jouguet Detonations", NASA-SP-273, 1971, (1976 program version).*

[6] *T. P. Coffee and J. M. Heimerl, "Transport Algorithms for Premixed, Laminar, Steady-State Flames", to be published in Combustion and Flame.*

For future reference, we will discuss here the simplest transport algorithm used in the code. First assume that Ficks law,

$$\hat{Y}_k \hat{V}_k = - \hat{D}_{k\,m} \frac{\partial \hat{Y}_k}{\partial \hat{x}} \tag{21}$$

is valid. This is only strictly true for a binary mixture where thermal diffusion is negligible. $\hat{D}_{k\,m}$ is a multicomponent diffusion coefficient. Equation (21) can be written as

$$\rho\, Y_k\, V_k = - \frac{\rho^2\, D_{k\,m}}{\psi_\infty} \frac{\partial Y_k}{\partial \psi} . \tag{22}$$

As a first approximation, we can take $\rho^2\, D_{k\,m}$ and $\rho\, \lambda$ to be constant. Also, we can assume that $c_p = c_{p\,k}$, k=1,2,...N. Here c_p must be chosen so as to obtain the proper adiabatic temperature T_B. The procedure for choosing these constants and the justification of this approximation is given in reference 6. This level of approximation is reasonably accurate, if the constants are properly chosen.

Equations (15) and (16) now become

$$\frac{\partial Y_k}{\partial t} = - \frac{t_\infty}{\psi_\infty}\, m_o \frac{\partial Y_k}{\partial \psi} + \frac{t_\infty}{\psi_\infty}\, \rho^2 D_{km} \frac{\partial Y_K}{\partial \psi^2} + t_\infty\, R_k M_k / \rho \tag{23}$$

and

$$\frac{\partial I}{\partial t} = - \frac{t_\infty}{\psi_\infty}\, m_o \frac{\partial T}{\partial \psi} + \frac{t_\infty}{c_p} \left\{ \frac{\rho \lambda}{\psi_\infty^2} \frac{\partial^2 T}{\partial \psi^2} - \sum_{k=1}^{N} \frac{R_k M_k\, h_k}{\rho\, T_\infty} \right\}. \tag{24}$$

Equation (5) now simplifies to

$$h_k = h_k^o + c_p (T - T_o) , \tag{25}$$

so the required input parameters for this level of approximation are the constants $\rho^2 D_{k\,m}$, $\rho\, \lambda$, c_p, h_k^o and the functions $R_k(T , Y_1 , \ldots , Y_N)$.

III. THE NUMERICAL METHOD - PDECOL

The package PDECOL, developed by Madsen and Sincovec[7], was used to solve the equations. This package is designed to solve a general system of N nonlinear partial differential equations of at most second order on a finite interval. In our coordinate system the appropriate form is

$$\frac{\partial \vec{u}}{\partial t} = \vec{f}\,(t\ ,\ \psi\ ,\ \vec{u}\ ,\ \vec{u}_{\psi}\ ,\ \vec{u}_{\psi\psi})\ , \tag{26}$$

where

$$\begin{aligned} \vec{u} &= (Y_1\ ,\ \ldots,\ Y_{N-1},\ T) \\ \vec{u}_{\psi} &= \left(\frac{\partial Y_1}{\partial \psi},\ldots,\ \frac{\partial Y_{N-1}}{\partial \psi}\ ,\ \frac{\partial T}{\partial \psi}\right)\ , \\ \vec{u}_{\psi\psi} &= \left(\frac{\partial^2 Y_1}{\partial \psi^2}\ ,\ldots,\ \frac{\partial^2 Y_{N-1}}{\partial \psi^2}\ ,\ \frac{\partial^2 T}{\partial \psi^2}\right)\ . \end{aligned} \tag{27}$$

Fairly general boundary conditions of the form

$$\vec{b}\ (\vec{u},\ \vec{u}_{\psi}) = \vec{Z}\ (t) \tag{28}$$

are allowed, where $\vec{b}$ and $\vec{Z}$ are arbitrary vector valued functions with N components. Each solution component is assumed to be a known function of ψ at the initial time $t = t_o$. That is, $Y_k(t_o, \psi)$, k=1,2,...N-1, and $T\ (t_o, \psi)$ are known functions. The initial conditions must be consistent with the boundary conditions.

The spatial discretization is accomplished by finite element collocation methods based on B-splines[8]. The user must supply a set of

---

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

[8] *C. de Boor, "Package for Calculating with B-Splines", Siam, J. Numer. Anal. 14, 441-472, (1977).*

NB breakpoints, that is, a set of strictly increasing locations where the polynomials are joined. He must also supply the order KORD of the splines and the number of continuity conditions, NCC, to be applied at the breakpoints. That is, if NCC=1, the approximating function is continuous; if NCC=2, it is continuous and smooth; and so on. PDECOL then generates a set of NC = KORD(NB-1) - NCC(NB-2) basis functions and collocation points. The basis functions $B_i(\psi)$ are piecewise polynomials of order KORD - 1. The basic assumption is that the solution can be written in the form

$$u_k = \sum_{i=1}^{N} c_k^{(i)}(t) \quad B_i(\psi) \ , \ k=1,\ldots N \ , \tag{29}$$

where the basis functions $B_i(\psi)$ 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 expansions above satisfy the given boundary conditions and that they satisfy the partial differential equations exactly at the (N-2) interior (collocation) points. Since by definition a B-spline is zero except over a small interval, at any collocation point no more than KORD of the B-splines are non-zero. So the system of ODE's for the coefficients $c_k^{(i)}$ will not be fully coupled.

The boundary conditions (28) must also be changed into ordinary differential equations. This can be done by taking the derivative with respect to time, which results in

$$\sum_{j=1}^{N} \left[\frac{\partial b_k}{\partial u_j} \frac{\partial u_j}{\partial t} + \frac{\partial b_k}{\partial u_{j\psi}} \frac{\partial u_{j\psi}}{\partial t} \right] = \frac{dZ_k}{dt} \ , \quad k=1,2\ldots N. \tag{30}$$

When equation (29) is substituted into equation (30), a set of ODE's in the $c_k^{(i)}$ results.

Unlike a finite difference code, the program can be run with no boundary condition at either the left or the right boundary. The program simply collocates at the boundary, using the same procedure as for the interior points. It must be, of course, the user's responsibility to define a mathematically meaningful PDE problem.

This system of ODE's is integrated in time, using a variant of the Gear stiff integrator[9]. This is a fully implicit, predictor-corrector method. The required banded Jacobian is generated internally by the program. Once the integrator has reached a desired output time, the values of Y_k and T can be obtained for any ψ by substituting into the expansions (29). The accuracy of the time integration is determined by a user supplied error tolerance ε.

For the actual integration, the code requires the values of the basis functions and the first two space derivatives only at the collocation points. Since computing these values is complicated, this is done once at the beginning of the program and the values are saved.

In general, a user must supply a main program and three subroutines. The program MAIN sets the values of the required parameters, calls the integrator, and writes any desired output. The subroutine UINIT gives the initial conditions. Given a collocation point ψ, the subroutine must return the vector $\vec{u}(t_o, \psi)$. The subroutine F evaluates the function $\vec{f}$ given by Eq. (26). Given t, ψ, u, u_ψ, and $u_{\psi\psi}$, the routine must return the vector $\partial u/\partial t$. The program automatically converts this to the corresponding set of ODE's involving the $c_k^{(i)}$. The subroutine BNDRY specifies the boundary conditions. That is, given t, ψ, $\vec{u}$, and $\vec{u}_\psi$, the subroutine returns the quantities $\partial b_k/\partial u_j$, $\partial b_k/\partial u_j$ $\partial Z_k/\partial t$, $j,k = 1,2\ldots N$, for both ψ_L and ψ_R.

IV. MODIFICATIONS OF PDECOL

The time integration is controlled by a user supplied error tolerance ε. Single step error estimates divided by $CMAX_k^{(i)}$ will be kept less than ε in the root-mean-square norm. In PDECOL, $CMAX_k^{(i)}$ is initially set to the maximum of $|c_k^{(i)}|$ and 1.0. Thereafter, $CMAX_k^{(i)}$ is the largest value of $c_k^{(i)}$ seen so far, or the initial $CMAX_k^{(i)}$ if that is larger. However, this error criterion does not produce the desired accuracy for flame simulations, because radical species with small concentrations will control the flame, and must be computed accurately. But mass fractions much smaller than 1.0 will not be computed accurately using the original PDECOL criterion. An alternate criterion is the purely relative error criterion, that is, $CMAX_k^{(i)} = |c_k^{(i)}|$. This criterion is also unacceptable because some species at some locations will approach zero, and excess computation will result

[9] *A. C. Hindmarsh, "Preliminary Documentation of GEARIB: Solution of Implicit Systems of Ordinary Differential Equations with Banded Jacobian", Rep. UCID-30130, Lawrence Livermore Laboratory, (1976).*

in accurately computing these negligible concentrations. Consequently a semi-relative error control is used. $CMAX_k^{(i)}$ is chosen as the maximum of $c_k^{(i)}$ and a user supplied parameter SREC. Thus, mass fractions less than SREC will be computed less accurately. We have normally used $SREC=10^{-6}$.

The original program PDECOL is fully implicit. That is, it generates a set of NO = N X NC ordinary differential equations of the form

$$A \frac{d\vec{c}}{dt} = \vec{g}(t, \vec{c}) \quad , \tag{31}$$

where $c = (c_1^{(1)} \ldots c_N^{(1)}, c_1^{(2)} \ldots c_N^{(2)} \ldots)$. The resulting Jacobian $\partial\vec{g}/\partial\vec{c}$ is a banded NO by NO matrix. The band width is 3 X ML+1, where ML = N (KORD - 1) - 1. To advance a time step, the program first computes an explicit predictor for the values of the $c_k^{(i)}$ at the next time step. Then it solves a set of linear algebraic equations involving the Jacobian to correct the values. For a stiff system, such as one involving chemistry, this allows time steps orders of magnitude larger than those of an explicit method[9]. The drawback is the storage and execution time required to work with the Jacobian.

For example, consider a system where KORD = 4, N = 10, and NC = 20. The bandwidth of the Jacobian is 88, and we have a system of 300 ODE's. We then have 26,400 possible non-zero elements in the banded matrix. The amount of storage required also increases rapidly. For instance, if N = 20 instead of 10 the bandwidth will be 178, and we will require 106,800 words to store the Jacobian. Since we want to be able to solve systems at least this large, the storage requirements become almost prohibitive. In addition, solving such large systems of linear algebraic equations is very time consuming.

To avoid this problem, we essentially uncouple the partial differential equations and solve them successively. The basic procedure is illustrated below. Suppose we are at time t_n and we want to advance a time step to t_{n+1}. We first integrate the equation for Y_1, under the assumption that $Y_2, \ldots, Y_{N-1}$, T are constant at their t_n values. This uncouples the first PDE from the system. Subsequently, we integrate the second PDE for Y_2, using the new value of Y_1 at t_{n+1}, and the old values for the other variables. Continuing this process, we finally solve for $T(t_{n+1})$, using updated values for all the mass fractions.

In general, this method is restricted to smaller step sizes than a fully implicit method. However, as steady state is approached, all the variables approach constants with respect to time, and the temporal coupling vanishes.

Now consider how this assumption affects the associated system (31) of ODE's. In the Jacobian, we will have

$$\partial g_k^{(i)}/\partial c_\ell^{(j)} = 0 \quad \text{if } k \neq \ell.$$

That is, changes in the ℓ^{th} PDE will not affect the k^{th} PDE if $k \neq \ell$. So most of the terms in the Jacobian will become zero.

The remaining nonzero elements are not in banded form. However, we can accomplish this by rearranging the vector $\vec{c} = (c_1^{(1)}, c_1^{(2)}, \ldots c_1^{(NC)}, c_2^{(1)}, \ldots c_2^{(NC)} \ldots)$. That is, the coefficients for each PDE are grouped together instead of the coefficients for each collocation point.

The Jacobian matrix can now be decomposed into N smaller NC X NC matrices on the main diagonal. Moreover, the band width of these matrices is only 3XML+1, where ML = KORD - 2. The Jacobian is essentially N separate Jacobians, each for one PDE with NC collocation points.

The savings in storage space is dramatic. For our previous example of KORD = 4, N = 10, and NC = 30, we have 2100 nonzero elements instead of 26,400. If N = 20, we have only 4200 nonzero elements instead of 106,800.

The following procedure is used to actually integrate a time step. The predictor of the predictor-corrector method is used to obtain first estimates for all the $c_k^{(i)}$. Then the corrected $c_1^{(i)}$ are computed, using the first small Jacobian and assuming that the other $c_k^{(i)}$ do not change. Then the corrected values for $c_1^{(i)}$ and the predicted values for $c_k^{(i)}$, $k > 2$, are used, and the corrected $c_2^{(i)}$ are computed. This process continues through the N PDE's. The estimated error is calculated. If necessary, the above procedure is iterated. It is more efficient to use the predicted values rather than the values at the previous time step. In using the procedure, it is more efficient to integrate the minor species first, since they change most rapidly, then the major species, and finally the temperature.

The above procedure is similar to one developed by Spalding and Stephenson for use in a finite difference code[10].

[10] *D.B. Spalding and P.L. Stephenson, "Laminar Flame Propagation in Hydrogen + Bromine Mixtures", Proc. R. Soc. Lond. A. 324, 315-337 (1971).*

PDECOL was rewritten to use the successive calculation method. This also means that we must change the user supplied routines F and BNDRY. Because of the rearrangement of the vector $\vec{c}$, the core integrater will call F first at each collocation point, asking for the values of $\partial Y_1/\partial t$. It will then repeat for each PDE. The routine F is written so as to return only the desired time derivative.

Since the problem can no longer directly handle coupling terms, the boundary conditions (28) must be uncoupled, that is, the boundary conditions must be of the form

$$b_k(u_k, u_{k\psi}) = Z_k(t) \quad . \tag{32}$$

The rewritten subroutine BNDRY evaluates the quantities $\partial b_k/\partial u_k$, $\partial b_k/\partial u_{k\psi}$, $\partial Z_k/\partial t$, $k = 1,2...N$.

Comparisons of the execution time of the fully implicit method versus the successive calculation method have not been made. However, our main purpose was not to reduce the execution time but to reduce the storage requirements. This iterative procedure accomplishes this goal and simultaneously gives accurate results in reasonable run times.

V. ADDITIONS TO PDECOL

The computer program PDECOL requires a set of user supplied subroutines. We have written a set of subroutines that casts the equations into a computationally efficient form and which generates the required output. These routines are the first seven listed in the appendix; namely MAIN, F, UINIT, FLSP, BNDRY, BKPT, and RT. In addition, several auxiliary programs are mentioned, but without providing a listing. These routines generate sets of input data or actual subroutines that are used frequently, or analyze the output of the flame code.

The code can be run with several different options. We first describe the case of an unbounded flame (NBURN = 0) using the simplest constant transport algorithm (NTRAN = 1). There is no information available about the flame speed or the solution profiles (NSTART = 1). Later in this section we describe restarting the integration (NSTART = 2), and using a more complicated transport algorithm (NTRAN = 2). Burner stabilized flames (NBURN = 1) are discussed in the next section.

Initially, a chemistry scheme must be chosen. In the appendix an H_2 - O_2 - N_2 system is used, with nine chemical species (H, OH, O, HO_2, H_2O_2, H_2, O_2, H_2O, N_2) and a set of thirty reactions involving these species, each with a rate constant of the form $a\,T^b \exp(c/T)$ (either b or c can be zero). This information is used as input to an auxiliary

code. This code writes the subroutine RT that computes the chemistry terms $R_k M_k/\rho$. The procedure is analogous to that used to write the transport subroutines[6]. This subroutine is attached to the flame code in the job stream and can be used for any problem involving this set of kinetics.

The actual subroutine RT has three main parts. The rate constants are evaluated for the current temperature and stored in the vector RK. Since the code uses a successive calculation method, the subroutine will be called N times at each time step and each collocation point. The rate constants are only recomputed if the temperature has been changed. Otherwise this section is skipped. Each rate constant is multiplied by the concentrations of the appropriate reactants divided by ρ to obtain r_j/ρ. The terms $R_k M_k/\rho$ (stored in the vector R) are calculated by adding the rates for the reactions in which Y_k is a product, subtracting the rates for which Y_k is a reactant, and multiplying by M_k.

The choice of a transport algorithm determines the form of the subroutine F. The version given in the appendix is for constant transport. Common statements are used to make the appropriate constants available for all subroutines. The chemistry terms required are found by calling RT.

The initial temperature T_U and mass fractions Y_{kU} of the unburned gas are input quantities. Another auxiliary code determines the adiabatic temperatures T_B and mass fractions Y_{kB}, and the constants $\rho^2 D_{km}$, h_k^o, $\rho\lambda$, and c_p needed by the constant transport algorithm (see reference 6). This information is saved on a data file and attached to TAPE 11 when the code is run.

The remaining data necessary to run the program is read in on cards (TAPE 5). The pressure p and the normalizing constants t_∞, ψ_∞, and T_∞ are specified. The option parameters NSTART, NTRAN, and NBURN are chosen. The numerical parameters required by the code are specified; that is, ψ_L and ψ_R, the final integration time t_{FINAL}, the time integration error control parameters ε and SREC, and a set of breakpoints.

The major difficulty in efficiently solving the flame equations is choosing an appropriate set of breakpoints. These must be close enough that spatial errors are minimized and yet not so dense that one's computer resources are exceeded. The breakpoints should be densest in the flame front, where the gradients are very steep.

The technique in the code is to use a static mesh, with the breakpoints most closely spaced near the center of the interval of integration. The flame front is then forced to remain near the center of the interval. This is done by adjusting m_o, the mass flux through the origin. At steady state, the mass flux through the flame is a constant. So m_o is

iteratively modified to match the steady state mass flow through the flame. This leads to a coordinate system in which the flame front is at rest. The transient behavior can cause the flame front to drift away from the center. This can also be corrected by modifying m_o.

This method requires a procedure for tracking the flame front. To do this the position of a specific temperature T_{cn} is monitored. The code attempts to keep this temperature located at the center of the interval of integration. As a heuristic rule this temperature is defined by

$$T_{cn} = T_U + 0.4\ (T_B - T_u) \quad . \tag{33}$$

The average of T_U and T_B is not used. This is because the mixture will not normally reach the adiabatic temperature at the end of the flame front. Rather, there is a radical overshoot, and the recombination of these radicals will very slowly raise the temperature. T_{cn} as defined will usually be close to the center of the flame front. The details of this iterative procedure are discussed in reference 11.

So the breakpoints may be chosen to be densest in the center of the interval of integration. However, it is tedious to have to choose an entire breakpoint sequence for each problem. We have developed a procedure to generate an appropriate type of breakpoint sequence from a small number of parameters. By varying three parameters, a wide variety of breakpoint sequences can be generated.

The user must supply NINT, the number of intervals (NB = NINT+1), NCN, the number of intervals of equal length that will be at the center of the interval, and FC, the ratio between the longest intervals (on the boundaries) and the shortest intervals. Also let L be the total length of the interval of integration, that is, $L = \psi_R - \psi_L$. The program generates a set of intervals whose lengths increase by a constant factor α, where

$$a = \log^{-1}\ [2(\log FC)/(NINT-NCN)] \quad . \tag{34}$$

The common length LC of the NCN center shortest intervals is

$$LC = L/[NCN + 2\ \alpha(\alpha^{(NINT-NCN)/2}-1)/(\alpha-1)]. \tag{35}$$

The procedure can best be seen by example. Suppose we have $\psi_L = 0$,

---

[11] *T.P. Coffee and J.M. Heimerl, "A Method for Computing the Flame Speed of a Laminar, Premixed, One Dimensional Flame", BRL Technical Report ARBRL-TR-02212, January 1980.*

ψ_R = 10, NINT = 12, NCN = 4, and FC = 6. Then L = 10, α = 1.5651, and LC = .3155. The resulting breakpoint sequence is given in Table 1. Note that the 4 center intervals are of the same length, the length of the intervals then increases by a factor of α, and the two intervals by the boundaries are 6 times as long as the central intervals. So the procedure automatically generates a set of breakpoints that are closest together near the center, with the spacing increasing smoothly toward the boundaries.

Some experimentation is necessary to choose the proper value of the above parameters. However, choosing NINT = 12, NCN = 4, and FC between 4 and 8 has worked in most of the cases we have tried. Normally we experiment using the constant transport algorithm, and then use a more realistic transport subroutine once we have a good breakpoint sequence.

TABLE 1. THE SET OF BREAKPOINTS GENERATED BY L = 10, NINT = 12, NCN = 4 and FC = 6.

Breakpoints	Interval Lengths
0.0	
	1.8929
1.8929	
	1.2094
3.1024	
	.7728
3.8752	
	.4937
4.3689	
	.3155
4.6844	
	.3155
5.0000	
	.3155
5.3155	
	.3155
5.6310	
	.4937
6.1247	
	.7728
6.8975	
	1.2094
8.1069	
	1.8929
10.0000	

The breakpoint sequence is written by the subroutine BKPT. It also generates a larger set of evaluation points by interpolating between the breakpoints. Using Eq. (29), the subroutine VALUES can evaluate the Y_k and T at this larger set of points. This information is useful in generating detailed output, such as graphs, or in performing numerical integrations (see below).

The subroutine UINIT writes the initial profiles, where

$$\psi_1 = \psi_L + 0.24\ (\psi_R - \psi_L)$$
$$\psi_2 = \psi_L + 0.64\ (\psi_R - \psi_L) \tag{36}$$

and

$$Y_k(t_o,\psi) = \begin{cases} Y_{kU}, & \psi_L \le \psi \le \psi_1. \\ y_{kU} + (y_{kB} - y_{kU}) \sin\left[\frac{\pi}{2}\left(\frac{\psi-\psi_1}{\psi_2-\psi_1}\right)^2\right]^2, & \psi_1 \le \psi \le \psi_2 \\ Y_{kB}, & \psi_2 \le \psi \le \psi_R. \end{cases} \tag{37}$$

The temperature T is defined similarly.

This particular definition will give us $T = T_{cn}$ at $\psi_{cn} = 0.5(\psi_L+\psi_R)$. The choice of the particular function that defines the Y_k and T between ψ_1 and ψ_2 is not important. We have used a straight line with success, but defining a smooth function is slightly more efficient.

A requirement of PDECOL is that the initial profiles satisfy the boundary conditions, in our case given by Eqs. (17) and (18). The above initial profiles have the proper unburned values at ψ_L and are constant (space derivative zero) near ψ_R.

For this case the subroutine BNDRY has a simple form. At ψ_L, the subroutine returns the values $\partial b_k/\partial u_k = 1$, $\partial b_k/\partial u_{k\psi} = 0$, and $\partial Z_k/\partial t = 0$. At ψ_R, the conditions are $\partial b_k/\partial u_j = 0$, $\partial b_k/\partial u_{k\psi} = 1$, and $\partial Z_k/\partial t = 0$.

To begin the integration, a starting value for m_o, the mass flux through the origin is required. To do this, the code ignores the time dependent terms, assumes that the mass flux ρv is constant, and uses Eq. (20) to obtain a value of ρv. This gives a reasonable starting value for m_o.

The evaluation of Eq. (20) is carried out in the subroutine FLSP. The integral is approximated using the trapezoidal rule, where the integrand $R_k M_k/\rho$ is calculated at the evaluation points. The flame speed $v_k(-\infty)$ is evaluated for each species k and for several intervals (a,b), where $a = \psi_L$. The initial value for m_o is based on species N-1 integrated over the entire interval (ψ_L, ψ_R).

The integration is performed over several intervals (a,b) because of the behavior of the minor species. These are normally close to zero at ψ_L, reach a peak in the flame front, and are close to zero again at ψ_R. For these species, integrating from the left boundary to the flame front is much more accurate.

The chemistry terms in Eq. (20) are found by calling RT. The diffusion terms for this case are found using Ficks law, Eq. (22). Since we are assuming that $\rho^2 D_{km}$ is constant, this value is stored in the vector R2D. The subroutine FLSP also computes the x values from the relation

$$x(t, \psi) = \psi_\infty \int_{\psi_L}^{\psi} \rho^{-1} \, d\psi, \tag{38}$$

using the trapezoidal rule, and computes an estimate of the flame thickness.

As the time integration proceeds, control returns to MAIN at a series of output times. The present spatial location of the temperature T_{cn} is found, and this is used to find the average value of the mass flux since the last output time. The function $m_o(t)$ is redefined at these times so as to keep the flame front in roughly the same position. The output times are chosen by the program so that the flame front will not drift too far between evaluations. Also FLSP is called so the user can see if the flame speeds computed from the different species profiles are approaching a common value.

At the final time t_{FINAL}, FLSP also writes an output file. It consists of all the evaluation points ψ_i, the corresponding x_i, the

values of Y_k and T, plus their first and second derivatives with respect to ψ. This file can be attached to an output routine. By also attaching F and RT, we can compute and print out any quantity in the steady state solution in which we are interested. This file can also be attached to a graphics routine.

Similarly, the program MAIN writes a restart file. This consists of the present location of T_{cn}, the present value of m_o, and the values of Y_k and T at the collocation points. This file can be used to restart the time integration. It is attached to TAPE1, and the parameter NSTART is set equal to 2. The input parameters read in on cards can now be changed if desired. UINIT will translate these input values to center the flame front, and will use interpolation to find the appropriate values of the starting profiles at the new collocation points. The old value of m_o is used to start the integration.

So far only the constant transport case (NTRAN = 1) has been considered. For more realistic algorithms F is written by an auxiliary code and attached to the flame code (NTRAN) = 2). Several different levels of approximation can be used.[6]

In these more complicated algorithms, the diffusion velocities V_k are coupled, and they must be computed simultaneously. But since the code uses successive calculation, only the value of one V_k is required on each call to F. Because the computation is time-consuming, it is preferable not to compute the V_k N times at each time step for each collocation point. To economize computer time, all the thermodynamic and transport quantities required at all the collocation points for k = 1 are computed, and stored in vectors. For k > 1, we use the same values, even though some of the Y_k terms have changed slightly. Only the chemistry terms are reevaluated. Since the chemistry normally changes much more rapidly than the transport, this will still be a good approximation.

In generating an approximation to the Jacobian (using finite differences) it is necessary to recompute the transport each time F is called, since what is of interest is the effect of changes in Y_k and T on the time derivatives. Ignoring the changes in transport leads to an inaccurate Jacobian. However, the rate constants k_j and the thermodynamic quantities c_p and h_k need be recomputed only if T is changed, since they only depend on temperature.

As will be seen in the next section it will still be useful to formulate the mass flux in a Ficks law form Eq. (22). For the more complicated transport subroutines, we define

$$\rho,2\ D_{km} = - \frac{\psi_\infty \rho Y_k V_k}{\partial Y_k / \partial \psi} . \qquad (39)$$

This new quantity $\rho^2 D_{km}$ is no longer constant with respect to space or time.

VI. BURNER STABILIZED FLAMES

So far we have only discussed flames that propagate in an unbounded medium. In actual experiments, the flame will usually be stabilized by a burner. It is useful to be able to model this type of experiment so that we can compare experimental and calculated profiles.

Our basic idealization is of a cylindrical, porous plug burner. The premixed gas exits the plug with a constant velocity v, but the plug prevents back diffusion of the products into the burner. When the gas is ignited, a flame front develops and propagates toward the burner. The gas velocity v at the burner must be less than the flame velocity, or the flame will be blown off the burner. In the model this will look like an unbounded flame. In an experiment the flame will be extinguished by the surrounding atmosphere. As the flame approaches the burner, the burner surface acts as a heat sink for the flame. The loss of heat slows down the flame velocity, until the flame stabilizes near the surface of the burner at the gas velocity. As the gas velocity v is decreased, the flame loses more heat to the burner, and stabilizes closer to the burner surface. If v is made too small in an experiment, the flame can flash back into the burner.

For this kind of burner, air is entrained along the outside edges of the flame. However, the center of the flame will correspond closely to a premixed, laminar, one-dimensional flame.

The only change in the equations is in the left boundary condition at the surface of the burner. Because of back diffusion from the flame to the burner surface, the mass fractions of the unburned mixture are not conserved. However, due to conservation of mass, the mass flux fractions, defined as

$$\epsilon_k = \frac{\rho Y_k v + \rho Y_k V_k}{\rho v} = Y_k + \frac{\rho Y_k V_k}{\rho v} \tag{40}$$

are conserved. Within the burner, the diffusion velocities are zero, and $\epsilon_{kU} = Y_{kU}$. So the appropriate boundary conditions at the burner surface are

$$\epsilon_k = Y_{kU}. \tag{41}$$

The boundary conditions for the temperature equation depends on how heat

is extracted. We assume that the burner is maintained at a constant temperature. Then the boundary condition is the same as for an unbounded flame,

$$T = T_U. \tag{42}$$

This idealization is discussed by Hirshfelder, Curtiss and Bird[12].

To implement this in the flame code, the input parameter NBURN is set equal to 1. The fluid velocity at the burner must be specified. Then m_0 is a predetermined constant instead of an adjustable parameter. All the other input data remains the same. However, the code will handle this data differently.

The breakpoints are generated differently, since the flame front will be near the left boundary instead of in the center of the interval of integration. NCN is the number of intervals of equal length at the left boundary. FC is the ratio between the longest interval (at the right boundary) and the shortest interval (at the left boundary). The constant α is now defined by

$$\alpha = \log^{-1} [(\log FC)/(NINT-NCN)]. \tag{43}$$

The common length LC of the NCN shortest intervals is now

$$LC = L/[NCN + \alpha(\alpha^{(NINT-NCN)}-1)/(\alpha-1)]. \tag{44}$$

As an example, consider the same input parameters that we used for an unbounded flame; $\psi_L = 0$, $\psi_R = 10$, NINT = 12, NCN =4, and FC = 6. Then L = 10, α = 1.2510 and LC = .4181. The resulting breakpoint sequence is given in Table 2. The shortest intervals are near the burner surface, where fairly steep gradients are expected.

In general, it is possible to choose a shorter interval of integration L for a burner stabilized flame. There is no longer a need for a relatively long interval to the left of the flame front in order to approach the unburned conditions. The number of breakpoints can then also be reduced.

The initial profiles are also chosen differently. The code sets

[12] *J.O. Hirshfelder, C.F. Curtiss, and R.B. Bird,* op. cit., *pp. 761-763.*

TABLE 2. THE SET OF BREAKPOINTS GENERATED BY L=10, NINT = 12, NCN = 4 and FC = 6 (NBURN = 1)

Breakpoints	Interval Lengths
0	
	.3458
.3458	
	.3458
.6916	
	.3458
1.0374	
	.3458
1.3832	
	.4326
1.8159	
	.5412
2.3571	
	.6771
3.0342	
	.8471
3.8812	
	1.0597
4.9409	
	1.3257
6.2666	
	1.6585
7.9251	
	2.0749
10.0000	

$$\psi_1 = \psi_L + 0.1\ (\psi_R - \psi_L)$$

$$\psi_2 = \psi_L + 0.3\ (\psi_R - \psi_L) \quad , \tag{45}$$

and then uses Eq. (37) as before. This flame front is more likely to be close to the position of the final steady state stabilized flame. Also, the flame front is chosen to be narrower. If Eq. (36) is used, the initial transient flame velocity may be less than the velocity of the fluid. The flame then drifts toward the right, and can be carried completely outside the interval of integration before it stabilizes . Choosing a steeper flame front leads to a larger initial flame velocity.

The boundary condition, Eq. (41), can cause some difficulty. It is fairly straightforward for the constant transport case. Using Fick's law, Eq. (22), it can be rewritten as

$$Y_k - \left(\frac{\rho^2 D_{km}}{\psi_\infty} m_o \frac{\partial Y_k}{\partial \psi} \right) \Big|_{\psi_L} = Y_U \quad . \tag{46}$$

Recall that PDECOL converts this to a time dependent equation by taking the time derivative. In the form used by PDECOL, Eq. (46) becomes

$$\left[\frac{\partial Y_k}{\partial t} - \frac{\rho^2 D_{km}}{\psi_\infty m_o} \frac{\partial}{\partial t} \left(\frac{\partial Y_k}{\partial \psi} \right) \right] \Big|_{\psi_L} = 0 \; . \tag{47}$$

So the subroutine BNDRY must now return the values $\partial b_k/\partial u_k = 1$, $\partial b_k/\partial u_k = - \rho^2 D_{km}/\psi_\infty m_o$, and $\partial Z_k/\partial t = 0$ at ψ_L for $k = 1,2 \ldots N - 1$. The mass flux fractions will then have the proper values as the integration proceeds.

For more realistic transport algorithms the mass flux $\rho Y_k V_k$ is a complicated function of all the mass fractions. But by using Eq. (39) to define $\rho^2 D_{km}$, the boundary condition can be put in the same form as Eq. (46), except that $\rho^2 D_{km}$ is not a constant. Taking the time derivative we obtain

$$\left\{ \frac{\partial Y_k}{\partial t} - \frac{\rho^2 D_{km}}{\psi_\infty m_o} \frac{\partial}{\partial t} \left(\frac{\partial Y_k}{\partial \psi} \right) - \frac{1}{\psi_\infty m_o} \frac{\partial}{\partial t} \left(\rho^2 D_{km} \right) \frac{\partial Y_k}{\partial \psi} \right\} \Big|_{\psi_L} = 0. \tag{48}$$

The time derivative of $\rho^2 D_{km}$ can not be evaluated. It is not possible to write it as an explicit function of the Y_k and $\partial Y_k/\partial \psi$. Moreover, since the code uses successive calculation, cross coupling terms between the different species are not allowed. By necessity, the last term in Eq. (48) must be ignored.

It is possible to just use the expression

$$\left\{ \frac{\partial Y_k}{\partial t} - \frac{\rho^2 D_{km}}{\psi_\infty m_o} \frac{\partial}{\partial t} \left(\frac{\partial Y_k}{\partial \psi} \right) \right\} \Big|_{\psi_L} = 0 \tag{49}$$

as the boundary condition. At steady state, the mass flux fractions will approach constants. But because of the transient behavior of $\rho^2 D_{km}$,

the mass flux fractions will converge to incorrect values.

What is required is a correction term that will cause the code to converge to the proper steady state limit. Moreover, it must be of a form that can be used in the code. This can be done by choosing the boundary condition

$$\left\{ \frac{\partial Y_k}{\partial t} - \frac{\rho^2 D_{km}}{\psi_\infty m_o} \frac{\partial}{\partial t} \left(\frac{\partial V_k}{\partial \psi} \right) \right\}_{\psi_L} = \frac{\partial Z}{\partial t} , \tag{50}$$

where

$$\frac{\partial Z}{\partial t} = \frac{Y_{kU} - \varepsilon_k|_{\psi_L}}{0.1 \; t_{FINAL}} . \tag{51}$$

The rationale is that $\rho^2 D_{km}$ changes rather slowly. As it changes the mass flux fraction at ψ_L will change slowly from the proper value. We use a heuristically defined function $\partial Z/\partial t$ to modify the value of ε_k until it again equals Y_k. So as the time integration proceeds, the value of ε_k at ψ_L will vary slightly, but will approach the proper steady state limit.

It is necessary to check that the boundary conditions are still consistent with the initial conditions. For the initial profiles defined by Eq. (37) this will be the case. The mass fractions Y_k are constant near ψ_L, so $\partial Y_k/\partial \psi$ at ψ_L will be zero. Also $\partial T/\partial \psi$ at ψ_L is zero. For any transport algorithm all the V_k will be zero at ψ_L, and $\varepsilon_k = Y_k$. So the initial profile is consistent with the boundary conditions.

When the integration is restarted (NSTART = 2), the space derivatives at ψ_L are normally non-zero. It is necessary to modify the way PDECOL determines the initial values of $c_k^{(i)}$. Normally this is done by the subroutine INITAL. It calls the user supplied subroutine UINIT to obtain the values $u_k(t_0, \psi_j)$, where the ψ_j, $j = 1,2...$ NC are the collocation points. Then by substituting into the expansion (29), it obtains a set of linear algebraic equations of the form

$$u_k(t_o, \psi_j) = \sum_{i=1}^{NC} c_k^{(i)}(t_o) \; B_i(\psi_j) \quad k = 1,2...N , \quad j = 1,2...NC . \tag{52}$$

These systems of equations are solved to obtain the initial $c_k^{(i)}(t_o)$.

For a burner stabilized flame, the appropriate values of the mass fractions at $\psi_1 = \psi_L$ are not known but the values of the mass flux fractions ϵ_k are. Thus, we use the boundary condition (46), substitute the expansion (29), and obtain

$$Y_{kU} = \sum_{i=1}^{NC} c_k^{(i)}(t_o) \left[B_i(\psi_1) - \frac{\rho^2 D_{km}}{\psi_\infty} \frac{\partial B_i(\psi_1)}{\partial \psi} \right], \quad k=1,2...N-1. \quad (53)$$

The values of $\rho^2 D_{km}$ from the last run of the code are used. These values are also saved in the restart file. So for a burner stabilized flame, the subroutine INITAL has been changed so that it generates this new set of equations for j = 1, k = 1,2...N - 1.

Note that the above procedure is not strictly necessary. Since we have a correction term $\partial Z/\partial t$, the boundary conditions will approach the proper values, even if they are incorrect at the start of the integration. But it is more efficient to begin with the boundary conditions as accurate as possible.

To actually make comparisons with experiments, some of the boundary conditions may have to be further modified. For instance, evidence exists that hydrogen atoms H combine very rapidly on the burner surface to form H_2.[13] For practical purposes, we consider this to be instantaneous. Then if our first species is H and our second species is H_2, their boundary conditions are

$$Y_1 = 0 \quad \text{at } \psi_L \quad (54)$$

and

$$\epsilon_1 + \epsilon_2 = Y_{2u} \quad \text{at } \psi_L \, . \quad (55)$$

These changes can easily be made in UINIT and INITAL.

As the time integration proceeds, control returns to the routine MAIN at five equally spaced output times. The code calls subroutine FLSP so the integration process can be monitored. There is no need to adjust m_o, since this is now a predetermined constant.

---

[13] *J. Warnatz, "Calculation of the Structure of Laminar Flat Flames III: Structure of Burner-Stabilized Hydrogen-Oxygen and Hydrogen-Fluorine Flames", Verlog Chemie GmbH, BdB 8/78 E 4018.*

VII. NUMERICAL CONSIDERATIONS

This code has been applied to the H_2-O_2-N_2 system, with nine chemical species.[6] An earlier version of the code has been applied to the ozone system, with three species[14]. Several calculations have been performed for methane-air flames. In each case the code integrates in time until the steady state solution has been reached. What is desired is accurate values for the flame speed and accurate temperature and species profiles. Results are not discussed here, since they are given in the papers referenced above. Instead, we will discuss some of the considerations necessary to obtain accurate results, based on our experiences with the above systems.

The flame speed can be calculated from any of the species profiles using Eq. (20). Normally the time integration proceeds until the flame speeds based on the major species (reactants and products) agree to within a small fraction of a percent. At this point the solution is very close to steady state. The values of the flame speed calculated from the minor species (radicals) are very sensitive, and require more accuracy (both spatial and temporal) in order to achieve the same agreement.

In general, it is easier to obtain an accurate solution for a fast flame than for a slow flame. The slow flame requires many more integration steps before all the oscillations die out and a steady state solution is achieved.

To obtain an accurate solution, both the spatial and temporal accuracy must be sufficient. If the temporal accuracy is low, the calculated flame speeds can oscillate around the correct value. If the spatial accuracy is low, the problem may come very close to convergence, but to the wrong value, since the communication between the collocation points is inadequate. As a consequence of the spatial or temporal accuracy becoming too small, the integration may break down completely.

Spatial accuracy is more important in the flame front, where we have steep gradients. Hence, our algorithms for choosing breakpoints concentrates them in the region to be occupied by the flame front.

The need for spatial and temporal accuracy is connected. Suppose the number of breakpoints is increased, but the temporal error tolerance ε is not decreased. The calculated solutions will oscillate, and eventually the solution will break up. On the other hand, suppose ε is decreased but the breakpoint sequence is not changed. We can still obtain an

---

[14] *J.M. Heimerl and T.P. Coffee, "The Detailed Modeling of Premixed, Laminar Steady-State Flames. I. Ozone", Combustion and Flame, Volume 39, pp. 301-315, 1980.*

answer, but the integration will take longer, and the accuracy of the solution is not increased. At this stage, the appropriate value of ε for a given breakpoint sequence is a matter of trial and error.

An additional problem can occur with very slow flames. If the interval of integration is chosen too small, there may be a noticeable gradient in the temperature profile at the cold boundary. Since the gas is entering through the cold boundary quite slowly, this can result in a substantial heat loss through the boundary. This heat loss will slow down the flame. The integration will converge, but the computed flame speed will be too low. This effect can occur for any flame. However, for most flames, a noticeable heat loss is due to a fairly steep temperature gradient at the boundary. It is then obvious that the interval of integration must be increased. But for a slow flame (less than 20 cm/sec) a noticeable error in the flame speed can result (5 to 10 percent), even when the gradient at the cold boundary appears negligible.

Most of the cases we have run have been with NINT = 12, KORD = 4, and NCC = 2 (13 breakpoints, 26 collocation points). We have let $\varepsilon = 10^{-3}$, and SREC = 10^{-6}. The number of central intervals NCN was 4, and the ratio FC between the longest and shortest intervals was between 4 and 8.

The length of the interval of integration L, as well as the optimum value for FC, is a matter of trial and error. However, a reasonable set of breakpoints can be found fairly easily by experimenting with the above scheme, using the constant transport subroutine. The code can then be run with a more realistic transport algorithm, using the restart option.

To make sure that convergence was obtained, the code was also run with NINT = 16 and $\varepsilon = 3 \times 10^{-4}$ for a number of cases. Only negligible differences occurred.

The program was run on the BRL CYBER 76. For hydrogen-oxygen flames, the run time varied from about 30 seconds with the simpliest transport subroutine to 5 minutes or more for the most complicated case.

An earlier version of this code was applied to the solution of an unbounded ozone flame[14]. The successive calculation procedure was not implemented at that time. However, since the number of species is so small, the savings in reducing the size of the Jacobian is not very important.

At that time we reported using NINT = 57 to NINT = 70, KORD = 6 and NCC = 5. The number of collocation points was then between 62 and 75. We have now been able to solve this case with equal accuracy using NINT = 12, KORD = 4, and NCC = 2. Partly this is due to choosing NCC = 2. The program then generates a special choice of collocation points (Gauss - Legendre quadrature points in each subinterval) which gives increased spatial accuracy. More important is the improved algorithm for choosing breakpoints. By using a little more care, a much smaller set of breakpoints can still reproduce the flame with sufficient accuracy. So the careful choice of breakpoints is probably the most important consideration in running the code efficiently.

REFERENCES

1. F. A. Williams, Combustion Theory, Addison-Wesley, Reading, MA, Chapter 1, 1965.

2. R. B. Bird, W. E. Stewart and E. N. Lightfoot, Transport Phenomena, John Wiley and Sons, NY, 1960, Chapter 18.

3. R. M. Fristram and A. A. Westenberg, Flame Structure, McGraw-Hill, NY, Chapter V-1, 1965.

4. J. O. Hirshfelder, C. F. Curtiss and R. B. Bird, Molecular Theory of Gases and Liquids, 2nd Printing corrected, with notes, John Wiley and Sons, NY, 1964, Chapter 11.1.

5. S. Gordon and B. J. McBride, "Computer Program for Calculation of Complex Chemical Equilibrium Compositions, Rocket Performance, Incident and Reflected Shocks, and Chapman-Jouguet Detonations", NASA-SP-273, 1971, (1976 program version).

6. T. P. Coffee and J. M. Heimerl, "Transport Algorithms for Premixed, Laminar, Steady-State Flames", to be published in Combust. and Flame.

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

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

9. A. C. Hindmarsh, "Preliminary Documentation of GEARIB: Solution of Implicit Systems of Ordinary Differential Equations with Banded Jacobian", Rep. UCID-30130, Lawrence Livermore Laboratory, 1976.

10. D. B. Spalding and P. L. Stephenson, "Laminar Flame Propagation in Hydrogen + Bromine Mixtures", Proc. R. Soc. Lond. A., Volume 324, 315-337, 1971.

11. T. P. Coffee and J. M. Heimerl, "A Method for Computing the Flame Speed of a Laminar, Premixed, One Dimensional Flame", BRL Technical Report ARBRL-TR-02212, January 1980.

12. J. O. Hirshfelder, C. F. Curtiss, and R. B. Bird, op. cit., pp. 761-763.

13. J. Warnatz, "Calculation of the Structure of Laminar Flat Flames III: Structure of Burner-Stabilized Hydrogen-Oxygen and Hydrogen-Flourine Flames", Verlog Chemie GmbH, BdB 8/78 E 4018.

REFERENCES (continued)

14. J. M. Heimerl and T. P. Coffee, "The Detailed Modeling of Premixed, Laminar Steady-State Flames. I. Ozone", Combustion and Flame, Volume 39, pp. 301-315, 1980.

APPENDIX A

APPENDIX A

A listing of the computer code follows. The subroutines MAIN, F, UINIT, FLSP, BNDRY, BKPT, and RT are written for laminar flame problems. The subroutine F given is for the constant transport assumption. The subroutine RT given is for a H_2-O_2-N_2 flame. There are nine chemical species and thirty reactions.

The rest of the subroutines are from PDECOL. They have been modified as discussed in the text.

After the listing, the job stream and output is given for a typical hydrogen-oxygen flame. The initial unburned gas is 50% H_2 and 50% air, where air is 21% O_2 and 79% N_2. The nitrogen is considered to be a diluent, and does not react. The required starting information is attached to TAPE11. The main program and the subroutine RT are attached in a compiled form. The integration begins from the initial profiles (37). At the final output time, the flame speed and the temperature and species profiles are given, as well as the corresponding x values. The restart file is written on TAPE 2, and the output file on TAPE 9. Both files are catalogued for possible future use.

```
***  03/10/80  SCOPE 2.1.5  B R L  VER 00B  ***   04/15/80       80106

SYS DEVICES 819/ 4/PF FLS=377K FLL=1750K MXS=300K MXL=1305K MXH=1105H

HH.MM.SS CPU SECOND ORIGIN
         11.06.56.MFA. A2    BRL NOS/BE 1.3 L499 VER 004    03/17/80
11.07.03 00000.003 MFZ.     -TERRY,STMFZ,T30,P1.     PRINT PLFLSC, CY=5 FOR BRL REPORT.
11.07.03 00000.003 JOB.     -ACCOUNT,PD***.
11.07.04 00000.026 JOB.     -ATTACH,OLDPL1,PLFLSC1,ID=SBEJMH.
11.07.05 00000.029 MFZ.        PF254 - CYCLE     5 ATTACHED FROM SN=SYSTEM
11.07.05 00000.030 LOD.     -UPDATE(P=OLDPL1,O,L=1)
11.07.42 00000.045 USR.      READING INPUT
11.07.44 00000.171 USR.      UPDATE COMPLETED
11.07.44 00000.172 LOD.     -FTN(I=COMPILE,SL,R=0,T)
11.08.53 00001.890 USR.           1.714 CP SECONDS COMPILATION TIME
11.08.54 00001.895 JOB.     -ATTACH,OLDPL3,PLH202RTW30N&,ID=SBEJMH.
11.08.54 00001.899 MFZ.        PF254 - CYCLE     1 ATTACHED FROM SN=SYSTEM
11.08.54 00001.899 LOD.     -UPDATE(P=OLDPL3,O,L=1)
11.09.01 00001.914 USR.      READING INPUT
11.09.01 00001.941 USR.      UPDATE COMPLETED
11.09.01 00001.942 LOD.     -FTN(I=COMPILE,SL,R=0,T)
11.09.56 00002.416 USR.           .470 CP SECONDS COMPILATION TIME
11.09.56 00002.422 JOB.     -ATTACH,OLDPL2,PLFLSC2,ID=SBEJMH.
11.09.57 00002.425 MFZ.        PF254 - CYCLE     5 ATTACHED FROM SN=SYSTEM
11.09.57 00002.426 LOD.     -UPDATE(P=OLDPL2,O,L=1)
11.10.11 00002.440 USR.      READING INPUT
11.10.35 00002.901 USR.      UPDATE COMPLETED
11.10.35 00002.902 LOD.     -FTN(I=COMPILE,SL,R=0,T)
11.20.41 00006.645 USR.           3.739 CP SECONDS COMPILATION TIME
11.20.41 00006.647 MFZ.        JM166 - MAXIMUM USER SCM          47000B WORDS
11.20.41 00006.647 MFZ.        JM167 - MAXIMUM USER LCM           7000B WORDS
11.20.41 00006.647 MFZ.        JM170 - MAXIMUM JS*IO LCM             6IB BUFFERS
11.20.41 00006.648 MFZ.        RM770 - MAXIMUM ACTIVE FILES                    5
11.20.41 00006.648 MFZ.        RM771 - OPEN/CLOSE CALLS                       74
11.20.41 00006.648 MFZ.        RM772 - DATA TRANSFER CALLS                37,754
11.20.41 00006.648 MFZ.        RM773 - CONTROL/POSITIONING CALLS             245
11.20.41 00006.648 MFZ.        RM774 - BM DATA TRANSFER CALLS              3,348
11.20.41 00006.648 MFZ.        RM775 - BM CONTROL/POSITIONING CALLS          342
11.20.41 00006.649 MFZ.        RM776 - QUEUE MANAGER CALLS                   746
11.20.41 00006.649 MFZ.        RM777 - RECALL CALLS                          574
11.20.41 00006.649 MFZ.        SCM                133.251  KWS
11.20.41 00006.649 MFZ.        LCM                112.623  KWS
11.20.41 00006.649 MFZ.        I/O                  0.242  MW
11.20.41 00006.650 MFZ.        RMS                  0.177  MWS
11.20.41 00006.650 MFZ.        USER                 3.201  SEC
11.20.41 00006.650 MFZ.        JOB                  6.657  SEC
11.20.41 00006.650 MFZ.        DIO              1 604.967  KW
11.20.41 00006.650 MFZ.        SS                   8.158  SEC
11.20.41 00006.651 MFZ.        COST ESTIMATE       $1.24
11.20.41 00006.651 MFZ.        SC050 - 000060 SC/LC SWAPS
```

```
      PROGRAM MAIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE3=OUTPUT,TAPE1,
     * TAPE2,TAPE4,TAPE9,TAPE10,TAPE11)
C  ************************
C  THIS VERSION USES SUCCESSIVE CALCULATION.
C  EACH PDE IS INTEGRATED IN TIME INDEPENDENTLY.
C  ************************
      COMMON/ENDPT/PH0,PH5
      COMMON/TABP/PRESS,PSR,NPDEM
      COMMON/TARSM/SMALL
      COMMON/GEAR0/DTUSED,NQ,NSTEPS,NF,NJ
      COMMON/GEAR1/DUM(5),UROUND,IOUM(4)
      COMMON/OPTION/NOGAUS,MAXDER
      COMMON/IOUNIT/LOUT
      COMMON/GEARSS/HMX
      COMMON/TARIN/PH2,PH3,PHCTO,PHCT
      COMMON/START/NSTART,NBURN,NTRAN,KORD,NSKIP
      COMMON/TARAB/ASP,BSP,TPN,PHN,TMU,TMSPH,TMSPH2,TMSTP,TPENT
      COMMON/TAREM/EM
      COMMON/TAROM/NDIM1,NDIM2
      DIMENSION LBINP(7)
      COMMON/TARBK/PHBKPT(31)
      COMMON/TABLB/LB(21)
      DIMENSION UPH(20),UUF(20)
      COMMON/TARCT/RL,CPMX,H0(20),R2D(20),R2DM(20)
      COMMON/TARNT/CPINV,R2UF(20)
      COMMON/TABMF/RHUV(1000),DRHUV(1000)
      COMMON/TARV/PHVAL(401),UN(401),U(20,401),UC(20),UB(20),W(20)
      COMMON/MAIN/SCTCH(25),WORK(12000),IWORK(750)
      COMMON/ISTART/IW1,IW2,IW3,IW4,IW5,IW6,IW7,IW8,IW9,IW10,
     *  IW11,IW12,IW13,IW14,IW15,IW16,IW17,IW18
      COMMON/TARCH/DZM
      DATA LOUT,NOGAUS,MAXDER,UROUND/3,0,5,7.1E-15/
      DATA NDIM1,NDIM2/20,401/
      DATA IWORK(1),IWORK(2)/12000,750/
C  ************************
C  THE FOLLOWING INPUT DATA MUST BE SPECIFIED.
C  DIMENSIONING IN MAIN,F,UINIT,FLSP,PL,BNDRY,BKPT,AND INITAL
C  MUST BE CONSISTENT.
C  SUBROUTINE PT MUST BE WRITTEN.
C  IF NTRAN = 2, SUBROUTINE F MUST BE WRITTEN.
C  ************************
      READ(5,402)LBINP
  402 FORMAT(7A10)
      WRITE(3,450)LBINP
  450 FORMAT(//7A10//)
      READ(5,430)PRESS
      WRITE(3,7)PRESS
    7 FORMAT(/5X,10HPRESSURE =,1PE14.4,2X,4HATM./)
      READ(11,400)NPDE
  400 FORMAT(I4)
      WRITE(3,405)NPDE
  405 FORMAT(/10X,6HNPDE =,I4/)
      WRITE(3,455)
  455 FORMAT(//26X,1HW,11X,2HUC,10X,2HUB,10X,2HHO,10X,3HR2D/)
      DO 425 K=1,NPDE
      READ(11,410)LB(K),W(K),UC(K),UB(K),H0(K),R2D(K)
  410 FORMAT(A10,1PSE14.8)
```

```
      WRITE(3,412)K,LH(K),H(K),UC(K),UO(K),HO(K),H2O(K)                 MAIN      59
  412 FORMAT(I8,4X,A10,1P5E12.4)                                        MAIN      60
  425 CONTINUE                                                          MAIN      61
      NPDEP=NPDE+1                                                      MAIN      62
      READ(5,410)LB(NPDEP)                                              MAIN      63
      READ(11,431)UC(NPDE),UB(NPDE)                                     MAIN      64
  430 FORMAT(1P10E12.4)                                                 MAIN      65
      READ(5,430)TPN,PHN,TMN                                            MAIN      66
      READ(11,431)CPMX,RL,TPENT                                         MAIN      67
  431 FORMAT(1P3E16.8)                                                  MAIN      68
      UC(NPDE)=UC(NPDE)/TPN                                             MAIN      69
      UB(NPDE)=UB(NPDE)/TPN                                             MAIN      70
      WRITE(3,460)UC(NPDE),UB(NPDE)                                     MAIN      71
  460 FORMAT(//10X,4HTC =,1PE12.4,5X,4HTB =,1PE12.4/)                   MAIN      72
      WRITE(3,465)TPN,PHN,TMN                                           MAIN      73
  465 FORMAT(/10X,5HTPN =,1PE12.4,5X,5HPHN =,1PE12.4,5X,5HTMN =,        MAIN      74
     * 1PE12.4/)                                                        MAIN      75
      WRITE(3,470)CPMX,RL,TPENT                                         MAIN      76
  470 FORMAT(/10X,6HCPMX =,1PE12.4,5X,4HRL =,1PE12.4,5X,                MAIN      77
     *  7HTPENT =,1PE12.4/)                                            MAIN      78
C  PH0 IS THE LEFT BOUNDARY.  PH5 IS THE RIGHT BOUNDARY.                MAIN      79
C  ERROR IS MEASURED BY (YC-YP)/MAX(YC,SREC)                            MAIN      80
C  IF NSTART = 1, BEGIN FROM A GUESSED PROFILE.                         MAIN      81
C  IF NSTART = 2, BEGIN FROM A PROFILE ON TAPE1.                        MAIN      82
C  IF NBURN = 0, UNBOUNDED FLAME.                                       MAIN      83
C  IF NBURN = 1, FLAT FLAME BURNER AT PH = 0.                           MAIN      84
C  IF NBURN = 1, SPECIFY THE INCOMING FLUID VELOCITY(CM/SEC).           MAIN      85
C  IF NTRAN = 1, THE SHORT VERSION OF F IS USED.                        MAIN      86
C  IF NTRAN = 2, A LONG VERSION IS USED.                                MAIN      87
      READ(5,430)PH0,PH5,TFINAL,EPS,SREC                                MAIN      88
      READ(5,432)NSTART,NINT,NBURN,FLDSP                                MAIN      89
  432 FORMAT(3I4,1PE12.4)                                               MAIN      90
      WRITE(3,480)PH0,PH5,TFINAL                                        MAIN      91
  480 FORMAT(/10X,5HPH0 =,1PE12.4,5X,5HPH5 =,1PE12.4,5X,                MAIN      92
     *  8HTFINAL =,1PE12.4/)                                           MAIN      93
      WRITE(3,485)EPS,SREC                                              MAIN      94
  485 FORMAT(/10X,5HEPS =,1PE12.4,5X,6HSREC =,1PE12.4/)                 MAIN      95
      WRITE(3,490)NSTART,NINT,NBURN,FLDSP                               MAIN      96
  490 FORMAT(/10X,8HNSTART =,I4,5X,6HNINT =,I4,5X,7HNBURN =,I4,         MAIN      97
     *  5X,7HFLDSP =,1PE12.4/)                                         MAIN      98
      READ(5,434)NTRAN,TPRINT                                           MAIN      99
  434 FORMAT(I4,1PE12.4)                                                MAIN     100
      WRITE(3,495)NTRAN,TPRINT                                          MAIN     101
  495 FORMAT(/10X,7HNTRAN =,I4,5X,8HTPRINT =,1PE12.4/)                  MAIN     102
C  TMAX IS THE MAXIMUM RUN TIME ALLOWED.                               MAIN     103
      READ(5,436)TMAX                                                   MAIN     104
  436 FORMAT(1P10E12.4)                                                 MAIN     105
      WRITE(3,497)TMAX                                                  MAIN     106
  497 FORMAT(/10X,6HTMAX =,1PE12.4/)                                    MAIN     107
      READ(5,437)NCN,FC                                                 MAIN     108
  437 FORMAT(I4,1PE12.4)                                                MAIN     109
      WRITE(3,498)NCN,FC                                                MAIN     110
  498 FORMAT(/10X,5HNCN =,I6,5X,4HFC =,1PE12.4/)                        MAIN     111
      SMALL=1.0E-30                                                     MAIN     112
      KOMD=4                                                            MAIN     113
      NCC=7                                                             MAIN     114
      T0=0.0                                                            MAIN     115
```

```
      MF=22                                                              MAIN      116
      PSR=PRESS/82.05                                                    MAIN      117
      TMSPH=TMN/PHN                                                      MAIN      118
      TMSPH2=TMSPH/PHN                                                   MAIN      119
      TMSTP=TMN/TPN                                                      MAIN      120
      DZM=10.0/TFINAL                                                    MAIN      121
C  RENORMALIZE DIFFUSION AND THERMAL CONDUCTIVITY TERMS.                  MAIN      122
      DO 9 K=1,NPDE                                                      MAIN      123
9     R2D(K)=R2D(K)/PHN                                                  MAIN      124
C  ALL R2D = RH*RH*DIM/PHN.                                              MAIN      125
C  R2D IS THE VALUE AT THE LEFT BOUNDARY.                                MAIN      126
C  R2DM IS THE VALUE AT THE RIGHT END OF AN INTERVAL USED FOR            MAIN      127
C  COMPUTING THE FLAME SPEED.                                            MAIN      128
C  FOR NTRAN = 1, THEY ARE THE SAME.                                     MAIN      129
      NCPTS=(KORD-NCC)*NINT+NCC                                          MAIN      130
      DO 602 K=1,NPDE                                                    MAIN      131
602   R2DM(K)=R2D(K)                                                     MAIN      132
C  FIND THE APPROPRIATE NORMALIZED VALUES FOR USE IN F (NTRAN=1).         MAIN      133
      CPINV=TMSTP/CPMX                                                   MAIN      134
      DO 605 K=1,NPDE                                                    MAIN      135
605   R2DF(K)=R2D(K)*TMSPH                                               MAIN      136
      RL=RL*TMSPH2/CPMX                                                  MAIN      137
      NPDEM=NPDE-1                                                       MAIN      138
      NPDEP=NPDE+1                                                       MAIN      139
      NCPTS=(KORD-NCC)*NINT+NCC                                          MAIN      140
      YSSM=0.0                                                           MAIN      141
      YN=1.0                                                             MAIN      142
      DO 5 K=1,NPDEM                                                     MAIN      143
      YN=YN-UC(K)                                                        MAIN      144
5     YSSM=YSSM+UC(K)/W(K)                                               MAIN      145
      YSSM=YSSM+YN/W(NPDE)                                               MAIN      146
      T=UC(NPDE)*TPN                                                     MAIN      147
      RHO=PSR/(T*YSSM)                                                   MAIN      148
      WRITE(3,8)RHO                                                      MAIN      149
8     FORMAT(/5X,17H1NITIAL DENSITY =,1PE12.4/)                          MAIN      150
      NPTS=NINT+1                                                        MAIN      151
      DT=TFINAL*1.0E-10                                                  MAIN      152
      TOUT=0.0                                                           MAIN      153
C  HMX IS THE LARGEST TIME STEP ALLOWED IN THE INTEGRATION.               MAIN      154
      HMX=TFINAL/20.0                                                    MAIN      155
      INDEX=1                                                            MAIN      156
      CALL BKPT(NINT,NCN,KC,NVPTS)                                       MAIN      157
C  DEFINE THE CENTER OF THE FLAME.                                       MAIN      158
      KCEN=NPDE                                                          MAIN      159
      VCEN=UC(KCEN)+0.4*(UH(KCEN)-UC(KCEN))                              MAIN      160
      WRITE(3,573)KCEN,VCEN                                              MAIN      161
573   FORMAT(/10X,6HKCEN =,I6,5X,6HVCEN =,1PE12.4/)                      MAIN      162
C  INITIAL VALUE FOR THE SPEED OF THE ORIGIN (MASS FLOW).                MAIN      163
      IF(NSTART.EQ.2)READ(1,255)SPEEDO                                   MAIN      164
      IF(NSTART.EQ.2)WRITE(3,552)SPEEDO                                  MAIN      165
      IF(NSTART.EQ.1)SPEEDO=0.0                                          MAIN      166
      ASP=SPEEDO                                                         MAIN      167
      IF(NBURN.EQ.0.0)GO TO 550                                          MAIN      168
      FM=RHO*FLDSP                                                       MAIN      169
      WRITE(3,47)FM,FLDSP                                                MAIN      170
      SPEEDO=-FM*TMSPH                                                   MAIN      171
      SPN1=SPEEDO                                                        MAIN      172
```

```
                ASP=SPEEDO                                                      MAIN      173
                BSP=0.0                                                         MAIN      174
          550   CONTINUE                                                        MAIN      175
          552   FORMAT(/5X,8HSPEEDO =,1PE12.4/)                                 MAIN      176
          555   FORMAT(1PE14.6)                                                 MAIN      177
                FT=SECOND(CP)                                                   MAIN      178
          C  CALL INTEGRATOR AND WRITE OUTPUT.                                  MAIN      179
          20    CALL PDECOL(TO,TOUT,DT,PHHKPT,EPS,NINT,KORD,NCC,NPDE,MF,        MAIN      180
                *  INDEX,WORK,IWORK,SPEC)                                       MAIN      181
                IF(INDEX.NE.0)GO TO 70                                          MAIN      182
                GT=SECOND(CP)                                                   MAIN      183
                HT=GT-FT                                                        MAIN      184
                IF(HT.GT.TMAX)TFINAL=TOUT                                       MAIN      185
                WRITE(3,30)TOUT,DTUSED,NSTEPS                                   MAIN      186
          30    FORMAT(//10X,3HT =,1PE12.4,4X,4HDT =,1PE12.4,4X,                MAIN      187
                * 13HTOTAL STEPS =,I4/)                                         MAIN      188
                WRITE(3,32)NF,NJ                                                MAIN      189
          32    FORMAT(/10X,4HNF =,I4,5X,4HNJ =,I4/)                           MAIN      190
                WRITE(3,45)HT                                                   MAIN      191
          45    FORMAT(/10X,10HRUN TIME =,1PE12.4/)                             MAIN      192
                CALL VALUES(PHVAL,U,SCTCH,NDIM1,NDIM2,NVPTS,0,WORK)             MAIN      193
                DO 25 K=1,NVPTS                                                 MAIN      194
                UN(K)=1.0                                                       MAIN      195
                DO 25 J=1,NPDEM                                                 MAIN      196
          25    UN(K)=UN(K)-U(J,K)                                              MAIN      197
                IF(TOUT.LT.TPRINT)GO TO 65                                      MAIN      198
                DO 35 K=1,NPDEM                                                 MAIN      199
                WRITE(3,37)LB(K)                                                MAIN      200
          37    FORMAT(/10X,A10/)                                               MAIN      201
                WRITE(3,39)(U(K,I),I=1,NVPTS,NSKIP)                             MAIN      202
          39    FORMAT(1P10E12.4)                                               MAIN      203
          35    CONTINUE                                                        MAIN      204
                WRITE(3,37)LB(NPDE)                                             MAIN      205
                WRITE(3,39)(UN(I),I=1,NVPTS,NSKIP)                              MAIN      206
                WRITE(3,37)LB(NPDEP)                                            MAIN      207
                WRITE(3,39)(U(NPDE,I),I=1,NVPTS,NSKIP)                          MAIN      208
          65    CONTINUE                                                        MAIN      209
                CALL FLSP(NPDE,TOUT,TPRINT,TFINAL,NVPTS,NCPTS,                  MAIN      210
                * PHO,KCEN,FSP,UPH,UDF)                                         MAIN      211
          443   FORMAT(/5X,8HSPEEDO =,1PE12.4,6X,4HDS =,1PE12.4/)               MAIN      212
                IF(TOUT.GE.TPRINT)TPRINT=TPRINT+TFINAL/5.0                      MAIN      213
                IF(NHURN.EQ.0)GO TO 100                                         MAIN      214
                DO 610 J=1,NPDEM                                                MAIN      215
          610   UDF(J)=U(J,1)-UPH(J)*R2D(J)/FM                                  MAIN      216
                UDF(NPDE)=1.0                                                   MAIN      217
                DO 612 J=1,NPDEM                                                MAIN      218
          612   UDF(NPDE)=UDF(NPDE)-UDF(J)                                      MAIN      219
                WRITE(3,615)(UDF(L),L=1,NPDE)                                   MAIN      220
          615   FORMAT(/2X,29HSPECIES MASS FLUX FRACTIONS =,1PE12.4//)          MAIN      221
          C  FIND THE FORMULA FOR ADJUSTING THE ORIGIN SPEED AND CENTERING      MAIN      222
          C  THE FLAME.                                                         MAIN      223
          100   DO 110 K=1,NVPTS                                                MAIN      224
                KP=K+1                                                          MAIN      225
                UMAX=AMAX1(U(KCEN,K),U(KCEN,KP))                                MAIN      226
                UMIN=AMIN1(U(KCEN,K),U(KCEN,KP))                                MAIN      227
                IF(UMAX.GT.VCEN.AND.UMIN.LT.VCEN)KS=K                           MAIN      228
          110   IF(UMAX.GT.VCEN.AND.UMIN.LT.VCEN)GO TO 115                      MAIN      229
```

```
      115   KP=KS+1                                                          MAIN      230
230         PC=(VCEN-U(KCEN,KS))/(U(KCEN,KP)-U(KCEN,KS))                     MAIN      231
            PHNEW=PHVAL(KS)+PC*(PHVAL(KP)-PHVAL(KS))                         MAIN      232
            IF(NBURN.EQ.1)GO TO 560                                          MAIN      233
            IF(TOUT.GT.0.0)GO TO 150                                         MAIN      234
            FM=RHO*FSP                                                       MAIN      235
235         IF(NSTART.EQ.1)SPEEDO=-FM*TMSPH                                  MAIN      236
            WRITE(3,447)FM,FSP                                               MAIN      237
      447   FORMAT(/5X,4HMO =,1PE12.4,6X,6HFLSP =,1PE12.4/)                  MAIN      238
            DS=0.0                                                           MAIN      239
            IF(SPEEDO.GT.0.0)SPEEDO=0.0                                      MAIN      240
240         WRITE(3,443)SPEEDO,DS                                            MAIN      241
            TINC=TFINAL/100.0                                                MAIN      242
            TOLD=TOUT                                                        MAIN      243
            TOUT=TINC                                                        MAIN      244
            DPH=PHNEW-PHCT                                                   MAIN      245
245         DT=TOUT-TOLD                                                     MAIN      246
            PHOLD=PHNEW-SPEEDO*DT                                            MAIN      247
            BSP=0.0                                                          MAIN      248
            ASP=SPEEDO                                                       MAIN      249
            SPEEDN=ASP+BSP*TOUT                                              MAIN      250
250         SPN1=SPEEDN                                                      MAIN      251
            GO TO 20                                                         MAIN      252
      150   SPEEDO=SPEEDN                                                    MAIN      253
            SPO1=SPN1                                                        MAIN      254
            SPN1=(PHNEW-PHOLD)/(TOUT-TOLD)                                   MAIN      255
255         FM=-SPN1/TMSPH                                                   MAIN      256
            FSP=FM/RHO                                                       MAIN      257
            WRITE(3,47)FM,FSP,PHNEW                                          MAIN      258
      47    FORMAT(/5X,4HMO =,1PE12.4,6X,6HFLSP =,1PE12.4,6X,                MAIN      259
           *  7HPHNEW =,1PE12.4/)                                           MAIN      260
260         DS=SPN1-SPO1                                                     MAIN      261
            TOLD=TOUT                                                        MAIN      262
      C  ESTIMATE CHANGE IN FLAME SPEED PER UNIT TIME.                       MAIN      263
            DSG=ABS(DS)/TINC                                                 MAIN      264
      C  CHOOSE A NEW TINC SO THAT THE ESTIMATED DRIFT OF THE FLAME          MAIN      265
265   C  FROM THE CENTER IS VERY SMALL.                                      MAIN      266
            DPHMX=(PH3-PH2)/50.0                                             MAIN      267
            TINC=SQRT(DPHMX/DSG)                                             MAIN      268
            TINC=AMAX1(TINC,TFINAL/100.0)                                    MAIN      269
            TINC=AMIN1(TINC,TFINAL/4.0)                                      MAIN      270
270         TOUT=TOUT+TINC                                                   MAIN      271
            IF(TOUT.GT.TFINAL)TOUT=TFINAL                                    MAIN      272
            IF(TOLD.EQ.TFINAL)GO TO 201                                      MAIN      273
            DT=TOUT-TOLD                                                     MAIN      274
            DPH=PHNEW-PHCT                                                   MAIN      275
275         SPN2=2.0*SPN1-SPEEDO+2.0*DPH/DT                                  MAIN      276
            SPEEDN=0.5*(SPN1+SPN2)                                           MAIN      277
      C  IF DPH IS SMALL ENOUGH, DON'T BOTHER TO RECENTER THE FLAME.         MAIN      278
            IF(ABS(DPH).LT.0.01*(PH5-PH0))SPEEDN=SPN1                        MAIN      279
      C  FLAME SPEED CANNOT BE NEGATIVE.                                     MAIN      280
280         IF(SPEEDN.GT.0.0)SPEEDN=0.0                                      MAIN      281
            BSP=(SPEEDN-SPEEDO)/(TOUT-TOLD)                                  MAIN      282
            ASP=SPEEDO-TOLD*BSP                                              MAIN      283
            DS=SPEEDN-SPEEDO                                                 MAIN      284
            WRITE(3,43)SPN1,SPN2,SPEEDN,DS                                   MAIN      285
285   43    FORMAT(/5X,6HSPN1 =,1PE12.4,6X,6HSPN2 =,1PE12.4,6X,              MAIN      286
```

```
      *  8HSPEEDN =,1PE12.4,6X,4HDS =,1PE12.4/)                          MAIN      287
       PHOLD=PHNEW-DT*(ASP+BSP*(TOUT+TOLD)*0.5)                          MAIN      288
       GO TO 570                                                         MAIN      289
  560  TOUT=TOUT+TFINAL/5.0                                              MAIN      290
  570  CONTINUE                                                          MAIN      291
  290  IF(TOUT.LE.TFINAL)GO TO 20                                        MAIN      292
  201  CONTINUE                                                          MAIN      293
C  CREATE DATA FILE FOR RESTART                                         MAIN      294
       WRITE(3,220)SPN1                                                  MAIN      295
  295  WRITE(2,220)SPN1                                                  MAIN      296
       WRITE(3,220)PHNEW                                                 MAIN      297
       WRITE(2,220)PHNEW                                                 MAIN      298
       WRITE(3,200)                                                      MAIN      299
  200  FORMAT(/10X,18HCOLLOCATION POINTS/)                               MAIN      300
  300  WRITE(2,210)NCPTS                                                 MAIN      301
       WRITE(3,210)NCPTS                                                 MAIN      302
  210  FORMAT(I8)                                                        MAIN      303
       IEND=IW3+NCPTS-1                                                  MAIN      304
       WRITE(2,220)(WORK(L),L=IW3,IEND)                                  MAIN      305
  305  WRITE(3,220)(WORK(L),L=IW3,IEND)                                  MAIN      306
       CALL VALUES(WORK(IW3),0,SCTCH,ND1M1,ND1M2,NCPTS,0,WORK)           MAIN      307
       DO 250 K=1,NPDE                                                   MAIN      308
       WRITE(2,220)(U(K,L),L=1,NCPTS)                                    MAIN      309
       WRITE(3,220)(U(K,L),L=1,NCPTS)                                    MAIN      310
  310  250  CONTINUE                                                     MAIN      311
       WRITE(2,220)(R2D(L),L=1,NPDE)                                     MAIN      312
       WRITE(3,220)(R2D(L),L=1,NPDE)                                     MAIN      313
  220  FORMAT(1PRE14.6)                                                  MAIN      314
  500  CONTINUE                                                          MAIN      315
  315  70   WRITE(3,80)INDEX                                             MAIN      316
  80   FORMAT(/10X,7HINDEX =,I3)                                         MAIN      317
       STOP                                                              MAIN      318
       END                                                               MAIN      319
```

```
      SUBROUTINE F(TIME,PH,U,UPH,UPH2,FVAL,NPDE,KPDE,IC,KSKT,KSKR)                F       2
      DIMENSION U(NPDE),UPH(NPDE),UPH2(NPDE)                                      F       3
      COMMON/TABAB/ASP,BSP,TPN,PHN,TMR,TMSPH,TMSPH2,TMSTP,TPENT                   F       4
      COMMON/TABP/PRESS,PSW,NPDEM                                                 F       5
      DIMENSION H(20),R(20)                                                       F       6
      COMMON/TABCT/RL,CPMX,H0(20),R2U(20),R2UH(20)                                F       7
      COMMON/TABNT/CPINV,R2UF(20)                                                 F       8
      COMMON/TABRY/T,PH,Y1,Y2,Y3,Y4,Y5,Y6,Y7,Y8,Y9,Y10                            F       9
C     ************************                                                    F      10
C     AT EACH CALL THE TIME RATE OF CHANGE FOR ONE PDE IS RETURNED IN FVAL.       F      11
C     ************************                                                    F      12
      YN=1.0                                                                      F      13
      DO 10 K=1,NPDEM                                                             F      14
   10 YN=YN-U(K)                                                                  F      15
      CALL RT(U,YN,R,NPDE,KPDE,IC,KSKR)                                           F      16
      SP=ASP+HSP*TIME                                                             F      17
   35 IF(KPDE.EQ.NPDE)GO TO 50                                                    F      18
      FVAL=SP*UPH(KPDE)+R2UF(KPDE)*UPH2(KPDE)+TMN*R(KPDE)                         F      19
      RETURN                                                                      F      20
   50 DT=TPN*U(NPDE)-TPENT                                                        F      21
      DO 20 K=1,NPDE                                                              F      22
   20 H(K)=H0(K)+CPMX*DT                                                          F      23
      RTEMP=0.0                                                                   F      24
      DO 25 K=1,NPDE                                                              F      25
   25 RTEMP=RTEMP-R(K)*H(K)                                                       F      26
      FVAL=SP*UPH(NPDE)+RL*UPH2(NPDE)+RTEMP*CPINV                                 F      27
      RETURN                                                                      F      28
      END                                                                         F      29
```

```
      SUBROUTINE UINIT(PH,U,NPDE,KPDE)                                  UINIT      2
      COMMON/TABIN/PH2,PH3,PHCTO,PHCT                                   UINIT      3
      COMMON/ENDPT/PH0,PH5                                              UINIT      4
      COMMON/TABAR/ASP,HSP,TPN,PHN,TMJ,TMSPH,TMSPH2,TMSTP,TPENT         UINIT      5
      COMMON/START/NS,NBURN,NTRAN,KORD,NSKIP                            UINIT      6
      COMMON/TABV/PHVAL(401),PHS(401),UP(20,401),UC(20),UB(20),W(20)    UINIT      7
C  PIS2=PI / 2.0.                                                       UINIT      8
      DATA PIS2/1.5707963/                                              UINIT      9
C  *************************                                          UINIT     10
C  DETERMINE THE INITIAL STARTING PROFILE.                            UINIT     11
C  X WILL BE A COLLOCATION POINT.                                     UINIT     12
C  U STORES THE CORRESPONDING FUNCTION VALUES.                        UINIT     13
C  *************************                                          UINIT     14
      IF(NS.GT.1)GO TO 50                                               UINIT     15
C  DETERMINE AN INITIAL GUESSED PROFILE FROM VALUES AT THE            UINIT     16
C  BURNED AND UNBURNED ENDS.                                          UINIT     17
      IF(PH.GT.PH2)GO TO 5                                              UINIT     18
      U=UC(KPDE)                                                        UINIT     19
      RETURN                                                            UINIT     20
5     IF(PH.GT.PH3)GO TO 10                                             UINIT     21
      PHD=(PH-PH2)/(PH3-PH2)                                            UINIT     22
      PHM=PIS2*(PHD**2)                                                 UINIT     23
      UD=UB(KPDE)-UC(KPDE)                                              UINIT     24
      U=UC(KPDE)+UD*((SIN(PHM))**2)                                     UINIT     25
      RETURN                                                            UINIT     26
10    U=UB(KPDE)                                                        UINIT     27
      RETURN                                                            UINIT     28
50    IF(NS.GT.2)GO TO 100                                              UINIT     29
C  READ THE COLLOCATION VALUES AND THE CORRESPONDING FUNCTION         UINIT     30
C  VALUES FROM A PREVIOUS RUN.                                        UINIT     31
      READ(1,65)PHCTO                                                   UINIT     32
      READ(1,55)NPS                                                     UINIT     33
55    FORMAT(I8)                                                        UINIT     34
      NM=NPS-1                                                          UINIT     35
      READ(1,65)(PHS(L),L=1,NPS)                                        UINIT     36
      DO 60 K=1,NPDE                                                    UINIT     37
      READ(1,65)(UP(K,L),L=1,NPS)                                       UINIT     38
C  WE DO NOT WANT ANY NEGATIVE CONCENTRATIONS.                        UINIT     39
      DO 58 L=2,NM                                                      UINIT     40
      IF(UP(K,L).GE.0.0)GO TO 58                                        UINIT     41
      LM=L-1                                                            UINIT     42
      LP=L+1                                                            UINIT     43
      UP(K,L)=0.5*(UP(K,LM)+UP(K,LP))                                   UINIT     44
58    CONTINUE                                                          UINIT     45
60    CONTINUE                                                          UINIT     46
65    FORMAT(1PE14.6)                                                   UINIT     47
      IF(NBURN.EQ.1)GO TO 75                                            UINIT     48
C  CENTER THE FLAME.                                                  UINIT     49
      DO 70 K=1,NPS                                                     UINIT     50
      PHS(K)=PHS(K)+PHCT-PHCTO                                          UINIT     51
70    CONTINUE                                                          UINIT     52
75    CONTINUE                                                          UINIT     53
      PHXL=PHS(2)-PH0                                                   UINIT     54
      PHXR=PH5-PHS(NPS)                                                 UINIT     55
      NS=3                                                              UINIT     56
      U=UC(KPDE)                                                        UINIT     57
      RETURN                                                            UINIT     58
```

```
      C  DETERMINE THE NEW VALUES BY INTERPOLATION.                    UINIT      59
      100   K=2                                                        UINIT      60
   60         IF(PH.GT.PH0)GO TO 102                                   UINIT      61
              U=UC(KPDE)                                               UINIT      62
              RETURN                                                   UINIT      63
      102   IF(PH.GT.PHS(2))GO TO 104                                  UINIT      64
              P=(PH-PH0)/PHXL                                          UINIT      65
   65         U=UC(KPDE)+P*(UP(KPDE,2)-UC(KPDE))                       UINIT      66
              RETURN                                                   UINIT      67
      104   IF(PH.LT.PHS(NPS))GO TO 108                                UINIT      68
              IF(PHXR.NE.0.0)P=(PH-PHS(NPS))/PHXR                      UINIT      69
              IF(PHXR.EQ.0.0)P=0.0                                     UINIT      70
   70         U=UP(KPDE,NPS)+P*(U0(KPDE)-UP(KPDE,NPS))                 UINIT      71
              RETURN                                                   UINIT      72
      108   CONTINUE                                                   UINIT      73
      110   IF(PH.LE.PHS(K))GO TO 120                                  UINIT      74
              K=K+1                                                    UINIT      75
   75         GO TO 110                                                UINIT      76
      120   KM=K-1                                                     UINIT      77
              P=(PHS(K)-PH)/(PHS(K)-PHS(KM))                           UINIT      78
              U=UP(KPDE,K)-P*(UP(KPDE,K)-UP(KPDE,KM))                  UINIT      79
              RETURN                                                   UINIT      80
   80         END                                                      UINIT      81
```

```
      SUBROUTINE FLSP(NPDE,TOUT,TPRINT,TFINAL,NVPTS,NCPTS,                FLSP      2
     * RHO,KCEN,FSP,UPH,UDF)                                              FLSP      3
C     ************************                                            FLSP      4
C     DETERMINE THE FLAME SPEED BY NUMERICALLY INTEGRATING THE SPECIES    FLSP      5
C     EQUATIONS,USING THE TRAPEZOIDAL RULE.                               FLSP      6
C     ************************                                            FLSP      7
      DIMENSION UPH(NPDE),UDF(NPDE)                                       FLSP      8
      COMMON/ENDPT/PHL,PHR                                                FLSP      9
      COMMON/TABIN/PHFFL,PHFFR,PHCTU,PHCT                                 FLSP     10
      COMMON/TABAB/ASP,HSP,TPN,PHN,TMP,TMSPH,TMSPH2,TMSTP,TPENT           FLSP     11
      COMMON/TABP/PRESS,PSR,NPDEM                                         FLSP     12
      COMMON/TABDM/NDIM1,NDIM2                                            FLSP     13
      COMMON/START/NSTART,NHURN,NTRAN,KORD,NSKIP                          FLSP     14
      COMMON/MAIN/SCTCH(25),WORK(12000),IWORK(750)                        FLSP     15
      COMMON/ISTART/IW1,IW2,IW3,IDUM(15)                                  FLSP     16
      COMMON/TABCT/RL,CPMA,HO(20),R2D(20),R2DM(20)                        FLSP     17
      COMMON/TABV/PHVAL(401),UN(401),U(20,401),UC(20),UR(20),W(20)        FLSP     18
      COMMON/TABMF/RHUV(1000),DRHUV(1000)                                 FLSP     19
      COMMON/TABSM/SMALL                                                  FLSP     20
      COMMON/TABLB/LB(21)                                                 FLSP     21
      DIMENSION RINT(20),RINTI(20),RI(20)                                 FLSP     22
      DIMENSION ULT(20,3),URT(20,3),FLS (20),UPH2(20)                     FLSP     23
      DIMENSION X(401),R(20),D(20)                                        FLSP     24
      DO 5 J=1,NPDE                                                       FLSP     25
    5 RI(J)=0.0                                                           FLSP     26
C  INTEGRATE FROM THE LEFT BOUNDARY TO VARIOUS POINTS.                    FLSP     27
      CALL VALUES(PHL,ULT,SCTCH,NDIM1,1,2,WORK)                           FLSP     28
  128 FORMAT(/2X,13HR2D AT LEFT =,1P10E12.4/)                             FLSP     29
      DO 10 J=1,NPDEM                                                     FLSP     30
      UPH(J)=ULT(J,2)                                                     FLSP     31
   10 UPH2(J)=ULT(J,3)                                                    FLSP     32
      UPH(NPDE)=0.0                                                       FLSP     33
      UPH2(NPDE)=0.0                                                      FLSP     34
      DO 12 J=1,NPDEM                                                     FLSP     35
      UPH(NPDE)=UPH(NPDE)-UPH(J)                                          FLSP     36
   12 UPH2(NPDE)=UPH2(NPDE)-UPH2(J)                                       FLSP     37
      IF(ABS(UPH(NPDE)).LT.SMALL)UPH(NPDE)=SMALL                          FLSP     38
      R2D(NPDE)=-RHUV(NPDE)/UPH(NPDE)                                     FLSP     39
      WRITE(3,128)(R2D(L),L=1,NPDE)                                       FLSP     40
      WRITE(3,126)(UPH(L),L=1,NPDE),ULT(NPDE,2)                           FLSP     41
  126 FORMAT(/2X,13HUPH AT LEFT =,1P10E12.4/)                             FLSP     42
      WRITE(3,127)(UPH2(L),L=1,NPDE),ULT(NPDE,3)                          FLSP     43
  127 FORMAT(/2X,14HUPH2 AT LEFT =,1P10E12.4/)                            FLSP     44
   16 NS=32                                                               FLSP     45
      NSPL=NS+1                                                           FLSP     46
      DO 155 NMID=NSPL,NVPTS,NS                                           FLSP     47
      NMIDO=NMID-NS                                                       FLSP     48
      DO 100 K=NMIDO,NMID                                                 FLSP     49
      CALL RT(U(1,K),UN(K),R,NPDE,1,1,1)                                  FLSP     50
      DO 20 J=1,NPDE                                                      FLSP     51
      RINT(J)=R(J)                                                        FLSP     52
   20 CONTINUE                                                            FLSP     53
      IF(K.EQ.NMIDO)GO TO 50                                              FLSP     54
      KM=K-1                                                              FLSP     55
      DPH=PHVAL(K)-PHVAL(KM)                                              FLSP     56
      DO 30 J=1,NPDE                                                      FLSP     57
   30 RI(J)=RI(J)+0.5*(RINT(J)+RINTI(J))*DPH                              FLSP     58
```

```
                50    DO 60 J=1,NPDE                                          FLSP   59
                60    RINTI(J)=RINT(J)                                        FLSP   60
      60       100    CONTINUE                                                FLSP   61
                      DO 110 J=1,NPDE                                         FLSP   62
               110    FLS (J)=PHN*RI(J)                                       FLSP   63
               C  FIND APPROPRIATE VALUES AT NMID.                            FLSP   64
                      CALL VALUES(PHVAL(NMID),URT,SCTCH,NDIM1,1,1,2,WORK)     FLSP   65
      65              IF(NTRAN.EQ.1)GO TO 118                                 FLSP   66
               C  IF THE LONG VERSION OF F IS USED, WE NEED TO FIND OUT WHAT THE VALUE FLSP   67
               C  OF R2D IS AT THE APPROPRIATE POINT IN THE FLAME.            FLSP   68
                      IC=NCPTS+1                                              FLSP   69
                      CALL F(TOUT,PHVAL(NMID),URT(1,1),URT(1,2),URT(1,3),FVAL,FLSP   70
      70             * NPDE,1,IC,1,1)                                         FLSP   71
                      IST=NCPTS*NPDE                                          FLSP   72
                      DO 115 J=1,NPDEM                                        FLSP   73
               115    R2DM(J)=-RHOV(IST+J)/URT(J,2)                           FLSP   74
               118    DO 120 J=1,NPDEM                                        FLSP   75
      75       120    D(J)=R2DM(J)*URT(J,2) -R2D(J)*URT(J,2)                  FLSP   76
               C  THE SUM OF THE DIFFUSION TERMS MUST ADD UP TO ZERO.         FLSP   77
                      D(NPDE)=0.0                                             FLSP   78
                      DO 124 J=1,NPDEM                                        FLSP   79
                      D(NPDE)=D(NPDE)-D(J)                                    FLSP   80
      80       124    CONTINUE                                                FLSP   81
                      DO 130 J=1,NPDE                                         FLSP   82
               130    FLS (J)=FLS (J)+D(J)                                    FLSP   83
               135    CONTINUE                                                FLSP   84
                      DIFF=UN(NMID)-UN(1)                                     FLSP   85
      85              IF(ABS(DIFF).LE.SMALL)FLS(NPDE)=0.0                     FLSP   86
                      IF(ABS(DIFF).GT.SMALL)FLS(NPDE)=FLS(NPDE)/(RHO*DIFF)    FLSP   87
                      DO 150 J=1,NPDEM                                        FLSP   88
                      DIFF=U(J,NMID)-U(J,1)                                   FLSP   89
                      IF(ABS(DIFF).LE.SMALL)FLS(J)=0.0                        FLSP   90
      90              IF(ABS(DIFF).GT.SMALL)FLS(J)=FLS(J)/(RHO*DIFF)          FLSP   91
               150    CONTINUE                                                FLSP   92
                      WRITE(3,152)(FLS (L),L=1,NPDE)                          FLSP   93
               152    FORMAT(/2X,13HFLAME SPEED =,1P10E12.4///)               FLSP   94
               155    CONTINUE                                                FLSP   95
      95              FSP=FLS(NPDEM)                                          FLSP   96
               C  FIND THE X VALUES USING THE TRAPEZOIDAL RULE.               FLSP   97
                      RI(1)=0.0                                               FLSP   98
                      X(1)=0.0                                                FLSP   99
                      DO 200 K=1,NVPTS                                        FLSP  100
     100              T=U(NPDE,K)*TPN                                         FLSP  101
                      YSSM=UN(K)/W(NPDE)                                      FLSP  102
                      DO 160 J=1,NPDEM                                        FLSP  103
               160    YSSM=YSSM+U(J,K)/W(J)                                   FLSP  104
                      RH=PSR/(T*YSSM)                                         FLSP  105
     105              RINT(1)=1.0/RH                                          FLSP  106
                      IF(K.EQ.1)GO TO 190                                     FLSP  107
                      KM=K-1                                                  FLSP  108
                      DPH=PHVAL(K)-PHVAL(KM)                                  FLSP  109
                      RI(1)=RI(1)+0.5*(RINT(1)+RINTI(1))*DPH                  FLSP  110
     110              X(K)=PHN*RI(1)                                          FLSP  111
               190    RINTI(1)=RINT(1)                                        FLSP  112
               200    CONTINUE                                                FLSP  113
                      IF(TOUT.LT.TPRINT)GO TO 250                             FLSP  114
                      WRITE(3,201)                                            FLSP  115
```

```
      201   FORMAT(/10X,7HX IN CM/)                                          FLSP  116
            WRITE(3,202)(X(L),L=1,NVPTS,NSKIP)                               FLSP  117
      202   FORMAT(1P10E12.4)                                                FLSP  118
      250   CONTINUE                                                         FLSP  119
      C  COMPUTE THE FLAME THICKNESS.                                        FLSP  120
            TMAX=AMAX1(U(NPDE,NVPTS),U(NPDE,1))                              FLSP  121
            TMIN=AMIN1(U(NPDE,NVPTS),U(NPDE,1))                              FLSP  122
            DT=0.1*(TMAX-TMIN)                                               FLSP  123
            TH=TMAX-DT                                                       FLSP  124
            TL=TMIN+DT                                                       FLSP  125
            DO 310 K=1,NVPTS                                                 FLSP  126
            KP=K+1                                                           FLSP  127
            IF(TL.GT.U(NPDE,K).AND.TL.LT.U(NPDE,KP))KL=K                     FLSP  128
            IF(TH.GT.U(NPDE,K).AND.TH.LT.U(NPDE,KP))KH=K                     FLSP  129
      310   CONTINUE                                                         FLSP  130
            KLP=KL+1                                                         FLSP  131
            KHP=KH+1                                                         FLSP  132
            PL=(U(NPDE,KL)-TL)/(U(NPDE,KL)-U(1,KLP))                         FLSP  133
            PH=(U(NPDE,KH)-TH)/(U(NPDE,KH)-U(1,KHP))                         FLSP  134
            PHIL=PHVAL(KL)+PL*(PHVAL(KLP)-PHVAL(KL))                         FLSP  135
            PHIH=PHVAL(KH)+PH*(PHVAL(KHP)-PHVAL(KH))                         FLSP  136
            XLW=X(KL)+PL*(X(KLP)-X(KL))                                      FLSP  137
            XH=X(KH)+PH*(X(KHP)-X(KH))                                       FLSP  138
            FTH=ABS(XLW-XH)                                                  FLSP  139
            WRITE(3,314)PHIL,PHIH                                            FLSP  140
            PHCEN=0.5*(PHL+PHR)                                              FLSP  141
            PHFFL=AMIN1(PHIL,PHIH)                                           FLSP  142
            PHFFR=AMAX1(PHIL,PHIH)                                           FLSP  143
      314   FORMAT(/10X,23HFLAME FRONT FROM PHI* =,1PE12.4,2X,               FLSP  144
           * 9HTO PHI* =,1PE12.4/)                                           FLSP  145
            WRITE(3,318)XLW,XH                                               FLSP  146
      318   FORMAT(/10X,20HFLAME FRONT FROM X =,1PE12.4,2X,                  FLSP  147
           * 6HTO X =,1PE12.4/)                                              FLSP  148
            WRITE(3,330)FTH                                                  FLSP  149
      330   FORMAT(/10X,17HFLAME THICKNESS =,1PE12.4,2X,2HCM/)               FLSP  150
            IF(TOUT.LT.TFINAL)RETURN                                         FLSP  151
      C  CREATE OUTPUT FILE.                                                 FLSP  152
            WRITE(9,1208)NPDE,NVPTS                                          FLSP  153
            WRITE(3,1208)NPDE,NVPTS                                          FLSP  154
      1208  FORMAT(5I4)                                                      FLSP  155
            DO 1215 K=1,NPDE                                                 FLSP  156
            WRITE(9,1212)LAB(K),X(K),HO(K),RZ(K)                             FLSP  157
            WRITE(3,1212)LAB(K),X(K),HO(K),RZ(K)                             FLSP  158
      1212  FORMAT(A10,1P3E16.8)                                             FLSP  159
      1215  CONTINUE                                                         FLSP  160
            LHT=10HT                                                         FLSP  161
            WRITE(9,1212)LHT                                                 FLSP  162
            WRITE(3,1212)LHT                                                 FLSP  163
            WRITE(9,1210)TOUT                                                FLSP  164
            WRITE(3,1210)TOUT                                                FLSP  165
            WRITE(9,1210)ASP,HSP,TPN,PHN,TM                                  FLSP  166
            WRITE(3,1210)ASP,HSP,TPN,PHN,TM                                  FLSP  167
            WRITE(9,1210)PRESS,PSR                                           FLSP  168
            WRITE(9,1210)CPMX,RL,TPENT                                       FLSP  169
            WRITE(3,1210)CPMX,RL,TPENT                                       FLSP  170
            WRITE(3,1210)PRESS,PSR                                           FLSP  171
      1210  FORMAT(1P6E16.8)                                                 FLSP  172
```

```
                      DO 1200 K=1,NVPTS                                          FLSP      173
                      CALL VALUES(PHVAL(K),URT,SCTCH,M,IM,1,1,2,WORK)            FLSP      174
                      WRITE(9,1210)PHVAL(K),X(K),(URT(L,1),L=1,NPDE),            FLSP      175
     175             * (URT(L,2),L=1,NPDE),(URT(L,3),L=1,NPDE)                   FLSP      176
                1200  CONTINUE                                                   FLSP      177
                      RETURN                                                     FLSP      178
                      END                                                        FLSP      179
```

```
      SUBROUTINE BNDRY(TIME,PH,U,UPH,DUDU,DBDUPH,DZDT,NPDE)                    BNDRY      2
      DIMENSION U(NPDE),UPH(NPDE),DZDT(NPDE)                                   BNDRY      3
      DIMENSION DBDU(NPDE),DBDUPH(NPDE)                                        BNDRY      4
      COMMON/ENDPT/PHL,PHR                                                     BNDRY      5
      COMMON/TABP/PRESS,PSR,NPDEM                                              BNDRY      6
C  FOR BOUNDARY CONDITIONS, SEE HIRSHFELDER, CURTISS,                          BNDRY      7
C  AND BIRD, PP. 762,763.                                                       BNDRY      8
      COMMON/TABCT/RL,CPMA,HO(20),R2D(20),R2DM(20)                             BNDRY      9
      COMMON/TABMF/RHUV(1000),DRHUV(1000)                                      BNDRY     10
      COMMON/TABSM/SMALL                                                       BNDRY     11
      COMMON/TABFM/FM                                                          BNDRY     12
      COMMON/START/NSTART,NBURN,NTRAN,KORD                                     BNDRY     13
      COMMON/TABCH/DZM                                                         BNDRY     14
      COMMON/TABV/PHVAL(401),UN(401),UT(20,401),UC(20),UR(20),W(20)            BNDRY     15
      DO 5 I=1,NPDE                                                            BNDRY     16
      DZDT(I)=0.0                                                              BNDRY     17
      DBDU(I)=0.0                                                              BNDRY     18
5     DBDUPH(I)=0.0                                                            BNDRY     19
      IF(PH.GT.PHL)GO TO 50                                                    BNDRY     20
      IF(NBURN.EQ.1)GO TO 7                                                    BNDRY     21
      DO 10 J=1,NPDE                                                           BNDRY     22
10    DBDU(J)=1.0                                                              BNDRY     23
      RETURN                                                                   BNDRY     24
C  BURNER STABILIZED FLAME.  WE NEED TO FIND THE MASS FLUX                      BNDRY     25
C  FRACTIONS AT THE COLD BOUNDARY.                                              BNDRY     26
7     IF(NTRAN.EQ.1)GO TO 12                                                   BNDRY     27
C  FIND R2D AT THE LEFT BOUNDARY.                                               BNDRY     28
      DO 8 J=1,NPDEM                                                           BNDRY     29
      IF(ABS(UPH(J)).LT.SMALL)UPH(J)=SMALL                                     BNDRY     30
8     R2D(J)=-RHUV(J)/UPH(J)                                                   BNDRY     31
12    CONTINUE                                                                 BNDRY     32
      DBDU(NPDE)=1.0                                                           BNDRY     33
      DO 20 J=1,NPDEM                                                          BNDRY     34
      DBDU(J)=1.0                                                              BNDRY     35
20    DBDUPH(J)=-R2D(J)/FM                                                     BNDRY     36
C  IF THE MASS FLUX FRACTION EP DOES NOT HAVE THE CORRECT VALUE,                BNDRY     37
C  WE USE DZDT TO MOVE IT TOWARDS Y COLD.                                       BNDRY     38
C  DZM IS CHOSEN SO THAT, IF R2D DOES NOT CHANGE, EP WOULD EQUAL                BNDRY     39
C  Y COLD AFTER A TIME INTERVAL TFINAL/10.0.                                    BNDRY     40
      DO 25 J=1,NPDEM                                                          BNDRY     41
      EP=U(J)-R2D(J)*UPH(J)/FM                                                 BNDRY     42
25    DZDT(J)=(UC(J)-EP)*DZM                                                   BNDRY     43
      RETURN                                                                   BNDRY     44
50    DO 30 J=1,NPDE                                                           BNDRY     45
30    DBDUPH(J)=1.0                                                            BNDRY     46
      RETURN                                                                   BNDRY     47
      END                                                                      BNDRY     48
```

```
      SUBROUTINE BKPT(NINT,NCN,FC,NVPTS)                                BKPT      2
      COMMON/ENDPT/PH0,PH5                                              BKPT      3
      COMMON/TABBK/PHBKPT(31)                                           BKPT      4
      COMMON/TABV/PHVAL(401),UN(401),V(20,401),UC(20),US(20),W(20)      BKPT      5
      COMMON/TABIN/PH2,PH3,PHCTO,PHCT                                   BKPT      6
      COMMON/START/NSTART,NBURN,NTRAN,KORD,NSKIP                        BKPT      7
C  COMPUTE THE BREAKPOINTS.                                             BKPT      8
      IF(NBURN.EQ.1)GO TO 650                                           BKPT      9
C  UNBOUNDED FLAME.                                                     BKPT     10
      PHCT=PH0+0.5*(PH5-PH0)                                            BKPT     11
      PH2=PH0+0.24*(PH5-PH0)                                            BKPT     12
      PH3=PH0+0.64*(PH5-PH0)                                            BKPT     13
      ALP=2.0*ALOG10(FC)/FLOAT(NINT-NCN)                                BKPT     14
      ALP=10.0**ALP                                                     BKPT     15
      I=(NINT-NCN)/2                                                    BKPT     16
      P=(PH5-PH0)/(FLOAT(NCN)+2.0*ALP*(ALP**I-1.0)/(ALP-1.0))           BKPT     17
      PHBKPT(1)=PH0                                                     BKPT     18
      DP=FC*P                                                           BKPT     19
      NU=(NINT-NCN)/2                                                   BKPT     20
      DO 11 K=1,NU                                                      BKPT     21
      KP=K+1                                                            BKPT     22
      PHBKPT(KP)=PHBKPT(K)+DP                                           BKPT     23
11    DP=DP/ALP                                                         BKPT     24
      NL=NU+1                                                           BKPT     25
      NU=NU+NCN                                                         BKPT     26
      DP=P                                                              BKPT     27
      DO 12 K=NL,NU                                                     BKPT     28
      KP=K+1                                                            BKPT     29
12    PHBKPT(KP)=PHBKPT(K)+DP                                           BKPT     30
      NL=NU+1                                                           BKPT     31
      NU=NINT                                                           BKPT     32
      DO 13 K=NL,NU                                                     BKPT     33
      KP=K+1                                                            BKPT     34
      DP=DP*ALP                                                         BKPT     35
13    PHBKPT(KP)=PHBKPT(K)+DP                                           BKPT     36
      PHBKPT(NINT+1)=PH5                                                BKPT     37
      GO TO 660                                                         BKPT     38
C  BURNER STABILIZED FLAME.                                             BKPT     39
650   PH2=PH0+0.1*(PH5-PH0)                                             BKPT     40
      PH3=PH0+0.3*(PH5-PH0)                                             BKPT     41
      ALP=ALOG10(FC)/FLOAT(NINT-NCN)                                    BKPT     42
      ALP=10.0**ALP                                                     BKPT     43
      I=NINT-NCN                                                        BKPT     44
      P=(PH5-PH0)/(FLOAT(NCN)+ALP*(ALP**I-1.0)/(ALP-1.0))               BKPT     45
      PHBKPT(1)=PH0                                                     BKPT     46
      DO 653 K=1,NCN                                                    BKPT     47
      KP=K+1                                                            BKPT     48
653   PHBKPT(KP)=PHBKPT(K)+P                                            BKPT     49
      NL=NCN+1                                                          BKPT     50
      NU=NINT                                                           BKPT     51
      DO 657 K=NL,NU                                                    BKPT     52
      KP=K+1                                                            BKPT     53
      P=P*ALP                                                           BKPT     54
657   PHBKPT(KP)=PHBKPT(K)+P                                            BKPT     55
660   CONTINUE                                                          BKPT     56
      NU=NU+1                                                           BKPT     57
      WRITE(3,39)(PHBKPT(L),L=1,NU)                                     BKPT     58
```

```
      C  DEFINE THE EVALUATION POINTS.                                  BKPT   59
           NSP=8                                                        BKPT   60
           NVPTS=NSP*NINT+1                                             BKPT   61
           DO 17 K=1,NVPTS                                              BKPT   62
           KV=1+(K-1)*NSP                                               BKPT   63
        17 PHVAL(KV)=PHBKPT(K)                                          BKPT   64
           NSPP=NSP+1                                                   BKPT   65
           DO 19 K=NSPP,NVPTS,NSP                                       BKPT   66
           KM=K-NSP                                                     BKPT   67
           DP=(PHVAL(K)-PHVAL(KM))/FLOAT(NSP)                           BKPT   68
           DO 19 J=2,NSP                                                BKPT   69
           KT=KM+J-1                                                    BKPT   70
           PHVAL(KT)=PHVAL(KM)+DP*FLOAT(J-1)                            BKPT   71
        19 CONTINUE                                                     BKPT   72
           NSKIP=NINT/10                                                BKPT   73
           IF(NSKIP.LT.1)NSKIP=1                                        BKPT   74
           WRITE(3,39)(PHVAL(I),I=1,NVPTS,NSKIP)                        BKPT   75
        39 FORMAT(1P10E12.4)                                            BKPT   76
           RETURN                                                       BKPT   77
           END                                                          BKPT   78
```

```
SUBROUTINE RT          76/76   OPT=1 ROUND=*-*/ TRACE          FTN 4.8*498          04/15/80  11.09.01          AGE     1

      1                SUBROUTINE RT(U,YN,K,NPDE,KPDE,IC,KSKIP)                                          RT          2
                       DIMENSION U(1),R(1)                                                               RT          3
                       COMMON/TARP/PRESS,PSR,NPDEM                                                       RT          4
                       COMMON/TARAB/ASP,HSP,TPN,PHN,TMSP,TMSPH,TMSPH2,TMSTP,TPENT                        RT          5
      5                COMMON/TABRY/T,RH,Y1,Y2,Y3,Y4,Y5,Y6,Y7,Y8,Y9,Y10                                   RT          6
                       COMMON/TARAM/YAB,WAB                                                              RT          7
                 C  THE DIMENSION OF RK MUST BE AT LEAST THE NUMBER OF REACTIONS TIMES                   RT          8
                 C  THE NUMBER OF COLLOCATION POINTS.                                                    RT          9
                       DIMENSION RK(2000)                                                                RT         10
     10          C  H2-O2 KINETICS.      WARNATZ, LLYOD'S.                                               RT         11
                       IST=  30*(IC-1)                                                                   RT         12
                       T=U(NPDE)*TPN                                                                     RT         13
                       Y 1=U( 1)/  1.00                                                                  RT         14
                       Y 2=U( 2)/ 17.00                                                                  RT         15
     15                Y 3=U( 3)/ 16.00                                                                  RT         16
                       Y 4=U( 4)/ 33.00                                                                  RT         17
                       Y 5=U( 5)/ 34.00                                                                  RT         18
                       Y 6=U( 6)/  2.00                                                                  RT         19
                       Y 7=U( 7)/ 32.00                                                                  RT         20
     20                Y 8=U( 8)/ 18.00                                                                  RT         21
                       Y 9=YN/ 28.00                                                                     RT         22
                       Y10=Y1+Y2+Y3+Y4+Y5+Y6+Y7+Y8+Y9                                                    RT         23
                       Y11=Y6+0.4*Y7+6.0*Y8+0.4*Y9                                                       RT         24
                       RH=PSR/(T*Y10)                                                                    RT         25
     25                RH2=RH*RH                                                                         RT         26
                       IF(KSKIP.GT.1)GO TO 10                                                            RT         27
                       RK(IST+  1)=  2.2000E+13*EXP( -2.5900E+03/T)                                      RT         28
                       RK(IST+  2)=  9.3000E+13*EXP( -1.0250E+04/T)                                      RT         29
                       RK(IST+  3)=  2.2000E+14*EXP( -8.4500E+03/T)                                      RT         30
     30                RK(IST+  4)=  1.0000E+13                                                          RT         31
                       RK(IST+  5)=  1.8000E+10*(T**(  1.0000E+00))*EXP( -4.4800E+03/T)                  RT         32
                       RK(IST+  6)=  8.3000E+09*(T**(  1.0000E+00))*EXP( -3.5000E+03/T)                  RT         33
                       RK(IST+  7)=  6.3000E+12*EXP( -5.5000E+02/T)                                      RT         34
                       RK(IST+  8)=  6.8000E+13*EXP( -4.2400E+03/T)                                      RT         35
     35                RK(IST+  9)=  9.0000E+17*(T**( -1.0000E+00))                                      RT         36
                       RK(IST+ 10)=  2.2000E+22*(T**( -2.0000E+00))                                      RT         37
                       RK(IST+ 11)=  5.0000E+15*EXP(  5.0000E+02/T)                                      RT         38
                       RK(IST+ 12)=  2.5000E+14*EXP( -4.5000E+02/T)                                      RT         39
                       RK(IST+ 13)=  2.5000E+13*EXP( -3.5000E+02/T)                                      RT         40
     40                RK(IST+ 14)=  1.5000E+13*EXP( -5.0000E+02/T)                                      RT         41
                       RK(IST+ 15)=  5.0000E+13*EXP( -5.0000E+02/T)                                      RT         42
                       RK(IST+ 16)=  5.0000E+12*EXP( -5.0000E+02/T)                                      RT         43
                       RK(IST+ 17)=  9.1000E+14*EXP(  2.5500E+03/T)                                      RT         44
                       RK(IST+ 18)=  3.2000E+14*EXP( -4.5000E+03/T)                                      RT         45
     45                RK(IST+ 19)=  8.5000E+12*EXP( -5.0000E+02/T)                                      RT         46
                       RK(IST+ 20)=  8.8000E+14*EXP( -4.8300E+04/T)                                      RT         47
                       RK(IST+ 21)=  7.0000E+13*EXP(  9.0000E+02/T)                                      RT         48
                       RK(IST+ 22)=  7.0000E+15*EXP( -2.3000E+04/T)                                      RT         49
                       RK(IST+ 23)=  1.2000E+13*EXP( -2.0200E+04/T)                                      RT         50
     50                RK(IST+ 24)=  7.3000E+11*EXP( -9.4000E+03/T)                                      RT         51
                       RK(IST+ 25)=  1.7000E+12*EXP( -1.9000E+03/T)                                      RT         52
                       RK(IST+ 26)=  1.2000E+17*EXP( -2.2900E+04/T)                                      RT         53
                       RK(IST+ 27)=  2.8000E+13*EXP( -3.2000E+03/T)                                      RT         54
                       RK(IST+ 28)=  2.8000E+13*EXP( -3.2000E+03/T)                                      RT         55
     55                RK(IST+ 29)=  1.0000E+13*EXP( -1.1000E+02/T)                                      RT         56
                       RK(IST+ 30)=  2.8000E+13*EXP( -1.6500E+04/T)                                      RT         57
                  10   CONTINUE                                                                          RT         58
```

```
      R  1=RH*RK(IST+  1)*Y 2*Y 6                                       RT         59
      R  2=RH*RK(IST+  2)*Y 1*Y 8                                       RT         60
   60 R  3=RH*RK(IST+  3)*Y 1*Y 7                                       RT         61
      R  4=RH*RK(IST+  4)*Y 3*Y 2                                       RT         62
      R  5=RH*RK(IST+  5)*Y 3*Y 6                                       RT         63
      R  6=RH*RK(IST+  6)*Y 1*Y 2                                       RT         64
      R  7=RH*RK(IST+  7)*Y 2*Y 2                                       RT         65
   65 R  8=RH*RK(IST+  8)*Y 3*Y 8                                       RT         66
      R  9=RH2*RK(IST+  9)*Y 1*Y 1*Y11                                  RT         67
      R 10=RH2*RK(IST+ 10)*Y 1*Y 2*Y11                                  RT         68
      R 11=RH2*RK(IST+ 11)*Y 1*Y 7*Y11                                  RT         69
      R 12=RH*RK(IST+ 12)*Y 1*Y 4                                       RT         70
   70 R 13=RH*RK(IST+ 13)*Y 1*Y 4                                       RT         71
      R 14=RH*RK(IST+ 14)*Y 1*Y 4                                       RT         72
      R 15=RH*RK(IST+ 15)*Y 2*Y 4                                       RT         73
      R 16=RH*RK(IST+ 16)*Y 3*Y 4                                       RT         74
      R 17=RH2*RK(IST+ 17)*Y 2*Y 2*Y10                                  RT         75
   75 R 18=RH*RK(IST+ 18)*Y 1*Y 5                                       RT         76
      R 19=RH*RK(IST+ 19)*Y 4*Y 4                                       RT         77
      R 20=RH*RK(IST+ 20)*Y 6*Y11                                       RT         78
      R 21=RH2*RK(IST+ 21)*Y 3*Y 3*Y10                                  RT         79
      R 22=RH*RK(IST+ 22)*Y 4*Y11                                       RT         80
   80 R 23=RH*RK(IST+ 23)*Y 2*Y 2                                       RT         81
      R 24=RH*RK(IST+ 24)*Y 4*Y 6                                       RT         82
      R 25=RH*RK(IST+ 25)*Y 1*Y 5                                       RT         83
      R 26=RH*RK(IST+ 26)*Y 5*Y10                                       RT         84
      R 27=RH*RK(IST+ 27)*Y 3*Y 5                                       RT         85
   85 R 28=RH*RK(IST+ 28)*Y 3*Y 5                                       RT         86
      R 29=RH*RK(IST+ 29)*Y 2*Y 5                                       RT         87
      R 30=RH*RK(IST+ 30)*Y 4*Y 8                                       RT         88
      R( 1)=  1.00*                                                     RT         89
     *(+R  1-R  2-R  3+R  4+R  5-R  6-2.*R  9-R 10-R 11-R 12-R 13       RT         90
   90 * -R 14-R 18+2.*R 20+R 22+R 23+R 24-R 25)                         RT         91
      R( 2)= 17.00*                                                     RT         92
     *(-R  1+R  2+R  3-R  4+R  5-R  6-2.*R  7+2.*R  8-R 10+2.*R 12      RT         93
     * -R 15+R 16-2.*R 17+R 18-2.*R 24+2.*R 26+R 27-R 29+R 30)          RT         94
      R( 3)= 16.00*                                                     RT         95
   95 *(+R  3-R  4-R  5+R  6+R  7-R  8+R 14-R 16-2.*R 21-R 27-R 28      RT         96
     * )                                                               RT         97
      R( 4)= 33.00*                                                     RT         98
     *(+R 11-R 12-R 13-R 14-R 15-R 16-2.*R 19-R 22+R 23-R 24+R 25       RT         99
     * +R 27+R 29-R 30)                                                RT        100
  100  R( 5)= 34.00*                                                    RT        101
     *(+R 17-R 18+R 19+R 24-R 25-R 26-R 27-R 28-R 29+R 30)              RT        102
      R( 6)=  2.00*                                                     RT        103
     *(-R  1+R  2-R  5+R  6+R  9+R 13-R 20-R 24+R 25)                   RT        104
      R( 7)= 32.00*                                                     RT        105
  105 *(-R  3+R  4-R 11+R 13+R 15+R 16+R 19+R 21+R 22+R 28)              RT        106
      R( 8)= 18.00*                                                     RT        107
     *(+R  1-R  2+R  7-R  8+R 10+R 14+R 15+R 19+R 24+R 29-R 30)         RT        108
      R( 9)=  0.00                                                      RT        109
      RETURN                                                            RT        110
  110 END                                                               RT        111
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
14	I	U	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
15	I	U	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
16	I	U	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
17	I	U	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
18	I	U	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
19	I	U	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
20	I	U	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
91	I	K	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
94	I	K	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
97	I	K	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
100	I	K	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
102	I	K	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
104	I	K	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
105	I	K	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
108	I	K	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.

```
      SUBROUTINE PDECOL(T0,TOUT,DT,XLEFT,XRIGHT,EPS,NINT,KORD,NCC,NPDE,MF,     PDECOL     2
     *     INDEX,WORK,IWORK,SREC)                                               PDECOL     3
C                                                                               PDECOL     4
C-------------------------------------------------------------------------      PDECOL     5
C-------------------------------------------------------------------------      PDECOL     6
C                                                                               PDECOL     7
C THIS IS THE SEPT 23, 1977 VERSION OF PDECOL.                                  PDECOL     8
C                                                                               PDECOL     9
C THIS PACKAGE WAS CONSTRUCTED SO AS TO CONFORM TO AS MANY ANSI-FORTRAN         PDECOL    10
C RULES AS WAS CONVIENTLY POSSIBLE.  THE FORTRAN USED VIOLATES ANSI             PDECOL    11
C STANDARDS IN THE TWO WAYS LISTED BELOW....                                    PDECOL    12
C                                                                               PDECOL    13
C   1. SUBSCRIPTS OF THE GENERAL FORM C*V1 + V2 + V3 ARE USED                   PDECOL    14
C      (POSSIBLY IN A PERMUTED ORDER), WHERE C IS AN INTEGER CONSTANT           PDECOL    15
C      AND V1, V2, AND V3 ARE INTEGER VARIABLES.                                PDECOL    16
C                                                                               PDECOL    17
C   2. ARRAY NAMES APPEAR SINGLY IN DATA STATEMENTS IN THE ROUTINES             PDECOL    18
C      BSPLVN AND COSET.                                                        PDECOL    19
C                                                                               PDECOL    20
C-------------------------------------------------------------------------      PDECOL    21
C-------------------------------------------------------------------------      PDECOL    22
C                                                                               PDECOL    23
C-------------------------------------------------------------------------      PDECOL    24
C PDECOL IS THE DRIVER ROUTINE FOR A SOPHISTICATED PACKAGE OF                   PDECOL    25
C SUBROUTINES WHICH IS DESIGNED TO SOLVE THE GENERAL SYSTEM OF                  PDECOL    26
C NPDE NONLINEAR PARTIAL DIFFERENTIAL EQUATIONS OF AT MOST SECOND               PDECOL    27
C ORDER ON THE INTERVAL (XLEFT,XRIGHT) FOR T .GT. T0 WHICH IS OF THE            PDECOL    28
C FORM....                                                                      PDECOL    29
C                                                                               PDECOL    30
C       DU/DT  =  F( T, X, U, UX, UXX )                                         PDECOL    31
C                                                                               PDECOL    32
C WHERE                                                                         PDECOL    33
C                                                                               PDECOL    34
C            U  =  ( U(1),  U(2), ... ,  U(NPDE) )                              PDECOL    35
C           UX  =  ( UX(1), UX(2), ... , UX(NPDE) )                             PDECOL    36
C          UXX  =  (UXX(1),UXX(2), ... ,UXX(NPDE) ) .                           PDECOL    37
C                                                                               PDECOL    38
C EACH U(K) IS A FUNCTION OF THE SCALAR QUANTITIES T AND X.                     PDECOL    39
C UX(K) REPRESENTS THE FIRST PARTIAL DERIVATIVE OF U(K) WITH RESPECT            PDECOL    40
C TO THE VARIABLE X.  UXX(K) REPRESENTS THE SECOND PARTIAL DERIVATIVE           PDECOL    41
C OF U(K) WITH RESPECT TO THE VARIABLE X. AND DU/DT IS THE VECTOR OF            PDECOL    42
C PARTIAL DERIVATIVES OF U WITH RESPECT TO THE TIME VARIABLE T.                 PDECOL    43
C F  REPRESENTS AN ARBITRARY VECTOR VALUED FUNCTION WHOSE NPDE                  PDECOL    44
C COMPONENTS DEFINE THE RESPECTIVE PARTIAL DIFFERENTIAL EQUATIONS OF            PDECOL    45
C THE PDE SYSTEM.  SEE SUBROUTINE F DESCRIPTION BELOW.                          PDECOL    46
C                                                                               PDECOL    47
CC  THE PROGRAM IS MODIFIED TO SOLVE THE SYSTEM USING                           PDECOL    48
CC  SUCCESSIVE CALCULATION.                                                     PDECOL    49
CC  THAT IS, THE PDE'S ARE UNCOUPLED AND SOLVED IN ORDER.                       PDECOL    50
CC  THE EQUATIONS THAT VARY THE MOST( RADICALS) SHOULD BE LISTED FIRST.         PDECOL    51
C BOUNDARY CONDITIONS                                                           PDECOL    52
C                                                                               PDECOL    53
C   DEPENDING ON THE TYPE OF PDE(S), 0, 1, OR 2 BOUNDARY CONDITIONS             PDECOL    54
C   ARE REQUIRED FOR EACH PDE IN THE SYSTEM. THESE ARE IMPOSED AT XLEFT         PDECOL    55
C   AND/OR XRIGHT AND EACH MUST BE OF THE FORM....                              PDECOL    56
C                                                                               PDECOL    57
C       R(U,UX)  =  /(T)                                                        PDECOL    58
```

```
C                                                                       PDECOL  59
C   WHERE  B  AND  Z  ARE ARBITRARY VECTOR VALUED FUNCTIONS WITH        PDECOL  60
C   NPDE COMPONENTS AND  U, UX, AND T  ARE AS ABOVE.  THESE BOUNDARY    PDECOL  61
C   CONDITIONS MUST BE CONSISTENT WITH THE INITIAL CONDITIONS WHICH ARE PDECOL  62
C   DESCRIBED NEXT.                                                     PDECOL  63
C                                                                       PDECOL  64
C INITIAL CONDITIONS                                                    PDECOL  65
C                                                                       PDECOL  66
C   EACH SOLUTION COMPONENT  U(K)  IS ASSUMED TO BE A KNOWN (USER       PDECOL  67
C   PROVIDED) FUNCTION OF  X  AT THE INITIAL TIME T = T0.  THE          PDECOL  68
C   INITIAL CONDITION FUNCTIONS MUST BE CONSISTENT WITH THE BOUNDARY    PDECOL  69
C   CONDITIONS ABOVE, I.E. THE INITIAL CONDITION FUNCTIONS MUST         PDECOL  70
C   SATISFY THE BOUNDARY CONDITIONS FOR T = T0.  SEE SUBROUTINE UINIT   PDECOL  71
C   DESCRIPTION BELOW.                                                  PDECOL  72
C-----------------------------------------------------------------------PDECOL  73
C                                                                       PDECOL  74
C REQUIRED USER SUPPLIED SUBROUTINES                                    PDECOL  75
C                                                                       PDECOL  76
C THE USER IS REQUIRED TO CONSTRUCT THREE SUBPROGRAMS AND A MAIN        PDECOL  77
C PROGRAM WHICH DEFINE THE PDE PROBLEM WHOSE SOLUTION IS TO BE          PDECOL  78
C ATTEMPTED.  THE THREE SUBPROGRAMS ARE...                              PDECOL  79
C                                                                       PDECOL  80
CC  1)  SUBROUTINE F(T,X,U,UX,UXX,FVAL,NPDE,KPDE,IC,KSKT,KSKX)         PDECOL  81
CC    DIMENSION U(NPDE),UX(NPDE),UXX(NPDE)                             PDECOL  82
C       THIS ROUTINE DEFINES THE DESIRED PARTIAL DIFFERENTIAL           PDECOL  83
C       EQUATIONS TO BE SOLVED.  THE PACKAGE PROVIDES VALUES OF THE     PDECOL  84
C       INPUT SCALARS T AND X AND INPUT ARRAYS (LENGTH NPDE) U, UX,     PDECOL  85
C       AND UXX, AND THE USER MUST CONSTRUCT THIS ROUTINE TO COMPUTE    PDECOL  86
C       THE OUTPUT ARRAY FVAL (LENGTH NPDE) WHICH CONTAINS THE          PDECOL  87
C       CORRESPONDING VALUES OF THE RIGHT HAND SIDES OF THE DESIRED     PDECOL  88
C       PARTIAL DIFFERENTIAL EQUATIONS, I.E.                            PDECOL  89
C                                                                       PDECOL  90
C       FVAL(K) = THE VALUE OF THE RIGHT HAND SIDE OF THE K-TH PDE IN   PDECOL  91
C                 THE PDE SYSTEM ABOVE, FOR K = 1 TO NPDE.              PDECOL  92
C                                                                       PDECOL  93
C       THE INCOMING VALUE OF THE SCALAR QUANTITY X WILL BE A           PDECOL  94
C       COLLOCATION POINT VALUE (SEE INITAL AND COLPNT) AND THE         PDECOL  95
C       INCOMING VALUES IN THE ARRAYS U, UX AND UXX CORRESPOND TO THIS  PDECOL  96
C       POINT X AND TIME T.                                             PDECOL  97
C     RETURN                                                            PDECOL  98
C     END                                                               PDECOL  99
C                                                                       PDECOL 100
C 2)  SUBROUTINE BNDRY( T, X, U, UX, DBDU, DBDUX, DZDT, NPDE )          PDECOL 101
C     DIMENSION U(NPDE), UX(NPDE), DZDT(NPDE)                           PDECOL 102
CC    DIMENSION DBDU(NPDE),DBDUX(NPDE)                                  PDECOL 103
C       THIS ROUTINE IS USED TO PROVIDE THE PDE PACKAGE WITH NEEDED     PDECOL 104
C       INFORMATION ABOUT THE BOUNDARY CONDITION FUNCTIONS B AND Z      PDECOL 105
C       ABOVE.  THE PACKAGE PROVIDES VALUES OF THE INPUT VARIABLES      PDECOL 106
C       T, X, U, AND UX, AND THE USER IS TO DEFINE THE CORRESPONDING    PDECOL 107
C       OUTPUT VALUES OF THE DERIVATIVES OF THE FUNCTIONS B AND Z       PDECOL 108
C       WHERE....                                                       PDECOL 109
CC       DBDU(K)=PARTIAL DERIVATIVE OF THE K*TH COMPONENT OF B          PDECOL 110
CC       WITH RESPECT TO U(K).                                          PDECOL 111
CC       DBDUX(K)=PARTIAL DERIVATIVE OF THE K*TH COMPONENT OF B         PDECOL 112
CC       WITH RESPECT TO UX(K).                                         PDECOL 113
CC       DZDT(K)=DERIVATIVE OF THE K*TH COMPONENT OF Z                  PDECOL 114
CC       WITH RESPECT TO T.                                             PDECOL 115
```

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

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

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

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

```
C                 TO BE SOLVED (USED ONLY ON FIRST CALL).                    PDECOL  344
C     MF     =  THE METHOD FLAG  (USED ONLY ON FIRST CALL. UNLESS            PDECOL  345
C                 INDEX = 4).  ALLOWED VALUES ARE 11, 12, 21, 22.            PDECOL  346
C                 FOR FIRST ATTEMPTS WE RECOMMEND THE USE OF MF = 22.        PDECOL  347
C                 MF HAS TWO DECIMAL DIGITS, METH AND MITER                  PDECOL  348
C                 (MF = 10*METH + MITER).                                    PDECOL  349
C                 METH IS THE BASIC METHOD INDICATOR..                       PDECOL  350
C                   METH = 1  MEANS THE ADAMS METHODS (GENERALIZATIONS OF    PDECOL  351
C                             CRANK-NICOLSON).                               PDECOL  352
C                   METH = 2  MEANS THE BACKWARD DIFFERENTIATION             PDECOL  353
C                             FORMULAS (BDF), OR STIFF METHODS OF GEAR.      PDECOL  354
C                 MITER IS THE ITERATION METHOD INDICATOR                    PDECOL  355
C                 AND DETERMINES HOW THE JACOBIAN MATRIX IS                  PDECOL  356
C                 TO BE COMPUTED..                                           PDECOL  357
C                   MITER = 1 MEANS CHORD METHOD WITH ANALYTIC JACOBIAN.     PDECOL  358
C                             FOR THIS USER SUPPLIES SUBROUTINE DERIVF.      PDECOL  359
C                             SEE DESCRIPTION ABOVE.                         PDECOL  360
C                   MITER = 2 MEANS CHORD METHOD WITH JACOBIAN CALCULATED    PDECOL  361
C                             INTERNALLY BY FINITE DIFFERENCES.  SEE         PDECOL  362
C                             SUBROUTINES PSETIB AND DIFFF.                  PDECOL  363
C     INDEX  =  INTEGER USED ON INPUT TO INDICATE TYPE OF CALL.              PDECOL  364
C                 WITH THE FOLLOWING VALUES AND MEANINGS..                   PDECOL  365
C                    1    THIS IS THE FIRST CALL FOR THIS PROBLEM.          PDECOL  366
C                    0    THIS IS NOT THE FIRST CALL FOR THIS PROBLEM,      PDECOL  367
C                         AND INTEGRATION IS TO CONTINUE.                   PDECOL  368
C                    2    SAME AS 0 EXCEPT THAT TOUT IS TO BE HIT           PDECOL  369
C                         EXACTLY (NO INTERPOLATION IS DONE).  SEE NOTE      PDECOL  370
C                         BELOW.  ASSUMES TOUT .GE. THE CURRENT T.          PDECOL  371
C                         IF TOUT IS .LT. THE CURRENT TIME, THEN TOUT IS    PDECOL  372
C                         RESET TO THE CURRENT TIME AND CONTROL IS          PDECOL  373
C                         RETURNED TO THE USER.  A CALL TO VALUES WILL      PDECOL  374
C                         PRODUCE ANSWERS FOR THE NEW VALUE OF TOUT.        PDECOL  375
C                    3    SAME AS 0 EXCEPT CONTROL RETURNS TO CALLING       PDECOL  376
C                         PROGRAM AFTER ONE STEP.  TOUT IS IGNORED AND      PDECOL  377
C                         DT MUST BE SET .GT. 0 TO A MAXIMUM ALLOWED        PDECOL  378
C                         DT VALUE. SEE ABOVE.                              PDECOL  379
C                    4    THIS IS NOT THE FIRST CALL FOR THE PROBLEM,       PDECOL  380
C                         AND THE USER HAS RESET EPS AND/OR MF.             PDECOL  381
C                 SINCE THE NORMAL OUTPUT VALUE OF INDEX IS 0,              PDECOL  382
C                 IT NEED NOT BE RESET FOR NORMAL CONTINUATION.             PDECOL  383
C                                                                           PDECOL  384
C  NOTE.. THE PACKAGE MUST HAVE TAKEN AT LEAST ONE SUCCESSFUL TIME          PDECOL  385
C STEP BEFORE A CALL WITH INDEX = 2 OR 4 IS ALLOWED.                        PDECOL  386
C AFTER THE INITIAL CALL, IF A NORMAL RETURN OCCURRED AND A NORMAL          PDECOL  387
C CONTINUATION IS DESIRED, SIMPLY RESET TOUT AND CALL AGAIN.                PDECOL  388
C ALL OTHER PARAMETERS WILL BE READY FOR THE NEXT CALL.                     PDECOL  389
C A CHANGE OF PARAMETERS WITH INDEX = 4 CAN BE MADE AFTER                   PDECOL  390
C EITHER A SUCCESSFUL OR AN UNSUCCESSFUL RETURN PROVIDED AT LEAST           PDECOL  391
C ONE SUCCESSFUL TIME STEP HAS BEEN MADE.                                   PDECOL  392
C                                                                           PDECOL  393
C     WORK   =  FLOATING POINT WORKING ARRAY FOR PDECOL.  WE RECOMMEND      PDECOL  394
C                 THAT IT BE INITIALIZED TO ZERO BEFORE THE FIRST CALL      PDECOL  395
C                 TO PDECOL.  ITS TOTAL LENGTH MUST BE AT LEAST             PDECOL  396
C                                                                           PDECOL  397
CC        KORD * (3*NPDE + NCPTS*(3*MAXORD+2) +                             PDECOL  398
CC        NPDE*NCPTS*(3*ML+MAXORD+7)                                        PDECOL  399
C                                                                           PDECOL  400
```

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

```
C       -12    ILLEGAL NCC VALUE.                                        PDECOL  458
C       -13    ILLEGAL NPDE VALUE.                                       PDECOL  459
C       -14    ILLEGAL MF VALUE.                                         PDECOL  460
C       -15    ILLEGAL BREAKPOINTS -- NOT STRICTLY INCREASING.           PDECOL  461
C       -16    INSUFFICIENT STORAGE FOR WORK OR IWORK.                   PDECOL  462
C-------------------------------------------------------------------------PDECOL  463
C                                                                         PDECOL  464
C                                                                         PDECOL  465
C SUMMARY OF ALL PACKAGE ROUTINES                                         PDECOL  466
C                                                                         PDECOL  467
C PDECOL - STORAGE ALLOCATION, ERROR CHECKING, INITIALIZATION, REPEATED   PDECOL  468
C          CALLS TO STIFIH TO ADVANCE THE TIME.                           PDECOL  469
C                                                                         PDECOL  470
C INTERP - INTERPOLATES COMPUTED BASIS FUNCTION COEFFICIENTS TO THE       PDECOL  471
C          DESIRED OUTPUT TIMES, TOUT, FOR USE BY VALUES.                 PDECOL  472
C                                                                         PDECOL  473
C INITAL - INITIALIZATION, GENERATION AND STORAGE OF PIECEWISE POLY-       PDECOL  474
C          NOMIAL SPACE BASIS FUNCTION VALUES AND DERIVATIVES, DET-       PDECOL  475
C          ERMINES THE BASIS FUNCTION COEFFICINTS OF THE PIECEWISE        PDECOL  476
C          POLYNOMIALS WHICH INTERPOLATE THE USERS INITIAL CONDITIONS.    PDECOL  477
C                                                                         PDECOL  478
C COLPNT - GENERATION OF REQUIRED COLLOCATION POINTS.                     PDECOL  479
C                                                                         PDECOL  480
C BSPLVD - B-SPLINE PACKAGE ROUTINES WHICH ALLOW FOR EVALUATION OF        PDECOL  481
C BSPLVN   ANY B-SPLINE BASIS FUNCTION OR DERIVATIVE VALUE.               PDECOL  482
C INTERV                                                                  PDECOL  483
C                                                                         PDECOL  484
C VALUES - GENERATION AT ANY POINT(S) OF VALUES OF THE COMPUTED           PDECOL  485
C          APPROXIMATE SOLUTION AND ITS DERIVATIVES WHICH ARE             PDECOL  486
C          PIECEWISE POLYNOMIALS.  THE SUBROUTINE IS CALLED ONLY BY       PDECOL  487
C          THE USER.                                                      PDECOL  488
C                                                                         PDECOL  489
C STIFIH - CORE INTEGRATOR, TAKES SINGLE TIME STEPS TO ADVANCE THE        PDECOL  490
C          TIME.  ASSEMBLES AND SOLVES THE PROPER NONLINEAR EQUATIONS     PDECOL  491
C          WHICH ARE RELATED TO USE OF ADAMS OR GEAR TYPE INTEGRATION     PDECOL  492
C          FORMULAS.  CHOOSES PROPER STEP SIZE AND INTEGRATION FORMULA    PDECOL  493
C          ORDER TO MAINTAIN A DESIRED ACCURACY.  DESIGNED FOR ODE        PDECOL  494
C          PROBLEMS OF THE FORM A * (DY/DT) = G(T,Y).                     PDECOL  495
C                                                                         PDECOL  496
C COSET  - GENERATES INTEGRATION FORMULA AND ERROR CONTROL COEFFICIENTS.   PDECOL  497
C                                                                         PDECOL  498
C RES    - COMPUTES RESIDUAL VECTORS USED IN SOLVING THE NONLINEAR        PDECOL  499
C          EQUATIONS BY A MODIFIED NEWTON METHOD.                         PDECOL  500
C                                                                         PDECOL  501
C DIFFUN - COMPUTES A**-1 * G(T,Y) WHERE A AND G ARE AS ABOVE (STIFIH).   PDECOL  502
C                                                                         PDECOL  503
C ADDA   - ADDS THE A MATRIX TO A GIVEN MATRIX IN BAND FORM.              PDECOL  504
C                                                                         PDECOL  505
C EVAL   - EVALUATES THE COMPUTED PIECEWISE POLYNOMIAL SOLUTION AND       PDECOL  506
C          DERIVATIVES AT COLLOCATION POINTS.                             PDECOL  507
C                                                                         PDECOL  508
C GFUN   - EVALUATES THE FUNCTION G(T,Y) BY CALLING EVAL AND THE USER     PDECOL  509
C          SUBROUTINES F AND BNDRY.                                       PDECOL  510
C                                                                         PDECOL  511
C PSETIH - GENERATES PROPER JACOBIAN MATRICES REQUIRED BY THE MODIFIED    PDECOL  512
C          NEWTON METHOD.                                                 PDECOL  513
C                                                                         PDECOL  514
```

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

```
C Y                (MAXDER+1)  AND THEIR SCALED TIME DERIVATIVES.          PDECOL  572
C WORK(IW10)                                                               PDECOL  573
C                                                                          PDECOL  574
CC DFDU            NPDE        WORKING STORAGE USED TO COMPUTE THE         PDECOL  575
C WORK(IW11)                   JACOBIAN MATRIX.                            PDECOL  576
C                                                                          PDECOL  577
CC DFDUX           NPDE        WORKING STORAGE USED TO COMPUTE THE         PDECOL  578
C WORK(IW12)                   JACOBIAN MATRIX.                            PDECOL  579
C                                                                          PDECOL  580
CC DFDUXX          NPDE        WORKING STORAGE USED TO COMPUTE THE         PDECOL  581
C WORK(IW13)                   JACOBIAN MATRIX.                            PDECOL  582
C                                                                          PDECOL  583
CC DBDU            NPDE        BOUNDARY CONDITION INFORMATION              PDECOL  584
C WORK(IW14)                                                               PDECOL  585
C                                                                          PDECOL  586
CC DBDUX           NPDE        BOUNDARY CONDITION INFORMATION              PDECOL  587
C WORK(IW15)                                                               PDECOL  588
C                                                                          PDECOL  589
C DZDT             NPDE        BOUNDARY CONDITION INFORMATION.             PDECOL  590
C WORK(IW16)                                                               PDECOL  591
C                                                                          PDECOL  592
C PW               NPDE*NCPTS* STORAGE AND PROCESSING OF THE JACOBIAN      PDECOL  593
C WORK(IW17)         (3*ML+1)  MATRIX.                                     PDECOL  594
C                                                                          PDECOL  595
C ILEFT            NCPTS       POINTERS TO BREAKPOINT SEQUENCE FOR         PDECOL  596
C IWORK                        GENERATION OF BASIS FUNCTION VALUES.        PDECOL  597
C                                                                          PDECOL  598
C IPIV             NPDE*NCPTS  PIVOT INFORMATION FOR THE LU DECOMPOSED     PDECOL  599
C IWORK(IW18)                  JACOBIAN MATRIX PW.                         PDECOL  600
C                                                                          PDECOL  601
C WHERE...                                                                 PDECOL  602
C                                                                          PDECOL  603
C      NCPTS = KORD*NINT - NCC*(NINT-1)                                    PDECOL  604
C  ML=KORD + IQUAD - 2                                                     PDECOL  605
CC  ASSUME IQUAD = 0.                                                      PDECOL  606
C      IQUAD = 1 IF KORD = 3 AND A NULL BOUNDARY CONDITION EXISTS          PDECOL  607
C      IQUAD = 0 OTHERWISE                                                 PDECOL  608
C     MAXDER = 5 UNLESS OTHERWISE SET BY THE USER INTO /OPTION/.           PDECOL  609
C                                                                          PDECOL  610
C THE COMMON BLOCK /OPTION/ CONTAINS THE VARIABLES NOGAUS AND MAXDER.      PDECOL  611
C NOGAUS IS SET .EQ. 0 IN THE BLOCK DATA.  IT CAN BE CHANGED TO BE         PDECOL  612
C SET .EQ. 1 IF THE GAUSS-LEGENDRE COLLOCATION POINTS ARE NOT              PDECOL  613
C DESIRED WHEN NCC = 2 (SEE ABOVE AND COLPNT).  MAXDER IS SET              PDECOL  614
C .EQ. 5 IN THE BLOCK DATA AND ITS VALUE REPRESENTS THE                    PDECOL  615
C MAXIMUM ORDER OF TIME INTEGRATION FORMULA ALLOWED.  ITS VALUE            PDECOL  616
C AFFECTS THE STORAGE REQUIRED IN WORK AND MAY BE CHANGED IF               PDECOL  617
C DESIRED.  SEE COSET FOR RESTRICTIONS.  THESE CHANGES MAY BE MADE BY      PDECOL  618
C THE USER BY ACCESSING /OPTION/ IN HIS CALLING PROGRAM (BEFORE THE        PDECOL  619
C FIRST CALL TO PDECOL) OR BY CHANGING THE DATA STATEMENT IN               PDECOL  620
C THE BLOCK DATA.                                                          PDECOL  621
C                                                                          PDECOL  622
C--------------------------------------------------------------------------PDECOL  623
C                                                                          PDECOL  624
C                                                                          PDECOL  625
C COMMUNICATION                                                            PDECOL  626
C                                                                          PDECOL  627
C EACH SUBROUTINE IN THE PACKAGE CONTAINS A COMMUNICATION SUMMARY          PDECOL  628
```

```
C AS INDICATED BELOW.                                                    PDECOL  629
C                                                                        PDECOL  630
C PACKAGE ROUTINES CALLED..   EVAL,INITAL,INTERP,STIFIB                  PDECOL  631
C USER ROUTINES CALLED..      BNDRY                                      PDECOL  632
C CALLED BY..                 USERS MAIN PROGRAM                         PDECOL  633
C FORTRAN FUNCTIONS USED..    ABS,AMAX1,FLOAT,SQRT                       PDECOL  634
C--------------------------------------------------------------------    PDECOL  635
      DIMENSION WORK(1),IWORK(1),XHKPT(1)                                PDECOL  636
      COMMON /GEAR0/ DTUSED,NQUSED,NSTEP,NFE,NJE                         PDECOL  637
      COMMON /GEAR1/ T,DTC,DTMN,DTMX,EPSC,UROUND,N,MFC,KFLAG,JSTART      PDECOL  638
      COMMON/GEAR9/EPSJ,RO,ML,MU,MW,NML,NML,NJUW,NOWI                    PDECOL  639
      COMMON /OPTION/ NOGAUS,MAXDER                                      PDECOL  640
      COMMON /SIZES/ NIG,KOR,NC,NPD,NCPTS,NEQN,IQUAD                     PDECOL  641
      COMMON /ISTART/ IW1,IW2,IW3,IW4,IW5,IW6,IW7,IW8,IW9,IW10,          PDECOL  642
     *                IW11,IW12,IW13,IW14,IW15,IW16,IW17,IW18            PDECOL  643
      COMMON /IOUNIT/ LOUT                                               PDECOL  644
      IF (INDEX .EQ. 0) GO TO 60                                         PDECOL  645
      IF (INDEX .EQ. 2) GO TO 70                                         PDECOL  646
      IF (INDEX .EQ. 4) GO TO 80                                         PDECOL  647
      IF (INDEX .EQ. 3) GO TO 90                                         PDECOL  648
C--------------------------------------------------------------------    PDECOL  649
C SEVERAL CHECKS ARE MADE HERE TO DETERMINE IF THE INPUT PARAMETERS      PDECOL  650
C HAVE LEGAL VALUES.  ERROR CHECKS ARE MADE ON INDEX, EPS, (T0-TOUT)*DT, PDECOL  651
C DT, NINT, KORD, NCC, NPDE, MF, WHETHER THE BREAKPOINT SEQUENCE IS      PDECOL  652
C STRICTLY INCREASING, AND WHETHER THERE IS SUFFICIENT STORAGE           PDECOL  653
C PROVIDED FOR WORK AND IWORK.  PROBLEM DEPENDENT PARAMETERS ARE         PDECOL  654
C CALCULATED AND PLACED IN COMMON.                                       PDECOL  655
C--------------------------------------------------------------------    PDECOL  656
      IERID = -6                                                         PDECOL  657
      IF (INDEX .NE. 1) GO TO 320                                        PDECOL  658
      IERID = IERID - 1                                                  PDECOL  659
      IF (EPS .LE. 0.) GO TO 320                                         PDECOL  660
      IERID = IERID - 1                                                  PDECOL  661
      IF ((T0-TOUT)*DT .GT. 0.) GO TO 320                                PDECOL  662
      IERID = IERID - 1                                                  PDECOL  663
      IF (DT .EQ. 0.0) GO TO 320                                         PDECOL  664
      IERID = IERID - 1                                                  PDECOL  665
      NIN = NINT                                                         PDECOL  666
      IF (NIN .LT. 1) GO TO 320                                          PDECOL  667
      IERID = IERID - 1                                                  PDECOL  668
      KOR = KORD                                                         PDECOL  669
      IF (KOR .LT. 3  .OR.  KOR .GT. 20) GO TO 320                       PDECOL  670
      IERID = IERID - 1                                                  PDECOL  671
      NC = NCC                                                           PDECOL  672
      IF (NCC .LT. 2  .OR.  NCC .GE. KOR) GO TO 320                      PDECOL  673
      IERID = IERID - 1                                                  PDECOL  674
      NPD = NPDE                                                         PDECOL  675
      NPDE2 = NPD*NPD                                                    PDECOL  676
      IF (NPDE .LT. 1) GO TO 320                                         PDECOL  677
      IERID = IERID - 1                                                  PDECOL  678
      IF (MF.NE.22.AND.MF.NE.21.AND.MF.NE.12.AND.MF.NE.11) GO TO 320     PDECOL  679
      IERID = IERID - 1                                                  PDECOL  680
      DO 10 K=1,NIN                                                      PDECOL  681
        IF (XHKPT(K) .GE. XHKPT(K+1)) GO TO 320                          PDECOL  682
   10 CONTINUE                                                           PDECOL  683
      NCPTS = KOR + (NIN - 1) * (KOR - NCC)                              PDECOL  684
      NEQN = NPDE * NCPTS                                                PDECOL  685
```

```
      ML=KORD-2                                                          PDECOL   686
      MU = ML                                                            PDECOL   687
      MW = ML + ML + 1                                                   PDECOL   688
      NDW=NEQN*(3*ML+1)                                                  PDECOL   689
      NDW1=NCPTS*(3*ML+1)                                                PDECOL   690
      IWSAVE = IWORK(1)                                                  PDECOL   691
      IISAVE = IWORK(2)                                                  PDECOL   692
      IW1=4*NPDE+1                                                       PDECOL   693
      IW2 = IW1 + 3*KORD*NCPTS                                           PDECOL   694
      IW3 = IW2 + NCPTS + KORD                                           PDECOL   695
      IW4 = IW3 + NCPTS                                                  PDECOL   696
      IW5 = IW4 + NEQN                                                   PDECOL   697
      IW6 = IW5 + NEQN                                                   PDECOL   698
      IW7 = IW6 + NEQN                                                   PDECOL   699
      IW8 = IW7 + NEQN                                                   PDECOL   700
      IW9 = IW8 + NEQN                                                   PDECOL   701
      IW10 = IW9 + 3*NPDE                                                PDECOL   702
      IW11 = IW10 + NEQN*(MAXDER+1)                                      PDECOL   703
      IW12=IW11+NPDE                                                     PDECOL   704
      IW13=IW12+NPDE                                                     PDECOL   705
      IW14=IW13+NPDE                                                     PDECOL   706
      IW15=IW14+NPDE                                                     PDECOL   707
      IW16=IW15+NPDE                                                     PDECOL   708
      IW17 = IW16 + NPDE                                                 PDECOL   709
      IW18 = NCPTS + 1                                                   PDECOL   710
      IERID = IERID - 1                                                  PDECOL   711
      IWSTOR = IW17 + NEQN*(3*ML+1) - 1                                  PDECOL   712
      IISTOR = IW18 + NEQN - 1                                           PDECOL   713
      IF ( IWSAVE .LT. IWSTOR  .OR.  IISAVE .LT. IISTOR ) GO TO 335      PDECOL   714
C-----------------------------------------------------------------------PDECOL   715
C PERFORM INITIALIZATION TASKS.  IF KORD .EQ. 3 THEN CALCULATE THE BAND-PDECOL   716
C WIDTH OF THE ASSOCIATED MATRIX PROBLEM BY DETERMINING THE TYPE OF    PDECOL   717
C BOUNDARY CONDITIONS. THEN CHECK FOR SUFFICIENT STORAGE AGAIN.        PDECOL   718
C-----------------------------------------------------------------------PDECOL   719
      CALL INITAL(KORD,WORK(IW1),WORK(IW6),XBKPT,WORK(IW2),WORK(IW3),  PDECOL   720
     *            WORK(IW17),IWORK(IW18),IWORK)                        PDECOL   721
      IF(IQUAD .NE. 0) GO TO 280                                         PDECOL   722
      IWSTOR = IW17 + NEQN*(3*ML+1) - 1                                  PDECOL   723
      IF ( IWSAVE .LT. IWSTOR ) GO TO 335                                PDECOL   724
C-----------------------------------------------------------------------PDECOL   725
C IF INITIAL VALUES OF CMAX OTHER THAN THOSE SET BELOW ARE DESIRED,    PDECOL   726
C THEY SHOULD BE SET HERE.  ALL CMAX(I) MUST BE POSITIVE.              PDECOL   727
C HAVING PROPER VALUES OF CMAX FOR THE PROBLEM BEING SOLVED IS AS      PDECOL   728
C IMPORTANT AS CHOOSING EPS (SEE ABOVE), SINCE ERRORS ARE              PDECOL   729
C MEASURED RELATIVE TO CMAX.  IF VALUES FOR DTMIN OR DTMAX, THE        PDECOL   730
C BOUNDS ON ABS(DT), OTHER THAN THOSE BELOW ARE DESIRED, THEY          PDECOL   731
C SHOULD BE SET BELOW.                                                 PDECOL   732
C-----------------------------------------------------------------------PDECOL   733
CC  SREC MODIFIES THE ERROR CONTROL.                                   PDECOL   734
CC  AN ABSOLUTE ERROR CRITERION IS USED FOR QUANTITIES LESS THAN SREC. PDECOL   735
      DO 50 I = 1,NEQN                                                   PDECOL   736
        II = I - 1                                                       PDECOL   737
      WORK(IW4+I)=AMAX1(ABS(WORK(IW5+I)),SREC)                           PDECOL   738
   50   WORK(IW10+I) = WORK(IW6+I)                                       PDECOL   739
      N = NEQN                                                           PDECOL   740
      T = TO                                                             PDECOL   741
      DTC = DT                                                           PDECOL   742
```

```
                  DTMN = ABS(DT)                                                PDECOL   743
                  DTUSED = 0.                                                   PDECOL   744
                  EPSC = EPS                                                    PDECOL   745
745               MFC = MF                                                      PDECOL   746
                  JSTART = 0                                                    PDECOL   747
                  EPSJ =  SQRT(UROUND)                                          PDECOL   748
                  NM1 = NEQN - 1                                                PDECOL   749
                  NOML = NEQN*ML                                                PDECOL   750
750               NMCUT = 0                                                     PDECOL   751
                  KFLAG = 0                                                     PDECOL   752
                  TOUTP = T0                                                    PDECOL   753
                  IF ( T0 .EQ. TOUT ) GO TO 360                                 PDECOL   754
               60 DTMX = ABS(TOUT-TOUTP)*10.                                    PDECOL   755
755               GO TO 140                                                     PDECOL   756
            C                                                                   PDECOL   757
               70 DTMX = ABS(TOUT-TOUTP)*10.                                    PDECOL   758
                  IF ((T-TOUT)*DTC .GE. 0.) GO TO 340                           PDECOL   759
                  GO TO 150                                                     PDECOL   760
760         C                                                                   PDECOL   761
               80 IF ((T-TOUT)*DTC .GE. 0.) GO TO 300                           PDECOL   762
                  JSTART = -1                                                   PDECOL   763
                  EPSC = EPS                                                    PDECOL   764
                  MFC = MF                                                      PDECOL   765
765               GO TO 100                                                     PDECOL   766
            C                                                                   PDECOL   767
               90 DTMX = DT                                                     PDECOL   768
              100 IF ((T+DTC) .EQ. T) WRITE(LOUT,110)                           PDECOL   769
              110 FORMAT(36H WARNING..  T + DT = T ON NEXT STEP.)                PDECOL   770
770         C------------------------------------------------------------------ PDECOL   771
            C TAKE A TIME STEP BY CALLING THE INTEGRATOR.                       PDECOL   772
            C------------------------------------------------------------------ PDECOL   773
                  CALL STIFIB(NEQN,TOUT,WORK(IWI0),WORK(Iw4),WORK(IW5),WORK(IW6),PDECOL   774
                 *       WORK(IW7),WORK(IW8),WORK(Iw17),WORK(Iw18),WORK,IWORK)  PDECOL   775
775         C                                                                   PDECOL   776
                  KGO = 1 - KFLAG                                               PDECOL   777
                  GO TO (120, 160, 220, 260, 280), KGO                          PDECOL   778
            C KFLAG =   0,  -1,  -2,  -3   -4                                   PDECOL   779
            C                                                                   PDECOL   780
780           120 CONTINUE                                                      PDECOL   781
            C------------------------------------------------------------------ PDECOL   782
            C NORMAL RETURN FROM INTEGRATOR.                                    PDECOL   783
            C                                                                   PDECOL   784
            C THE WEIGHTS CMAX(I) ARE UPDATED.  IF DIFFERENT VALUES ARE DESIRED,PDECOL   785
785         C THEY SHOULD BE SET HERE.  A TEST IS MADE FOR EPS BEING TOO SMALL  PDECOL   786
            C FOR THE MACHINE PRECISION.                                        PDECOL   787
            C                                                                   PDECOL   788
            C ANY OTHER TESTS OR CALCULATIONS THAT ARE REQUIRED AFTER EVERY     PDECOL   789
            C STEP SHOULD BE INSERTED HERE.                                     PDECOL   790
790         C                                                                   PDECOL   791
            C IF INDEX = 3, SAVE1 IS SET TO THE CURRENT C VALUES ON RETURN.     PDECOL   792
            C IF INDEX = 2, DT IS CONTROLLED TO HIT TOUT (WITHIN ROUNDOFF       PDECOL   793
            C ERROR), AND THEN THE CURRENT C VALUES ARE PUT IN SAVE1 ON RETURN. PDECOL   794
            C FOR ANY OTHER VALUE OF INDEX, CONTROL RETURNS TO THE INTEGRATOR   PDECOL   795
795         C UNLESS TOUT HAS BEEN REACHED.  THEN INTERPOLATED VALUES OF C ARE  PDECOL   796
            C COMPUTED AND STORED IN SAVE1 ON RETURN.                           PDECOL   797
            C IF INTERPOLATION IS NOT DESIRED, THE CALL TO INTERP SHOULD BE     PDECOL   798
            C REMOVED AND CONTROL TRANSFERRED TO STATEMENT 340 INSTEAD OF 360.  PDECOL   799
```

```
                C-----------------------------------------------------------------------  PDECOL  A00
      A00             D = 0.                                                              PDECOL  A01
                      DO 130 I = 1,NEQN                                                   PDECOL  A02
                        II = I - 1                                                        PDECOL  A03
                        AYI = ABS(WORK(IW10+II))                                          PDECOL  A04
                      WORK(IW4+I))=AMAX1(SREC*AYI)                                        PDECOL  A05
      A05       130   D = D + (AYI/WORK(IW4+II))**2                                       PDECOL  A06
                      D = D*(UROUND/EPS)**2                                               PDECOL  A07
                      IF (D .GT.  FLOAT(NEQN)) GO TO 240                                  PDECOL  A08
                      IF (INDEX .EQ. 3) GO TO 340                                         PDECOL  A09
                      IF (INDEX .EQ. 2) GO TO 150                                         PDECOL  A10
      A10       140 IF ((T-TOUT)*DTC .LT. 0.) GO TO 100                                 PDECOL  A11
                      CALL INTERP(TOUT,WORK(IW10),NEQ,,WORK(IW6))                         PDECOL  A12
                      GO TO 360                                                           PDECOL  A13
                C                                                                         PDECOL  A14
                  150 IF (((T+DTC)-TOUT)*DTC .LE. 0.) GO TO 100                         PDECOL  A15
      A15             IF (ABS(T-TOUT) .LE. 100.*UROUND*ABS(DTMX)) GO TO 340             PDECOL  A16
                      IF ((T-TOUT)*DTC .GE. 0.) GO TO 340                               PDECOL  A17
                      DTC = (TOUT - T)*(1. - 4.*UROUND)                                 PDECOL  A18
                      JSTART = -1                                                       PDECOL  A19
                      GO TO 100                                                         PDECOL  A20
      A20             C-----------------------------------------------------------------------  PDECOL  A21
                C ON AN ERROR RETURN FROM INTEGRATOR, AN IMMEDIATE RETURN OCCURS IF   PDECOL  A22
                C KFLAG = -2 OR -4, AND RECOVERY ATTEMPTS ARE MADE OTHERWISE.         PDECOL  A23
                C TO RECOVER, DT AND DTMN ARE REDUCED BY A FACTOR OF .1 UP TO 10      PDECOL  A24
                C TIMES BEFORE GIVING UP.                                             PDECOL  A25
      A25       C-----------------------------------------------------------------------  PDECOL  A26
                  160 WRITE (LOUT,170) T                                              PDECOL  A27
                  170 FORMAT(//35H KFLAG = -1 FROM INTEGRATOR AT T = ,E16.8/           PDECOL  A28
                     *   40H  ERROR TEST FAILED WITH ABS(DT) = DTMIN/)                  PDECOL  A29
                  180 IF (NHCUT .EQ. 10) GO TO 200                                    PDECOL  A30
      A30             NHCUT = NHCUT + 1                                               PDECOL  A31
                      DTMN = .1*DTMN                                                  PDECOL  A32
                      DTC = .1*DTC                                                    PDECOL  A33
                      WRITE (LOUT,190) DTC                                            PDECOL  A34
                  190 FORMAT(25H  DT HAS BEEN REDUCED TO ,E16.8,                      PDECOL  A35
      A35            *   26H  AND STEP WILL BE RETRIED//)                             PDECOL  A36
                      JSTART = -1                                                     PDECOL  A37
                      GO TO 100                                                       PDECOL  A38
                C                                                                     PDECOL  A39
                  200 WRITE (LOUT,210)                                                PDECOL  A40
      A40         210 FORMAT(//44H PROBLEM APPEARS UNSOLVABLE WITH GIVEN INPUT//)     PDECOL  A41
                      GO TO 340                                                       PDECOL  A42
                C                                                                     PDECOL  A43
                  220 WRITE (LOUT,230) T,DTC                                          PDECOL  A44
                  230 FORMAT(//35H KFLAG = -2 FROM INTEGRATOR AT T = ,E16.8,6H  DT =, PDECOL  A45
      A45            *  E16.8/52H  THE REQUESTED ERROR IS SMALLER THAN CAN BE HANDLED//) PDECOL  A46
                      GO TO 340                                                       PDECOL  A47
                C                                                                     PDECOL  A48
                  240 WRITE (LOUT,250) T                                              PDECOL  A49
                  250 FORMAT(//37H INTEGRATION HALTED BY DRIVER AT T = ,E16.8/         PDECOL  A50
      A50            *   56H  EPS TOO SMALL TO BE ATTAINED FOR THE MACHINE PRECISION/) PDECOL  A51
                      KFLAG = -2                                                      PDECOL  A52
                      GO TO 340                                                       PDECOL  A53
                C                                                                     PDECOL  A54
                  260 WRITE (LOUT,270) T                                              PDECOL  A55
      A55         270 FORMAT(//35H KFLAG = -3 FROM INTEGRATOR AT T = ,E16.8/           PDECOL  A56
```

```
                     *   45H  CORRECTOR CONVERGENCE COULD NOT BE ACHIEVED//)                PDECOL   857
                      GO TO 180                                                            PDECOL   858
                C                                                                          PDECOL   859
                  280 WRITE (LOUT,290)                                                     PDECOL   860
        860       290 FORMAT(//28H SINGULAR MATRIX ENCOUNTERED,                            PDECOL   861
                     *         35H PROBABLY DUE TO STORAGE OVERWRITES//)                   PDECOL   862
                      KFLAG = -4                                                           PDECOL   863
                      GO TO 340                                                            PDECOL   864
                C                                                                          PDECOL   865
                  300 WRITE(LOUT,310) T,TOUT,DTC                                           PDECOL   866
        865       310 FORMAT(//45H INDEX = -1 ON INPUT WITH (T-TOUT)*DT .GE. 0./           PDECOL   867
                     *   4H T =,E16.8,9H   TOUT =,E16.8,8H   DTC =,E16.8/                  PDECOL   868
                     *   44H INTERPOLATION WAS DONE AS ON NORMAL RETURN./                  PDECOL   869
                     *   41H DESIRED PARAMETER CHANGES WERE NOT MADE.)                     PDECOL   870
                      CALL INTERP(TOUT,WORK(IW10),NEQN,WORK(IW6))                          PDECOL   871
        870           INDEX = -5                                                           PDECOL   872
                      RETURN                                                               PDECOL   873
                C                                                                          PDECOL   874
                  320 WRITE(LOUT,330) IERID                                                PDECOL   875
                  330 FORMAT(//24H ILLEGAL INPUT...INDEX= ,I3//)                           PDECOL   876
        875           INDEX = IERID                                                        PDECOL   877
                      RETURN                                                               PDECOL   878
                C                                                                          PDECOL   879
                  335 WRITE(LOUT,336) IWSTOR,IWSAVE,IISTOR,IISAVE                          PDECOL   880
                  336 FORMAT(//21H INSUFFICIENT STORAGE/24H  WORK MUST BE OF LENGTH,       PDECOL   881
                     *  I10,5X,12HYOU PROVIDED,I10/24H IWORK MUST BE OF LENGTH,I10,5X,     PDECOL   882
                     *  12HYOU PROVIDED,I10//)                                            PDECOL   883
                      INDEX = IERID                                                        PDECOL   884
                      RETURN                                                               PDECOL   885
        885       C                                                                        PDECOL   886
                  340 TOUT = T                                                             PDECOL   887
                      DO 350 I = 1,NEQN                                                    PDECOL   888
                        II = I - 1                                                         PDECOL   889
                  350   WORK(IW6+II) = WORK(IW10+II)                                      PDECOL   890
        890       360 INDEX = KFLAG                                                        PDECOL   891
                      TOUTP = TOUT                                                         PDECOL   892
                      DT = DTUSED                                                          PDECOL   893
                      IF (KFLAG .NE. 0) DT = DTC                                           PDECOL   894
                      RETURN                                                               PDECOL   895
        895           END                                                                  PDECOL   896
```

CARD NR. SEVERITY DETAILS DIAGNOSIS OF PROBLEM

691 I IWORK ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.

```
      SUBROUTINE VALUES(X,USOL,SCTCH,NDIM1,NDIM2,NPTS,NDERV,WORK)
C-----------------------------------------------------------------------
C SUBROUTINE VALUES COMPUTES THE SOLUTION U AND THE FIRST NDERV
C DERIVATIVES OF U AT THE NPTS POINTS X AND AT TIME TOUT AND RETURNS
C THEM IN THE ARRAY USOL.  THIS ROUTINE MUST BE USED TO OBTAIN
C SOLUTION VALUES SINCE PDECOL DOES NOT RETURN ANY SOLUTION VALUES
C TO THE USER.  SEE PDECOL.
C
C THE CALLING PARAMETERS ARE...
C   X      =  AN ARBITRARY VECTOR OF SPATIAL POINTS OF LENGTH NPTS AT
C             WHICH THE SOLUTION AND THE FIRST NDERV DERIVATIVE VALUES
C             ARE TO BE CALCULATED.  IF X .LT. XLEFT ( X .GT. XRIGHT )
C             THEN THE PIECEWISE POLYNOMIAL OVER THE LEFTMOST ( RIGHT-
C             MOST ) INTERVAL IS EVALUATED TO CALCULATE THE SOLUTION
C             VALUES AT THIS UNUSUAL VALUE OF  X.  SEE PDECOL.
C
C   USOL   =  AN ARRAY WHICH CONTAINS THE SOLUTION AND THE FIRST
C             NDERV DERIVATIVES OF THE SOLUTION AT ALL THE POINTS IN
C             THE INPUT VECTOR X.  IN PARTICULAR, USOL(J,I,K) CONTAINS
C             THE VALUE OF THE (K-1)-ST DERIVATIVE OF THE J-TH PDE
C             COMPONENT AT THE I-TH POINT OF THE X VECTOR FOR
C             J = 1 TO NPDE, I = 1 TO NPTS, AND K = 1 TO NDERV+1.
C
C   SCTCH  =  A USER SUPPLIED WORKING STORAGE ARRAY OF LENGTH AT LEAST
C             KORD*(NDERV+1).  SEE BELOW AND PDECOL FOR DEFINITIONS OF
C             THESE PARAMETERS.
C
C   NDIM1  =  THE FIRST DIMENSION OF THE OUTPUT ARRAY USOL IN THE CALLING
C             PROGRAM.  NDIM1 MUST BE .GE. NPDE.
C
C   NDIM2  =  THE SECOND DIMENSION OF THE OUTPUT ARRAY USOL IN THE
C             CALLING PROGRAM.  NDIM2 MUST BE .GE. NPTS.
C
C   NPTS   =  THE NUMBER OF POINTS IN THE X VECTOR.
C
C   NDERV  =  THE NUMBER OF DERIVATIVE VALUES OF THE SOLUTION THAT ARE
C             TO BE CALCULATED.  NDERV SHOULD BE LESS THAN KORD SINCE
C             THE KORD-TH DERIVATIVE OF A POLYNOMIAL OF DEGREE KORD-1
C             IS EQUAL TO ZERO.  SEE PDECOL.
C
C   WORK   =  THE USERS WORKING STORAGE ARRAY WHICH IS USED IN THIS CASE
C             TO PROVIDE THE CURRENT BASIS FUNCTION COEFFICIENTS AND THE
C             PIECEWISE POLYNOMIAL BREAKPOINT SEQUENCE.
C
C PACKAGE ROUTINES CALLED..  HSPLVD,INTERV
C USER ROUTINES CALLED..     NONE
C CALLED BY..                USERS MAIN PROGRAM
C FORTRAN FUNCTIONS USED..   NONE
C
C-----------------------------------------------------------------------
      DIMENSION USOL(NDIM1,NDIM2,NDERV),X(NPTS),SCTCH(1),WORK(1)
      COMMON /SIZES/ NINT,KORD,NCC,NPDE,NCPTS,NEQN,IQUAD
      COMMON /ISTART/ IW1,IW2,IW3,IW4,IW5,IW6,IDUM(12)
      DATA ILEFT/0/, MFLAG/0/
      NDERV1 = NDERV + 1
      DO 20 IPTS=1,NPTS
        CALL INTERV(WORK(IW2),NCPTS,X(IPTS),ILEFT,MFLAG)
```

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

```
      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) DEFINING 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 DECH-SOLH.      INITAL    23
C ILEFT  = POINTERS TO BREAKPOINT SEQUENCE GENERATED BY INITAL.         INITAL    24
C                                                                       INITAL    25
C PACKAGE ROUTINES CALLED..  BSPLVD,COLPNT,DECH,INTERV,SOLH             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,RO,ML,MU,IDUM(3),NOW,NOWI                       INITAL    33
CC  FOR BURNER STABILIZED FLAMES,                                       INITAL    34
CC  THE MASS FLUX FRACTION THROUGH THE LEFT BOUNDARY IS SET EQUAL       INITAL    35
CC  TO THE INITIAL COLD BOUNDARY MASS FRACTIONS.                        INITAL    36
      COMMON/TAHCT/RL,CPMX,HO(20),H2O(20),H2OM(20)                      INITAL    37
      COMMON/TAHFM/FM                                                   INITAL    38
      COMMON/START/NSTART,NBURN,NTRAN,KRD                               INITAL    39
      MFLAG = -2                                                        INITAL    40
      IER = 0                                                           INITAL    41
C-----------------------------------------------------------------------INITAL    42
C SET UP THE PIECEWISE POLYNOMIAL SPACE BREAKPOINT SEQUENCE.            INITAL    43
C-----------------------------------------------------------------------INITAL    44
      KRPT = KORD - NCC                                                 INITAL    45
      DO 10 I=1,KORD                                                    INITAL    46
        XT(NCPTS+I) = X(NINT+1)                                         INITAL    47
   10   XT(I) = X(1)                                                    INITAL    48
      DO 20 I=2,NINT                                                    INITAL    49
        II = (I-2)*KRPT + KORD                                          INITAL    50
        DO 20 J=1,KRPT                                                  INITAL    51
   20     XT(II+J) = X(I)                                              INITAL    52
C-----------------------------------------------------------------------INITAL    53
C SET UP COLLOCATION POINTS ARRAY XC.                                   INITAL    54
C-----------------------------------------------------------------------INITAL    55
      CALL COLPNT(X, XC, XT)                                            INITAL    56
C-----------------------------------------------------------------------INITAL    57
C GENERATE THE ILEFT ARRAY.  STORE THE BASIS FUNCTION VALUES IN THE     INITAL    58
```

```
C ARRAY A.  THE ARRAY A IS DIMENSIONED A(KORD,3,NCPTS) AND A(K,J,I)      INITAL  59
C CONTAINS THE VALUE OF THE (J-1)-ST DERIVATIVE (J = 1,2,3) OF THE K-TH  INITAL  60
C NONZERO BASIS FUNCTION (K = 1, ... ,KORD) AT THE I-TH COLLOCATION      INITAL  61
C POINT (I = 1, ... ,NCPTS).  SET UP RHS FOR INTERPOLATING THE INITIAL   INITAL  62
C CONDITIONS AT THE COLLOCATION POINTS.  SET THE INTERPOLATION MATRIX     INITAL  63
C INTO THE BANDED MATRIX PW.                                             INITAL  64
C-----------------------------------------------------------------------  INITAL  65
      DO 35 I=1,NCPTS                                                    INITAL  66
        CALL INTERV(XT,NCPTS,XC(I),ILEFT(I),MFLAG)                       INITAL  67
        CALL BSPLVD(XT,KORD,XC(I),ILEFT(I),A(1,1,I),3)                   INITAL  68
35    CONTINUE                                                           INITAL  69
CC  IF NSTART = 1,                                                       INITAL  70
CC  THE STARTING VALUES FOR THE C'S ARE JUST THE INCOMING Y VALUES,      INITAL  71
CC  RATHER THAN SOLVING THE SYSTEM OF EQUATIONS,                         INITAL  72
CC  THIS PREVENTS THE SYSTEM FROM OVERWORKING TRYING TO GET AROUND       INITAL  73
CC  SHARP CORNERS OR GENERATING NEGATIVE VALUES,                         INITAL  74
      NM=NCPTS-1                                                         INITAL  75
      DO 50 KPDE=1,NPDE                                                  INITAL  76
      DO 40 I=1,NCPTS                                                    INITAL  77
      IP=I+(KPDE-1)*NCPTS                                                INITAL  78
40    CALL UINIT(XC(I),RHS(IP),NPDE,KPDE)                                INITAL  79
      IF(NSTART.EQ.1)GO TO 50                                            INITAL  80
C  IF NHURN = 1, WE NEED THE CORRECT VALUE FOR R2U AT THE LEFT           INITAL  81
C  BOUNDARY TO GET THE PROPER MASS FLUX FRACTION,                        INITAL  82
      IF(NTRAN.EQ.1)GO TO 230                                            INITAL  83
      IF(KPDE.GT.1)GO TO 230                                             INITAL  84
      READ(1,220)(R2U(L),L=1,NPDE)                                       INITAL  85
220   FORMAT(1P8E14.6)                                                   INITAL  86
      WRITE(3,225)(R2U(L),L=1,NPDE)                                      INITAL  87
225   FORMAT(/2X,5HR2U =,1P6E14.6/)                                      INITAL  88
230   CONTINUE                                                           INITAL  89
      DO 30 I=1,NOW1                                                     INITAL  90
30    PW(I)=0.0                                                          INITAL  91
      DO 38 I=1,NM                                                       INITAL  92
      II=I-1                                                             INITAL  93
      ICOL=ILEFT(I)-I-1                                                  INITAL  94
      JL=MAX0(1,I+2-NCPTS)                                               INITAL  95
      JU=MIN0(KORD,KORD+I-2)                                             INITAL  96
      DO 38 J=JL,JU                                                      INITAL  97
      J1=II+NCPTS*(ICOL+J-1)                                             INITAL  98
      PW(1,J1)=A(J,1,I)                                                  INITAL  99
      IF(NHURN.EQ.0)GO TO 38                                             INITAL 100
      IF(I.EQ.1.AND.KPDE.LT.NPDE)PW(1,J1)=                               INITAL 101
     *     A(J,1,I)-R2U(KPDE)*A(J,2,I)/FM                                INITAL 102
38    CONTINUE                                                           INITAL 103
CC  WE WANT THE SPACE DERIVATIVE AT THE RIGHT TO BE ZERO                 INITAL 104
      ICOL=ILEFT(NCPTS)-NCPTS-1                                          INITAL 105
      DO 39 J=2,KORD                                                     INITAL 106
      J1=NM+NCPTS*(ICOL+J-1)                                             INITAL 107
39    PW(1,J1)=A(J,2,I)                                                  INITAL 108
      IP=NCPTS*KPDE                                                      INITAL 109
      RHS(IP)=0.0                                                        INITAL 110
      CALL DECH(NCPTS,NCPTS,ML,MU,PW,IPIV,IER)                           INITAL 111
      IF(IER.NE.0)RETURN                                                 INITAL 112
      IST=(KPDE-1)*NCPTS+1                                               INITAL 113
      CALL SOLH(NCPTS,NCPTS,ML,MU,PW,RHS(IST),IPIV)                      INITAL 114
50    CONTINUE                                                           INITAL 115
```

```
115                RETURN                                              INITAL    116
                   END                                                 INITAL    117
```

```
      SUBROUTINE COLPNT(X, XC, XT)                                      COLPNT     2
C-----------------------------------------------------------------------COLPNT     3
C COLPNT IS CALLED ONLY ONCE BY INITAL TO DEFINE THE REQUIRED COLLOCA-   COLPNT     4
C TION POINTS WHICH ARE TO BE USED WITH THE USER SELECTED PIECEWISE      COLPNT     5
C POLYNOMIAL SPACE.  THE COLLOCATION POINTS ARE CHOSEN SUCH THAT THEY    COLPNT     6
C ARE EITHER THE POINTS AT WHICH THE PIECEWISE POLYNOMIAL SPACE BASIS    COLPNT     7
C FUNCTIONS ATTAIN THEIR UNIQUE MAXIMUM VALUES, OR, THE GAUSS-LEGENDRE  COLPNT     8
C QUADRATURE POINTS WITHIN EACH PIECEWISE POLYNOMIAL SPACE SUBINTERVAL.  COLPNT     9
C DEPENDING UPON THE SPACE BEING USED AND THE DESIRE OF THE USER.        COLPNT    10
C                                                                       COLPNT    11
C X  = USER DEFINED PIECEWISE POLYNOMIAL BREAKPOINTS.                   COLPNT    12
C XC = COLLOCATION POINTS DEFINED BY COLPNT.                            COLPNT    13
C XT = PIECEWISE POLYNOMIAL BREAKPOINT SEQUENCE.                        COLPNT    14
C                                                                       COLPNT    15
C PACKAGE ROUTINES CALLED..  BSPLVD,INTERV                              COLPNT    16
C USER ROUTINES CALLED..     NONE                                       COLPNT    17
C CALLED BY..                INITAL                                     COLPNT    18
C FORTRAN FUNCTIONS USED..   NONE                                       COLPNT    19
C-----------------------------------------------------------------------COLPNT    20
      DIMENSION RHO(40),X(1),XC(1),AT(1)                                COLPNT    21
      COMMON /SIZES/ NINT,KORD,NCC,NPOF,NCPTS,NEQN,IQUAD                COLPNT    22
      COMMON /OPTION/ NOGAUS,MAXDER                                     COLPNT    23
      DATA ILEFT/0/                                                     COLPNT    24
C-----------------------------------------------------------------------COLPNT    25
C IF THE VARIABLE NOGAUS IN THE COMMON BLOCK /OPTION/ IS SET .EQ. 1.    COLPNT    26
C THE USE OF THE GAUSS-LEGENDRE POINTS IS PROHIBITED FOR ALL CASES.     COLPNT    27
C NOGAUS IS CURRENTLY SET .EQ. 0 BY A DATA STATEMENT IN THE BLOCK DATA. COLPNT    28
C THE USER MAY CHANGE THIS AS DESIRED.                                  COLPNT    29
C-----------------------------------------------------------------------COLPNT    30
      IF ( NCC .NE. 2  .OR. NOGAUS .EQ. 1 ) GO TO 200                   COLPNT    31
C-----------------------------------------------------------------------COLPNT    32
C COMPUTE THE COLLOCATION POINTS TO BE AT THE GAUSS-LEGENDRE POINTS IN   COLPNT    33
C EACH PIECEWISE POLYNOMIAL SPACE SUBINTERVAL.  THE ARRAY RHO IS SET TO COLPNT    34
C CONTAIN THE GAUSS-LEGENDRE POINTS FOR THE STANDARD INTERVAL (-1,1).    COLPNT    35
C-----------------------------------------------------------------------COLPNT    36
      IPTS = KORD - 2                                                   COLPNT    37
      GO TO (10,20,30,40,50,60,70,80,90,100,110,120,130,140,150,160,170,COLPNT    38
     *    180),IPTS                                                    COLPNT    39
   10 RHO(1) = 0.                                                       COLPNT    40
      GO TO 190                                                         COLPNT    41
   20 RHO(2) = .577350269189626E-00                                     COLPNT    42
      RHO(1) = - RHO(2)                                                 COLPNT    43
      GO TO 190                                                         COLPNT    44
   30 RHO(3) = .774596669241483E-00                                     COLPNT    45
      RHO(1) = - RHO(3)                                                 COLPNT    46
      RHO(2) = 0.                                                       COLPNT    47
      GO TO 190                                                         COLPNT    48
   40 RHO(3) = .339981043584856E-00                                     COLPNT    49
      RHO(2) = - RHO(3)                                                 COLPNT    50
      RHO(4) = .861136311594053E-00                                     COLPNT    51
      RHO(1) = - RHO(4)                                                 COLPNT    52
      GO TO 190                                                         COLPNT    53
   50 RHO(4) = .538469310105683E-00                                     COLPNT    54
      RHO(2) = - RHO(4)                                                 COLPNT    55
      RHO(5) = .906179845938664E-00                                     COLPNT    56
      RHO(1) = - RHO(5)                                                 COLPNT    57
      RHO(3) = 0.                                                       COLPNT    58
```

```
                      GO TO 190                                                  COLPNT  59
                   60 RHO(4) = .2386191860831976-00                              COLPNT  60
   60                 RHO(3) = - RHO(4)                                          COLPNT  61
                      RHO(5) = .6612093864662656-00                              COLPNT  62
                      RHO(2) = - RHO(5)                                          COLPNT  63
                      RHO(6) = .9324695142031526-00                              COLPNT  64
                      RHO(1) = - RHO(6)                                          COLPNT  65
   65                 GO TO 190                                                  COLPNT  66
                   70 RHO(5) = .4058451513773976-00                              COLPNT  67
                      RHO(3) = - RHO(5)                                          COLPNT  68
                      RHO(6) = .7415311855993946-00                              COLPNT  69
                      RHO(2) = - RHO(6)                                          COLPNT  70
   70                 RHO(7) = .9491079123427596-00                              COLPNT  71
                      RHO(1) = - RHO(7)                                          COLPNT  72
                      RHO(4) = 0.                                                COLPNT  73
                      GO TO 190                                                  COLPNT  74
                   80 RHO(5) = .1834346424956506-00                              COLPNT  75
   75                 RHO(4) = - RHO(5)                                          COLPNT  76
                      RHO(6) = .5255324099163296-00                              COLPNT  77
                      RHO(3) = - RHO(6)                                          COLPNT  78
                      RHO(7) = .7966647741362706-00                              COLPNT  79
                      RHO(2) = - RHO(7)                                          COLPNT  80
   80                 RHO(8) = .9602898564975366-00                              COLPNT  81
                      RHO(1) = - RHO(8)                                          COLPNT  82
                      GO TO 190                                                  COLPNT  83
                   90 RHO( 5) = .0                                               COLPNT  84
                      RHO( 6) = .3242534243038096-00                             COLPNT  85
   85                 RHO( 7) = .6133714327005906-00                             COLPNT  86
                      RHO( 8) = .8360311073266366-00                             COLPNT  87
                      RHO( 9) = .9681602395076266-00                             COLPNT  88
                      DO 95 I=1,4                                                COLPNT  89
                   95   RHO(I) = -RHO(10-I)                                      COLPNT  90
   90                 GO TO 190                                                  COLPNT  91
                  100 RHO( 6) = .1488743389816316-00                             COLPNT  92
                      RHO( 7) = .4333953941292476-00                             COLPNT  93
                      RHO( 8) = .6794095682990246-00                             COLPNT  94
                      RHO( 9) = .8650633666889846-00                             COLPNT  95
   95                 RHO(10) = .9739065285171726-00                             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
  100                 RHO( 7) = .2695431559523456-00                             COLPNT 101
                      RHO( 8) = .5190961292068126-00                             COLPNT 102
                      RHO( 9) = .7301520055740496-00                             COLPNT 103
                      RHO(10) = .8870625997680956-00                             COLPNT 104
                      RHO(11) = .9782246581460576-00                             COLPNT 105
  105                 DO 115 I=1,5                                               COLPNT 106
                  115   RHO(I) = -RHO(12-I)                                      COLPNT 107
                      GO TO 190                                                  COLPNT 108
                  120 RHO( 7) = .1252334085114696-00                             COLPNT 109
                      RHO( 8) = .3678314989981806-00                             COLPNT 110
  110                 RHO( 9) = .5873179542866176-00                             COLPNT 111
                      RHO(10) = .7699026741943056-00                             COLPNT 112
                      RHO(11) = .9041172563704756-00                             COLPNT 113
                      RHO(12) = .9815606342467196-00                             COLPNT 114
                      DO 125 I=1,6                                               COLPNT 115
```

```
115               125   RHO(I) = -RHO(13-I)                                  COLPNT  116
                        GO TO 190                                            COLPNT  117
                  130 RHO( 7) = .0                                           COLPNT  118
                      RHO( 8) = .2304583159551 35E-00                        COLPNT  119
                      RHO( 9) = .4484927510364 47E-00                        COLPNT  120
120                   RHO(10) = .6423493394403 40E-00                        COLPNT  121
                      RHO(11) = .8015780907333 10E-00                        COLPNT  122
                      RHO(12) = .9175983992229 78E-00                        COLPNT  123
                      RHO(13) = .9841830547185 88E-00                        COLPNT  124
                      DO 135 I=1,6                                           COLPNT  125
125               135   RHO(I) = -RHO(14-I)                                  COLPNT  126
                        GO TO 190                                            COLPNT  127
                  140 RHO( 8) = .1080549487073 44E-00                        COLPNT  128
                      RHO( 9) = .3191123649278 90E-00                        COLPNT  129
                      RHO(10) = .5152486363581 54E-00                        COLPNT  130
130                   RHO(11) = .6872929048116 85E-00                        COLPNT  131
                      RHO(12) = .8272013150697 65E-00                        COLPNT  132
                      RHO(13) = .9284348836635 74E-00                        COLPNT  133
                      RHO(14) = .9862838086968 12E-00                        COLPNT  134
                      DO 145 I=1,7                                           COLPNT  135
135               145   RHO(I) = -RHO(15-I)                                  COLPNT  136
                        GO TO 190                                            COLPNT  137
                  150 RHO( 8) = .0                                           COLPNT  138
                      RHO( 9) = .2011940939974 35E-00                        COLPNT  139
                      RHO(10) = .3941513470775 63E-00                        COLPNT  140
140                   RHO(11) = .5709721726085 39E-00                        COLPNT  141
                      RHO(12) = .7244177313601 70E-00                        COLPNT  142
                      RHO(13) = .8482065834104 27E-00                        COLPNT  143
                      RHO(14) = .9372733924007 06E-00                        COLPNT  144
                      RHO(15) = .9879925180204 85E-00                        COLPNT  145
145                   DO 155 I = 1,7                                         COLPNT  146
                  155   RHO(I) = -RHO(16-I)                                  COLPNT  147
                        GO TO 190                                            COLPNT  148
                  160 RHO( 9) = .9501250983763 74E-01                        COLPNT  149
                      RHO(10) = .2816035507792 59E-00                        COLPNT  150
150                   RHO(11) = .4580167776572 27E-00                        COLPNT  151
                      RHO(12) = .6178762444026 44E-00                        COLPNT  152
                      RHO(13) = .7554044083550 03E-00                        COLPNT  153
                      RHO(14) = .8656312023878 32E-00                        COLPNT  154
                      RHO(15) = .9445750230732 33E-00                        COLPNT  155
155                   RHO(16) = .9894009349916 50E-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
160                   RHO(10) = .1784841814958 48E-00                        COLPNT  161
                      RHO(11) = .3512317634538 76E-00                        COLPNT  162
                      RHO(12) = .5126905370864 77E-00                        COLPNT  163
                      RHO(13) = .6576711592166 91E-00                        COLPNT  164
                      RHO(14) = .7815140038968 01E-00                        COLPNT  165
165                   RHO(15) = .8802391537269 86E-00                        COLPNT  166
                      RHO(16) = .9506755217687 68E-00                        COLPNT  167
                      RHO(17) = .9905754753144 17E-00                        COLPNT  168
                      DO 175 I=1,8                                           COLPNT  169
                  175   RHO(I) = -RHO(18-I)                                  COLPNT  170
170                     GO TO 190                                            COLPNT  171
                  180 RHO(10) = .8477501304173 53E-01                        COLPNT  172
```

```
                RHO(11) = .25188622569150GE-00                                  COLPNT   173
                RHO(12) = .41175116146284JE-00                                  COLPNT   174
                RHO(13) = .55977083107394RE-00                                  COLPNT   175
      175       RHO(14) = .69168704306035E-00                                   COLPNT   176
                RHO(15) = .80370495897252JE-00                                  COLPNT   177
                RHO(16) = .89260246649755RE-00                                  COLPNT   178
                RHO(17) = .95582394957139RE-00                                  COLPNT   179
                RHO(18) = .99156516842093IE-00                                  COLPNT   180
      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
      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
      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
      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   200 ITOP = NCPTS - 1                                              COLPNT   200
      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
      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
      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
      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
      220     240 XC(I) = XR                                                    COLPNT   221
              RETURN                                                            COLPNT   222
              END                                                               COLPNT   223
```

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	1		CONSTANT TOO LONG. HIGH ORDER DIGITS RETAINED. BUT SOME PRECISION LOST.
161	1		CONSTANT TOO LONG. HIGH ORDER DIGITS RETAINED. BUT SOME PRECISION LOST.
162	1		CONSTANT TOO LONG. HIGH ORDER DIGITS RETAINED. BUT SOME PRECISION LOST.
163	1		CONSTANT TOO LONG. HIGH ORDER DIGITS RETAINED. BUT SOME PRECISION LOST.
164	1		CONSTANT TOO LONG. HIGH ORDER DIGITS RETAINED. BUT SOME PRECISION LOST.
165	1		CONSTANT TOO LONG. HIGH ORDER DIGITS RETAINED. BUT SOME PRECISION LOST.
166	1		CONSTANT TOO LONG. HIGH ORDER DIGITS RETAINED. BUT SOME PRECISION LOST.
167	1		CONSTANT TOO LONG. HIGH ORDER DIGITS RETAINED. BUT SOME PRECISION LOST.
171	1		CONSTANT TOO LONG. HIGH ORDER DIGITS RETAINED. BUT SOME PRECISION LOST.
173	1		CONSTANT TOO LONG. HIGH ORDER DIGITS RETAINED. BUT SOME PRECISION LOST.
174	1		CONSTANT TOO LONG. HIGH ORDER DIGITS RETAINED. BUT SOME PRECISION LOST.
175	1		CONSTANT TOO LONG. HIGH ORDER DIGITS RETAINED. BUT SOME PRECISION LOST.
176	1		CONSTANT TOO LONG. HIGH ORDER DIGITS RETAINED. BUT SOME PRECISION LOST.
177	1		CONSTANT TOO LONG. HIGH ORDER DIGITS RETAINED. BUT SOME PRECISION LOST.
178	1		CONSTANT TOO LONG. HIGH ORDER DIGITS RETAINED. BUT SOME PRECISION LOST.
179	1		CONSTANT TOO LONG. HIGH ORDER DIGITS RETAINED. BUT SOME PRECISION LOST.

```
      SUBROUTINE BSPLVD ( AT, 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,DIGITAL,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,A,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,A,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 + K - M                                                   BSPLVD    55
        DIFF = XT(IPKMD) - AT(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
                        A(L,J) = (A(L,J) - A(L,J-1))/DIFF*FKMJ                   BSPLVD     60
                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(I,1) = A(I,1)/DIFF*FKMJ                              BSPLVD     65
                65    90  DO 110 I=1,K                                          BSPLVD     66
                          V = 0.                                                 BSPLVD     67
                          JLOW = MAXO(I,M)                                       BSPLVD     68
                          DO 100 J=JLOW,K                                        BSPLVD     69
                     100    V = A(I,J)*VNIKX(J,M) + V                           BSPLVD     70
                70   110  VNIKX(I,M) = V                                        BSPLVD     71
                     120 RETURN                                                  BSPLVD     72
                         END                                                     BSPLVD     73
```

```
      SUBROUTINE BSPLVN ( XT, JHIGH, INDEX, X, ILEFT, VNIKX )
C-----------------------------------------------------------------------
C THIS SUBROUTINE IS PART OF THE B-SPLINE PACKAGE FOR THE STABLE
C EVALUATION OF ANY B-SPLINE BASIS FUNCTION OR DERIVATIVE VALUE.
C SEE REFERENCE BELOW.
C
C CALCULATES THE VALUE OF ALL POSSIBLY NONZERO B-SPLINES AT THE
C POINT X OF ORDER MAX(JHIGH,(J+1)(INDEX-1)) FOR THE BREAKPOINT SEQ-
C UENCE XT.  ASSUMING THAT XT(ILEFT) .LE. X .LE. XT(ILEFT+1), THE ROUT-
C INE RETURNS THE B-SPLINE VALUES IN THE ONE DIMENSIONAL ARRAY VNIKX.
C
C FOR DEFINITIONS OF CALLING ARGUMENTS SEE ABOVE AND BSPLVD.
C
C REFERENCE
C
C    DEBOOR, C., PACKAGE FOR CALCULATING WITH B-SPLINES, SIAM J.
C      NUMER. ANAL., VOL. 14, NO. 3, JUNE 1977, PP. 441-472.
C
C PACKAGE ROUTINES CALLED..  NONE
C USER ROUTINES CALLED..     NONE
C CALLED BY..                BSPLVD
C FORTRAN FUNCTIONS USED..   NONE
C-----------------------------------------------------------------------
      DIMENSION XT(1),VNIKX(1)
      DIMENSION DELTAM(20),DELTAP(20)
      DATA J/1/,DELTAM/20*0.E-00/,DELTAP/20*0.E-00/
      GO TO (10,20),INDEX
   10 J = 1
      VNIKX(1) = 1.
      IF (J .GE. JHIGH) GO TO 40
   20 IPJ = ILEFT+J
      DELTAP(J) = XT(IPJ) - X
      IMJP1 = ILEFT-J+1
      DELTAM(J) = X - XT(IMJP1)
      VMPREV = 0.
      JP1 = J+1
      DO 30 L=1,J
        JP1ML = JP1-L
        VM = VNIKX(L)/(DELTAP(L) + DELTAM(JP1ML))
        VNIKX(L) = VM*DELTAP(L) + VMPREV
   30   VMPREV = VM*DELTAM(JP1ML)
      VNIKX(JP1) = VMPREV
      J = JP1
      IF (J .LT. JHIGH) GO TO 20
   40 RETURN
      END
```

BSPLVN	2
BSPLVN	3
BSPLVN	4
BSPLVN	5
BSPLVN	6
BSPLVN	7
BSPLVN	8
BSPLVN	9
BSPLVN	10
BSPLVN	11
BSPLVN	12
BSPLVN	13
BSPLVN	14
BSPLVN	15
BSPLVN	16
BSPLVN	17
BSPLVN	18
BSPLVN	19
BSPLVN	20
BSPLVN	21
BSPLVN	22
BSPLVN	23
BSPLVN	24
BSPLVN	25
BSPLVN	26
BSPLVN	27
BSPLVN	28
BSPLVN	29
BSPLVN	30
BSPLVN	31
BSPLVN	32
BSPLVN	33
BSPLVN	34
BSPLVN	35
BSPLVN	36
BSPLVN	37
BSPLVN	38
BSPLVN	39
BSPLVN	40
BSPLVN	41
BSPLVN	42
BSPLVN	43
BSPLVN	44
BSPLVN	45
BSPLVN	46
BSPLVN	47

CARD NR. SEVERITY DETAILS DIAGNOSIS OF PROBLEM

21 1 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,INITIAL,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(NO,TOUT,Y,YMAX,ERROR,SAVE1,SAVE2,SAVE3,          STIFIB      2
     *                  P,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 NO 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   NO       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.                                        STIFIB     59
C   ERROR      AN ARRAY OF N ELEMENTS.  ERROR(I)/TQ(2) IS THE ESTIMATED         STIFIB     60
C              ONE-STEP ERROR IN Y(I).                                          STIFIB     61
C   SAVE1,SAVE2,SAVE3   THREE WORKING STORAGE ARRAYS, EACH OF LENGTH N.        STIFIB     62
C   PW         A BLOCK OF LOCATIONS USED FOR THE CHORD ITERATION              STIFIB     63
C              MATRIX.  SEE DESCRIPTION IN PDECOL.                            STIFIB     64
C   IPIV       AN INTEGER ARRAY OF LENGTH N FOR PIVOT INFORMATION.            STIFIB     65
C   ML,MU      THE LOWER AND UPPER HALF BANDWIDTHS, RESPECTIVELY, OF          STIFIB     66
C              THE CHORD ITERATION MATRIX.  SEE DESCRIPTION IN PDECOL.        STIFIB     67
C   WORK,IWORK    WORKING ARRAYS WHICH ARE USED TO PASS APPROPRIATE            STIFIB     68
C                 ARRAYS TO OTHER SUBROUTINES.                                STIFIB     69
C                                                                             STIFIB     70
C PACKAGE ROUTINES CALLED..  COSET,DIFFUN,PSETIB,PES,SOLB                     STIFIB     71
C USER ROUTINES CALLED..     NONE                                             STIFIB     72
C CALLED BY..                PDECOL                                           STIFIB     73
C FORTRAN FUNCTIONS USED..   ABS,AMAX1,AMIN1,FLOAT                           STIFIB     74
C-------------------------------------------------------------------------    STIFIB     75
      DIMENSION Y(NO,1),YMAX(NO),ERROR(NO),SAVE1(NO),SAVE2(NO),               STIFIB     76
     *          SAVE3(NO),PW(1),IPIV(1),WORK(1),IWORK(1)                      STIFIB     77
      COMMON /SIZES/ NINT,KORD,NCC,NPDE,NCPTS,NEQN,IQUAD                      STIFIB     78
      COMMON /ISTART/ IW1,IW2,IW3,IW4,IW5,IW6,IW7,IW8,IW9,IW10,IW11,          STIFIB     79
     *                IW12,IW13,IW14,IW15,IW16,IW17,IW18                       STIFIB     80
      COMMON /GEAR1/ T,H,HMIN,HMAX,EPS,UROUND,N,MF,KFLAG,JSTART               STIFIB     81
      COMMON/GEAR9/EPSJ,RO,ML,MU,MW,NML,NOML,NOW,NOWI                         STIFIB     82
      COMMON /GEAR0/ NUSED,NQUSED,NSTEP,NFE,NJE                               STIFIB     83
      COMMON/GEARSS/HMX                                                       STIFIB     84
      COMMON /OPTION/ NOGAUS,MAXDER                                           STIFIB     85
      COMMON/TARSM/SMALL                                                      STIFIB     86
      DIMENSION EL(13),TQ(4)                                                  STIFIB     87
      DATA EL(2)/1./, OLDLO/1./, TQ(1)/0./, IER/0/                            STIFIB     88
      KFLAG = 0                                                               STIFIB     89
      TOLD = T                                                                STIFIB     90
      IF (JSTART .GT. 0) GO TO 200                                            STIFIB     91
      IF (JSTART .NE. 0) GO TO 120                                            STIFIB     92
C-------------------------------------------------------------------------    STIFIB     93
C ON THE FIRST CALL, THE ORDER IS SET TO 1, AND THE INITIAL YDOT IS            STIFIB     94
C CALCULATED.  RMAX IS THE MAXIMUM RATIO BY WHICH H CAN BE INCREASED           STIFIB     95
C IN A SINGLE STEP.  IT IS INITIALLY 1.E4 TO COMPENSATE FOR THE SMALL          STIFIB     96
C INITIAL H, BUT THEN IS NORMALLY EQUAL TO 10.  IF A FAILURE                   STIFIB     97
C OCCURS (IN CORRECTOR CONVERGENCE OR ERROR TEST), RMAX IS SET AT 2            STIFIB     98
C FOR THE NEXT INCREASE.                                                       STIFIB     99
C-------------------------------------------------------------------------    STIFIB    100
      NQ = 1                                                                  STIFIB    101
      IER = 0                                                                 STIFIB    102
      CALL DIFFUN(N,T,Y,SAVE1,IER,PW,IPIV,WORK,IWORK)                         STIFIB    103
      IF ( IER .NE. 0 ) GO TO 685                                             STIFIB    104
      DO 110 I = 1,N                                                          STIFIB    105
  110   Y(I,2) = H*SAVE1(I)                                                   STIFIB    106
      METH = MF/10                                                            STIFIB    107
      MITER = MF - 10*METH                                                    STIFIB    108
      L = 2                                                                   STIFIB    109
      IDOUB = 3                                                               STIFIB    110
      RMAX = 1.E+04                                                           STIFIB    111
      RC = 0.                                                                 STIFIB    112
      CRATE = 1.                                                              STIFIB    113
      EPSOLD = EPS                                                            STIFIB    114
      HMXI = HMX                                                              STIFIB    115
```

```
      115               MFOLD = MF                                                STIFIB  116
                        NOLD = N                                                  STIFIB  117
                        NSTEP = 0                                                 STIFIB  118
                        NSTEPJ = 0                                                STIFIB  119
                        NFE = 0                                                   STIFIB  120
      120               NJE = 1                                                   STIFIB  121
                        IRET = 3                                                  STIFIB  122
                        GO TO 130                                                 STIFIB  123
                  C-------------------------------------------------------------- STIFIB  124
                  C IF THE CALLER HAS CHANGED METH, COSET IS CALLED TO SET       STIFIB  125
      125         C THE COEFFICIENTS OF THE METHOD.  IF THE CALLER HAS CHANGED   STIFIB  126
                  C N, EPS, OR METH, THE CONSTANTS E, EUN, EUP, AND RND MUST BE RESET.  STIFIB  127
                  C E IS A COMPARISON FOR ERRORS OF THE CURRENT ORDER NQ. EUP IS STIFIB  128
                  C TO TEST FOR INCREASING THE ORDER.  EUN FOR DECREASING THE ORDER.  STIFIB  129
                  C RND IS USED TO TEST FOR CONVERGENCE OF THE CORRECTOR ITERATES.  STIFIB  130
      130         C IF THE CALLER HAS CHANGED H, Y MUST BE RESCALED.             STIFIB  131
                  C IF H OR METH HAS BEEN CHANGED, IDOUB IS RESET TO L + 1 TO PREVENT  STIFIB  132
                  C FURTHER CHANGES IN H FOR THAT MANY STEPS.                    STIFIB  133
                  C-------------------------------------------------------------- STIFIB  134
                    120 IF (MF .EQ. MFOLD) GO TO 150                              STIFIB  135
      135               MEO = METH                                                STIFIB  136
                        MIO = MITER                                               STIFIB  137
                        METH = MF/10                                              STIFIB  138
                        MITER = MF - 10*METH                                      STIFIB  139
                        MFOLD = MF                                                STIFIB  140
      140               IF (MITER .NE. MIO) IWEVAL = MITER                        STIFIB  141
                        IF (METH .EQ. MEO) GO TO 150                              STIFIB  142
                        IDOUB = L + 1                                             STIFIB  143
                        IRET = 1                                                  STIFIB  144
                    130 CALL COSET (METH, NQ, EL, TQ)                             STIFIB  145
      145               LMAX = MAXDER + 1                                         STIFIB  146
                        RC = RC*EL(1)/OLDL0                                       STIFIB  147
                        OLDL0 = EL(1)                                             STIFIB  148
                    140 FN =  FLOAT(N)                                            STIFIB  149
                        EUN = FN*(TQ(1)*EPS)**2                                   STIFIB  150
      150               E   = FN*(TQ(2)*EPS)**2                                   STIFIB  151
                        EUP = FN*(TQ(3)*EPS)**2                                   STIFIB  152
                        RND = FN*(TQ(4)*EPS)**2                                   STIFIB  153
                        GO TO (160, 170, 200), IRET                               STIFIB  154
                    150 IF ((EPS .EQ. EPSOLD) .AND. (N .EQ. NOLD)) GO TO 160      STIFIB  155
      155               EPSOLD = EPS                                              STIFIB  156
                        NOLD = N                                                  STIFIB  157
                        IRET = 1                                                  STIFIB  158
                        GO TO 140                                                 STIFIB  159
                    160 IF (H .EQ. HOLD) GO TO 200                                STIFIB  160
      160               RH = H/HOLD                                               STIFIB  161
                        H = HOLD                                                  STIFIB  162
                        IREDO = 3                                                 STIFIB  163
                        GO TO 175                                                 STIFIB  164
                    170 RH = AMAX1(RH,HMIN/ ABS(H))                               STIFIB  165
      165           175 RH = AMIN1(RH,HMAX/ ABS(H),RMAX)                          STIFIB  166
                        RH=AMIN1(RH,HMX/ABS(H))                                   STIFIB  167
                        R1 = 1.                                                   STIFIB  168
                        DO 180 J = 2,L                                            STIFIB  169
                          R1 = R1*RH                                              STIFIB  170
      170                 DO 180 I = 1,N                                        STIFIB  171
                    180     Y(I,J) = Y(I,J)*R1                                    STIFIB  172
```

```
                H = H*RH                                                        STIFIB   173
                RC = RC*RH                                                      STIFIB   174
                IDOUB = L + 1                                                   STIFIB   175
      175       IF (IREDO .EQ. 0) GO TO 690                                     STIFIB   176
              C-------------------------------------------------------------    STIFIB   177
              C THIS SECTION COMPUTES THE PREDICTED VALUES BY EFFECTIVELY       STIFIB   178
              C MULTIPLYING THE Y ARRAY BY THE PASCAL TRIANGLE MATRIX.          STIFIB   179
              C RC IS THE RATIO OF NEW TO OLD VALUES OF THE COEFFICIENT  H*EL(1).STIFIB   180
      180     C WHEN RC DIFFERS FROM 1 BY MORE THAN 30 PERCENT, OR THE CALLER HAS STIFIB   181
              C CHANGED MITER, IWEVAL IS SET TO MITER TO FORCE PW TO BE UPDATED. STIFIB   182
              C IN ANY CASE, PW IS UPDATED AT LEAST EVERY 40-TH STEP.           STIFIB   183
              C PW IS THE CHORD ITERATION MATRIX A - H*EL(1)*(DG/DY).           STIFIB   184
              C-------------------------------------------------------------    STIFIB   185
      185       200 IF ( ABS(RC-1.) .GT. 0.3) IWEVAL = MITER                    STIFIB   186
                IF (NSTEP .GE. NSTEPJ+40) IWEVAL = MITER                        STIFIB   187
                T = T + H                                                       STIFIB   188
                DO 210 J1 = 1,NQ                                                STIFIB   189
                  DO 210 J2 = J1,NQ                                             STIFIB   190
      190           J = (NQ + J1) - J2                                          STIFIB   191
                    DO 210 I = 1,N                                              STIFIB   192
              210     Y(I,J) = Y(I,J) + Y(I,J+1)                                STIFIB   193
              220   CONTINUE                                                    STIFIB   194
                    IF (IWEVAL .LE. 0) GO TO 350                                STIFIB   195
      195     C-------------------------------------------------------------    STIFIB   196
              C IF INDICATED, THE MATRIX PW IS REEVALUATED BEFORE STARTING THE  STIFIB   197
              C CORRECTOR ITERATION.  IWEVAL IS SET TO 0 AS AN INDICATOR        STIFIB   198
              C THAT THIS HAS BEEN DONE.  PW IS COMPUTED AND PROCESSED IN PSETIB.STIFIB   199
              C-------------------------------------------------------------    STIFIB   200
      200 S   IWEVAL = 0                                                        STIFIB   201
              RC = 1.                                                           STIFIB   202
              NJE = NJE + 1                                                     STIFIB   203
              NSTEPJ = NSTEP                                                    STIFIB   204
              CON = -H*EL(1)                                                    STIFIB   205
      205     CALL PSETIB (Y, PW, NQ, CON, MITER, IER, WORK(IW1), IWORK,        STIFIB   206
             *      WORK(IW3),WORK(IW9),SAVE2,IPIV,YMAX,WORK(IW11),WORK(IW12), STIFIB   207
             *      WORK(IW13),WORK(IW16),WORK(IW14),WORK(IW15),WORK,NPDE)      STIFIB   208
              IF (IER .NE. 0) GO TO 420                                         STIFIB   209
              350 CONTINUE                                                      STIFIB   210
      210     CC*****************************************************           STIFIB   211
              CC  UP TO 3 CORRECTOR ITERATIONS ARE TAKEN.  A CONVERGENCE TEST IS STIFIB   212
              CC  MADE ON THE R. M. S. NORM OF EACH CORRECTION, USING RND, WHICH STIFIB   213
              CC  IS DEPENDENT ON EPS.  THE SUM OF THE CORRECTIONS IS ACCUMULATED IN STIFIB   214
              CC  THE VECTOR ERROR(I).  THE Y ARRAY IS NOT ALTERED IN THE CORRECTOR STIFIB   215
      215     CC  LOOP.  THE UPDATED Y VECTOR IS STORED TEMPORARILY IN SAVE1.  THE STIFIB   216
              CC  UPDATED H*YDOT IS STORED IN SAVE2.                            STIFIB   217
              CC*****************************************************           STIFIB   218
                    DO 230 I=1,N                                                STIFIB   219
                    SAVE2(I)=Y(I,2)                                             STIFIB   220
      220           SAVE1(I)=Y(I,1)                                             STIFIB   221
              230   ERROR(I)=0.0                                                STIFIB   222
                    M=0                                                         STIFIB   223
              360   D=0                                                         STIFIB   224
                    DO 385 K=1,NPDE                                             STIFIB   225
      225           IST=(K-1)*NCPTS+1                                           STIFIB   226
                    IEND=K*NCPTS                                                STIFIB   227
                    NST=(K-1)*NQ*1+1                                            STIFIB   228
                    CALL RES(T,H,SAVE1,SAVE2,SAVE3,CODE,NCPTS,WORK(IW1),IWORK,  STIFIB   229
```

```
     *      WORK,WORK(IW14),WORK(IW15),WORK(IW16),WORK(IW3),                  STIFIB  230
     *      WORK(IW9),K)                                                     STIFIB  231
C-----------------------------------------------------------------------     STIFIB  232
C COMPUTE THE CORRECTOR ERROR, R SUB 1, AND SOLVE THE LINEAR SYSTEM          STIFIB  233
C WITH THAT AS RIGHT-HAND SIDE AND PW AS COEFFICIENT MATRIX,                 STIFIB  234
C USING THE LU DECOMPOSITION OF PW.                                          STIFIB  235
C-----------------------------------------------------------------------     STIFIB  236
      CALL SOLB(NCPTS,NCPTS,ML,MU,P(1,IST),SAVE3(IST),IPIV(IST))             STIFIB  237
      DO 380 I=IST,IEND                                                      STIFIB  238
        ERROR(I) = ERROR(I) + SAVE3(I)                                       STIFIB  239
        D = D + (SAVE3(I)/YMAX(I))**2                                        STIFIB  240
        SAVE1(I) = Y(I,1) + EL(1)*ERROR(I)                                   STIFIB  241
  380   SAVE2(I) = Y(I,2) + ERROR(I)                                         STIFIB  242
  385 CONTINUE                                                               STIFIB  243
      NFE=NFE+1                                                              STIFIB  244
C-----------------------------------------------------------------------     STIFIB  245
C TEST FOR CONVERGENCE.  IF M.GT.0, AN ESTIMATE OF THE CONVERGENCE           STIFIB  246
C RATE CONSTANT IS STORED IN CRATE, AND THIS IS USED IN THE TEST.            STIFIB  247
C-----------------------------------------------------------------------     STIFIB  248
  400 IF (M .NE. 0) CRATE = AMAX1(.9*CRATE,D/D1)                             STIFIB  249
      IF((D*AMIN1(1.0E0,2.*CRATE)).LE.END)GO TO 450                          STIFIB  250
      D1 = D                                                                 STIFIB  251
      M = M + 1                                                              STIFIB  252
      IF (M .EQ. 3) GO TO 410                                                STIFIB  253
      GO TO 360                                                              STIFIB  254
C-----------------------------------------------------------------------     STIFIB  255
C THE CORRECTOR ITERATION FAILED TO CONVERGE IN 3 TRIES.                     STIFIB  256
CC THE Y ARRAY IS RETRACTED TO ITS VALUES                                    STIFIB  257
C BEFORE PREDICTION, AND H IS REDUCED, IF POSSIBLE.  IF NOT, A               STIFIB  258
C NO-CONVERGENCE EXIT IS TAKEN.                                              STIFIB  259
C-----------------------------------------------------------------------     STIFIB  260
410   CONTINUE                                                               STIFIB  261
      IWEVAL=MITER                                                           STIFIB  262
  420 T = TOLD                                                               STIFIB  263
      RMAX = 2.                                                              STIFIB  264
      DO 430 J1 = 1,NQ                                                       STIFIB  265
        DO 430 J2 = J1,NQ                                                    STIFIB  266
          J = (NQ + J1) - J2                                                 STIFIB  267
          DO 430 I = 1,N                                                     STIFIB  268
  430       Y(I,J) = Y(I,J) - Y(I,J+1)                                       STIFIB  269
      IF ( ABS(H) .LE. HMIN*1.00001) GO TO 680                               STIFIB  270
      RH=0.60                                                                STIFIB  271
      IREDO = 1                                                              STIFIB  272
      GO TO 170                                                              STIFIB  273
C-----------------------------------------------------------------------     STIFIB  274
C THE CORRECTOR HAS CONVERGED.  IWEVAL IS SET TO -1 TO SIGNAL                STIFIB  275
C THAT PW MAY NEED UPDATING ON SUBSEQUENT STEPS.  THE ERROR TEST             STIFIB  276
C IS MADE AND CONTROL PASSES TO STATEMENT 500 IF IT FAILS.                   STIFIB  277
C-----------------------------------------------------------------------     STIFIB  278
 450  IWEVAL = -1                                                           STIFIB  279
      D = 0.                                                                 STIFIB  280
      DO 460 I = 1,N                                                         STIFIB  281
  460   D = D + (ERROR(I)/YMAX(I))**2                                        STIFIB  282
      IF (D .GT. E) GO TO 500                                                STIFIB  283
C-----------------------------------------------------------------------     STIFIB  284
C AFTER A SUCCESSFUL STEP, UPDATE THE Y ARRAY.                               STIFIB  285
C CONSIDER CHANGING H IF IDOUM = 1.  OTHERWISE DECREASE IDOUM BY 1.          STIFIB  286
```

```
C IF IDOUB IS THEN 1 AND NQ .LT. MAXORD, THEN ERROR IS SAVED FOR          STIFIB   287
C USE IN A POSSIBLE ORDER INCREASE ON THE NEXT STEP.                      STIFIB   288
C IF A CHANGE IN H IS CONSIDERED, AN INCREASE OR DECREASE IN ORDER        STIFIB   289
C BY ONE IS CONSIDERED ALSO.  A CHANGE IN H IS MADE ONLY IF IT IS BY A    STIFIB   290
CC FACTOR OF AT LEAST 1.1.  IF NOT, IDOUB IS SET TO 3 TO PREVENT          STIFIB   291
C TESTING FOR THAT MANY STEPS.                                            STIFIB   292
CC H IS NOT INCREASED NEAR TOUT.                                          STIFIB   293
C-------------------------------------------------------------------     STIFIB   294
      KFLAG = 0                                                           STIFIB   295
      IREDO = 0                                                           STIFIB   296
      NSTEP = NSTEP + 1                                                   STIFIB   297
      HUSED = H                                                           STIFIB   298
      NQUSED = NQ                                                         STIFIB   299
      DO 470 J = 1,L                                                      STIFIB   300
        DO 470 I = 1,N                                                    STIFIB   301
  470     Y(I,J) = Y(I,J) + EL(J)*ERROR(I)                                STIFIB   302
      IF(T+2.0*H.GT.TOUT)IDOUB=IDOUB+1                                    STIFIB   303
      IDOUB=IDOUB+M                                                       STIFIB   304
      IF (IDOUB .EQ. 1) GO TO 520                                         STIFIB   305
      IDOUB = IDOUB - 1                                                   STIFIB   306
      IF (IDOUB .GT. 1) GO TO 700                                         STIFIB   307
      IF (L .EQ. LMAX) GO TO 700                                          STIFIB   308
      DO 490 I = 1,N                                                      STIFIB   309
  490   Y(I,LMAX) = ERROR(I)                                              STIFIB   310
      GO TO 700                                                           STIFIB   311
C-------------------------------------------------------------------     STIFIB   312
C THE ERROR TEST FAILED.  KFLAG KEEPS TRACK OF MULTIPLE FAILURES.         STIFIB   313
C RESTORE T AND THE Y ARRAY TO THEIR PREVIOUS VALUES, AND PREPARE         STIFIB   314
C TO TRY THE STEP AGAIN.  COMPUTE THE OPTIMUM STEP SIZE FOR THIS OR       STIFIB   315
C ONE LOWER ORDER.                                                        STIFIB   316
C-------------------------------------------------------------------     STIFIB   317
  500 KFLAG = KFLAG - 1                                                   STIFIB   318
      T = TOLD                                                            STIFIB   319
      DO 510 J1 = 1,NQ                                                    STIFIB   320
        DO 510 J2 = J1,NQ                                                 STIFIB   321
          J = (NQ + J1) - J2                                              STIFIB   322
          DO 510 I = 1,N                                                  STIFIB   323
  510       Y(I,J) = Y(I,J) - Y(I,J+1)                                    STIFIB   324
      RMAX = 2.                                                           STIFIB   325
      IF ( ABS(H) .LE. HMIN*1.00001) GO TO 660                            STIFIB   326
      IF (KFLAG .LE. -3) GO TO 640                                        STIFIB   327
      IREDO = 2                                                           STIFIB   328
      ENQ2=.5/FLOAT(L)                                                    STIFIB   329
      PR2=((D/E)**ENQ2)*2.0+2.0E-06                                       STIFIB   330
      RH=1.0/PR2                                                          STIFIB   331
      GO TO 170                                                           STIFIB   332
C-------------------------------------------------------------------     STIFIB   333
C REGARDLESS OF THE SUCCESS OR FAILURE OF THE STEP, FACTORS               STIFIB   334
C PR1, PR2, AND PR3 ARE COMPUTED, BY WHICH H COULD BE DIVIDED             STIFIB   335
C AT ORDER NQ - 1, ORDER NQ, OR ORDER NQ + 1, RESPECTIVELY.               STIFIB   336
C IN THE CASE OF FAILURE, PR3 = 1.E20 TO AVOID AN ORDER INCREASE.         STIFIB   337
C THE SMALLEST OF THESE IS DETERMINED AND THE NEW ORDER CHOSEN            STIFIB   338
C ACCORDINGLY.  IF THE ORDER IS TO BE INCREASED, WE COMPUTE ONE           STIFIB   339
C ADDITIONAL SCALED DERIVATIVE.                                           STIFIB   340
C-------------------------------------------------------------------     STIFIB   341
  520 PR3 = 1.E+20                                                        STIFIB   342
      IF (L .EQ. LMAX) GO TO 540                                          STIFIB   343
```

```
                      D1 = 0.                                                    STIFIH   344
                      DO 530 I = 1,N                                             STIFIH   345
  345               530   D1 = D1 + ((ERROR(I) - Y(I,LMAX))/YMAX(I))**2          STIFIH   346
                      ENO3 = .5/ FLOAT(L+1)                                      STIFIH   347
                      PR3 = ((D1/EUP)**ENO3)*1.4 + 1.4E-06                       STIFIH   348
                    540 ENO2 = .5/ FLOAT(L)                                      STIFIH   349
                      PR2 = ((D/E)**ENO2)*1.2 + 1.2E-06                          STIFIH   350
  350                 PR1 = 1.E+20                                               STIFIH   351
                      IF (NQ .EQ. 1) GO TO 560                                   STIFIH   352
                      D = 0.                                                     STIFIH   353
                      DO 550 I = 1,N                                             STIFIH   354
                    550   D = D + (Y(I,L)/YMAX(I))**2                           STIFIH   355
  355                 ENO1 = .5/ FLOAT(NQ)                                       STIFIH   356
                      PR1 = ((D/EON)**ENO1)*1.3 + 1.3E-06                        STIFIH   357
                    560 IF (PR2 .LE. PR3) GO TO 570                              STIFIH   358
                      IF (PR3 .LT. PR1) GO TO 590                                STIFIH   359
                      GO TO 580                                                  STIFIH   360
  360               570 IF (PR2 .GT. PR1) GO TO 580                              STIFIH   361
                      NEWQ = NQ                                                  STIFIH   362
                      RH = 1./PR2                                                STIFIH   363
                      GO TO 620                                                  STIFIH   364
                    580 NEWQ = NQ - 1                                            STIFIH   365
  365                 RH = 1./PR1                                                STIFIH   366
                      GO TO 620                                                  STIFIH   367
                    590 NEWQ = L                                                 STIFIH   368
                      RH = 1./PR3                                                STIFIH   369
                      IF (RH .LT. 1.1) GO TO 610                                 STIFIH   370
  370                 DO 600 I = 1,N                                             STIFIH   371
                    600   Y(I,NEWQ+1) = ERROR(I)*EL(L)/ FLOAT(L)                 STIFIH   372
                      GO TO 630                                                  STIFIH   373
                    610   IDOUB=3                                                STIFIH   374
                      GO TO 700                                                  STIFIH   375
  375               620 IF ((KFLAG .EQ. 0) .AND. (RH .LT. 1.1)) GO TO 610      STIFIH   376
                    C-----------------------------------------------------------------  STIFIH   377
                    C IF THERE IS A CHANGE OF ORDER, RESET NQ, L, AND THE COEFFICIENTS.  STIFIH   378
                    C IN ANY CASE H IS RESET ACCORDING TO RH AND THE Y ARRAY IS RESCALED.  STIFIH   379
                    C THEN EXIT FROM 690 IF THE STEP WAS OK, OR REDO THE STEP OTHERWISE.  STIFIH   380
  380               C-----------------------------------------------------------------  STIFIH   381
                      IF (NEWQ .EQ. NQ) GO TO 170                                STIFIH   382
                    630 NQ = NEWQ                                                STIFIH   383
                      L = NQ + 1                                                 STIFIH   384
                      IRET = 2                                                   STIFIH   385
  385                 GO TO 130                                                  STIFIH   386
                    C-----------------------------------------------------------------  STIFIH   387
                    C CONTROL REACHES THIS SECTION IF 3 OR MORE FAILURES HAVE OCCURED.  STIFIH   388
                    C IT IS ASSUMED THAT THE DERIVATIVES THAT HAVE ACCUMULATED IN THE  STIFIH   389
                    C Y ARRAY HAVE ERRORS OF THE WRONG ORDER.  HENCE THE FIRST        STIFIH   390
  390               C DERIVATIVE IS RECOMPUTED, AND THE ORDER IS SET TO 1.  THEN      STIFIH   391
                    C H IS REDUCED BY A FACTOR OF 10, AND THE STEP IS RETRIED.        STIFIH   392
                    C AFTER A TOTAL OF 7 FAILURES, AN EXIT IS TAKEN WITH KFLAG = -2.  STIFIH   393
                    C-----------------------------------------------------------------  STIFIH   394
                    640 IF (KFLAG .EQ. -7) GO TO 670                             STIFIH   395
  395                 RH = .1                                                    STIFIH   396
                      RH = AMAX1(HMIN/ ABS(H),RH)                                STIFIH   397
                      H = H*RH                                                   STIFIH   398
                      IER = 0                                                    STIFIH   399
                      CALL DIFFUN (N, T, Y, SAVE1, IER, RW, IPIV, WORK, IWORK)   STIFIH   400
```

```
                      IF ( IER .NE. 0 ) GO TO 685                                STIFIB   401
                      NJE = NJE + 1                                              STIFIB   402
                      DO 650 I = 1,N                                             STIFIB   403
                650     Y(I,2) = H*SAVE1(I)                                      STIFIB   404
                      IWEVAL = MITER                                             STIFIB   405
                      IDOUB=3                                                    STIFIB   406
                      IF (NQ .EQ. 1) GO TO 200                                   STIFIB   407
                      NQ = 1                                                     STIFIB   408
                      L = 2                                                      STIFIB   409
                      IHET = 3                                                   STIFIB   410
                      GO TO 130                                                  STIFIB   411
                C-------------------------------------------------------------   STIFIB   412
                C ALL RETURNS ARE MADE THROUGH THIS SECTION.  H IS SAVED IN HOLD  STIFIB   413
                C TO ALLOW THE CALLER TO CHANGE H ON THE NEXT STEP.               STIFIB   414
                C-------------------------------------------------------------   STIFIB   415
                660   KFLAG = -1                                                 STIFIB   416
                      GO TO 700                                                  STIFIB   417
                670   KFLAG = -2                                                 STIFIB   418
                      GO TO 700                                                  STIFIB   419
                680   KFLAG = -3                                                 STIFIB   420
                      GO TO 700                                                  STIFIB   421
                685   KFLAG = -4                                                 STIFIB   422
                      GO TO 700                                                  STIFIB   423
                690   RMAX=10.0                                                  STIFIB   424
                700   HOLD = H                                                   STIFIB   425
                      JSTART = NQ                                                STIFIB   426
                C  WE DO NOT WANT ANY NEGATIVE CONCENTRATIONS.                   STIFIB   427
                      RETURN                                                     STIFIB   428
                      END                                                        STIFIB   429
```

CARD NR. SEVERITY DETAILS DIAGNOSIS OF PROBLEM

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

```
      SUBROUTINE GFUN (T,C,UDOT,NPDE, CPTS,A,BC,DBDU,DBDUX,DZDT,
     *     XC,UVAL,ILEFT,KPDE,KSKT,KSKR)
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(1),UDOT(1)
      DIMENSION A(1),BC(NPDE,4),XC(1),UVAL(NPDE,3),ILEFT(1)
      DIMENSION DZDT(NPDE),DBDU(NPDE),DBDUX(NPDE)
      COMMON /SIZES/ NINT,KORD,IDUM(4),IQUAD
      DO 10 J=1,4
10    BC(KPDE,J)=0.0
C-----------------------------------------------------------------------
C UPDATE THE LEFT BOUNDARY VALUES.  SAVE LEFT BOUNDARY CONDITION
CC  INFORMATION IN THE FIRST 2*NPDE LOCATIONS OF BC.
C
C NOTE.. UVAL(K,1) = U(K), UVAL(K,2) = UX(K), AND UVAL(K,3) = UXX(K).
C-----------------------------------------------------------------------
      KS=(KPDE-1)*NCPTS+1
      CALL EVAL(1,NPDE,C,UVAL,A,ILEFT)
      CALL F(T,XC(1),UVAL,UVAL(1,2),UVAL(1,3),UDOT(KS),
     *     NPDE,KPDE,1,KSKT,KSKR)
      CALL BNDRY(T,XC(1),UVAL,UVAL(1,2),DBDU,DBDUX,DZDT,NPDE)
      ILIM = KORD + 2
      BC(KPDE,1)=1.0
      IF (DBDU(KPDE).EQ.0.0.AND.DBDUX(KPDE).EQ.0.0)GO TO 30
      UDOT(KS)=DZDT(KPDE)
      BC(KPDE,2)=A(ILIM)*DBDUX(KPDE)
      BC(KPDE,1)=DBDU(KPDE)-BC(KPDE,2)
   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)
      KS=KS+1
      CALL F(T,XC(I),UVAL,UVAL(1,2),UVAL(1,3),UDOT(KS),
     *     NPDE,KPDE,I,KSKT,KSKR)
   40 CONTINUE
C-----------------------------------------------------------------------
C UPDATE THE RIGHT BOUNDARY VALUES.  SAVE THE RIGHT BOUNDARY CONDITION
CC  INFORMATION IN THE LAST 2*NPDE LOCATIONS IN BC.
C-----------------------------------------------------------------------
      CALL EVAL(NCPTS,NPDE,C,UVAL,A,ILEFT)
      KS=KS+1
      CALL F(T,XC(NCPTS),UVAL,UVAL(1,2),UVAL(1,3),UDOT(KS),
     *     NPDE,KPDE,NCPTS,KSKT,KSKR)
```

```
                CALL HNDRY(T,XC(NCPTS),UVAL,UVAL(1,2),DHDU,DHDUX,DZDT,NPDE)      GFUN      59
                ILIM = NCPTS * J * KORD - KORD - 1                               GFUN      60
60              BC(KPDE,4)=1.0                                                   GFUN      61
                IF(DBDU(KPDE).EQ.0.0.AND.DBDUX(KPDE).EQ.0.0)GO TO 60             GFUN      62
                UDOT(KS)=DZDT(KPDE)                                              GFUN      63
                BC(KPDE,3)=A(ILIM)*DBDUX(KPDE)                                   GFUN      64
                BC(KPDE,4)=DBDU(KPDE)-BC(KPDE,3)                                 GFUN      65
65            60 CONTINUE                                                        GFUN      66
                RETURN                                                           GFUN      67
                END                                                              GFUN      68
```

```
      SUBROUTINE EVAL(ICPT,NPDE,C,UVAL,A,ILEFT)                          EVAL      2
C-----------------------------------------------------------------------  EVAL      3
C CALLING ARGUMENTS ARE DEFINED BELOW AND IN PDECOL.                       EVAL      4
C                                                                          EVAL      5
C SUBROUTINE EVAL EVALUATES U(K), UX(K), AND UXX(K), K=1 TO NPDE,          EVAL      6
C AT THE COLLOCATION POINT WITH INDEX ICPT USING THE VALUES OF             EVAL      7
C THE BASIS FUNCTION COEFFICIENTS IN C AND THE BASIS FUNCTION VALUES        EVAL      8
C STORED IN A.  THE RESULTS ARE STORED IN UVAL AS FOLLOWS..                EVAL      9
C UVAL(K,1) = U(K), UVAL(K,2) = UX(K), AND UVAL(K,3) = UXX(K).             EVAL     10
C                                                                          EVAL     11
C PACKAGE ROUTINES CALLED..  NONE                                          EVAL     12
C USER ROUTINES CALLED..     NONE                                          EVAL     13
C CALLED BY..                GFUN,PDECOL,PSETIN                            EVAL     14
C FORTRAN FUNCTIONS USED..   NONE                                          EVAL     15
C-----------------------------------------------------------------------  EVAL     16
      DIMENSION C(1),UVAL(NPDE,3),A(1),ILEFT(1)                            EVAL     17
      COMMON/SIZES/NINT,KORD,IDUM(2),NCPTS,NEQN,IQUAD                      EVAL     18
      IK = ILEFT(ICPT) - KORD                                              EVAL     19
      IC = 3*KORD*(ICPT-1)                                                 EVAL     20
      DO 10 M=1,3                                                          EVAL     21
        ICC = IC + KORD*(M-1)                                              EVAL     22
        DO 10 J=1,NPDE                                                     EVAL     23
        JC=(J-1)*NCPTS                                                     EVAL     24
          UVAL(J,M) = 0.                                                   EVAL     25
          DO 10 I=1,KORD                                                   EVAL     26
      UVAL(J,M)=UVAL(J,M)+C(JC+I+IK)*A(I+ICC)                              EVAL     27
   10 CONTINUE                                                             EVAL     28
      RETURN                                                               EVAL     29
      END                                                                  EVAL     30
```

```
      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..                STIFIH
C FORTRAN FUNCTIONS USED..   NONE
C-----------------------------------------------------------------------
      DIMENSION Y(N),YDOT(N),PW(1),IPIV(1),WORK(1),IWORK(1)
      COMMON/GEAR9/EPSJ,RU,ML,MU,MW,NM1,NOML,NOW,NOW1
      COMMON /SIZES/ NINT,KORD,NCC,NPDE,NCPTS,NEQN,IQUAD
      COMMON /ISTART/ IW1,IW2,IW3,IQUAD(5),IW9,IW10,IW11,IW12,IW13,IW14,
     *                IW15,IW16,IW17,IW18
      DO 10 I = 1,NOW
   10   PW(I) = 0.
      DO 20 K=1,NPDE
      CALL GFUN(T,Y,YDOT,NPDE,NCPTS,WORK(IW1),WORK,WORK(IW14),
     *     WORK(IW15),WORK(IW16),WORK(IW3),WORK(IW9),IWORK,K,K,K)
      NST=(K-1)*NOW1+1
      IST=(K-1)*NCPTS+1
      CALL ADDA(PW(NST),NCPTS,WORK(IW1),IWORK,WORK,NPDE,K)
      CALL DECB(NCPTS,NCPTS,ML,MU,PW(NST),IPIV(IST),IER)
      IF ( IER .NE. 0 ) RETURN
      CALL SOLB(NCPTS,NCPTS,ML,MU,PW(NST),YDOT(IST),IPIV(IST))
   20 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ADDA(PW,NO,A,ILEFT,BC,NPDE,KPDE)                          ADDA      2
C-----------------------------------------------------------------------   ADDA      3
C CALLING ARGUMENTS ARE DEFINED BELOW AND IN PDECOL AND STIFIB.            ADDA      4
C                                                                          ADDA      5
C SUBROUTINE ADDA ADDS THE MATRIX  A  TO THE MATRIX STORED IN  PW  IN      ADDA      6
C BAND FORM.  PW IS STORED BY DIAGONALS WITH THE LOWERMOST DIAGONAL        ADDA      7
C STORED IN THE FIRST COLUMN OF THE ARRAY.                                 ADDA      8
CC  ONLY THE PART OF PW CORRESPONDING TO THE KPDE*TH PDE IS                ADDA      9
CC  WORKED WITH.                                                           ADDA     10
C                                                                          ADDA     11
C PACKAGE ROUTINES CALLED..  NONE                                          ADDA     12
C USER ROUTINES CALLED..     NONE                                          ADDA     13
C CALLED BY..                DIFFUN,PSETIB                                 ADDA     14
C FORTRAN FUNCTIONS USED..   NONE                                          ADDA     15
C-----------------------------------------------------------------------   ADDA     16
      DIMENSION PW(NO,1),A(1),ILEFT(1),BC(NPDE,4)                          ADDA     17
      COMMON /SIZES/ NINT,KORD,NCC,NPDE,NCPTS,NEQN,IQUAD                   ADDA     18
C-----------------------------------------------------------------------   ADDA     19
C ADD THE BOUNDARY CONDITION PORTIONS OF THE A MATRIX TO PW ( THE FIRST     ADDA     20
CC  AND LAST ROWS).                                                        ADDA     21
C-----------------------------------------------------------------------   ADDA     22
      ICOL=ILEFT(1)*IQUAD-1                                                ADDA     23
      PW(1,ICOL)=PW(1,ICOL)+BC(KPDE,1)                                     ADDA     24
      PW(1,ICOL+1)=PW(1,ICOL+1)+BC(KPDE,2)                                 ADDA     25
      PW(NCPTS,ICOL-1)=PW(NCPTS,ICOL-1)+BC(KPDE,3)                         ADDA     26
      PW(NCPTS,ICOL)=PW(NCPTS,ICOL)+BC(KPDE,4)                             ADDA     27
C-----------------------------------------------------------------------   ADDA     28
C UPDATE THE REMAINING ROWS OF PW BY ADDING THE APPROPRIATE VALUES         ADDA     29
C IN A TO PW.                                                              ADDA     30
C-----------------------------------------------------------------------   ADDA     31
      IND = NCPTS - 1                                                      ADDA     32
      DO 20 I=2,IND                                                        ADDA     33
        I2 = (I-1) * KORD * 3                                              ADDA     34
        ICOL = ILEFT(I) - 1 * IQUAD - 1                                    ADDA     35
        DO 20 J=1,KORD                                                     ADDA     36
      J1=ICOL+J                                                            ADDA     37
          J2 = I2 + J                                                      ADDA     38
   20 PW(I,J1)=PW(I,J1)+A(J2)                                              ADDA     39
      RETURN                                                               ADDA     40
      END                                                                  ADDA     41
```

```
      SUBROUTINE RES(T,H,C,V,R,NPDE,NCPTS,A,ILEFT,BC,DBDU,DBDUX,DZDT,
     *     XC,UVAL,K)
C-----------------------------------------------------------------------
C CALLING ARGUMENTS ARE DEFINED BELOW AND IN PDECOL.
C
C SUBROUTINE RES COMPUTES THE RESIDUAL VECTOR R = H*G(C,T) - A(C,T)*V
C WHERE H IS THE CURRENT TIME STEP SIZE, G IS A VECTOR, A IS A
C MATRIX, V IS A VECTOR, AND T IS THE CURRENT TIME.
CC  ONLY THE PART OF R CORRESPONDING TO THE K-TH PDE IS CALCULATED
CC  IN A SINGLE CALL TO RES.
C
C PACKAGE ROUTINES CALLED..  GFUN
C USER ROUTINES CALLED..     NONE
C CALLED BY..                STIFIB
C FORTRAN FUNCTIONS USED..   NONE
C-----------------------------------------------------------------------
      DIMENSION C(1),R(1),V(1)
      DIMENSION A(1),ILEFT(1),BC(NPDE,4),XC(1),UVAL(1)
      DIMENSION DBDU(NPDE),DBDUX(NPDE),DZDT(NPDE)
      COMMON /SIZES/ NINT,KORD,NCC,IDUM(3),IOUAD
C-----------------------------------------------------------------------
C FORM G(C,T) AND STORE IN R.
C-----------------------------------------------------------------------
      CALL GFUN(T,C,R,NPDE,NCPTS,A,BC,DBDU,DBDUX,DZDT,XC,UVAL,
     *     ILEFT,K,K,K)
C-----------------------------------------------------------------------
C FORM THE FIRST AND LAST BLOCK ROWS OF THE RESIDUAL VECTOR
C WHICH ARE DEPENDENT ON THE BOUNDARY CONDITIONS.
C-----------------------------------------------------------------------
      ILIM = NCPTS - 1
      IST=(K-1)*NCPTS+1
      IEND=K*NCPTS
      SUM1=BC(K,1)*V(IST)+BC(K,2)*V(IST+1)
      SUM2=BC(K,3)*V(IEND-1)+BC(K,4)*V(IEND)
      R(IST)=H*R(IST)-SUM1
      R(IEND)=H*R(IEND)-SUM2
C-----------------------------------------------------------------------
C FORM THE REMAINING COMPONENTS OF THE RESIDUAL VECTOR.
C-----------------------------------------------------------------------
      DO 50 ICPTS=2,ILIM
        I2 = (ICPTS-1) * KORD * 3
        ICOL = ILEFT(ICPTS) - KORD
          SUM1 = 0.
          DO 30 J=1,KORD
      SUM1=SUM1+A(I2+J)*V(IST-1+ICOL+J)
   30       CONTINUE
        R(IST-1+ICPTS)=H*R(IST-1+ICPTS)-SUM1
   40   CONTINUE
   50 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PSETIB (C, PW, NO, CO , MITER, IER, A, ILEFT, XC, UVAL,  PSETIB      2
     *    SAVE2, IPIV, CMAX, DFDU, DFDUX, DFDUXX, DZDT, DBDU, DBDUX, RC, NPDE)  PSETIB      3
C-----------------------------------------------------------------------  PSETIB      4
C CALLING ARGUMENTS ARE DEFINED BELOW AND IN PDECOL AND STIFIB.           PSETIB      5
C                                                                         PSETIB      6
C PSETIB IS CALLED BY STIFIB TO COMPUTE AND PROCESS THE MATRIX            PSETIB      7
C PW = A - H*EL(1)*(DG/DC), WHERE A AND DG/DC ARE TREATED IN BAND         PSETIB      8
C FORM.  DG/DC IS COMPUTED, EITHER WITH THE AID OF THE USER-SUPPLIED      PSETIB      9
C ROUTINE DERIVF IF MITER = 1, OR BY FINITE DIFFERENCING WITH THE AID     PSETIB     10
C OF THE PACKAGE-SUPPLIED ROUTINE DIFFF IF MITER = 2.  FINALLY,           PSETIB     11
C PW IS SUBJECTED TO LU DECOMPOSITION IN PREPARATION FOR LATER            PSETIB     12
C SOLUTION OF LINEAR SYSTEMS WITH PW AS COEFFICIENT MATRIX.               PSETIB     13
C SEE SUBROUTINES DECH AND SOLH.                                          PSETIB     14
CC  FOR THE FLAME CODE, MITER ALWAYS EQUALS 2.                            PSETIB     15
C                                                                         PSETIB     16
C IN ADDITION TO VARIABLES DESCRIBED PREVIOUSLY, COMMUNICATION            PSETIB     17
C WITH PSETIB USES THE FOLLOWING..                                        PSETIB     18
C   EPSJ    = SQRT(UROUND), USED IN THE NUMERICAL JACOBIAN INCREMENTS.    PSETIB     19
C   MW      = ML + MU + 1.                                                PSETIB     20
C   NM1     = NO - 1.                                                     PSETIB     21
C   NOML    = NO*ML.                                                      PSETIB     22
C   NOW     = NO*MW.                                                      PSETIB     23
C                                                                         PSETIB     24
C PACKAGE ROUTINES CALLED..  ADDA, DECH, DIFFF, EVAL, GFUN                PSETIB     25
C USER ROUTINES CALLED..     BNDRY, DERIVF                                PSETIB     26
C CALLED BY..                STIFIB                                       PSETIB     27
C FORTRAN FUNCTIONS USED..   ABS, FLOAT, MAX0, MIN0, SQRT                  PSETIB     28
C-----------------------------------------------------------------------  PSETIB     29
      DIMENSION PW(1), C(1), CMAX(1)                                      PSETIB     30
      DIMENSION A(1), ILEFT(1), BC(1), AC(1), UVAL(NPDE,3), SAVE2(1), IPIV(1)  PSETIB  31
      DIMENSION DFDU(NPDE), DFDUX(NPDE), DFDUXX(NPDE)                     PSETIB     32
      DIMENSION DZDT(NPDE), DBDU(NPDE), DBDUX(NPDE)                       PSETIB     33
      COMMON /SIZES/ NINT, KORD, NCC, NPDE, NCPTS, NEQN, IQUAD            PSETIB     34
      COMMON /GEAR1/ T, H, DUMMY(3), UROUND, N, IDUMMY(3)                 PSETIB     35
      COMMON/GEAR9/EPSJ, RO, ML, MU, MW, NM1, NOML, NOW, NOW1             PSETIB     36
      DO 10 I=1,NOW                                                       PSETIB     37
10    PW(I)=0.0                                                           PSETIB     38
      D=0.0                                                               PSETIB     39
      DO 100 K=1,NPDE                                                     PSETIB     40
      IST=(K-1)*NCPTS+1                                                   PSETIB     41
      IEND=K*NCPTS                                                        PSETIB     42
      CALL GFUN(T,C,SAVE2,NPDE,NCPTS,A,BC,DBDU,DBDUX,DZDT,XC,UVAL,        PSETIB     43
     *    ILEFT,K,K,K)                                                    PSETIB     44
      DO 20 I=IST,IEND                                                    PSETIB     45
20      D=D+SAVE2(I)**2                                                   PSETIB     46
100   CONTINUE                                                            PSETIB     47
      RO=ABS(H)*SQRT(D/FLOAT(NO))*1.0E+03*UROUND                          PSETIB     48
      DO 30 I=1,NCPTS                                                     PSETIB     49
        I2 = (I-1)*KORD*3                                                 PSETIB     50
        CALL EVAL(I,NPDE,C,UVAL,A,ILEFT)                                  PSETIB     51
      CALL DIFFF(T,XC(I),I,UVAL,UVAL(1,2),UVAL(1,3),                      PSETIB     52
     *    DFDU,DFDUX,DFDUXX,NPDE,CMAX,SAVE2)                             PSETIB     53
        ICOL = ILEFT(I) - I + IQUAD - 1                                   PSETIB     54
        KLOW = MAX0(1,I+2-NCPTS)                                          PSETIB     55
        KUP = MIN0(KORD,KORD+I-2)                                         PSETIB     56
      DO 30 K=1,NPDE                                                      PSETIB     57
      NST=(K-1)*NOW1+1                                                    PSETIB     58
```

```
      NEND=K*NOW1                                                       PSETIH    59
        DO 30 KBLK=KLOW,KUP                                             PSETIH    60
 60       J1=ICOL+KBLK                                                  PSETIH    61
          J2 = J2 + KBLK                                                PSETIH    62
          J3 = J2 + KOND                                                PSETIH    63
          J4 = J3 + KOND                                                PSETIH    64
      PW(NST-1+I+(J1-1)*NCPTS)=DFDU(K)*A(J2)+DFDUX(K)*A(J3)             PSETIH    65
     *      +DFDUXX(K)*A(J4)                                            PSETIH    66
   30 CONTINUE                                                          PSETIH    67
C-----------------------------------------------------------------------PSETIH    68
C MODIFY THE LAST AND THE FIRST BLOCK ROWS FOR THE BOUNDARY CONDITIONS. PSETIH    69
C CURRENT INFORMATION FOR THE RIGHT BOUNDARY CONDITION IS ALREADY IN    PSETIH    70
C THE ARRAYS DBDU, DBDUX AS A RESULT OF A PREVIOUS CALL TO GFUN.        PSETIH    71
C-----------------------------------------------------------------------PSETIH    72
      DO 50 K=1,NPDE                                                    PSETIH    73
      NST=(K-1)*NOW1+1                                                  PSETIH    74
      NEND=K*NOW1                                                       PSETIH    75
      IF(DBDU(K).EQ.0.0.AND.DBDUX(K).EQ.0.0)GO TO 50                    PSETIH    76
        DO 40 J=1,MW                                                    PSETIH    77
          PW(NST-1+J*NCPTS)=0.0                                         PSETIH    78
   40   CONTINUE                                                        PSETIH    79
   50 CONTINUE                                                          PSETIH    80
      CALL EVAL(1,NPDE,C,UVAL,A,ILEFT)                                  PSETIH    81
      CALL BNDRY(T,XC(1),UVAL,UVAL(1,2),DBDU,DBDUX,DZDT,NPDE)           PSETIH    82
      DO 70 K=1,NPDE                                                    PSETIH    83
      NST=(K-1)*NOW1+1                                                  PSETIH    84
      NEND=K*NOW1                                                       PSETIH    85
      IF(DBDU(K).EQ.0.0.AND.DBDUX(K).EQ.0.0)GO TO 70                    PSETIH    86
      DO 60 J=1,MW                                                      PSETIH    87
      PW(NST+(J-1)*NCPTS)=0.0                                           PSETIH    88
   60   CONTINUE                                                        PSETIH    89
   70 CONTINUE                                                          PSETIH    90
      DO 80 I=1,NOW                                                     PSETIH    91
 80   PW(I)=PW(I)*CON                                                   PSETIH    92
C-----------------------------------------------------------------------PSETIH    93
C ADD MATRIX A(C,T) TO Pw.                                              PSETIH    94
C-----------------------------------------------------------------------PSETIH    95
      DO 90 K=1,NPDE                                                    PSETIH    96
      NST=(K-1)*NOW1+1                                                  PSETIH    97
      NEND=K*NOW1                                                       PSETIH    98
      IST=(K-1)*NCPTS+1                                                 PSETIH    99
      CALL ADDA(PW(NST),NCPTS,A,ILEFT,C,NPDE,K)                         PSETIH   100
C-----------------------------------------------------------------------PSETIH   101
C DO LU DECOMPOSITION ON Pw.                                            PSETIH   102
C-----------------------------------------------------------------------PSETIH   103
      CALL DECH(NCPTS,NCPTS,ML,MU,PW(NST),IPIV(IST),IER)                PSETIH   104
 90   CONTINUE                                                          PSETIH   105
      RETURN                                                            PSETIH   106
      END                                                               PSETIH   107
```

```
      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
CC  STORED, AND RETURNED IN THE NPDE VECTORS DFDU,DFDUX,AND DFDUXX       DIFFF    11
CC  RESPECTIVELY, AT COLLOCATION POINT NUMBER IPT.  AGAIN, PARTIALS      DIFFF    12
CC  OF THE K*TH FUNCTION WITH RESPECT TO THE J*TH VARIABLE, K NOT        DIFFF    13
CC  EQUAL TO J, ARE ASSUMED TO BE NEGLIGABLE.                            DIFFF    14
C                                                                        DIFFF    15
C PACKAGE ROUTINES CALLED..  NONE                                        DIFFF    16
C USER ROUTINES CALLED..     F                                           DIFFF    17
C CALLED BY..                PSETIB                                      DIFFF    18
C FORTRAN FUNCTIONS USED..   AMAX1                                       DIFFF    19
C-----------------------------------------------------------------------DIFFF    20
      DIMENSION U(NPDE),UX(NPDE),UXX(NPDE),DFDU(NPDE)                    DIFFF    21
      DIMENSION DFDUX(NPDE),DFDUXX(NPDE),CMAX(1),SAVE2(1)                DIFFF    22
      COMMON/GEAR9/EPSJ,RO,ML,MU,MW,NM1,NDML,NUW,NUW1                    DIFFF    23
      COMMON/SIZES/NINT,KORD,NCC,NPD,NCPTS,NEQN,IOOAD                    DIFFF    24
      KSKR=2                                                             DIFFF    25
      KSKT=2                                                             DIFFF    26
      DO 40 K=1,NPDE                                                     DIFFF    27
      IF(K.EQ.NPDE)KSKR=1                                                DIFFF    28
      ID=(K-1)*NCPTS+IPT                                                 DIFFF    29
      UK=U(K)                                                            DIFFF    30
      R=EPSJ*CMAX(ID)                                                    DIFFF    31
        R = AMAX1(R,R0)                                                  DIFFF    32
      U(K)=U(K)+R                                                        DIFFF    33
        RINV = 1. / R                                                    DIFFF    34
      CALL F(T,X,U,UX,UXX,DFDU  (K),NPDE,K,IPT,1,KSKR)                   DIFFF    35
      DFDU(K)=(DFDU(K)-SAVE2(ID))*RINV                                   DIFFF    36
      U(K)=UK                                                            DIFFF    37
      UK=UX(K)                                                           DIFFF    38
      UX(K)=UX(K)+R                                                      DIFFF    39
      CALL F(T,X,U,UX,UXX,DFDUX (K),NPDE,K,IPT,1,KSKR)                   DIFFF    40
      DFDUX(K)=(DFDUX(K)-SAVE2(ID))*RINV                                 DIFFF    41
      UX(K)=UK                                                           DIFFF    42
      UK=UXX(K)                                                          DIFFF    43
      UXX(K)=UXX(K)+R                                                    DIFFF    44
      CALL F(T,X,U,UX,UXX,DFDUXX(K),NPDE,K,IPT,1,KSKR)                   DIFFF    45
      DFDUXX(K)=(DFDUXX(K)-SAVE2(ID))*RINV                               DIFFF    46
      UXX(K)=UK                                                          DIFFF    47
   40 CONTINUE                                                           DIFFF    48
      RETURN                                                             DIFFF    49
      END                                                                DIFFF    50
```

```
      SUBROUTINE INTERP (TOUT, Y, NQ, Y0)                                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 Y0.  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                   Y0(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 Y0(NQ),Y(NQ,1)                                             INTERP     19
      COMMON /GEAR1/ T,H,DUMMY(4),N,DUMMY(2),JSTART                        INTERP     20
      DO 10 I = 1,N                                                        INTERP     21
   10   Y0(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     Y0(I) = Y0(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(1)#10) 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.,37.89,53.33,70.08,87.97,106.9,             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   59
C WHERE          K = FACTORIAL(NQ)*(1 + 1/2 + ... + 1/NQ).               COSET   60
C                                                                        COSET   61
C THE ORDER IN WHICH THE GROUPS APPEAR BELOW IS..                        COSET   62
C IMPLICIT ADAMS METHODS OF ORDERS 1 TO 12.                              COSET   63
C BDF METHODS OF ORDERS 1 TO 5.                                          COSET   64
C-------------------------------------------------------------------     COSET   65
  101  EL(1) = 1.0E-00                                                   COSET   66
       GO TO 900                                                         COSET   67
  102  EL(1) = 0.5E-00                                                   COSET   68
       EL(3) = 0.5E-00                                                   COSET   69
       GO TO 900                                                         COSET   70
  103  EL(1) = 4.1666666666667E-01                                       COSET   71
       EL(3) = 0.75E-00                                                  COSET   72
       EL(4) = 1.6666666666667E-01                                       COSET   73
       GO TO 900                                                         COSET   74
  104  EL(1) = 0.375E-00                                                 COSET   75
       EL(3) = 9.1666666666667E-01                                       COSET   76
       EL(4) = 3.3333333333333E-01                                       COSET   77
       EL(5) = 4.1666666666667E-02                                       COSET   78
       GO TO 900                                                         COSET   79
  105  EL(1) = 3.4861111111111E-01                                       COSET   80
       EL(3) = 1.0416666666667E-00                                       COSET   81
       EL(4) = 4.8611111111111E-01                                       COSET   82
       EL(5) = 1.0416666666667E-01                                       COSET   83
       EL(6) = 8.3333333333333E-03                                       COSET   84
       GO TO 900                                                         COSET   85
  106  EL(1) = 3.2986111111111E-01                                       COSET   86
       EL(3) = 1.1416666666667E-00                                       COSET   87
       EL(4) = 0.625E-00                                                 COSET   88
       EL(5) = 1.7708333333333E-01                                       COSET   89
       EL(6) = 0.025E-00                                                 COSET   90
       EL(7) = 1.3888888888889E-03                                       COSET   91
       GO TO 900                                                         COSET   92
  107  EL(1) = 3.1559193121693E-01                                       COSET   93
       EL(3) = 1.225E-00                                                 COSET   94
       EL(4) = 7.5185185185185E-01                                       COSET   95
       EL(5) = 2.5520833333333E-01                                       COSET   96
       EL(6) = 4.8611111111111E-02                                       COSET   97
       EL(7) = 4.8611111111111E-03                                       COSET   98
       EL(8) = 1.9841269841270E-04                                       COSET   99
       GO TO 900                                                         COSET  100
  108  EL(1) = 3.0422453703704E-01                                       COSET  101
       EL(3) = 1.2964285714286E-00                                       COSET  102
       EL(4) = 8.6851851851852E-01                                       COSET  103
       EL(5) = 3.3576388888889E-01                                       COSET  104
       EL(6) = 7.7777777777778E-02                                       COSET  105
       EL(7) = 1.0648148148148E-02                                       COSET  106
       EL(8) = 7.9365079365079E-04                                       COSET  107
       EL(9) = 2.4801587301587E-05                                       COSET  108
       GO TO 900                                                         COSET  109
  109  EL(1) = 2.9486800044092E-01                                       COSET  110
       EL(3) = 1.3589285714286E-00                                       COSET  111
       EL(4) = 9.7655423280423E-01                                       COSET  112
       EL(5) = 0.4171875E-00                                             COSET  113
       EL(6) = 1.1135416666667E-01                                       COSET  114
       EL(7) = 0.01875E-00                                               COSET  115
```

```
                      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
```

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

CARD NR. SEVERITY DETAILS DIAGNOSIS OF PROBLEM

46 1 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
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
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
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
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
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
85        100     IER = NR                                                      DECB  86
                  RETURN                                                        DECB  87
                  END                                                           DECB  88
```

```
      SUBROUTINE SOLB (NDIM, N, ML, MU, B, Y, IPIV)                     SOLB       2
C-----------------------------------------------------------------------SOLB       3
C SUBROUTINES DECB AND SOLB FORM A TWO SUBROUTINE PACKAGE FOR THE        SOLB       4
C DIRECT SOLUTION OF A SYSTEM OF LINEAR EQUATIONS IN WHICH THE           SOLB       5
C COEFFICIENT MATRIX IS REAL AND BANDED.                                 SOLB       6
C                                                                        SOLB       7
C    SOLUTION OF  A*X = C  GIVEN LU DECOMPOSITION OF A FROM DECB.        SOLB       8
C    Y  =  RIGHT-HAND VECTOR C, OF LENGTH N, ON INPUT,                   SOLB       9
C       =  SOLUTION VECTOR X ON OUTPUT.                                  SOLB      10
C    ALL THE ARGUMENTS ARE INPUT ARGUMENTS.                              SOLB      11
C    THE OUTPUT ARGUMENT IS  Y.                                          SOLB      12
C                                                                        SOLB      13
C PACKAGE ROUTINES CALLED..  NONE                                        SOLB      14
C USER ROUTINES CALLED..     NONE                                        SOLB      15
C CALLED BY..                DIFFUN,INITAL,STIFIB                        SOLB      16
C FORTRAN FUNCTIONS USED..   MIN0                                        SOLB      17
C-----------------------------------------------------------------------SOLB      18
      DIMENSION B(NDIM,1),Y(N),IPIV(N)                                   SOLB      19
      IF (N .EQ. 1) GO TO 60                                             SOLB      20
      N1 = N - 1                                                         SOLB      21
      LL = ML + MU + 1                                                   SOLB      22
      IF (ML .EQ. 0) GO TO 32                                            SOLB      23
      DO 30 NR = 1,N1                                                    SOLB      24
        IF (IPIV(NR) .EQ. NR) GO TO 10                                   SOLB      25
        J = IPIV(NR)                                                     SOLB      26
        XX = Y(NR)                                                       SOLB      27
        Y(NR) = Y(J)                                                     SOLB      28
        Y(J) = XX                                                        SOLB      29
10      KK = MIN0(N-NR,ML)                                               SOLB      30
        DO 20 I = 1,KK                                                   SOLB      31
20        Y(NR+I) = Y(NR+I) + Y(NR)*B(NR,LL+1)                           SOLB      32
30      CONTINUE                                                         SOLB      33
32    LL = LL - 1                                                        SOLB      34
      Y(N) = Y(N)*B(N,1)                                                 SOLB      35
      KK = 0                                                             SOLB      36
      DO 50 NB = 1,N1                                                    SOLB      37
        NR = N - NB                                                      SOLB      38
        IF (KK .NE. LL) KK = KK + 1                                      SOLB      39
        DP = 0.                                                          SOLB      40
        IF (LL .EQ. 0) GO TO 50                                          SOLB      41
        DO 40 I = 1,KK                                                   SOLB      42
40        DP = DP + B(NR,I+1)*Y(NR+I)                                    SOLB      43
50      Y(NR) = (Y(NR) - DP)*B(NR,1)                                     SOLB      44
      RETURN                                                             SOLB      45
60    Y(1) = Y(1)*B(1,1)                                                 SOLB      46
      RETURN                                                             SOLB      47
      END                                                                SOLB      48
```

```
***  03/10/80  SCOPE 2.1.5  H R L  VER 00H  ***    04/15/80       80106

SYS DEVICES 819/ 4/MF FLS=377K FLL=1750K MXS=300K MXL=1305K MXH=1305K

HH.MM.SS CPU SECOND ORIGIN
          11.09.36.MFA. A2    HRL NOS/HE 1.3 L499 VER 004     03/17/80
12.47.45 00000.005 MFZ.      -TERRY.STMFZ.T200.P1.      COFFEE      FLAME      PLFLSC.CY=5.
12.47.45 00000.005 JOB.      -ACCOUNT.PD***.
12.47.46 00000.026 JOB.      -MAP(OFF)
12.47.46 00000.028 JOB.      -ATTACH.TAPE11.INH250AIRW30.ID=SHLJMH.
12.47.46 00000.032 MFZ.         MF254 - CYCLE     2 ATTACHED FROM SN=SYSTEM
12.47.46 00000.032 JOB.      -ATTACH.RTSC.BH20ZRTW30N2L.ID=SHLJMH.
12.47.46 00000.036 MFZ.         MF254 - CYCLE     1 ATTACHED FROM SN=SYSTEM
12.47.46 00000.037 JOB.      -REQUEST.TAPE2.*PF.
12.47.46 00000.039 JOB.      -REQUEST.TAPE9.*PF.
12.47.46 00000.040 JOB.      -ATTACH.COLD.HFLSC.ID=SHLJMH.
12.47.46 00000.044 MFZ.         MF254 - CYCLE     5 ATTACHED FROM SN=SYSTEM
12.47.46 00000.045 JOB.      -SLOAD(COLD,MAIN)
12.47.46 00000.048 JOB.      -LOAD(RTSC)
12.47.46 00000.048 JOB.      -SLOAD(COLD,F)
12.47.46 00000.049 JOB.      -SLOAD(COLD,UINIT)
12.47.46 00000.050 JOB.      -SLOAD(COLD,FLSP)
12.47.46 00000.050 JOB.      -SLOAD(COLD,BNDRY)
12.47.46 00000.051 JOB.      -SLOAD(COLD,BKPT)
12.47.46 00000.051 JOB.      -SLOAD(COLD,PUECOL)
12.47.46 00000.052 JOB.      -SLOAD(COLD,STIFIB)
12.47.46 00000.052 JOB.      -SLOAD(COLD,INITAL)
12.47.46 00000.053 JOB.      -SLOAD(COLD,GFUN)
12.47.46 00000.053 JOB.      -SLOAD(COLD,VALUES)
12.47.46 00000.054 JOB.      -SLOAD(COLD,COLPNT)
12.47.46 00000.054 JOB.      -SLOAD(COLD,BSPLVU)
12.47.46 00000.055 JOB.      -SLOAD(COLD,BSPLVN)
12.47.46 00000.055 JOB.      -SLOAD(COLD,INTERV)
12.47.46 00000.056 JOB.      -SLOAD(COLD,EVAL)
12.47.46 00000.056 JOB.      -SLOAD(COLD,DIFFUN)
12.47.47 00000.057 JOB.      -SLOAD(COLD,ADDA)
12.47.47 00000.057 JOB.      -SLOAD(COLD,RES)
12.47.47 00000.058 JOB.      -SLOAD(COLD,PSETIB)
12.47.47 00000.059 JOB.      -SLOAD(COLD,DIFFF)
12.47.47 00000.059 JOB.      -SLOAD(COLD,INTERP)
12.47.47 00000.060 JOB.      -SLOAD(COLD,CUSET)
12.47.47 00000.060 JOB.      -SLOAD(COLD,DECH)
12.47.47 00000.061 JOB.      -SLOAD(COLD,SOLH)
12.47.47 00000.061 JOB.      -EXECUTE.
12.47.53 00000.440 MFZ.         LD610 - FLS REQUIRED TO LOAD - 0015323 00.COG
12.47.54 00000.442 MFZ.         LD603 - EXECUTION INITIATED OS.EXP
12.47.54 00000.442 USR.       FORTRAN LIBRARY 498          01/03/80
12.50.20 00034.849 USR.          STOP
12.50.20 00034.849 USR.          34.406 CP SECONDS EXECUTION TIME
12.50.20 00034.854 JOB.      -CATALOG.TAPE2.H250AIRW30CNH12.ID=SHLJMH.
12.50.21 00034.858 MFZ.         PF060 - CYCLE     3 CATALOGED ON SN=SYSTEM
12.50.21 00034.859 JOB.      -CATALOG.TAPE9.H250AIRW30CNH12OUT.ID=SHLJMH.
12.50.21 00034.863 MFZ.         PF060 - CYCLE     3 CATALOGED ON SN=SYSTEM
12.50.21 00034.867 MFZ.         JM166 - MAXIMUM USER SCM          111100B WORDS
12.50.21 00034.867 MFZ.         JM167 - MAXIMUM USER LCM               0B WORDS
12.50.21 00034.867 MFZ.         JM170 - MAXIMUM JS*IO LCM          173B BUFFERS
12.50.21 00034.867 MFZ.         RM770 - MAXIMUM ACTIVE FILES                 3
12.50.21 00034.867 MFZ.         RM771 - OPEN/CLOSE CALLS                    12
12.50.21 00034.867 MFZ.         RM772 - DATA TRANSFER CALLS              1,466
12.50.21 00034.868 MFZ.         RM773 - CONTROL/POSITIONING CALLS            4
12.50.21 00034.868 MFZ.         RM774 - RM DATA TRANSFER CALLS             664
12.50.21 00034.868 MFZ.         RM775 - RM CONTROL/POSITIONING CALLS       121
12.50.21 00034.868 MFZ.         RM776 - RMFM MANAGER CALLS                 153
```

```
12.50.21 00034.868 MFZ.        RM777 - RECALL CALLS                              124
12.50.21 00034.869 MFZ.        SCM                1 317.071  KWS
12.50.21 00034.869 MFZ.        LCM                  317.166  KWS
12.50.21 00034.869 MFZ.        I/O                    0.017  MW
12.50.21 00034.869 MFZ.        RMS                    0.285  MWS
12.50.21 00034.869 MFZ.        USER                  34.295  SEC
12.50.21 00034.870 MFZ.        JOB                   34.872  SEC
12.50.21 00034.870 MFZ.        DIO                  311.314  K#
12.50.21 00034.870 MFZ.        SS                    45.793  SEC
12.50.21 00034.870 MFZ.        COST ESTIMATE         66.99
12.50.21 00034.870 MFZ.        SC050 - 000015 SC/LC SWAPS
```

H2-AIR FLAME. 50 PERCENT H2. WARNATZ KINETICS.

PRESSURE = 1.0000E+00 ATM.

NPDE = 9

		W	UC	UH	HO	H2O
1	H	1.0000E+00	6.4851E-14	5.2691E-05	5.2103E+04	4.5434E-07
2	OH	1.7000E+01	1.1025E-10	5.2566E-05	5.5829E+02	1.3530E-07
3	O	1.6000E+01	1.0376E-10	1.0468E-07	3.7221E+03	1.3746E-07
4	HO2	3.3000E+01	2.1401E-10	1.9533E-11	1.5146E+02	9.0984E-08
5	H2O2	3.4000E+01	2.2049E-10	1.3187E-10	-9.5683E+02	9.0471E-08
6	H2	2.0000E+00	6.4851E-02	3.7564E-02	-1.5951E-01	2.3005E-07
7	O2	3.2000E+01	2.1790E-01	4.3495E-08	-3.5689E-02	9.4245E-08
8	H2O	1.8000E+01	1.1673E-08	2.4508E-01	-3.2110E+03	1.1838E-07
9	N2	2.8000E+01	7.1725E-01	7.1725E-01	-3.9514E-02	1.3902E-07

TC = 2.9800E-01 TH = 1.9366E+00

TPN = 1.0000E+03 PMN = 5.0000E-05 TMN = 1.0000E-03

CPMX = 4.7856E-01 RL = 7.8440E-08 TPERT = 2.9800E+02

PHO = 0. PHS = 6.0000E-01 TFINAL = 5.0000E-01

EPS = 1.0000E-03 SMEC = 1.0000E-06

NSTART = 1 NINT = 12 NMONI = 0 PLUSP = 0.

NTMAX = 1 TPRINT = 5.0000E-01

TMAX = 1.0000E+02

NCN = 4 FL = 4.0000E+00

INITIAL DENSITY = 6.3065E-04

0.	9.8018E-02	1.6733E-01	2.1634E-01	2.5099E-01	2.7550E-01	3.0000E-01	3.2450E-01	3.4901E-01	3.8366E-01
4.3267E-01	5.0198E-01	6.0000E-01							
0.	1.2252E-02	2.4505E-02	3.6757E-02	4.9009E-02	6.1261E-02	7.3514E-02	8.5766E-02	9.8018E-02	1.0668E-01
1.1535E-01	1.2401E-01	1.3267E-01	1.4134E-01	1.5000E-01	1.5866E-01	1.6733E-01	1.7345E-01	1.7958E-01	1.8571E-01
1.9183E-01	1.9796E-01	2.0408E-01	2.1021E-01	2.1634E-01	2.2067E-01	2.2500E-01	2.2933E-01	2.3366E-01	2.3800E-01
2.4233E-01	2.4666E-01	2.5099E-01	2.5405E-01	2.5712E-01	2.6018E-01	2.6324E-01	2.6631E-01	2.6937E-01	2.7243E-01
2.7550E-01	2.7856E-01	2.8162E-01	2.8469E-01	2.8775E-01	2.9081E-01	2.9387E-01	2.9694E-01	3.0000E-01	3.0306E-01
3.0613E-01	3.0919E-01	3.1225E-01	3.1532E-01	3.1838E-01	3.2144E-01	3.2450E-01	3.2757E-01	3.3063E-01	3.3369E-01
3.3676E-01	3.3982E-01	3.4288E-01	3.4595E-01	3.4901E-01	3.5334E-01	3.5767E-01	3.6200E-01	3.6634E-01	3.7067E-01
3.7500E-01	3.7933E-01	3.8366E-01	3.8979E-01	3.9592E-01	4.0204E-01	4.0817E-01	4.1429E-01	4.2042E-01	4.2655E-01
4.3267E-01	4.4134E-01	4.5000E-01	4.5866E-01	4.6733E-01	4.7599E-01	4.8465E-01	4.9332E-01	5.0198E-01	5.1423E-01
5.2649E-01	5.3874E-01	5.5099E-01	5.6324E-01	5.7550E-01	5.8775E-01	6.0000E-01			

KCEN = 9 VCEN = 9.5343E-01

T = 0. DT = 0. TOTAL STEPS = 0

NF = 0 NJ = 0

RUN TIME = 8.0000E-03

H2O AT LEFT = 9.0867E-03 2.7060E-03 2.7492E-03 1.8197E-03 1.8094E-03 4.6010E-03 1.8849E-03 2.3677E-03 0.

UPH AT LEFT = 0. 0. 0. 0. 0. 0. 0. 0. 1.0000E-30 0.

UPH2 AT LEFT = 2.0680E-25 0. 0. 8.4703E-22 0. 2.2737E-13 0. 2.7105E-20 -2.2737E-13 9.0949E-13

FLAME SPEED = -2.5516E+03 -8.6046E+02 2.2673E+02 -2.6130E+10 -3.7377E+06 1.6050E+02 8.8279E+01 8.2698E+01 6.4080E+09

FLAME SPEED = 7.6276E+02 -2.4724E+04 3.1661E+06 -6.7849E+09 -7.2797E+05 5.3024E+01 2.7720E+01 3.0761E+01 1.8349E+09

FLAME SPEED = 8.3493E+02 -2.5163E+04 4.8471E+06 -5.7309E+09 -5.1461E+05 8.0253E+00 9.6144E+00 8.0431E+00 0.

FLAME FRONT FROM PHI* = 2.5095E-01 TO PHI* = 4.5759E-01

FLAME FRONT FROM X = 2.0886E-02 TO X = 4.8464E-02

FLAME THICKNESS = 2.7577E-02 CM

MO = 5.0724E-03 FLSP = 8.0431E+00

SPEEDO = -1.0145E-01 DS = 0.

T = 5.0000E-03 DT = 2.7511E-04 TOTAL STEPS = 51

```
        RUN TIME =  5.8600E+00

H2O AT LEFT =  9.0867E-03  2.7060E-03  2.7492E-03  1.8197E-03  1.8094E-03  4.6010E-03  1.8849E-03  2.3677E-03  0.
UPH AT LEFT =  1.3411E-09 -1.7184E-09  5.4406E-10  5.1579E-08  6.2165E-09 -4.3168E-06 -3.1783E-05  1.9065E-05  1.6976E-05  7.6777E-05

UPH2 AT LEFT = -1.8652E-07 -5.3094E-07 -9.6195E-08 -3.6054E-05 -8.9577E-07  1.0648E-03  7.2452E-03 -6.4641E-03 -1.8076E-03 -3.8844E-02

FLAME SPEED =  1.7634E+02  1.8839E+02  4.5225E+02  2.8594E+02  7.3178E+02  2.0476E+02  9.6679E+01  1.1624E+02  6.6176E+02
FLAME SPEED =  5.9300E+02  4.6934E+02  7.5362E+02  6.6555E+03  5.3483E+04  7.3938E+01  3.8262E+01  4.3247E+01 -8.1220E+02
FLAME SPEED =  2.6860E+03  1.9864E+03  6.5519E+05 -4.7552E+09 -1.3298E+08  2.0165E+01  1.6072E+01  1.4593E+01 -3.0986E+02
        FLAME FRONT FROM PHI* =  2.4225E-01  TO PHI* =  3.6627E-01

        FLAME FRONT FROM X =  2.0247E-02  TO X =  5.2812E-02

        FLAME THICKNESS =  3.2565E-02  CM

   MU =  3.8372E-02      FLSP =  6.0845E+01      PHNEW =  2.9967E-01

   SPN1 = -7.6744E-01      SPN2 = -1.5648E+00      SPEED = -7.6744E-01      DS = -6.6600E-01

        T =  1.0000E-02    DT =  6.3976E-04    TOTAL STEPS =      62

        NF =      84     NJ =       9

        RUN TIME =  6.9270E+00

H2O AT LEFT =  9.0867E-03  2.7060E-03  2.7492E-03  1.8197E-03  1.8094E-03  4.6010E-03  1.8849E-03  2.3677E-03  0.
UPH AT LEFT =  4.0121E-09  8.8487E-09  5.6945E-09  1.3841E-07  3.1086E-08 -2.6803E-05 -3.7487E-05  6.0286E-05  3.8157E-06  8.3990E-04

UPH2 AT LEFT = -5.5150E-07 -1.8045E-06 -4.8160E-07 -8.2649E-05 -4.5260E-06  4.9272E-03  8.4479E-03 -1.2060E-02 -1.2443E-03 -1.4961E-01

FLAME SPEED =  2.5450E+02  3.5339E+02  4.0492E+02  3.0[illegible]E+02  5.0180E+02  1.7720E+02  9.3857E+01  1.0844E+02  3.5161E+02
FLAME SPEED =  5.4617E+02  4.8131E+02  6.7111E+02  5.4594E+03  8.3531E+04  8.6939E+01  5.1043E+01  5.5297E+01 -4.1936E+02
FLAME SPEED =  5.9564E+03  4.2421E+03  1.2790E+06  5.9301E+08 -2.8225E+08  4.5591E+01  3.5784E+01  3.1471E+01  2.4156E+02
        FLAME FRONT FROM PHI* =  2.3797E-01  TO PHI* =  3.7932E-01

        FLAME FRONT FROM X =  2.0047E-02  TO X =  5.[illegible]E-02
```

FLAME THICKNESS = 3.8379E-02 CM

MO = 2.8700E-02 FLSP = 4.5509E+01 PHNEW = 2.9897E-01

SPN1 = -5.7400E-01 SPN2 = -6.2066E-01 SPEED = -5.7400E-01 DS = 1.9344E-01

T = 1.8548E-02 DT = 7.6771E-04 TOTAL STEPS = 77

NF = 111 NJ = 11

RUN TIME = 9.0630E+00

H2O AT LEFT = 9.0867E-03 2.7060E-03 2.7492E-03 1.8197E-03 1.8094E-03 4.6010E-03 1.8849E-03 2.3677E-03 0.

OPH AT LEFT = 1.8169E-08 6.1181E-08 4.5057E-08 -5.4582E-07 1.7072E-07 -7.1372E-05 -1.0440E-04 1.7900E-04 -2.9720E-06 2.1453E-03

OPH2 AT LEFT = -2.5068E-06 -1.0467E-05 -7.6319E-06 -6.1669E-05 -2.6656E-05 1.1245E-02 1.8853E-02 -3.2268E-02 2.2788E-03 -4.0088E-01

FLAME SPEED = 2.3717E+02 3.1362E+02 3.6576E+02 3.2175E+02 4.7275E+02 1.5612E+02 1.0011E+02 1.0781E+02 2.2795E+02

FLAME SPEED = 4.4120E+02 4.5064E+02 5.6781E+02 5.0474E+03 2.4456E+05 1.2997E+02 9.1541E+01 9.4450E+01 -2.4342E+02

FLAME SPEED = 1.4396E+04 8.8076E+03 1.9968E+06 3.2018E+08 -6.0778E+08 1.2016E+02 9.4871E+01 9.0839E+01 1.5316E+02

FLAME FRONT FROM PHI* = 2.2925E-01 TO PHI* = 4.9590E-01

FLAME FRONT FROM X = 1.9410E-02 TO X = 6.5311E-02

FLAME THICKNESS = 4.5901E-02 CM

MO = 3.7339E-02 FLSP = 5.9206E+01 PHNEW = 2.9832E-01

SPN1 = -7.4677E-01 SPN2 = -1.1806E+00 SPEED = -7.4677E-01 DS = -1.7277E-01

T = 3.1390E-02 DT = 8.0203E-04 TOTAL STEPS = 95

NF = 143 NJ = 13

RUN TIME = 1.1-29E+01

H2O AT LEFT = 9.0867E-03 2.7060E-03 2.7492E-03 1.8197E-03 1.8094E-03 4.6010E-03 1.8849E-03 2.3677E-03 0.

OPH AT LEFT = 1.3634E-07 4.2629E-07 3.6534E-07 -1.2534E-05 2.4439E-06 -3.0510E-04 -2.2908E-04 4.0097E-04 1.4224E-04 5.8526E-03

```
UPH2 AT LEFT = -1.8791E-05 -7.3639E-05 -9.8905E-05  8.7888E-04 -3.7392E-04  1.5602E-02  4.0314E-02 -6.7902E-02  1.1674E-02 -7.3979E-01

FLAME SPEED =  2.2109E+02  3.3001E+02  3.0895E+02  1.9344E+02  5.6891E+02  1.6462E+02  1.3393E+02  1.3715E+02  1.9803E+02
FLAME SPEED =  3.2308E+02  3.7075E+02  5.0901E+02  9.2650E+02  5.7429E+05  1.8783E+02  1.5547E+02  1.5900E+02 -8.3599E+01
FLAME SPEED =  1.3330E+04  6.2534E+03  8.2793E+05  1.5214E+07  1.1597E+09  1.932HE+02  1.6055E+02  1.5731E+02  3.5897E+01
        FLAME FRONT FROM PHI* =  2.1631E-01  TO PHI* =  4.1426E-01

        FLAME FRONT FROM X =  1.8492E-02  TO X =  7.5316E-02

        FLAME THICKNESS =  5.6824E-02  CM

   MO =  7.7035E-02      FLSP =  1.2215E+02      PHNEW =  2.8702E-01

   SPN1 = -1.5407E+00      SPN2 = -5.5790E+00      SPEEDI = -3.5599E+00      DS = -2.8131E+00

        T =  3.9342E-02    DT =  8.0527E-04    TOTAL STEPS =     109

        NF =     169     NJ =      15

        RUN TIME =  1.3591E+01

H2O AT LEFT =  9.0867E-03  2.7060E-03  2.7492E-03  1.8197E-03  1.8094E-03  4.6010E-03  1.8849E-03  2.3677E-03  0.
UPH AT LEFT =  2.5742E-07  3.0457E-06  3.9378E-06 -6.4781E-05  1.0302E-05 -3.8862E-04 -6.5792E-04  9.3984E-04  1.5412E-04 -1.0333E-02

UPH2 AT LEFT = -3.4387E-05 -3.8602E-04 -5.0868E-04  6.1011E-03 -1.2096E-03  1.5902E-02  7.9617E-02 -1.1364E-01  1.4161E-02 -1.0406E+00

FLAME SPEED =  2.2151E+02  2.9369E+02  2.6650E+02  1.0419E+02  4.2863E+02  1.7817E+02  1.5631E+02  1.5835E+02  1.8888E+02
FLAME SPEED =  2.8461E+02  3.3919E+02  4.1628E+02 -1.9614E+03  4.8375E+05  2.0648E+02  1.8109E+02  1.8457E+02 -6.4147E+01
FLAME SPEED =  7.1908E+03  3.1005E+03  2.5715E+05 -6.6600E+07  1.2131E+08  2.1508E+02  1.8278E+02  1.8023E+02  9.6746E+00
        FLAME FRONT FROM PHI* =  2.1631E-01  TO PHI* =  4.3256E-01

        FLAME FRONT FROM X =  1.8463E-02  TO X =  8.1805E-02

        FLAME THICKNESS =  6.3344E-02  CM

   MO =  1.1640E-01      FLSP =  1.8458E+02      PHNEW =  2.8562E-01

   SPN1 = -2.3281E+00      SPN2 = -5.4343E+00      SPEEDI = -3.8412E+00      DS = -1.2111E-01
```

T = 4.6021E-02 DT = 6.3384E-04 TOTAL STEPS = 123

NF = 190 NJ = 17

RUN TIME = 1.5322E+01

H2O AT LEFT = 9.0867E-03 2.7060E-03 2.7492E-03 1.8197E-03 1.8094E-03 4.6010E-03 1.8849E-03 2.3677E-03 0.

UPH AT LEFT = 1.6379E-07 5.9733E-06 7.2189E-06 -1.4871E-04 1.0728E-05 -2.9831E-04 -1.0717E-03 1.2402E-03 2.5444E-04 1.0318E-02

UPH2 AT LEFT = -2.1751E-05 -6.9877E-04 -8.4307E-04 1.4327E-02 -1.0355E-03 1.3772E-02 1.0944E-01 -1.3197E-01 -3.0135E-03 -1.0441E+00

FLAME SPEED = 2.2124E+02 2.6829E+02 2.4199E+02 1.4614E+02 3.1293E+02 1.8574E+02 1.6963E+02 1.6967E+02 1.7904E+02

FLAME SPEED = 2.7012E+02 3.1793E+02 3.3499E+02 -1.0874E+03 2.8065E+05 2.1727E+02 1.9836E+02 2.0087E+02 -1.9702E+02

FLAME SPEED = 3.7140E+03 1.5632E+03 6.7018E+04 -6.9575E+06 2.4318E+07 2.3393E+02 2.0007E+02 1.9788E+02 3.7685E+00

FLAME FRONT FROM PHI* = 2.2497E-01 TO PHI* = 4.5854E-01

FLAME FRONT FROM X = 1.9091E-02 TO X = 8.8814E-02

FLAME THICKNESS = 6.9724E-02 CM

RO = 1.3188E-01 FLSP = 2.0912E+02 PHNEW = 2.9280E-01

SPN1 = -2.6376E+00 SPN2 = -2.8335E+00 SPEED = -2.7356E+00 DS = 1.1459E+00

T = 5.6025E-02 DT = 1.0479E-03 TOTAL STEPS = 139

NF = 209 NJ = 20

RUN TIME = 1.7066E+01

H2O AT LEFT = 9.0867E-03 2.7060E-03 2.7492E-03 1.8197E-03 1.8094E-03 4.6010E-03 1.8849E-03 2.3677E-03 0.

UPH AT LEFT = 9.1874E-08 2.9204E-06 4.5704E-06 -1.3639E-04 1.4728E-05 -2.9250E-04 -1.9684E-04 7.5415E-04 3.7922E-04 6.0855E-03

UPH2 AT LEFT = -1.2387E-05 -3.6947E-04 -5.7917E-04 1.2427E-02 -1.5432E-03 9.1537E-03 7.9139E-02 -8.2614E-02 -1.6102E-02 -6.5492E-01

FLAME SPEED = 2.3534E+02 2.3211E+02 2.3325E+02 1.9400E+02 2.7003E+02 1.9749E+02 1.9006E+02 1.8749E+02 1.8119E+02

FLAME SPEED = 2.7364E+02 2.9605E+02 3.0833E+02 -4.3174E+02 1.4435E+05 2.3272E+02 2.2068E+02 2.2164E+02 6.7612E+02

FLAME SPEED = 4.8871E+06 2.6179E+02 2.6246E+02 2.2271E+02 4.3955E+00

FLAME FRONT FROM PHI* = 2.2918E-01 TO PHI* = 4.9329E-01

FLAME FRONT FROM X = 1.9226E-02 TO X = 9.9404E-02

FLAME THICKNESS = 8.0583E-02 CM

MO = 1.4305E-01 FLSP = 2.2683E+02 PHNEW = 2.972HE-01

SPN1 = -2.8610E+00 SPN2 = -3.3408E+00 SPEED = -2.8610E+00 DS = -1.2543E-01

T = 7.1404E-02 DT = 1.4736E-03 TOTAL STEPS = 15H

NF = 233 NJ = 25

RUN TIME = 1.9431E+01

H2O AT LEFT = 9.0867E-03 2.7060E-03 2.7492E-03 1.6197E-03 1.6094E-03 4.6010E-03 1.8849E-03 2.3677E-03 0.

UPH AT LEFT = 1.2103E-07 2.8517E-06 3.6109E-06 -1.0434E-04 2.2467E-05 -1.9507E-04 -3.6975E-04 3.7517E-04 2.6495E-04 4.277HE-03

UPH2 AT LEFT = -1.6295E-05 -3.7184E-04 -4.8423E-04 1.0496E-02 -2.4414E-03 6.8793E-03 3.9632E-02 -4.5179E-02 -8.5144E-03 -4.6556E-01

FLAME SPEED = 2.5372E+02 2.6206E+02 2.4767E+02 1.7370E+02 2.7371E+02 2.1963E+02 2.2302E+02 2.1924E+02 2.0520E+02

FLAME SPEED = 2.8644E+02 3.0762E+02 3.1457E+02 -1.7588E+03 1.0182E+05 2.5447E+02 2.4909E+02 2.4860E+02 2.4890E+02

FLAME SPEED = 9.3608E+02 5.4921E+02 3.1791E+03 -7.0265E+06 8.1794E+05 2.9265E+02 2.5248E+02 2.5281E+02 -5.0941E+00

FLAME FRONT FROM PHI* = 2.2925E-01 TO PHI* = 5.0176E-01

FLAME FRONT FROM X = 1.9137E-02 TO X = 1.0250E-01

FLAME THICKNESS = 8.3358E-02 CM

MO = 1.5499E-01 FLSP = 2.4575E+02 PHNEW = 2.9264E-01

SPN1 = -3.0997E+00 SPN2 = -4.1240E+00 SPEED = -3.0119E+00 DS = -7.5047E-01

T = 9.0143E-02 DT = 1.4736E-03 TOTAL STEPS = 170

NF = 253 NJ = 25

RUN TIME = 2.0481E+01

```
H2O AT LEFT =  9.0867E-03  2.7060E-03  2.7492E-03  1.8197E-03  1.8094E-03  4.6010E-03  1.8849E-03  2.3677E-03  0.

OPH AT LEFT =  1.3954E-07  5.7160E-06  5.9374E-06 -2.2203E-04  4.2387E-05 -1.3166E-04 -3.5909E-04  3.9561E-04  2.6359E-04  3.9890E-03

OPH2 AT LEFT = -1.8579E-05 -6.8406E-04 -7.1016E-04  2.1099E-02 -4.2580E-03  6.6517E-03  1.6656E-02 -4.3869E-02 -1.5466E-02 -4.5566E-01

FLAME SPEED =  2.7093E+02  2.6658E+02  2.6395E+02  1.8981E+02  2.8215E+02  2.4401E+02  2.5285E+02  2.5002E+02  2.4116E+02

FLAME SPEED =  2.8827E+02  2.8993E+02  3.0330E+02 -2.2988E+03  7.8673E+04  2.6895E+02  2.6902E+02  2.6790E+02  1.8602E+02

FLAME SPEED =  5.6637E+02  4.3163E+02  1.2113E+03 -3.1134E+06  3.5688E+05  3.0557E+02  2.7084E+02  2.7380E+02 -6.0668E-01

        FLAME FRONT FROM PHI* =  2.2924E-01  TO PHI* =  5.0191E-01

        FLAME FRONT FROM X =  1.8953E-02  TO X =  1.0211E-01

        FLAME THICKNESS =  8.3159E-02  CM

   MO =  1.6643E-01      FLSP =  2.6390E+02      PHNEW =  2.9091E-01

   SPN1 = -3.3286E+00      SPN2 = -3.9057E+00      SPEED = -3.5172E+00      DS = -5.2937E-03

        T =  1.1127E-01    DT =  2.4007E-03    TOTAL STEPS =     189

        NF =     280     NJ =      28

        RUN TIME =  2.3101E+01

H2O AT LEFT =  9.0867E-03  2.7060E-03  2.7492E-03  1.8197E-03  1.8094E-03  4.6010E-03  1.8849E-03  2.3677E-03  0.

OPH AT LEFT =  9.7208E-08  4.2032E-06  5.3071E-06 -2.0362E-04  6.9178E-05 -7.5909E-05 -3.5980E-04  3.0913E-04  2.5142E-04  2.7671E-03

OPH2 AT LEFT = -1.2927E-05 -4.9446E-04 -6.2235E-04  1.8559E-02 -5.6652E-03  5.4452E-03  1.5077E-02 -3.3260E-02 -1.8026E-02 -3.3763E-01

FLAME SPEED =  2.9237E+02  2.2559E+02  2.6960E+02  2.5644E+02  2.9052E+02  2.6248E+02  2.6983E+02  2.6945E+02  2.7198E+02

FLAME SPEED =  2.8487E+02  2.5857E+02  2.8796E+02 -2.5412E+02  5.9917E+04  2.7553E+02  2.7765E+02  2.7586E+02  2.1981E+02

FLAME SPEED =  3.9628E+02  3.7477E+02  5.4246E+02 -3.9917E+05  2.5455E+05  2.9994E+02  2.7839E+02  2.8174E+02 -3.9135E-01

        FLAME FRONT FROM PHI* =  2.3360E-01  TO PHI* =  4.9202E-01

        FLAME FRONT FROM X =  1.9196E-02  TO X =  6.4560E-02

        FLAME THICKNESS =  4.5364E-02  CM
```

```
 MO =  1.732BE-01      FLSP =  2.7476E+02      PHNE# =  2.9406E-01

 SPN1 = -3.4656E+00      SPN2 = -3.8352E+00      SPEE... = -3.4656E+00      DS =  1.5157E-01

      T =  1.3407E-01    DT =  2.4007E-03    TOTAL STEPS =     199

      NF =     297     NJ =      28

      RUN TIME =  2.4287E+01

H2O AT LEFT =  9.0867E-03  2.7060E-03  2.7492E-03  1.8197E-03  1.8094E-03  4.6010E-03  1.8849E-03  2.3677E-03  0.
UPH AT LEFT =  8.7589E-08  3.1231E-06  4.1576E-06 -1.3019E-04  6.8511E-05 -5.2228E-05 -2.8264E-04  2.1760E-04  1.7158E-04  2.0362E-03

UPH2 AT LEFT = -1.1677E-05 -3.7422E-04 -4.9985E-04  1.1743E-02 -6.5888E-03  4.6128E-03  2.7916E-02 -2.4007E-02 -1.3040E-02 -2.6151E-01

FLAME SPEED =  3.0079E+02  2.1841E+02  2.7280E+02  2.6717E+02  2.9434E+02  2.7352E+02  2.7621E+02  2.7790E+02  2.8322E+02
FLAME SPEED =  2.8263E+02  2.5003E+02  2.8267E+02  5.0204E+01  5.0289E+04  2.7833E+02  2.7999E+02  2.7976E+02  2.6258E+02
FLAME SPEED =  3.2633E+02  3.1574E+02  3.0523E+02 -1.0908E+05  1.8487E+05  2.8950E+02  2.8019E+02  2.8173E+02 -5.8120E-01
      FLAME FRONT FROM PHI* =  2.3354E-01  TO PHI* =  3.8365E-01

      FLAME FRONT FROM X =  1.9115E-02  TO X =  5.7870E-02

      FLAME THICKNESS =  3.8754E-02  CM

 MO =  1.7602E-01      FLSP =  2.7912E+02      PHNE# =  2.9453E-01

 SPN1 = -3.5205E+00      SPN2 = -3.8851E+00      SPEE... = -3.5205E+00      DS = -5.4910E-02

      T =  1.6937E-01    DT =  2.4007E-03    TOTAL STEPS =     213

      NF =     321     NJ =      28

      RUN TIME =  2.5940E+01

H2O AT LEFT =  9.0867E-03  2.7060E-03  2.7492E-03  1.8197E-03  1.8094E-03  4.6010E-03  1.8849E-03  2.3677E-03  0.
UPH AT LEFT =  1.0117E-07  3.7514E-06  4.4957E-06 -1.5207E-04  7.4044E-05 -4.3936E-05 -2.7749E-04  2.2419E-04  1.8731E-04  2.0479E-03

UPH2 AT LEFT = -1.3478E-05 -4.4793E-04 -5.3465E-04  1.4115E-02 -7.1567E-03  4.6134E-03  2.7800E-02 -2.4783E-02 -1.3410E-02 -2.6520E-01
```

FLAME SPEED = 2.9724E+02 2.2471E+02 2.7451E+02 2.5922E+02 2.9639E+02 2.7920E+02 2.7795E+02 2.8046E+02 2.8295E+02

FLAME SPEED = 2.8156E+02 2.4841E+02 2.8111E+02 -2.8220E+02 5.1520E+04 2.7981E+02 2.8036E+02 2.8046E+02 2.8334E+02

FLAME SPEED = 2.9299E+02 2.6262E+02 2.1870E+02 -1.9914E+05 1.4745E+05 2.8100E+02 2.8040E+02 2.8023E+02 2.7016E+00

FLAME FRONT FROM PHI* = 2.3358E-01 TO PHI* = 3.9590E-01

FLAME FRONT FROM X = 1.9145E-02 TO X = 6.2759E-02

FLAME THICKNESS = 4.3614E-02 CM

MO = 1.7690E-01 FLSP = 2.8050E+02 PHNE# = 2.9295E-01

SPN1 = -3.5380E+00 SPN2 = -3.7295E+00 SPEED = -3.6337E+00 DS = -1.1321E-01

T = 2.5039E-01 DT = 7.3361E-03 TOTAL STEPS = 231

NF = 346 NJ = 31

RUN TIME = 2.8076E+01

H2O AT LEFT = 9.0867E-03 2.7060E-03 2.7492E-03 1.8197E-03 1.8094E-03 4.6010E-03 1.8849E-03 2.3677E-03 0.

UPH AT LEFT = 7.7641E-08 2.9957E-06 4.2660E-06 -1.1906E-04 7.6449E-05 -3.7055E-05 -2.9985E-04 2.1739E-04 1.5478E-04 1.9192E-03

UPH2 AT LEFT = -1.0324E-05 -3.5239E-04 -5.0182E-04 1.0[illegible]E-02 -7.2342E-03 4.2275E-03 2.9310E-02 -2.3551E-02 -1.2709E-02 -2.4592E-01

FLAME SPEED = 3.0737E+02 2.1350E+02 2.7354E+02 2.6935E+02 2.9288E+02 2.8121E+02 2.7799E+02 2.8121E+02 2.8070E+02

FLAME SPEED = 2.8136E+02 2.4861E+02 2.8218E+02 -7.5724E+01 3.7126E+04 2.8059E+02 2.8037E+02 2.8060E+02 2.8733E+02

FLAME SPEED = 2.8339E+02 2.4157E+02 1.9220E+02 -1.1633E+05 1.0940E+05 2.8005E+02 2.8039E+02 2.8059E+02 -2.1515E+01

FLAME FRONT FROM PHI* = 2.3796E-01 TO PHI* = 4.1429E-01

FLAME FRONT FROM X = 1.9535E-02 TO X = 6.8654E-02

FLAME THICKNESS = 4.9118E-02 CM

MO = 1.7700E-01 FLSP = 2.8067E+02 PHNE# = 2.9595E-01

SPN1 = -3.5401E+00 SPN2 = -3.5112E+00 SPEED = -3.5401E+00 DS = 9.3642E-02

T = 3.7539E-01 DT = 5.4542E-03 TOTAL STEPS = 256

NF = 391 NJ = 36

RUN TIME = 3.1810E+01

H2O AT LEFT = 9.0867E-03 2.7060E-03 2.7492E-03 1.8197E-03 1.8094E-03 4.6010E-03 1.8849E-03 2.3677E-03 0.

UPH AT LEFT = 4.3087E-08 1.5657E-06 2.6659E-06 -3.5304E-05 4.3792E-05 -2.8572E-05 -2.0169E-04 1.3041E-04 8.7095E-05 1.3225E-03

UPH2 AT LEFT = -5.7378E-06 -1.8641E-04 -3.1793E-04 3.1901E-03 -4.1940E-03 3.2514E-03 1.9885E-02 -1.4329E-02 -7.2934E-03 -1.7179E-01

FLAME SPEED = 3.2213E+02 2.3411E+02 2.7566E+02 2.5601E+02 2.8298E+02 2.8177E+02 2.7743E+02 2.8234E+02 2.8054E+02

FLAME SPEED = 2.8120E+02 2.6997E+02 2.8329E+02 -1.0309E+03 5.7995E+03 2.8065E+02 2.8043E+02 2.8064E+02 2.8061E+02

FLAME SPEED = 2.8218E+02 2.6031E+02 1.5519E+02 -7.1043E+05 2.1158E+04 2.8074E+02 2.8036E+02 2.8070E+02 2.8270E+02

FLAME FRONT FROM PHI* = 2.4219E-01 TO PHI* = 4.1427E-01

FLAME FRONT FROM X = 1.9814E-02 TO X = 6.6922E-02

FLAME THICKNESS = 4.7108E-02 CM

MO = 1.7695E-01 FLSP = 2.8058E+02 PHNEW = 3.0193E-01

SPN1 = -3.5390E+00 SPN2 = -3.5069E+00 SPEIGN = -3.5390E+00 DS = 1.0530E-03

T = 5.0000E-01 DT = 1.6201E-02 TOTAL STEPS = 272

NF = 418 NJ = 39

RUN TIME = 3.4065E+01

H

6.4851E-14	1.5344E-10	-2.1723E-10	-7.7488E-10	-1.1824E-09	-1.1027E-09	-1.9879E-10	1.8605E-09	5.4303E-09	3.9667E-09
-3.6888E-09	-1.2145E-08	-1.7213E-08	-1.3900E-08	2.5834E-09	3.7028E-08	9.4224E-08	6.3791E-08	-6.3010E-08	-1.9058E-07
-2.2332E-07	-6.5632E-08	3.7808E-07	1.2034E-06	2.5060E-06	3.2930E-06	4.1972E-06	6.3643E-06	1.0943E-05	1.9081E-05
3.1926E-05	5.0625E-05	7.6328E-05	9.9830E-05	1.2874E-04	1.6341E-04	2.0421E-04	2.5149E-04	3.0563E-04	3.6697E-04
4.3589E-04	5.1279E-04	5.9794E-04	6.9152E-04	7.9367E-04	9.0456E-04	1.0243E-03	1.1532E-03	1.2912E-03	1.4389E-03
1.5955E-03	1.7593E-03	1.9283E-03	2.1010E-03	2.2754E-03	2.4449E-03	2.6226E-03	2.7904E-03	2.9504E-03	3.1016E-03
3.2427E-03	3.3720E-03	3.4902E-03	3.5943E-03	3.6837E-03	3.7854E-03	3.8610E-03	3.9135E-03	3.9459E-03	3.9611E-03
3.9620E-03	3.9517E-03	3.9330E-03	3.8941E-03	3.8504E-03	3.8092E-03	3.7581E-03	3.7044E-03	3.6495E-03	3.5950E-03
3.5422E-03	3.4708E-03	3.4024E-03	3.3396E-03	3.2742E-03	3.2144E-03	3.1575E-03	3.1036E-03	3.0526E-03	2.9843E-03
2.9208E-03	2.8633E-03	2.8131E-03	2.7714E-03	2.7405E-03	2.7208E-03	2.7139E-03			

OH

1.1025E-10	6.9408E-09	-3.7700E-09	-2.1492E-08	-3.7130E-08	-3.9230E-08	-1.7948E-08	3.6907E-08	1.3588E-07	1.4454E-07
[illegible]									

1.0227E-06	3.1152E-06	7.7174E-06	1.5638E-05	2.7684E-05	4.5460E-05	7.9510E-05	1.2741E-04	1.8996E-04	2.6512E-04
3.5249E-04	4.5124E-04	5.6057E-04	6.3045E-04	6.8012E-04	7.1215E-04	7.2910E-04	7.3353E-04	7.2803E-04	7.1515E-04
6.9747E-04	6.7894E-04	6.6271E-04	6.4980E-04	6.4137E-04	6.3856E-04	6.4250E-04	6.5431E-04	6.7513E-04	7.0815E-04
7.5462E-04	8.1276E-04	8.8077E-04	9.5686E-04	1.0392E-03	1.1261E-03	1.2157E-03	1.3011E-03	1.3763E-03	1.4403E-03
1.4924E-03	1.5318E-03	1.5576E-03	1.5690E-03	1.5652E-03	1.5431E-03	1.5119E-03	1.4747E-03	1.4345E-03	1.3943E-03
1.3573E-03	1.3264E-03	1.3046E-03	1.2956E-03	1.2733E-03	1.2666E-03	1.2649E-03	1.2666E-03	1.2710E-03	1.2769E-03
1.2833E-03	1.2919E-03	1.2998E-03	1.3069E-03	1.3132E-03	1.3185E-03	1.3230E-03	1.3264E-03	1.3288E-03	1.3329E-03
1.3393E-03	1.3472E-03	1.3557E-03	1.3638E-03	1.3707E-03	1.3755E-03	1.3773E-03			

0

1.0376E-10	1.1736E-08	-6.7114E-09	-3.7870E-08	-6.4373E-08	-6.8952E-08	-3.3939E-08	5.7733E-08	2.2353E-07	2.7054E-07
1.8801E-07	8.3188E-08	5.3349E-08	2.3575E-07	7.0760E-07	1.5863E-06	2.9791E-06	3.7466E-06	4.1689E-06	4.9831E-06
6.9344E-06	1.0766E-05	1.7221E-05	2.7042E-05	4.0972E-05	5.5172E-05	7.4898E-05	1.0028E-04	1.3144E-04	1.6850E-04
2.1160E-04	2.6086E-04	3.1640E-04	3.5763E-04	3.9792E-04	4.3675E-04	4.7358E-04	5.0791E-04	5.3919E-04	5.6691E-04
5.9055E-04	6.1055E-04	6.2968E-04	6.5092E-04	6.7725E-04	7.1164E-04	7.5708E-04	8.1653E-04	8.9298E-04	9.9085E-04
1.1046E-03	1.2404E-03	1.3803E-03	1.5224E-03	1.6607E-03	1.7894E-03	1.9026E-03	1.9983E-03	2.0398E-03	2.0588E-03
2.0470E-03	2.0061E-03	1.9379E-03	1.8442E-03	1.7266E-03	1.5933E-03	1.3608E-03	1.1836E-03	1.0158E-03	8.6192E-04
7.2617E-04	6.1290E-04	5.2643E-04	4.3682E-04	3.6849E-04	3.1830E-04	2.8311E-04	2.5977E-04	2.4514E-04	2.3608E-04
2.2943E-04	2.2103E-04	2.1400E-04	2.0811E-04	2.0314E-04	1.9886E-04	1.9506E-04	1.9150E-04	1.8797E-04	1.8317E-04
1.7896E-04	1.7535E-04	1.7237E-04	1.7002E-04	1.6832E-04	1.6729E-04	1.6694E-04			

HO2

2.1401E-10	-2.1296E-07	-9.3423E-08	2.1863E-07	5.8302E-07	8.5956E-07	9.0807E-07	5.8838E-07	-2.3971E-07	-5.5285E-07
2.4280E-08	1.3715E-06	3.3684E-06	5.8951E-06	8.8314E-06	1.2056E-05	1.5450E-05	2.1539E-05	3.7805E-05	6.8544E-05
1.1806E-04	1.9065E-04	2.9061E-04	4.2223E-04	5.8981E-04	7.2696E-04	8.6568E-04	9.9355E-04	1.0982E-03	1.1671E-03
1.1879E-03	1.1483E-03	1.0357E-03	9.2899E-04	8.2631E-04	7.2900E-04	6.3838E-04	5.5575E-04	4.8245E-04	4.1978E-04
3.6907E-04	3.2768E-04	2.9170E-04	2.6056E-04	2.3360E-04	2.1044E-04	1.9031E-04	1.7270E-04	1.5701E-04	1.4279E-04
1.2986E-04	1.1808E-04	1.0730E-04	9.7388E-05	8.8198E-05	7.9588E-05	7.1415E-05	6.3671E-05	5.6443E-05	4.9728E-05
4.3525E-05	3.7832E-05	3.2646E-05	2.7966E-05	2.3790E-05	1.8729E-05	1.4578E-05	1.1233E-05	8.5898E-06	6.5445E-06
4.9933E-06	3.8321E-06	2.9572E-06	2.0118E-06	1.3172E-06	8.3318E-07	5.1980E-07	3.3701E-07	2.4480E-07	2.0314E-07
1.7200E-07	1.2472E-07	9.2013E-08	7.1245E-08	5.9773E-08	5.4954E-08	5.4143E-08	5.4700E-08	5.3980E-08	5.1073E-08
4.8660E-08	4.6711E-08	4.5190E-08	4.4065E-08	4.3303E-08	4.2869E-08	4.2732E-08			

H2O2

2.2049E-10	2.5113E-07	7.6061E-08	-3.2825E-07	-7.6506E-07	-1.0377E-06	-9.4929E-07	-3.0324E-07	1.0972E-06	1.5931E-06
9.0745E-07	-3.0915E-07	-1.4063E-06	-1.7334E-06	-6.4012E-07	2.5241E-06	8.4097E-06	1.0854E-05	1.0045E-05	9.5996E-06
1.3136E-05	2.4272E-05	4.6625E-05	8.3812E-05	1.3965E-04	2.0360E-04	2.9624E-04	4.0743E-04	5.2883E-04	6.4969E-04
7.6087E-04	8.5282E-04	9.1599E-04	9.3937E-04	9.4587E-04	9.3644E-04	9.1360E-04	8.7748E-04	8.2982E-04	7.7194E-04
7.0518E-04	6.3088E-04	5.5138E-04	4.6945E-04	3.9784E-04	3.0947E-04	2.3699E-04	1.7323E-04	1.2098E-04	8.1114E-05
5.1608E-05	3.0860E-05	1.7269E-05	9.2345E-06	5.1575E-06	3.4378E-06	2.4752E-06	1.5396E-06	9.2417E-07	5.6651E-07
4.0327E-07	3.7132E-07	4.0755E-07	4.4883E-07	4.3204E-07	3.4900E-07	2.8672E-07	2.4179E-07	2.1080E-07	1.9031E-07
1.7692E-07	1.6721E-07	1.5776E-07	1.4503E-07	1.3601E-07	1.3004E-07	1.2648E-07	1.2470E-07	1.2405E-07	1.2389E-07
1.2358E-07	1.2280E-07	1.2216E-07	1.2161E-07	1.2112E-07	1.2065E-07	1.2016E-07	1.1962E-07	1.1899E-07	1.1806E-07
1.1724E-07	1.1654E-07	1.1595E-07	1.1549E-07	1.1515E-07	1.1495E-07	1.1488E-07			

H2

6.4851E-02	6.4851E-02	6.4851E-02	6.4851E-02	6.4851E-02	6.4850E-02	6.4848E-02	6.4846E-02	6.4841E-02	6.4839E-02
6.4836E-02	6.4831E-02	6.4824E-02	6.4813E-02	6.4795E-02	6.4771E-02	6.4738E-02	6.4710E-02	6.4677E-02	6.4634E-02
6.4577E-02	6.4503E-02	6.4407E-02	6.4285E-02	6.4132E-02	6.4003E-02	6.3853E-02	6.3675E-02	6.3463E-02	6.3213E-02
6.2919E-02	6.2574E-02	6.2174E-02	6.1753E-02	6.1301E-02	6.1115E-02	6.0696E-02	6.0241E-02	5.9751E-02	5.9225E-02
5.8661E-02	5.8059E-02	5.7415E-02	5.6731E-02	5.6003E-02	5.5231E-02	5.4416E-02	5.3549E-02	5.2637E-02	5.1672E-02
5.0657E-02	4.9602E-02	4.8518E-02	4.7413E-02	4.6294E-02	4.5174E-02	4.4074E-02	4.2994E-02	4.1959E-02	4.0973E-02
4.0044E-02	3.9177E-02	3.8379E-02	3.7659E-02	3.7015E-02	3.6244E-02	3.5617E-02	3.5116E-02	3.4727E-02	3.4433E-02
3.4218E-02	3.4068E-02	3.3965E-02	3.3870E-02	3.3819E-02	3.3803E-02	3.3817E-02	3.3851E-02	3.3900E-02	3.3956E-02
3.4011E-02	3.4084E-02	3.4157E-02	3.4228E-02	3.4297E-02	3.4364E-02	3.4429E-02	3.4491E-02	3.4549E-02	3.4629E-02
3.4710E-02	3.4784E-02	3.4859E-02	3.4920E-02	3.4968E-02	3.4999E-02	3.5010E-02			

O2

2.1790E-01	2.1790E-01	2.1790E-01	2.1790E-01	2.1790E-01	2.1790E-01	2.1790E-01	2.1790E-01	2.1789E-01	2.1789E-01
2.1789E-01	2.1790E-01	2.1790E-01	2.1790E-01	2.1789E-01	2.1787E-01	2.1783E-01	2.1781E-01	2.1781E-01	2.1780E-01
2.1775E-01	2.1764E-01	2.1744E-01	2.1713E-01	2.1667E-01	2.1625E-01	2.1571E-01	2.1502E-01	2.1413E-01	2.1302E-01
2.1163E-01	2.0993E-01	2.0789E-01	2.0621E-01	2.0434E-01	2.0225E-01	1.9996E-01	1.9746E-01	1.9475E-01	1.9183E-01
1.8868E-01	1.8533E-01	1.8174E-01	1.7790E-01	1.7376E-01	1.6929E-01	1.6447E-01	1.5925E-01	1.5360E-01	1.4742E-01
1.4067E-01	1.3342E-01	1.2574E-01	1.1770E-01	1.0937E-01	1.0081E-01	9.2103E-02	8.3430E-02	7.4976E-02	6.6815E-02
5.9018E-02	5.1658E-02	4.4807E-02	3.8538E-02	3.2922E-02	2.8111E-02	2.4081E-02	1.5904E-02	1.2252E-02	9.3941E-03
7.2024E-03	5.5478E-03	4.3012E-03	2.9583E-03	1.9637E-03	1.2632E-03	8.0201E-04	5.2584E-04	3.8014E-04	3.1044E-04
2.6226E-04	1.9494E-04	1.4781E-04	1.1730E-04	9.9862E-05	9.1926E-05	8.9932E-05	9.0318E-05	8.9522E-05	8.6147E-05
8.3290E-05	8.0929E-05	7.9044E-05	7.7614E-05	7.6619E-05	7.6037E-05	7.5848E-05			

H2O

1.1673E-08	6.5015E-07	-1.1076E-07	-1.5378E-06	-2.8976E-06	-3.4570E-06	-2.4827E-06	7.5867E-07	7.0003E-06	8.7742E-06
5.2380E-06	4.6384E-08	-3.1459E-06	-6.8427E-07	1.1086E-05	3.5819E-05	7.7171E-05	9.4319E-05	9.1072E-05	9.3323E-05
1.2696E-04	2.1788E-04	3.9197E-04	6.7513E-04	1.0932E-03	1.4822E-03	1.9902E-03	2.6776E-03	3.6047E-03	4.8317E-03
6.4190E-03	8.4268E-03	1.0915E-02	1.2993E-02	1.5336E-02	1.7946E-02	2.0823E-02	2.3965E-02	2.7373E-02	3.1047E-02
3.4987E-02	3.9191E-02	4.3674E-02	4.8454E-02	5.3567E-02	5.9024E-02	6.4852E-02	7.1075E-02	7.7716E-02	8.4854E-02
9.2506E-02	1.0060E-01	1.0907E-01	1.1783E-01	1.2683E-01	1.3598E-01	1.4522E-01	1.5437E-01	1.6327E-01	1.7186E-01
1.8005E-01	1.8780E-01	1.9501E-01	2.0164E-01	2.0761E-01	2.1490E-01	2.2096E-01	2.2592E-01	2.2991E-01	2.3306E-01
2.3550E-01	2.3735E-01	2.3876E-01	2.4027E-01	2.4138E-01	2.4217E-01	2.4268E-01	2.4298E-01	2.4314E-01	2.4321E-01
2.4320E-01	2.4333E-01	2.4338E-01	2.4341E-01	2.4342E-01	2.4343E-01	2.4343E-01	2.4343E-01	2.4343E-01	2.4344E-01
2.4344E-01	2.4343E-01	2.4343E-01	2.4342E-01	2.4342E-01	2.4341E-01	2.4341E-01			

N2

7.1725E-01	7.1725E-01	7.1725E-01	7.1725E-01	7.1725E-01	7.1725E-01	7.1725E-01	7.1725E-01	7.1726E-01	7.1726E-01
7.1726E-01	7.1727E-01	7.1727E-01	7.1728E-01	7.1729E-01	7.1731E-01	7.1733E-01	7.1734E-01	7.1737E-01	7.1739E-01
7.1741E-01	7.1741E-01	7.1740E-01	7.1736E-01	7.1730E-01	7.1723E-01	7.1713E-01	7.1699E-01	7.1684E-01	7.1667E-01
7.1649E-01	7.1630E-01	7.1611E-01	7.1598E-01	7.1585E-01	7.1571E-01	7.1556E-01	7.1540E-01	7.1524E-01	7.1506E-01
7.1487E-01	7.1466E-01	7.1443E-01	7.1419E-01	7.1394E-01	7.1368E-01	7.1342E-01	7.1316E-01	7.1291E-01	7.1269E-01
7.1252E-01	7.1241E-01	7.1236E-01	7.1237E-01	7.1244E-01	7.1258E-01	7.1279E-01	7.1306E-01	7.1337E-01	7.1370E-01
7.1406E-01	7.1442E-01	7.1478E-01	7.1513E-01	7.1545E-01	7.1586E-01	7.1620E-01	7.1648E-01	7.1671E-01	7.1689E-01
7.1703E-01	7.1714E-01	7.1721E-01	7.1728E-01	7.1735E-01	7.1737E-01	7.1740E-01	7.1741E-01	7.1741E-01	7.1741E-01
7.1741E-01	7.1741E-01	7.1740E-01	7.1739E-01	7.1739E-01	7.1738E-01	7.1737E-01	7.1737E-01	7.1736E-01	7.1735E-01
7.1733E-01	7.1731E-01	7.1729E-01	7.1727E-01	7.1726E-01	7.1725E-01	7.1724E-01			

TEMP/1000

2.9800E-01	2.9801E-01	2.9799E-01	2.9798E-01	2.9797E-01	2.9798E-01	2.9801E-01	2.9808E-01	2.9819E-01	2.9824E-01
2.9822E-01	2.9821E-01	2.9830E-01	2.9854E-01	2.9901E-01	2.9980E-01	3.0096E-01	3.0185E-01	3.0282E-01	3.0425E-01
3.0650E-01	3.0996E-01	3.1499E-01	3.2198E-01	3.3130E-01	3.3979E-01	3.5039E-01	3.6330E-01	3.7874E-01	3.9692E-01
4.1804E-01	4.4231E-01	4.6994E-01	4.9147E-01	5.1444E-01	5.3875E-01	5.6431E-01	5.9101E-01	6.1877E-01	6.4748E-01
6.7704E-01	7.0738E-01	7.3842E-01	7.7003E-01	8.0211E-01	8.3454E-01	8.6721E-01	9.0001E-01	9.3282E-01	9.6553E-01
9.9796E-01	1.0300E+00	1.0614E+00	1.0920E+00	1.1217E+00	1.1503E+00	1.1777E+00	1.2036E+00	1.2279E+00	1.2504E+00
1.2718E+00	1.2915E+00	1.3097E+00	1.3263E+00	1.3415E+00	1.3608E+00	1.3777E+00	1.3926E+00	1.4057E+00	1.4172E+00
1.4275E+00	1.4367E+00	1.4451E+00	1.4561E+00	1.4661E+00	1.4755E+00	1.4841E+00	1.4921E+00	1.4997E+00	1.5069E+00
1.5138E+00	1.5232E+00	1.5322E+00	1.5407E+00	1.5488E+00	1.5565E+00	1.5638E+00	1.5706E+00	1.5770E+00	1.5861E+00
1.5955E+00	1.6046E+00	1.6136E+00	1.6212E+00	1.6273E+00	1.6313E+00	1.6327E+00			

H2O AT LEFT = 9.0867E-03 2.7060E-03 2.7492E-03 1.8197E-03 1.8094E-03 4.6010E-03 1.8849E-03 2.3677E-03 0.

UPM AT LEFT = 4.3075E-08 1.5587E-06 2.6494E-06 -3.47[illegible]E-05 4.3214E-05 -2.85[illegible]E-05 -1.9901E-04 1.2917E-04 8.571[illegible]E-05 1.3176E-03

UPM2 AT LEFT = -5.7364E-06 -1.8566E-04 -3.1607E-04 3.1[illegible]1E-03 -4.1481E-03 3.2465E-03 1.9662E-02 -1.4207E-02 -7.1967E-03 -1.7063E-01

FLAME SPEED = 3.2237E+02 2.3404E+02 2.7567E+02 2.5597E+02 2.8300E+02 2.8179E+02 2.7744E+02 2.8237E+02 2.8068E+02

FLAME SPEED = 2.8127E+02 2.6871E+02 2.8314E+02 -1.09[illegible]E+03 2.6273E+03 2.8065E+02 2.8044E+02 2.8066E+02 2.7885E+02

FLAME SPEED = [illegible]

```
      X IN CM
                                                                  4.8568E-03  5.8281E-03  6.7996E-03  7.7714E-03  8.4587E-03
0.          9.7140E-04  1.9428E-03  2.9142E-03  3.8855E-03  1.2587E-02  1.3279E-02  1.3769E-02  1.4262E-02  1.4756E-02
9.1461E-03  9.8334E-03  1.0521E-02  1.1208E-02  1.1897E-02  1.7694E-02  1.8089E-02  1.8498E-02  1.8922E-02  1.9365E-02
1.5252E-02  1.5754E-02  1.6262E-02  1.6779E-02  1.7304E-02  2.2050E-02  2.2491E-02  2.2953E-02  2.3435E-02  2.3940E-02
1.9830E-02  2.0320E-02  2.0838E-02  2.1224E-02  2.1628E-02  2.7452E-02  2.8122E-02  2.8815E-02  2.9533E-02  3.0276E-02
2.4466E-02  2.5016E-02  2.5589E-02  2.6186E-02  2.6807E-02  3.5211E-02  3.6108E-02  3.7024E-02  3.7957E-02  3.8907E-02
3.1042E-02  3.1831E-02  3.2643E-02  3.3478E-02  3.4334E-02  4.5313E-02  4.6782E-02  4.8267E-02  4.9765E-02  5.1274E-02
3.9872E-02  4.0851E-02  4.1843E-02  4.2847E-02  4.3861E-02  6.2467E-02  6.4695E-02  6.6934E-02  6.9184E-02  7.1444E-02
5.2794E-02  5.4324E-02  5.5861E-02  5.8049E-02  6.0252E-02  9.0012E-02  9.3318E-02  9.6638E-02  9.9970E-02  1.0470E-01
7.3714E-02  7.6940E-02  8.0183E-02  8.3443E-02  8.6720E-02  1.3362E-01  1.3850E-01
1.0946E-01  1.1425E-01  1.1906E-01  1.2390E-01  1.2875E-01

        FLAME FRONT FROM PHI* =  2.4219E-01   TO PHI* =  4.1427E-01

        FLAME FRONT FROM X =  1.9814E-02   TO X =  6.6924E-02

        FLAME THICKNESS =  4.7110E-02  CM

  9  97
  H        1.00000000E+00   5.21031718E+04   9.08674678E-03
  OH       1.70000000E+01   5.56288874E+02   2.70604230E-03
  O        1.60000000E+01   3.72206442E+03   2.74924008E-03
 HO2       3.30000000E+01   1.51456505E+02   1.81967208E-03
 2O2       3.40000000E+01  -9.56829781E+02   1.80941928E-03
  H2       2.00000000E+00  -1.59507953E-01   4.60099146E-03
  O2       3.20000000E+01  -3.56890803E-02   1.88489399E-03
 H2O       1.80000000E+01  -3.21097059E+03   2.36766026E-03
  N2       2.80000000E+01  -3.95183284E-02   0.

 5.00000000E-01
-3.54323848E+00   8.45000316E-03   1.00000000E+03   5.00000000E-05   1.00000000E-03
 4.78562202E-01   6.55632474E-02   2.98000000E+02
 1.00000000E+00   1.21876904E-02

    HO =  1.7698E-01       FLSP =  2.8063E+02        PHNE* =  3.0193E-01

-3.539570E+00
 3.019300E-01

         COLLOCATION POINTS

     26                                                               1.526805E-01   1.776841E-01   2.059795E-01   2.236597E-01
 0.             2.071366E-02   7.730441E-02   1.126648E-01   2.948216E-01   3.051784E-01   3.193261E-01   3.296829E-01
 2.436676E-01   2.561694E-01   2.703171E-01   2.806739E-01   4.223159E-01   4.473195E-01   4.873352E-01   5.226956E-01
 3.438306E-01   3.563324E-01   3.763403E-01   3.940205E-01
 5.792863E-01   6.000000E-01                                  1.109824E-08  -1.923730E-08   5.882144E-07   3.836122E-06
 6.485084E-14  -6.685182E-11   3.004216E-10  -8.303138E-10   1.063220E-03   1.546234E-03   2.329472E-03   2.901808E-03
 3.703242E-05   1.191935E-04   3.238188E-04   5.707043E-04   1.632524E-03   3.423251E-03   3.140542E-03   2.939868E-03
 3.523876E-03   3.840172E-03   3.959881E-03   3.869944E-03
 2.733145E-03   2.713920E-03                                  3.033194E-07   1.344389E-06   9.775728E-06   6.749463E-05
 1.102464E-10   8.217861E-10  -5.073133E-09   8.935328E-08   4.452574E-04   7.389152E-04   1.065715E-03   1.354165E-03
 3.818662E-04   6.667576E-04   7.247153E-04   6.674358E-04   1.272714E-03   1.297434E-03   1.324151E-03   1.337156E-03
 1.562675E-03   1.522345E-03   1.346901E-03   1.276444E-03
 1.372439E-03   1.377277E-03                                  4.310664E-07   4.031410E-06   1.966496E-05   6.819661E-05
 1.037614E-10   1.164168E-09  -1.244945E-08   2.204271E-07   7.738368E-04   1.070417E-03   1.701867E-03   2.027428E-03
 2.281753E-04   3.855889E-04   5.481731E-04   6.236708E-04   2.418863E-04   2.160423E-04   1.939397E-04   1.801976E-04
 1.911591E-03   1.416931E-03   6.885170E-04   3.875447E-04   [illegible]    [illegible]    [illegible]    [illegible]
 1.679260E-04   1.669388E-04
 2.160074E-10  -1.575110E-07   [illegible]    [illegible]
```

```
1.182723E-03  8.575691E-04  4.618628E-04  3.022893E-04  1.846216E-04  1.337297E-04  8.547893E-05  5.862451E-05
3.114448E-05  1.577175E-05  4.597169E-06  1.507748E-06  2.283767E-07  1.007329E-07  5.429287E-08  4.935581E-08
4.313688E-08  4.273245E-08
2.204929E-10  1.638916E-07 -8.181571E-07  1.206643E-06  8.730431E-08  1.042506E-05  5.637685E-05  2.651095E-04
7.914157E-04  9.455837E-04  8.129410E-04  5.763647E-04  2.162042E-04  5.972726E-05  4.448189E-06  1.084060E-06
4.227233E-07  3.039779E-07  1.736816E-07  1.384435E-07  1.239794E-07  1.223490E-07  1.200039E-07  1.174837E-07
1.150745E-07  1.148776E-07
6.485084E-02  6.485074E-02  6.484747E-02  6.483671E-02  6.478865E-02  6.468762E-02  6.437222E-02  6.390176E-02
6.281797E-02  6.161351E-02  5.959232E-02  5.761871E-02  5.415119E-02  5.097553E-02  4.595208E-02  4.227400E-02
3.814730E-02  3.579647E-02  3.416574E-02  3.383061E-02  3.391701E-02  3.413461E-02  3.444837E-02  3.468553E-02
3.497931E-02  3.500996E-02
2.178998E-01  2.178982E-01  2.179023E-01  2.178921E-01  2.178869E-01  2.178125E-01  2.173569E-01  2.154923E-01
2.111397E-01  2.049389E-01  1.938710E-01  1.828783E-01  1.628959E-01  1.428174E-01  1.067389E-01  7.756356E-02
4.280195E-02  2.210425E-02  6.639131E-03  2.237501E-03  3.529965E-04  1.604470E-04  8.996373E-05  8.411963E-05
7.639565E-05  7.584760E-05
1.167315E-08  2.300260E-07 -1.756189E-06  6.679335E-06  1.718319E-05  9.269714E-05  4.667241E-04  1.816660E-03
6.992510E-03  1.458276E-02  2.848159E-02  4.225573E-02  5.673452E-02  9.008757E-02  1.296462E-01  1.605500E-01
1.971313E-01  2.192041E-01  2.361270E-01  2.410771E-01  2.431679E-01  2.433655E-01  2.434309E-01  2.434373E-01
2.434170E-01  2.434128E-01
2.980000E-01  2.979990E-01  2.980245E-01  2.982263E-01  2.992196E-01  3.024931E-01  3.169305E-01  3.468703E-01
4.251998E-01  5.071887E-01  6.275526E-01  7.287487E-01  8.773472E-01  9.879669E-01  1.130709E+00  1.220536E+00
1.314973E+00  1.372683E+00  1.430418E+00  1.463114E+00  1.501979E+00  1.529452E+00  1.565941E+00  1.592574E+00
1.624770E+00  1.632729E+00
9.086747E-03  2.706042E-03  2.749240E-03  1.819672E-03  1.809419E-03  4.600991E-03  1.884894E-03  2.367660E-03
0.

        INDEX =  0
```

GLOSSARY

c_{pk} = specific heat of species k, cal-g^{-1} - K^{-1}.

c_p = specific heat of the mixture, cal-g^{-1} - K^{-1}.

D_{km} = diffusion coefficient of species k in the mixture, cm^2-s^{-1}.

h_k = specific enthalpy of species k, cal-g^{-1}.

k_j = rate constant for reaction j in centimeter - mole - second units.

m_o = mass flux of the mixture through the origin, g-cm^{-2}-s^{-1}.

M_k = molecular weight of species k, g-$mole^{-1}$.

N = number of chemical species (also number of PDE's).

NO = number of ODE's.

p = pressure, atmos.

r_j = rate of reaction j, mole-cm^{-1}-s^{-1}.

R = gas constant = 82.05 cm^3-atoms-$mole^{-1}$-K^{-1}.

R_k = rate of production of species k, mole-cm^{-3}-s^{-1}.

S_v = velocity of the flame relative to the unburned mixture, cm-s^{-1}.

$\hat{t}$ = temporal coordinate, s. ($\hat{t}$, $\hat{x}$ coordinate system).

$\tilde{t}$ = temporal coordinate, s. ($\tilde{t}$, $\tilde{\psi}$ coordinate system).

t = non dimensional temporal coordinate, with respect to t_∞ (t, ψ coordinate system).

$\hat{T}$ = temperature of the mixture, K.

$\tilde{T}$ = temperature of the mixture, K.

T = non dimensional temperature of the mixture, with respect to T_∞.

T_U = temperature of the unburned mixture.

T_B = temperature of the burned mixture (adiabatic temperature).

v = fluid velocity of the mixture, cm-s^{-1}.

V_k = diffusion velocity of species k, cm-s^{-1}.

$\hat{x}$ = spatial coordinate, cm.

Y_k = mass fraction of species k.

Y_{kU} = mass fraction of species k in the unburned mixture.

Y_{kB} = mass fraction of species k in the burned mixture.

ε_k = mass flux fraction of species k.

λ = heat conductivity of the mixture, cal-cm^{-1}-s^{-1}-K^{-1}.

ρ = density of the mixture, g-cm^{-3}.

$\hat{\psi}$ = transformed distance coordinate, g-cm^{-2}.

ψ = non dimensional transformed distance coordinate, with respect to ψ_∞.

DISTRIBUTION LIST

No. of Copies	Organization
1 2	Commander Defense Technical Info Center ATTN: DTIC-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 Analysis 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
4	Commander US Army Armament Research and Development Command ATTN: DRDAR-LCA, J. Lannon DRDAR-LC, T. Vladimiroff DRDAR-LCE, R.F. Walker DRDAR-SCA, L. Stiefel Dover, NJ 07801
2	Commander US Army Armament Research & Development Command ATTN: DRDAR-TSS Dover, NJ 07801
1	Commander US Army Armament Materiel Readiness Command ATTN: DRSAR-LEP-L, Tech Lib Rock Island, IL 61299
1	Commander US Army Watervliet Arsenal ATTN: Code SARWV-RD, R. Thierry Watervliet, NY 12189
1	Director US Army ARRADCOM Benet Weapons Laboratory ATTN: DRDAR-LCB-TL Watervliet, NY 12189
1	Commander US Army Aviation Research and Development Command ATTN: DRDAV-E 4300 Goodfellow Blvd St. Louis, MO 63120
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-PPA-SA Ft. Monmouth, NJ 07703
1	Commander US Army Electronics Research & Development Command Technical Support Activity ATTN: DELSD-L Fort Monmouth, NJ 07703

DISTRIBUTION LIST

No. of Copies	Organization
1	Commander US Army Missile Command ATTN: DRSMI-R Redstone Arsenal, AL 35809
1	Commander US Army Missile Command ATTN: DRSMI-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-DP-AL WSMR, NM 88002
1	Commander US Army Materiel & Mechanics Research Center ATTN: DRXMR-ATL Watertown, MA 02172
5	Commander US Army Research Office ATTN: Tech Lib D. Squire F. Schmiedeshaff R. Ghirardelli M. Ciftan Research Triangle Park, NC 27709
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 G. Neece 800 N. Quincy Street Arlington, VA 22217
1	Commander Naval Sea Systems Command ATTN: J.W. Murrin (NAVSEA-62R2) National Center Building 2, Room 6E08 Washington, DC 20362
2	Commander Naval Surface Weapons Center ATTN: S.J. Jacobs/Code 240 Code 730 Silver Springs, 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. Lasar/Code 5B331 Newport, RI 02840
2	Commander Naval Weapons Center ATTN: R. Derr C. Thelen China Lake, CA 93555

DISTRIBUTION LIST

No. of Copies	Organization
1	Commander Naval Research Laboratory ATTN: Code 6180 Washington, DC 20375
3	Superintendent Naval Post Graduate School ATTN: Tech Lib D. Netzer A. Fuhs Monterey, CA 93940
2	Commander Naval Ordnance Station ATTN: S. Mitchell Tech Lib Indian Head, MD 20640
4	AFOSR ATTN: D. Ball L. Caveny B. T. Wolfson Bolling AFB, CA 93523
2	AFRPL (DYSC) ATTN: D. George J.N. Levine Edwards AFB, CA 93523
2	National Bureau of Standards ATTN: J. Hastie T. Kashiwagi Washington, DC 20234
1	Lockheed Missile & Space Co. 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: D. 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
1	Calspan Corporation ATTN: E.B. Fisher A.P. Trippe P.O. Box 400 Buffalo, NY 14221
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 Flight Propulsion Division ATTN: Tech Lib Cincinnati, OH 45215

DISTRIBUTION LIST

No. of Copies	Organization
1	General Motors Research Laboratories Physics Department ATTN: C. Dasch Warren, MI 48090
2	Hercules Incorporated Alleghany Ballistic Lab ATTN: R. Miller Tech Lib Cumberland, MD 21501
2	Hercules Incorporated Bacchus Works ATTN: B. Isom 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
2	Olin Corporation New Haven Plant 275 Winchester Avenue New Haven, CT 06504
1	Paul Gough Associates, Inc. ATTN: P.S. Gough P.O. Box 1614 Portsmouth, NH 03801
1	Physics International Company 2700 Merced Street Leandro, CA 94577
1	Pulsepower Systems, Inc. ATTN: L.C. Elmore 815 American Street San Carlos, CA 94070
3	Rockwell International Corp. Rocketdyne Division ATTN: C. Obert J.E. Flanagan A. Axeworthy 6633 Canoga Avenue Canoga Park, CA 91304
2	Hercules, Inc. Industrial Systems Dept. P. O. Box 548 McGregor, TX 76657
1	SAI ATTN: R. Farmer 21133 Victory Boulevard Suite 216 Canoga Park, CA 91303
1	Science Applications, Inc. ATTN: R.B. Edelman Combustion Dynamics & Propulsion Division 23146 Cumorah Crest Woodland Hills, CA 91364
1	Shock Hydrodynamics, Inc. ATTN: W.H. Anderson 4710-16 Vineland Avenue North Hollywood, CA 91602
1	Thiokol Corporation Elkton Division ATTN: E. Sutton Elkton, MD 21921

DISTRIBUTION LIST

No. of Copies	Organization
3	Thiokol Corporation Huntsville Division ATTN: D. Flanigan R. Glick Tech Lib Huntsville, AL 35807
2	Thiokol Corporation Wasatch Division ATTN: J. Peterson Tech Lib P.O. Box 524 Brigham City, UT 84302
1	TRW Systems Group ATTN: H. Korman One Space Park Redondo Beach, CA 90278
2	United Technologies ATTN: R. Brown Tech Lib P.O. Box 358 Sunnyvale, CA 94086
1	Universal Propulsion Co. ATTN: H.J. McSpadden 1800 West Deer Valley Rd. Phoenix AZ 85027
11	Battelle Memorial Institute ATTN: Tech Lib R. Bartlett (10 cys) 505 King Avenue Columbus, OH 43201
1	Brigham Young University Dept of Chemical Engineering ATTN: M. W. Beckstead Provo, UT 84601
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 Technology 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	Princeton University Princeton Combustion Rsch Lab ATTN: M. Summerfield 1041 US Highway One North Princeton, NJ 08540

DISTRIBUTION LIST

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 16801
1	Pennsylvania State University Dept of Material Sciences ATTN: H. Palmer University Park, PA 16802
1	Princeton University Forrestal Campus ATTN: I. Glassman Tech Lib P.O. Box 710 Princeton, NJ 08540
1	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	Southwest Research Institute Fire Research Section 8500 Culebra Road San Antonio, TX 78228
4	SRI International ATTN: Tech Lib J. Barker D. Crosley D. Golden 333 Ravenswood Avenue Menlo Park, CA 94025
1	Stevens Institute of Technology Davidson Laboratory ATTN: R. McAlevy, III Hoboken, NJ 07030
1	University of California, San Diego Ames Department ATTN: F. Williams P.O. Box 109 LaJolla, CA 92037
1	University of Illinois Dept of Aeronautical Engineering ATTN: H. Krier 144 MLB, 1206 W. Green St. Urbana, IL 61801
1	University of Minnesota Dept of Mechanical Engineering ATTN: E. Fletcher Minneapolis, MN 55455
1	University of Southern California Dept of Chemistry ATTN: S. Benson Los Angeles, CA 90007
2	University of Texas Dept of Chemistry ATTN: W. Gardiner H. Schaefer Austin, TX 78712

DISTRIBUTION LIST

No. of Copies	Organization
2	University of Utah Dept of Chemical Engineering ATTN: A. Baer G. Flandro Salt Lake City, UT 84112
3	Los Alamos National Laboratory Center for Non-Linear Studies, MS457 ATTN; Dr. Basil Nichols P.O. Box 1663 Los Alamos, New Mexico 87545
1	North Texas State University Mathematics Department ATTN: Dr. Robert R. Kallman Denton, Texas 76203

Aberdeen Proving Ground

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

Cdr, USATECOM
ATTN: DRSTE-TO-F

Dir, USACSL, Bldg. E3516, EA
ATTN: DRDAR-CLB-PA

USER EVALUATION OF REPORT

Please take a few minutes to answer the questions below; tear out this sheet, fold as indicated, staple or tape closed, and place in the mail. 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: _____

_____

_____